TÉCNICO LISBOA

STOP: Secure Transport lOcation Proofs for vehicle inspections

Henrique Figueiredo dos Santos

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisor: 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. José Manuel da Costa Alves Marques

December 2019

"The advance of technology is based on making it fit in so that you don't really even notice it, so it's part of everyday life." - Bill Gates

Acknowledgments

First of all, I would like to deeply thank my family, specially my parents for their support, care and encouragement over the years and my girlfriend Filipa for her continuous love, support and honesty, which makes me evolve every single day.

During these past five years at IST, I met incredible people and I am lucky to call some of them friends. A special thank you to the people I met in my first year, Diogo, Liliana, Mariana, Tiago, Nelson, Genebra, Nuno, Andre, just to name a few, for their friendship and academic support over the years. I was also part of LAGE2 and SET during the years where I met hard working and dedicated people, like Rita, Mariana, Regina and Valado, who taught me a lot.

I was very lucky to join the DSI Tagus team four years ago and I had the tremendous support from Professor Fernando, Bruno, Helena, Simão, Manuel, Luis and Ricardo during these years and specifically during the elaboration of this work. Additionally I was able to use resources from DSI Tagus, which hugely benefited this work and for which I am grateful.

Last but not least, I would like to express my deepest gratitude and appreciation to my supervisor, Professor Miguel Pardal, and to the SureThing team. The help, support and guidance from Professor Miguel were essential and the ideas shared and discussed with Rui, Gabriel, Pedro and Sheng made a big impact on this dissertation. Additionally I would like to thank Prof. Dr. Leonardo Rocha for the detailed revision of this work, which provided very useful feedback and consecutive improvements, and also to my longtime friends Bruno, Duarte and Miguel for participating in the evaluation of this project.

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

Resumo

As autoridades europeias e mundiais estão a desenvolver esforços para digitalizar os processos que suportam o transporte de mercadorias. Em várias partes do processo ainda são utilizados documentos em papel, mas as entidades governamentais estão a começar a implementar sistemas de "electronic Freight Transport Information (eFTI)" e as empresas são obrigadas a enviar informação de transporte antes do início do mesmo. No entanto, apesar destas melhorias, quando os veículos de transporte rodoviário são inspecionados, os inspetores ainda demoram a recolher as informações necessárias e a analisá-las enquanto o veículo está parado. Nesta dissertação, propomos *STOP*, um sistema de suporte à inspeção de veículos de transporte rodoviário. Este sistema utiliza dispositivos móveis e permite que os inspetores selecionem e notifiquem veículos para inspeção, enquanto recolhem as informações necessárias para preparar o procedimento de inspeção antecipadamente. Um servidor central controlado pelas autoridades coordena as interações do sistema, permitindo que o sistema esteja de acordo com a legislação e seja auditável. O trabalho relacionado foi apresentado para mostrar o que atualmente está a ser usado neste setor e como os dispositivos móveis podem ser usados para este propósito. Um protótipo do sistema STOP foi implementado com aplicações Android e avaliado com vários utilizadores. A avaliação focou-se na precisão e período da recolha de localização de dispositivos móveis, tempos de resposta do sistema, viabilidade da comunicação Bluetooth num cenário de inspeção e determinação dos melhores parâmetros para o sistema.

Palavras-chave: Inspeções inteligentes de veículos, Aplicações móveis, Provas de localização, Rastreamento de localização, Transformação digital no rodoviária

Abstract

An effort is being made by authorities around Europe and worldwide to digitize the processes that support the transportation of freight. Paper documents are still being used in several parts of the process, but governments are starting to implement electronic Freight Transport Information (eFTI) systems and enterprises are required to submit this information before the transportation begins. However despite these improvements, when road transportation vehicles are inspected, inspectors still take time to retrieve the information and analyze it whilst the vehicle is stopped. In this dissertation we propose *STOP*, a road transportation vehicle inspection support system. This system uses mobile devices and allows inspectors to select and notify vehicles for inspection while retrieving the needed information to prepare the inspection procedure beforehand. A central server owned by authorities coordinates the system interactions, allowing the system to be compliant and auditable. Related work was researched and presented to showcase what is currently being used in this sector and how mobile devices could be used for our purpose. A prototype of the STOP system was implemented, including Android applications and evaluated with several users. The evaluation focused on the location retrieval accuracy and rate of mobile devices, system response times, Bluetooth communication viability in an inspection scenario and on determining what are the best parameters for the system.

Keywords: Smart vehicle inspections, Mobile applications, Location proofs, Location tracking, Digital transformation in road transport

Contents

List of Tables

List of Figures

Glossary

Application Programming Interface (API) Application Programming Interface is a set of definitions and protocols for building and integrating application software.

Basic Service Set Identifier (BSSID) Basic Service Set Identifier is a unique identifier of a wireless network.

Database Management System (DBMS) Database Management System is a software system that enables users to define, create, maintain and control access to the database.

electronic Freight Transport Information (eFTI) Electronic Freight Transport Information is the digital representation of the details of a freight transportation.

Global Navigation Satellite System (GNSS) Global Navigation Satellite System is a satellite navigation system with global coverage.

Global Positioning System (GPS) Global Positioning System is a satellite-based radio-navigation system.

Global System for Mobile Communications (GSM) Global System for Mobile Communications is a standard for communication between mobile devices.

Inertial Navigation System (INS) Inertial Navigation System is a navigation system that uses motion and rotation sensors to calculate orientation and velocity.

JavaScript Object Notation (JSON) JavaScript Object Notation is a an open-standard file format.

Media Access Control (MAC) Media Access Control is a mechanism to support the identification and control of computers on a network.

On-board Diagnostics (OBD) On-board Diagnostics refers to the self-diagnostic and reporting capability of a vehicle.

Representational State Transfer (REST) Representational State Transfer is a software architectural style that defines a set of constraints to be used for creating Web services.

Service Set Identifier (SSID) Service Set Identifier is a customizable identifier of a wireless network, often refereed as the "network name".

Trusted Platform Module (TPM) Trusted Platform Module is a standard for a dedicated microcontroller designed to store encryption keys for hardware authentication.

User Interface (UI) User Interface is the space where interactions between humans and machines occur.

Chapter 1

Introduction

In recent years, around the world, there is a strong focus in implementing the digitalization of freight transport support processes. There has been a proposal for a protocol in 2008 for the United Nations Contract for the International Carriage of Goods by Road (CMR) to include electronic consignment notes (e-CMR) [35]. However, so far, only 22 countries have acceded to this [36]. In Europe, several reports have identified issues in this sector, as paper documents are still used in freight transport activities and there is lack of a legal framework requiring authorities to accept electronic Freight Transport Information (eFTI) [26, 30]. At the time of writing, the European Council has already announced procedures to mandate the implementation of such digitalization in the European Union [13, 12]. In Portugal, companies already have to submit freight transportation information electronically or by telephone before the transportation begins[1]. In case of inspection, the carrier only has to provide the identifier of the electronically submitted report with no usage of paper documents. The Portuguese Government has also implemented an electronic waste transportation report system, called e-GAR, where involved entities use an online platform to register and acknowledge transports of this kind [1, 3, 2]. In case of inspection, the carrier only has to present a PDF file in a mobile device, containing a QR code to identify the submitted information.

Despite the initial costs of this digitalization, both governmental parties and enterprises see this subject as a potential productivity enhancement to the industry, as pointed out by several reports and news outlets [30, 26, 37]. This can bring several positives outcomes, such as decreased environmental impact, less longsome bureaucratic procedures and significant savings.

With the amount of transportation data that can be gathered along with the openness of the transportation industry to digitize transportation processes, one can think of support solutions using recent technology. One scenario is the inspection of road transportation [29]. At

[1]Decreto-Lei n.º 198/2012 of August 24th

an inspection site, an inspector orders incoming transportation vehicles to stop to conduct an inspection, with no previous knowledge of what these vehicles are transporting. The first phase of this process is to request and analyze every legally required documents, such as the freight transportation information. Depending on the type or size of freight, the inspector has to adapt the procedure to the situation, possibly requesting colleagues to help. Naturally, this step may take additional time. If the selection and notification of vehicles for inspection could be done beforehand, inspectors would then have additional time to prepare the inspection procedure until the vehicle arrives. This can improve efficiency and reduce the duration of inspections. By enabling the location reporting of transportation vehicles to authorities, it would be possible to know the ongoing transportations and what vehicles are close to the inspection site.

A simple mobile device with Internet connection could be used by the inspector to retrieve eFTI, enabling the preparation beforehand. Additionally inspectors could submit inspection outcome reports digitally. A system of this nature could also prioritize the selection of vehicles that are transporting important goods, such as flammable material, to ensure the safety and compliance of transportation of this kind. By implementing an inspection assistant with mobile devices, hardware costs can be lower compared to a solution with proprietary devices, as the system could use devices already purchased for another means. Nevertheless, there has to be a focus on information security, as there is a concern regarding the forging of electronic transportation data [30] and also the prevention of unauthorized transportation data access. Additionally, mechanisms have to be implemented to ensure system reliability, specially regarding the location aspect, as it would depend on real-time information.

This dissertation proposes STOP - Secure Transport lOcation Proofs for vehicle inspections, a road transportation vehicle inspection support system. The system aims to have the real-time location position of on-going transportations in order to perform the automated and accurate selection of vehicles for inspection. Additionally, the digitalization of the required information for inspection and certified inspection outcome digital reports are also goals of the system. The document describes the system architecture, implementation and evaluation description of the solution that uses mobile devices and a central server. The roles described in the system cover the entities typically involved in the road transportation of goods and its inspection. The system uses the location from on-board mobile devices to track incoming vehicles to inspection sites and location proofing to digitally certify the occurrence of an inspection. The system procedures allow inspectors to automatically select vehicles for inspection, assisting inspection procedures. The evaluation made to the prototype not only provides insights regarding the feasibility of this type of system but also provides information regarding the location retrieval aspects of different

mobile phone devices.

The rest of this document is organized as follows. Chapter 2 presents the background and related work. Chapter 3 presents the proposed system in detail. Chapter 4 presents implementation details of the system prototype. Chapter 5 presents the experimental evaluation done. Finally Chapter 6 concludes the document with a summary of the contributions and opportunities for future work.

Chapter 2

Background and Related Work

This chapter presents background concepts to support our work, together with the most relevant related works. We were not able to find works with similar purposes, therefore we focused on research about location systems and their use cases, as a means to detail our system requirements and goals. We start in Section 2.1 by presenting location tracking systems and location-dependent mobile applications to describe how present solutions manage location retrieval using both proprietary and mobile devices. In Section 2.2 we present research regarding secure location certification using mobile devices, because we need to use sensitive and certified location data.

2.1 GPS-based Location Systems and Applications

First conceived for military use, the Global Positioning System (GPS) is offered free of charge, it is available worldwide and has been established as the primary Global Navigation Satellite System (GNSS). The GPS allows a tracker device to compute its location on Earth. This system is composed by a set of 31 operational satellites that emit radio signals that a GPS receiver can use to determine its position [5, 16]. The receiver locks to the signal of at least 4 satellites and calculates its position, taking into account the current time and the known coordinates of the satellites. Each GPS satellite continually broadcasts a signal that includes a pseudorandom code known to the receiver and a message that includes the time of transmission of the code and the satellite position at that time. The receiver calculates the time of arrival of a point by time-aligning a self-generated version and the received version of the code. With the time of arrival and time of transmission, the receiver calculates the time of flight that is approximate to the distance between the device and the satellite. The calculations done by the receiver are then converted to latitude, longitude and height coordinates. This procedure is done for a

start position of the receiver and performs best when the device is stationary because changing position while calculating distances to satellites will affect the result.

Most receivers have a track algorithm that uses sets of measurements done at consecutive times to predict the position of the receiver. This improves the position and time accuracy, detects bad measurements and estimates the speed of the device.

2.1.1 Vehicle Location Tracking Systems

A GPS tracker is a device that enables real time position tracking of attached objects [19]. This device continuously retrieves its location by retrieving satellite signals from GPS. Currently transportation companies use this type of devices to keep track of the location and other information of their vehicles. *Fleet management systems* receive and gather data from the devices to present real time information of the vehicles to the users [20, 10]. These solutions allow companies to monitor their fleet, ensuring secure transportation and reporting the delivery to a client as it happens. A proprietary location tracking device is installed onto the vehicle by the solution provider. These devices are often connected to the on-board computer of the vehicle, which allows other information such as fuel consumption to also be collected. The device transmits the collected information through Global System for Mobile Communications (GSM) cellular network to the servers of the provider. The transportation company can access this data typically through a web portal or computer software. With the data collected, these platforms present several insights regarding the usage of the vehicle, not only location tracking. There was no public documentation found regarding these platforms providing a web service to retrieve location tracking information from vehicles.

There are academic proposals of systems with GPS tracking for regulatory scenarios. *T-Box* [28] is a customized tachograph, a device fitted to a vehicle that automatically records its speed and distance, together with the activity of the driver, and it is mandatory in business vehicles in some countries. This work aims to improve digital tachographs mandatory in South Korea. This customized device was designed to record all driving data of the vehicle and remarkable events in a secure storage and transmit the logged information to a central server. The logging mechanism uses the Trusted Platform Module (TPM), a security chip added to the device, to determine the trustworthiness of the software stack and to ensure stored data has not been tampered and is protected. A driver registers the beginning of the trip in a drive check-in session and the T-Box sends a request message to the central server. This message contains data generated by operations of the TPM regarding the correctness of the device. During the drive time, T-Box periodically sends a log packet containing the logged data and the current tick-stamp,

generated by the TPM. Each log packet contains the value of a hash function calculated with the previous log packets sent, ensuring the *integrity* of all log packets. When the drive session ends, the device transmits the final log packet together with the current status of the TPM. This procedure ensures that the central server can confirm if it has received all logged data and check if the device or storage has been tampered between sessions.

Siegel proposed a system for remote monitoring of vehicles [33] using the standardized automotive On-board Diagnostics (OBD) port. The main motivation behind this work is to provide telemetry for governments to implement taxes based on the distance traveled by vehicles, instead of fuel tax, and to provide traffic congestion patterns. This system uses the OBD-II port of a vehicle to collect the relevant data. The first prototype of this system consisted of an OBD-II transceiver, connected to the corresponding port of the vehicle, that communicates through Bluetooth with a mobile device. The mobile device runs an application to store the transmitted data. However, the author considered that Bluetooth was a substantial bottleneck in the data logging process because of the amount and rate of the data transmitted and the battery usage was substantial. The final prototype consisted of an OBD reader with a GPS antenna and GSM modem to connect to a remote server. The module sends the raw data, from the port, and GPS data to the server, who processes and stores it. Additionally, Siegel created a web application with a data visualizer to provide an end-user interface where a user can check the location of the vehicle and other relevant information.

T-Box and Siegel's system have in common the use of a trusted and dedicated location tracking device that connects to a server via GSM.

2.1.2 Use of Location by Mobile Applications

GPS location is widely used across the majority of mobile devices in use today. Two of the most common uses are road navigation and ridesharing [14, 31]. These mobile applications rely on the location reported by devices to guide users to their destination for example. Google Android mobile devices retrieve their position, combining GPS signals with Wi-Fi and cell network signals [4]. Additionally, the smartphones are aided by Inertial Navigation System (INS) sensors, such as gyroscope, accelerometer and magnetometer sensors to increase the accuracy of the retrieved location.

One of the most popular mobile applications used for navigation is Waze[1]. The application provides navigation to the specified destination by taking into account data retrieved from other devices using the application [38]. The application takes advantage of knowing the destination,

[1] `https://www.waze.com/`

route and speed to predict the position of the vehicle in case the GPS signal is weak. Therefore the application always presents the vehicle on the road and is able to continue giving correct indications even in the presence of GPS location errors.

There are also automobile insurance companies proposing usage-based insurance where the device of the driver is used [21]. The driver installs the application in the device and it retrieves location data. This data can be used to calculate speed and be later used to understand what distance the vehicle does or if the driver drives over the speed limit often.

Such navigation applications have also been used in the transportation sector [24]. Every carrier wants to decrease route times and reduce costs with fuel consumption and vehicle maintenance. Therefore it is important to dynamically change routes according to traffic information. The use of a mobile application provides a low cost integration with any road route navigation system through mobile data. The main focus is to present useful information to the driver to achieve the reduced costs goal.

2.1.3 Security

Despite being widely used, GPS is not considered fully secure [27, 25]. A GPS spoofing attack aims to deceive GPS receivers by broadcasting incorrect signals. These are structured to resemble a set of normal GPS signals and they can be modified to cause the receiver to estimate its position where desired by the attacker. Inexpensive GPS spoofing devices are available in the market [17], therefore an attacker can easily purchase such devices. It is then possible to deceive mobile devices running road navigation applications [39], air drones [18], ships [34] or working vehicles [11].

2.2 Location Certification

Location proof, as defined by Saroiu and Wolman, is a mechanism to allow mobile devices to prove their location to applications and services [32]. There have been several systems that allow the creation of location proofs, namely, Saroiu and Wolman's work, APPLAUS, CREPUSCOLO and SureThing.

Saroiu and Wolman

Saroiu and Wolman considered that a component of an existent wireless infrastructure, such as a Wi-Fi Access Point (AP) or a cellular network tower, can issue meta-data which mobile devices can use to prove their location. A device can request a location proof from the infrastructure

and this proof can be sent to applications for verification. Therefore this system is based on a trusted infrastructure.

The scenario implemented takes advantage of beacon frames transmitted by a Wi-Fi AP when announcing its existence. The concept assumes the AP is a trusted witness. The authors suggest the use of APs with a GPS module, where a person places the AP outside of the building to setup the GPS coordinates and then places it in the desired indoor location. However this procedure requires substantial human intervention and it is not practical as most APs do not have this module and are directly placed at the desired location.

The system uses asymmetric cryptography to guarantee authentication and encryption, where each participating node contains a public and a private key. The holder of the private key sends messages encrypted with this key and other nodes use the paired public key of the sender to decrypt the message, authenticating the sender. These messages contain the computed hash value of the rest of the message, which allows the detection of any tampering, and as this value is encrypted with the private key of the sender, it is considered a *digital signature* of the message. Additionally other nodes can encrypt messages with the public key of one node, ensuring that these messages are only decrypted by this node, as it is the only holder of its private key.

The protocol starts when a client device receives a beacon frame from a AP and then sends a proof request containing its public key and the sequence number of the frame. The proof request is signed with the private key of the device. The sequence number prevents *replay attacks*, where requests are repeated or delayed by an attacker, and the signature prevents *integrity attacks*, where the message is tampered. After validating the request, the AP broadcasts a signed location proof containing its public key, the public key of the client, the current timestamp and the latitude and longitude geographical coordinates of the location. The AP does not check if the client received the location proof. Upon receiving the proof, the client signs it and transmits it to the application or service to use, who then decrypts the message with the public key of the client and checks the public keys contained within the content of this location proof. If the proof is validated, the application or service has a guarantee that the client device is at the reported location. This implementation is mainly suitable for indoor locations and the configuration of APs for this purpose is not trivial, because of the need to modify the firmware.

APPLAUS

Zhu and Cao proposed a location proof system called APPLAUS using mobile devices with Bluetooth [40]. A device can prove its location by requesting and receiving location proofs from nearby mobile devices. The system is therefore considered a neighbor-based proofing solution.

The main focus of this work was to create trustful location proofs and guarantee protection of the identity and location of the source, while only using mobile devices.

APPLAUS uses five entities to create and store location proofs, allowing untrusted mobile devices to be used. The *Prover* is the mobile device node who collects proofs from its neighbors. The *Witnesses* are the untrusted mobile device nodes that provide location proofs about Provers. The *Location Proof Server* is an untrusted node that stores submitted proofs. Additionally, the Location Proof Server cannot be relied to store tamper-resistant and reliable proofs. The *Certificate Authority* is an online service that registers the public and private key combination of every mobile node entering the network. The *Verifier* is an authorized third-party user or application who verifies a location proof of a Prover. Any mobile device can be a Prover or Witness.

APPLAUS provides some location privacy by using *pseudonyms* for the Prover and Witnesses to prevent an attacker from tracking specific devices. The Certificate Authority is the only party who knows the mapping between the real identity and pseudonyms of each device. As the Location Proof Server is considered untrusted, the stored proofs only contain pseudonyms. Therefore, if the Location Proof Server is compromised, the attacker cannot find the real source of a location proof.

To prevent two or more nodes to create false proofs between each other, otherwise known as *Collusion*, APPLAUS may ask the Prover to obtain a threshold number of Witness nodes, becoming more difficult for an attacker to have the number of devices asked to successfully create a false proof. However, since the Prover can claim it cannot find more neighbor nodes, the Location Proof Server can check that claim since it has information about the number of nodes in that particular time and location. Additionally, with this information, when a pair of location proofs is uploaded between the two same pseudonyms, the Location Proof Server can check if there are other concurrent and co-located proofs from other nodes that have not had any interaction with these two nodes, considering them suspicious of colluding, therefore an appropriate trust level is assigned to the submitted proofs. Furthermore, since multiple pseudonyms can be used by the same identity, the Certification Authority can attribute a trust level to each real identity based on the location proofs given.

The Prover triggers the protocol by broadcasting a location proof request through the Bluetooth interface. A witness accepts to create a proof, signs and sends it to the Prover who then sends it the Location Proof Server. An authorized Verifier can later query the Certificate Authority to retrieve location proofs of a specific Prover. The Certificate Authority will then proceed to convert the real identity of the Prover to the corresponding pseudonyms and retrieve

the desired location proofs from the Location Proof Server.

APPLAUS allows for a system where the Provers have their location proven by other nearby devices and at the same time their identity is protected. By using only Bluetooth, a location is only collected when there are devices in short range and attacks are only detected after analysis of submitted proofs which can become more difficult in a larger scale scenario.

CREPUSCOLO

Canlar et al. [9] created the CREPUSCOLO system to address both the *neighbor-based* type of proofing solutions, found in APPLAUS, and the *infrastructure-based* type, found in Saroiu and Wolman's work. CREPUSCOLO also extends the concept of proof by including elements such as photos that are characterized as being "indisputable".

The system consists of the same entities as in APPLAUS with the addition of the *Token Provider* entity, a trusted entity placed at a strategic location, with the main task of issuing tokens to confirm proofs acquired from witnesses. A Prover broadcasts a location claim, similar to a proof request, where it states its location and requests a location proof. A location proof created by a Witness contains pseudonyms of the Witness and Prover, a nonce, which is a pseudo-random number previously created for the corresponding location claim, the timestamp and location of the Witness and a hash value of the previous components, signed with the private key of the Witness. A token created by the Token Provider differs from a proof as it contains the known identity of the Token Provider instead of a pseudonym of a Witness and additionally a *proof* field signed with the public key of the Token Provider. A picture taken by a surveillance camera is an example of a proof field, to indisputably prove that the Prover was at its claimed location. All location proofs and tokens are encrypted with the public key of the Location Server.

The procedure to create location proofs is similar to the one in APPLAUS with the addition of the possibility of a proof being created by a Token Provider, a trusted entity. After a Prover has broadcast a location claim through Bluetooth, a Token Provider can send a generated Token which indisputably proves that the Prover was at its claimed location if the *proof* field was created physically together with the Prover. The Token contains the current location of the Prover and it can not be reconstructed at another location, since it contains the signature of a Token Provider, whose identity is publicly known.

The authors emphasized the *wormhole attack*. Two dishonest provers, P1 and P2, at different locations aim to make a Verifier believe that both are at the same location. Prover P1 forwards its location claim message to the other who then broadcasts it at its location. Honest witnesses

will create proofs stating they have sensed P1 at the location of P2. This is possible since the replayed location claim contains the pseudonym of P1. Then P2 sends the collected location proofs to the Location Server. Canlar et al. claim that neighbor-based solutions only consider simple collusion attacks where a dishonest Witness is evolved. In CREPUSCOLO, a Verifier mitigates the wormhole attack by checking location proofs submitted by P1 and the associated tokens. The *proof* field of these tokens, a picture for example, will show that P1 was not in fact at its claimed location and it was not involved in the creation of these tokens, therefore detecting the attack. Again, this relies on the assumption that the token is indisputable as evidence.

SureThing

Ferreira and Pardal [15] aim to provide correct location proofs to other applications and services, indoors or outdoors, improving the APPLAUS and CREPUSCOLO works. SureThing uses multiple entities similar to the ones presented in the two previous works, *Prover*, *Witness*, *Verifier* and *Certification Authority*, and it also uses geographical coordinates, Wi-Fi fingerprinting and Bluetooth beacons as location proof techniques. However there are some differences regarding some of the used entities: the Witness similar to the entity found in APPLAUS and CREPUSCOLO is called *Mobile Witness*; the Witness similar to the Token Provider entity in CREPUSCOLO trusted by the system is called *Master Witness*; the *Verifier* is the central entity of the system who validates all submitted location proofs.

A location proof contains the identifiers of the Prover and the Witness, the location of both entities, a nonce sent previously by the Verifier and the digital signature of the Witness. The Witness determines its own position by obtaining geographical information from GPS and Android Network Location Provider (ANLP)[2], Wi-Fi fingerprints and Bluetooth beacons.

SureThing proposed two methods for Collusion Avoidance. The *Witness Redundancy* mechanism forces the Prover to gather proofs for more than one Witness and chooses the number of witnesses according to the level of service possible and each proof has a different value associated with it. *Witness Decay* ensures that if a Prover is getting proofs from the same Witness, its proofs gradually become less valuable and the Verifier will not validate the location if the Prover can not gather proofs with enough value.

OTIT

Khan et al. proposed OTIT [22], a model for designing secure location provenance, the chronological history of the location of users. The authors defined seven features and requirements,

[2] `https://developer.android.com/guide/topics/location/strategies.html`

which they consider to be necessary for designing any secure location provenance scheme. We consider the chronological order preserving and tamper evident requirements of the model to be very important.

The authors considered six approaches to achieve the established requirements. Not all requirements were fulfilled by one single approach, however all approaches cover the chronological, order preserving, verifiable and tamper evident requirements. Performance evaluation was performed and results show that a block hash chain performs better. In such approach, a proof contains the hash value of the previous proof, producing a typical hash chain, and also a unique initialization vector. These vectors are also maintained in another hash chain of initialization vectors. Despite the performance of this approach, the authors consider a RSA chain, based on a RSA accumulator [7, 6, 8], the better choice, as it achieves more requirements than the other approaches with the disadvantage of taking more time.

2.3 Summary

The location retrieval and accuracy aspect is the most important for our purpose of creating a road transportation inspection system with mobile devices. With the presented works we conclude that, despite the reported vulnerabilities of GPS, there still a lot of systems using it and GPS can be considered reliable. However, as we propose an inspection procedure, it must be verifiable. The presented location certification works show us how interaction between devices can be made to exchange certified location information and also provide a location chronological order. This certification can be used to create certified inspection reports, replacing the need of paper reports, therefore digitizing the process of conducting an inspection. Additionally, records should be tamper-proof for future verification.

Chapter 3

STOP

In this chapter, we present a road transportation inspection support solution named STOP standing for Secure Transport lOcation Proofs. The main goal of STOP is to provide and register the required information to inspectors and drivers, by using mobile devices, assisting every entity involved. The location retrieval aspect of the solution is important as it needs the location of transportation vehicles close to an inspection site and also as a guarantee of the occurrence of an inspection. Section 3.1 describes the goals, participants, procedures and security policies of the system. Section 3.2 presents the components, interactions and security mechanisms. Section 3.4 summarizes the architecture of the system.

3.1 Inspection Process

The STOP system assists the road transportation inspection process. We have established the goals and security policies required in order to present a viable solution. Additionally, we have identified the entities involved in this process and how they may interact with STOP.

3.1.1 Goals

As discussed in Chapter 1, there is strong focus in implementing the digitalization of this process due to its benefits. Therefore we first consider that electronically submitting the required documentation and later on digitally presenting such information for inspection procedures is very important. Secondly, the automation of the selection of vehicles for inspection can assist the inspection process. This will allow inspectors to prepare the inspection procedure beforehand, reducing its duration, and also enable the system to retrieve the required documentation automatically without any user input. This automation needs to be accurate, as the as-is selection process is done with an inspector ordering a vehicle to stop at the inspection site, where there is

a guarantee that the selected vehicle is at the site and was ordered to stop for inspection. If the system selects vehicles for inspection that cannot reach the location, it will be very problematic. Therefore the location reporting of vehicles transporting goods is required for the selection. Lastly, inspectors are required to report the outcome of inspections. Typically these reports are done on paper and involve filling information regarding the inspected vehicle and the inspection outcome. By digitally creating this report with prefilled information and only requesting the inspector to register inspection outcome information, the report procedure is faster and it can be submitted immediately. Additionally, this report needs to be certified to indisputably prove that the inspection occurred and the report was created by the authorized inspector.

The related works presented in Chapter 2 fulfill some of the goals but not all, and also they were not design to cover this inspection procedure. The location systems and applications presented in Section 2.1 enable the location tracking of vehicles with the usage of Global Positioning System (GPS) and proprietary or mobile devices. However we have shown that, despite being reliable, there are vulnerabilities in GPS and we need to certificate the existence of inspections. By using the location proofing mechanisms presented in Section 2.2, we can therefore certify the presence of devices at the inspection site. This will prove that both the driver of the vehicle and the inspector were at the inspection site at the same time. Additionally, we consider that using mobile devices will make hardware costs lower compared to a solution with proprietary devices, as the system could use devices already purchased for another means. However it is important to assess if the usage of mobile devices is viable for our scenario. This assessment includes verifying the location reporting accuracy, the selection of vehicles for inspection and the location proofing procedure in transportation and inspection scenarios.

In sum, these are the goals and requirements of STOP:

- Digitalization of the required information for inspection procedures;
- Location reporting of on-going transportations;
- Automated and accurate selection of vehicles for inspection;
- Certified inspection outcome digital reports;

3.1.2 Participants

We consider that a transportation process starts with a company registering the description of the freight, known as electronic Freight Transport Information (eFTI), with the competent authorities. A carrier or the company itself performs the transportation, which can be inspected

by authorities at any point of the route. The process is finished when the goods are delivered to the reported receiver.

Therefore the system considers the following roles: *Authority*, *Inspector*, *Company* and *Carrier*. The *Authority* represents the entity responsible for inspecting goods in a country. The *Inspector* represents the authorized person conducting an inspection. The *Company* represents the enterprise sending goods to another enterprise or individual. The *Carrier* represents the entity transporting the reported goods.

Authority

The Authority sets up the STOP system by defining the transportation details that each company must submit beforehand and how the system must select vehicles for inspection. These details are defined according to the legislation of the country. If the government already has implemented a eFTI submission system, we consider that the STOP system can be integrated with such system to retrieve submitted eFTI through the usage of a web service. The Authority sets the configuration of the overall system. For example, it defines the maximum number of inspections per transportation for the selection of vehicles for inspection.

The STOP system considers that user authentication is delegated to a trusted authentication system maintained by the Authority, as many governments already have such systems implemented. The Authority associates specific users to the STOP roles, enabling the correct user authorization.

Inspector

The Inspector is a trusted user by the Authority, authorized to conduct inspections on heavy road vehicles transporting goods. This user creates in the system *checkpoints*, locations where inspections will take place at. The user defines a perimeter from inspection sites called *inspection selection range*. All vehicles inside that perimeter are considered for random selection. After creating the checkpoint, the inspector is allowed to request the selection of a vehicle for inspection. The inspector also registers the outcome of such inspection. Additionally, the user can report non-compliant users, such as drivers that have not stopped for inspection or are not using the STOP system.

Company and Carrier

The Company registers the upcoming freight transportation in the STOP system or in an external authority system, if available. It must register the details defined by the Authority in order

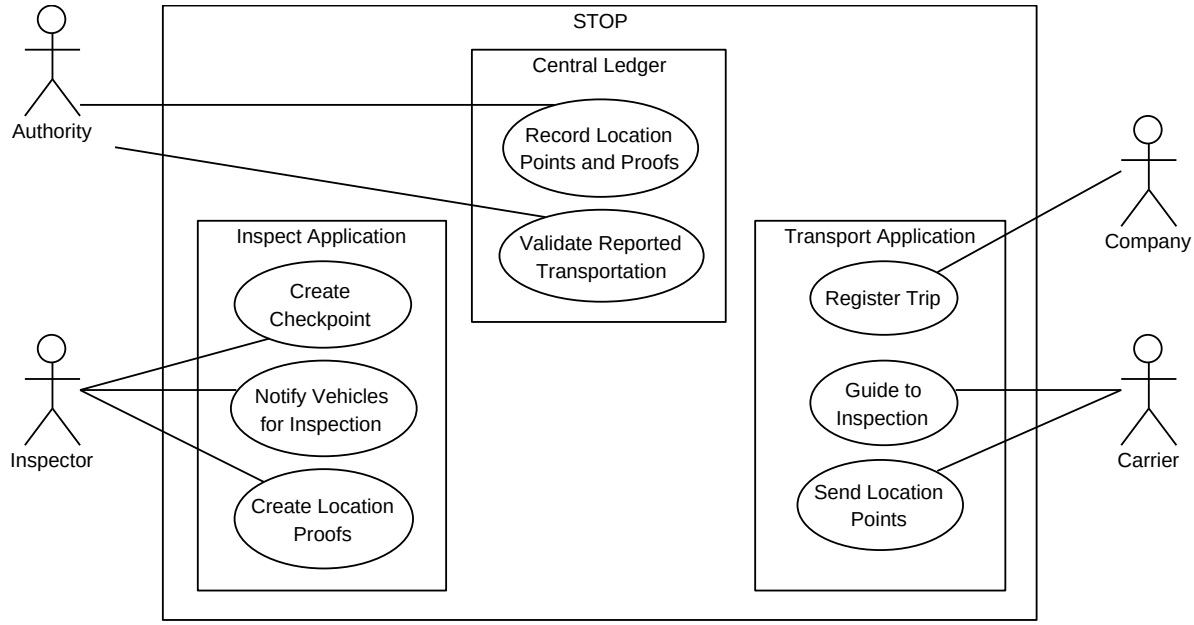

Figure 3.1: STOP Entities and Use Cases

to be legally compliant.

The Carrier is represented by the driver or drivers that drive the vehicle transporting the reported goods. This user has the following activities in the system:

- Reporting the initialization of the transportation;
- Acknowledging the inspection selection and driving the vehicle to the *checkpoint*;
- Reporting the end of the transportation.

The Carrier and Company roles may be played by the same entity, as some companies perform the transportation of their goods and do not subcontract an external company for that purpose.

3.1.3 Operation

We have defined three components for the STOP system. The *Central Ledger* is the main component responsible for registering information and coordinating the system procedures and it is used by the Authority. The *STOP Transport* mobile application is used by the Company and Carrier for registering and performing transportation, respectively. Inspectors use the *STOP Inspect* mobile application for the inspection procedures. These components are further detailed in Section 3.2. Figure 3.1 shows the interactions of each entity with the components of the system.

We now describe the system operation to showcase how the components interact with each other and with the users. For this description, we consider that trip registration is done in the STOP system. Therefore the transportation information parameters were defined in the Central Ledger setup.

Transportation

The Company registers an upcoming transportation in the Transport application with the parameters previously defined by the Authority. By default the system requests the following parameters for registration:

- The fiscal numbers of the entities sending and receiving the goods;
- The location coordinates of the locations where the goods are loaded into the vehicle and where the goods are delivered;
- The description of the freight, indicating the quantity and weight of products;
- The license plate of the vehicle transporting the goods.

After the vehicle is ready, the driver requests the initialization of the transportation through the User Interface (UI) of the Transport application. The Central Ledger authorizes such action and the driver begins the trip.

The on-board device will retrieve and send its location at the system-defined location retrieval rate. The location tracking is based on the works presented in Section 2.1. During the trip, the Transport application constantly checks if the vehicle was selected for inspection and, if so, notifies the driver to drive to the inspection site. The driver acknowledges the end of transportation in the UI.

Inspection

An inspector arrives at the site where inspections are going to occur. The inspector starts the Inspect application, logs in and creates a *checkpoint* in the application. The inspector defines the *inspection selection range* as previously mentioned. The action of creating a checkpoint will send the location coordinates of the inspection site and range value to the Central Ledger. This location is now registered as an active checkpoint in the STOP system.

When the inspector is ready to conduct inspections, he or she will request an inspection in the application. It will send a request to the Central Ledger, which will return the submitted transportation information regarding the selected vehicle. If there is no vehicle eligible for

inspection according to the selection parameters, then the inspector will try again later on. Meanwhile the on-board device of the selected vehicle retrieves the checkpoint information, which is presented to the driver. The inspector will analyze the eFTI while the vehicle arrives and the Inspect application displays the current location of the vehicle.

The driver will steer the vehicle to the checkpoint and will notify the application that the vehicle has arrived to the checkpoint. Both devices will now start the inspection certification procedure, based on the works presented in Section 2.2. The Transport device will communicate with the Inspect device and the inspector will be notified by the application to start conducting the inspection. Upon finishing the procedure, the inspector will register any relevant information in form of text or picture and approve the inspection. The Inspect device generates an *Inspector Location Proof* (ILP) and sends it to the Transport device and to the Central Ledger. When receiving the ILP, the Transport device creates a location proof, adding it to the location chain of the trip. This location proof is sent to the Central Ledger and the application notifies the driver to resume the trip, ending the inspection process. The communication protocol between the two devices is specified in detail in Section 3.3.3.

3.2 Solution Architecture

The STOP system is structured in three tiers: Presentation, Logic and Data tiers, as represented in Figure 3.2. This allows a separation of concerns for each tier, and the integration of new components such as different storage systems and user interfaces. The main components of the system are the *Central Ledger*, the *STOP Transport* and *STOP Inspect* mobile applications.

3.2.1 Central Ledger

The *Central Ledger* is a central server that receives data of transportations and inspections. As shown by Figure 3.2, all communication with the Central Ledger is done through the Representational State Transfer (REST)ful Application Programming Interface (API) web service provided by the server.

The interface provides operations that can be divided in two categories, *trip* and *inspection*, as they are used by the Transport and Inspect applications respectively. Trip operations cover the create, initialize, locate, show and end trip actions, as well as add the proof created after an inspection. The following objects are used:

- *Trip* - Contains the freight and trip information;
- *Location Chain Item* - Contains location data of the vehicle at a time point, explained in

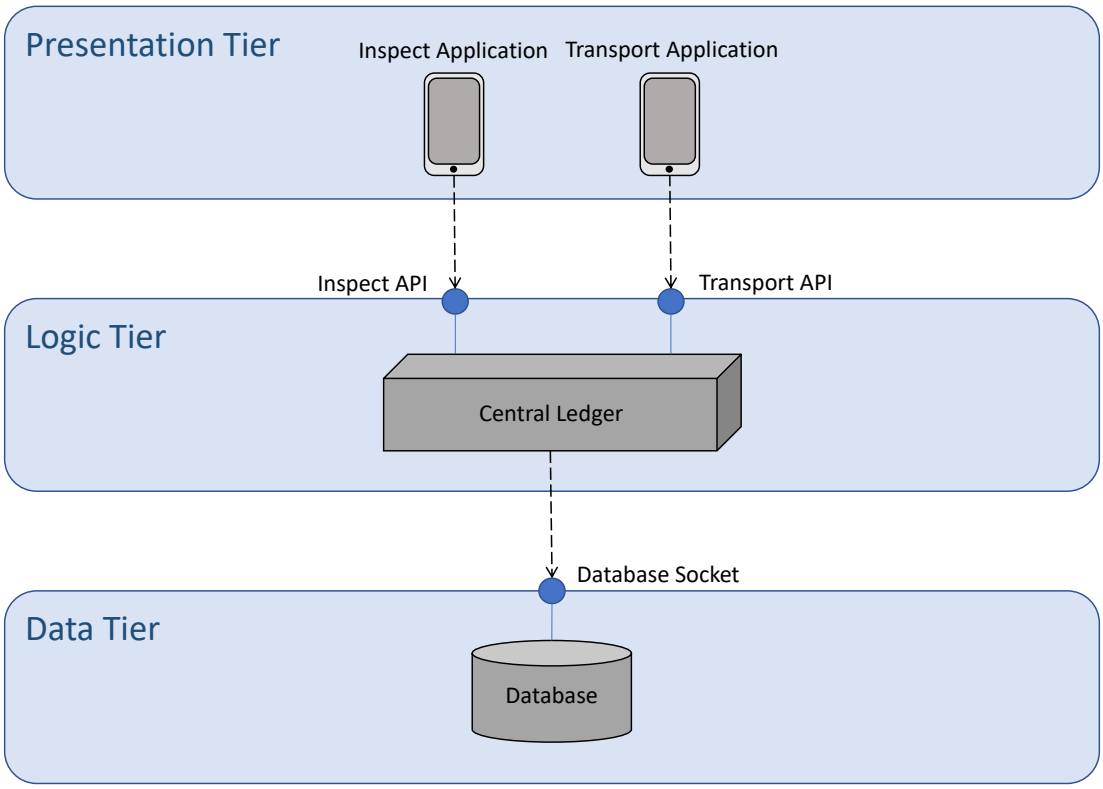

Figure 3.2: STOP Architecture

detail in Section 3.2.2.

Inspect operations allow creating, showing and ending checkpoints and inspections. The following objects are used:

- *Checkpoint* - Contains location information regarding the inspection area and inspections performed in this site;

- *Inspection* - Contains information regarding the selected vehicle and the proof that an inspection was conducted.

A detailed description of the interface was done in *OpenAPI*[1] description language format to have a first overview of the future implementation.

In order to keep information persistent, we consider that such functionality must be delegated to a dedicated Database Management System (DBMS). This reduces the complexity of the Central Ledger and several DBMSs already have mechanisms for database replication. Additionally, this option can enable multiple Central Ledger instances for increased availability and load balancing as a DBMS has concurrency control mechanisms.

[1]The OpenAPI format was made popular by the Swagger tool

3.2.2 Transport Application

The *STOP Transport* application is a mobile application, running on a mobile device inside of the vehicle transporting the reported goods.

The application presents an UI where the users with the Company and Carrier roles perform the activities described in Section 3.1.2. The device running this application must have an active Internet connection during the transportation process. The application uses the Central Ledger RESTful API to:

- Register trip information and receive registration approval;
- Request initialization of the transportation;
- Send location information of the device;
- Check if the vehicle was selected for inspection;
- Send location proofs generated by the device of the inspector;
- Report the end of transportation.

The application is responsible for building a valid *Location Chain* that can later be verified. The chain represents the location positions of the vehicle during the transportation in chronological order. A location chain item is either a *Location Point* or *Location Proof*, as illustrated in Figure 3.3. Both contain the signature of the previous location item. It enables the verification of the sequence of items in the location chain of the trip. By checking the previous signature in one location item, it is possible to assess if the previous items were modified or are missing, proving protection against record tampering. Every item is stored in the location chain instance of the Transport device and they are sent to the Central Ledger. The main difference between the two type of items is the source of the location position.

A location point contains the geographic coordinates retrieved by the Transport device at a time point of the trip. A location proof contains the geographic and time coordinates retrieved by an Inspect device at a checkpoint. It is intended to prove that the vehicle was inspected so it is digitally signed by an authorized inspector. In an inspection scenario, the Transport device receives a proof from the Inspect device, which is used for the location proof. This interaction is detailed in Section 3.1.3.

When the transportation ends, the Central Ledger has the complete *Location Chain* of the trip, further explained in Section 3.2.5.

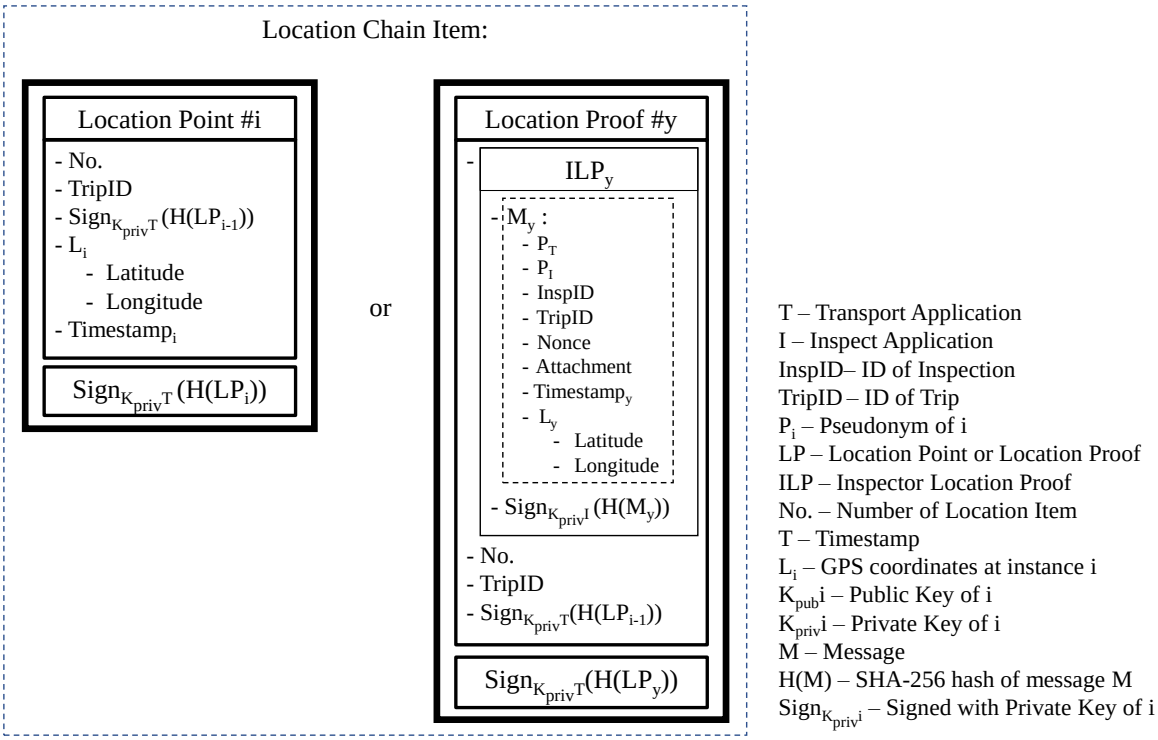

Figure 3.3: Types of Location Chain Item

3.2.3 Inspect Application

The *STOP Inspect* application is a mobile application used by a user with the Inspector role. The inspector runs this application on a mobile device at a location where the user considers suitable to conduct inspections on heavy road vehicles.

The UI of the application allows the user to conduct the activities described in Section 3.1.2. The application uses the Central Ledger RESTful API to:

- Register a *checkpoint*;
- Request the selection of a vehicle for inspection and retrieve information regarding the selected vehicle and the transportation;
- Submit the outcome of an inspection and the associated generated location proof;
- Report a non-compliant vehicle;
- Report leaving the checkpoint.

After a vehicle is selected, the application presents the respective eFTI for the inspector to analyze and the real-time location of the vehicle while it reaches the checkpoint.

The application communicates with the device inside of an inspected vehicle via short-range communication. This interaction guarantees that the correct vehicle is inspected. An *Inspector Location Proof* (ILP) is generated at the end of the inspection procedure, also shown in

Figure 3.3. The proof contains pseudonyms of the *Transport* and *Inspect* devices, trip and inspection identifiers and a random nonce generated by the Central Ledger for the occasion. The proof also contains an attachment parameter where an inspector may add text or a picture related to the inspection outcome. This proof can replace any paper report done by the inspector, as it proves the inspection was conducted.

3.2.4 Interactions

Figure 3.4 shows the interactions between the STOP components during the transportation and inspection procedures. The numbers show the sequence of the actions. The Transport application communicates with the *remote* Central Ledger to report locations. The Transport and Inspect applications communicate in *proximity*. The Inspect application also communicates with the *remote* Central Ledger.

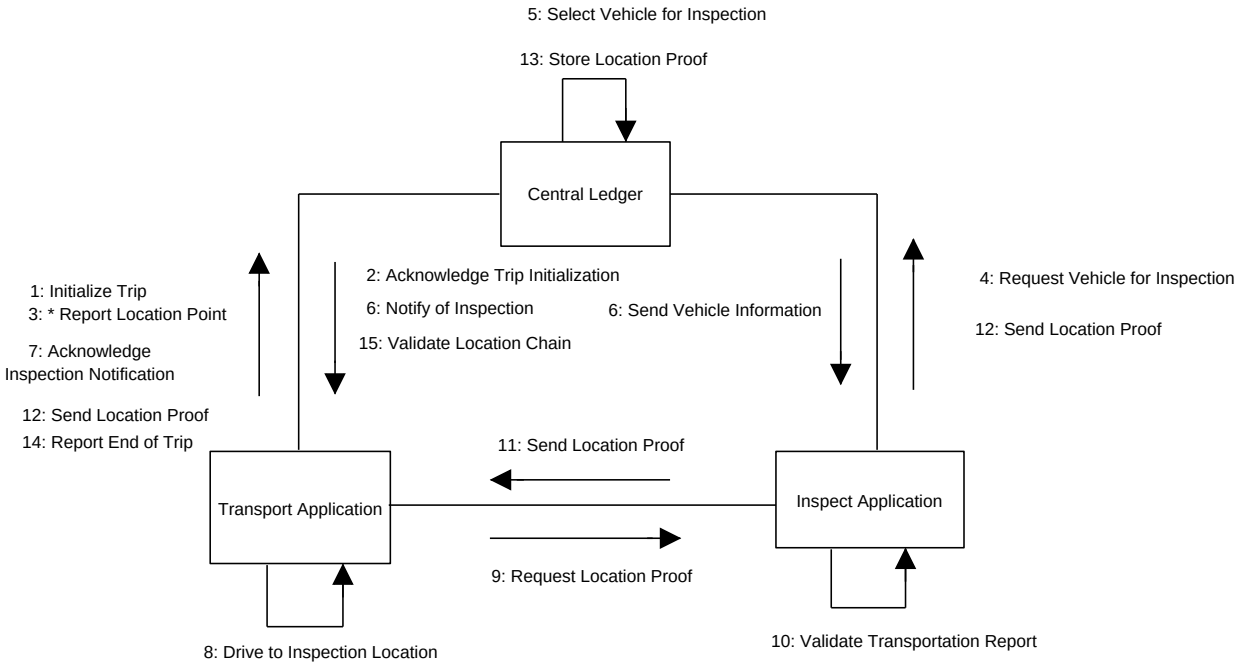

Figure 3.4: STOP Choreography

The *remote* communication between the applications and the Central Ledger is done through the provided RESTful API web service via cellular network. This API uses standard HTTP over TLS[2] to protect the messages [23].

The mobile applications keep persistent records of the objects and are able to submit them to the Central Ledger as soon as the connection is available, to tolerate momentary communication faults. However the system requires that devices are able to communicate frequently with the

[2] `https://tools.ietf.org/html/rfc8446`

Central Ledger, as they have to be informed of inspections and the ledger requires timely location information.

The Transport and Inspect applications communicate in *proximity*, through short-range communication at an inspection site. This interaction enables the certification of the occurrence of an inspection. The procedure is similar to the neighbor-based location proofing works presented in Section 2.2, where the Transport device will broadcast a proof request in the local network containing the information received from the Central Ledger. The short-range communication will ensure that the two devices are at the location. Additionally the information received by the two devices from the Central Ledger will ensure that the communication will be encrypted and protected against tampering and eavesdropping. The devices receive the public key certificates of the counterpart and encrypt and sign the short-range communication. When the inspection is concluded, the Inspect device then creates and sends the location proof to the Transport device, containing the information known to these two devices.

3.2.5 Location Chain

The *Location Chain* is the object used by the system to register the events of a transportation. It represents the trajectory made by the vehicle and the inspections performed during the transportation. Figure 3.5 shows an example of a Location Chain instance.

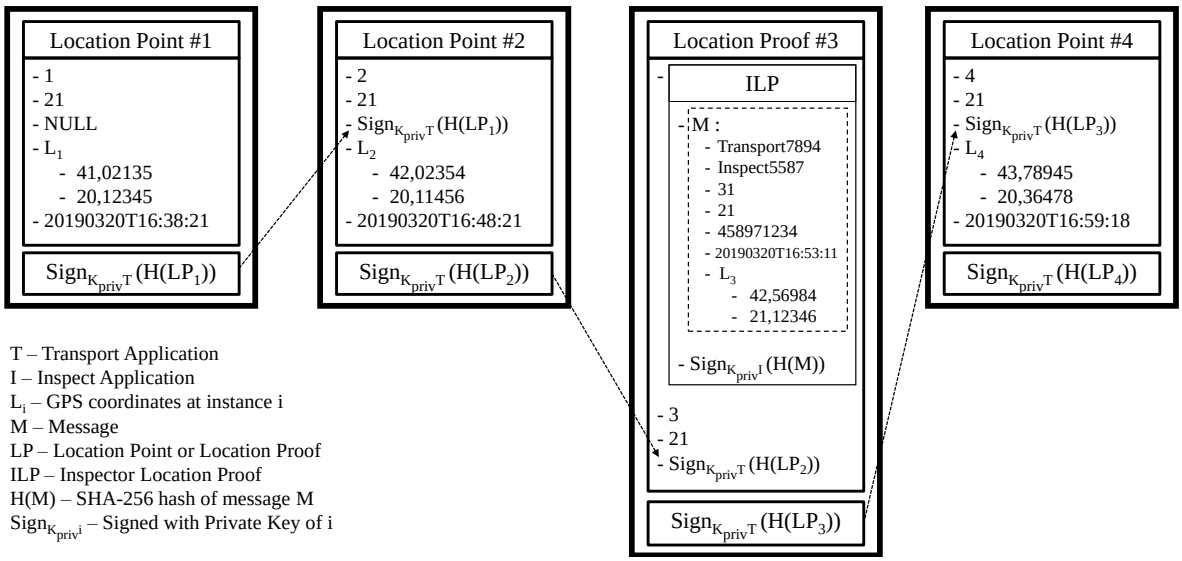

Figure 3.5: Location Chain Example

As previously described and showcased by Figure 3.5, a Location Chain is composed of items that can be Location Points or Location Proofs. The example shown contains three location points and one location proof. Every Location Chain item is created by the Transport device, who is responsible for guaranteeing that the item is signed and contains the correct order number,

the trip identifier and the signature of the hash value of the previous item. The source of the location details, such as location coordinates and timestamp, is the Transport or Inspect device, depending if the item is a Location Point or Proof, respectively. The combination of location points and proofs allows the reported trajectory to be validated by the Authority user, taking into account the trusted location details from Inspect devices.

The arrows in the figure represent how each location item is linked to the previous ones by containing the signature of the previous item. When one item is received by the Central Ledger, the comparison of the previous signature in this item with the signature of the last validated item will check if the Transport device and Central Ledger have the same instance of the location chain in the exact same order. This is specially important when the last location chain item is sent when the transportation is over. If all items were created correctly and received by the Central Ledger, the Central Ledger will only need to perform this validation. Additionally the signatures will guarantee that any tampering made by an attacker will be detected as the signatures are made with the private key of the Transport device.

3.3 Security Architecture

As the system uses sensitive information and has certified procedures, we have defined security requirements, specified an attacker model, and designed corresponding security mechanisms.

3.3.1 Policy

We consider that the system fully trusts Inspect users and the location information reported by the Carrier users is trusted enough to be used for inspection selection. Additionally we consider that the users are responsible for not giving out the credentials to another person.

Any submitted transportation information is only accessed by the Company user that registers it, the Carrier user responsible for performing the transportation, the Inspect user that conducts an inspection on such transportation and the Authority user. The same applies to the location information of the trip and inspection outcome reports. No other Company or Carrier user is able to assess information of other users.

At inspection sites, any information exchanged via short-range communication must not identify the Carrier or the Inspect users, despite the system having security mechanisms to prevent any eavesdropping or tampering. We consider the communication between the applications and the Central Ledger is secure, therefore sensitive transportation and inspection information is exchanged.

3.3.2 Attacker Model

For our system, we have considered two types of attackers: an *authenticated user* who wants to deceive the system and an *unauthenticated user* who wants to attack the system. We consider the first type as attacker A and the second type as attacker B, with the following intentions:

- A1 Report false location point;
- A2 Create false location proofs;
- A3 Turn off Transport device.

- B1 Impersonate an Inspector device at a checkpoint;
- B2 Impersonate an Transport device at a checkpoint;
- B3 Intercept communication between Transport and Inspect devices;
- B4 Intercept communication between devices and the central ledger.

3.3.3 Mechanisms

We now present the design of the security protocols for the STOP system and how the system prevents and mitigates attacks, considering the security requirements and attacker model presented.

Cryptographic Keys and Functions

Each user generates a pair of *RSA* public and private cryptographic keys in the device for asymmetric encryption and for signature. The public key is stored in the Central Ledger for encrypted communication and signature validation. The Central Ledger acts, effectively, as a Certification Authority (CA) for the public keys.

Every message or object requires a digital signature to be considered authentic. A signature is computed by calculating the hash value of the object with the *SHA-256* algorithm. It is then encrypted with the RSA private key of the device that created the message. The signature is validated by comparing a recomputed hash of the received object with the hash value decrypted using the corresponding public key of the sender.

Additionally, for each inspection, the Central Ledger generates random *pseudonyms* for the Carrier and Inspector users, used for short-range communication as transient device names. Each pseudonym has its unique key pair generated by the device using such pseudonym and the public key is certified by the Central Ledger.

Communication protocol between mobile applications

Figure 3.6 shows the interaction when a vehicle is selected for inspection. Upon receiving the details of the upcoming inspection, the Inspect and Transport devices obtain the public key certificate of the other device from the Central Ledger, along with a nonce and a pseudonym for each device. This is necessary to encrypt the communication between these devices and to prevent replay, eavesdropping and tampering attacks.

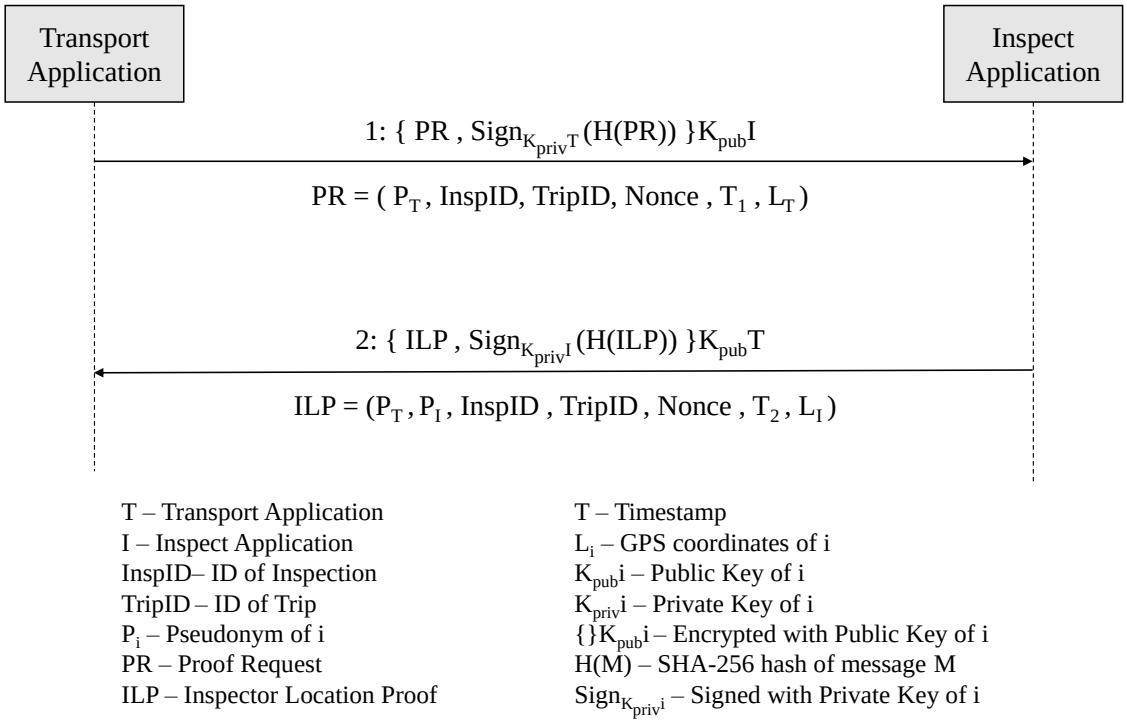

Figure 3.6: Inspection Protocol

When the vehicle arrives to the checkpoint, the Transport application starts searching for the device announcing as device name the pseudonym of the inspector. When found, the Transport device starts the communication by broadcasting a proof request. The broadcast message is encrypted with the public key of the inspector to guarantee that this message is only decrypted by the Inspect device. The broadcast message contains the proof request, represented in Figure 3.6 as PR, and the signature of the hash of the proof request, made with the private key of the Carrier. This guarantees that the proof request was created by the Carrier. The proof request contains pseudonyms of the devices, the ID's of the inspection and trip, the nonce generated by the Central Ledger, the timestamp of the Transport device and its GPS coordinates.

When the Inspect device receives a message from a device with the pseudonym of the Transport device, it validates if it is a correct proof request from the selected Carrier user and, if correct, notifies the inspector to conduct the inspection. If the device has received an invalid

proof request, it closes the communication socket. When the inspection is done, the outcome is reported in a message containing the proof, represented in Figure 3.6 as ILP, signed by the inspector. The message is encrypted with the public key of the Carrier. The message is then sent through the established socket to the Transport device. The Inspect device additionally sends a copy of the ILP to the Central Ledger.

The Transport device receives the Inspector Location Proof, decrypts and validates it. The device creates a location proof with the ILP as described in Section 3.2.2 and shown in Figure 3.3. It is then sent to the Central Ledger.

If the Transport device did not receive the proof after successfully sending a proof request, it will request the Central Ledger to produce a new nonce and pseudonym for that inspection. Duplicate messages with the same nonce, pseudonyms and identifiers are rejected as possible replay attacks. As the Inspect device is constantly retrieving information regarding the selected vehicle to present its current location to the inspector, the device will retrieve the new nonce and pseudonym.

Attack Mitigation and Prevention

We have implemented mitigation and prevention mechanisms considering the attacker model presented in Section 3.3.2. Regarding the *malicious authenticated user* attacks (attacker A): if the Carrier reports false location points (A1) or turns off the device (A3), the inspector will report detected vehicles that are not complying with the regulations and the company will be held accountable; if the Carrier or someone else tampers with proof data (A2), the digital signature can be used to detect the change; if the attacker tries to use an alternative key pair to produce false signatures, he cannot replace the legitimate key certified by the Central Ledger. Therefore, all the intentions of attacker A are stopped.

Regarding the *malicious unauthenticated attacks* (attacker B): the messages exchanged over short-range communication at the checkpoint are protected with confidentiality and integrity mechanisms, as described in section 3.3.3, preventing the interception (B3). If the attacker tries to impersonate a Carrier (B1) or an Inspector (B2) users the attacker is not able to decrypt received messages and send messages with the correct signature. The use of new pseudonyms and nonce values allows the devices to detect the incorrect reuse and reject the messages if the attacker tries to replay old messages. Finally, if the attacker tries to intercept the communication between devices and the Central Ledger (B4), it is protected by the industry standard HTTP over TLS. The server certificate is pinned by the applications. We can conclude that the malicious intentions of attacker B are also stopped.

3.4 Summary

STOP supports the road transportation inspection process, considering the Authority, Inspector, Company and Carrier roles. The system is composed of a central server and two mobile applications. The two mobile applications, Transport and Inspect, retrieve and submit information using the REST API of the Central Ledger. Companies are able to report upcoming transportations in the system. The Transport application constantly reports the location of the device in the form of location points, while the Inspect application is able to request the selection of vehicles for inspection. Upon an inspection procedure, the two applications exchange information through short-range communication in order to create a location proof, which certifies the occurrence of an inspection. The inspector is able to report the inspection outcome in this location proof, fulfilling the system goal of certified inspection reports.

The architecture considers that, upon implementation, the Authority chooses what information is submitted before each transportation and how the system selects vehicles for inspection. Additionally the Authority can configure the maximum number of inspections per transportation. Upon usage of the system, an inspector indicates the inspection selection range of each checkpoint.

The system has security mechanisms in order to prevent and mitigate malicious intents. As the system is owned by the Authority, it can audit the system and validate every procedure. We have considered the works presented in Chapter 2 in order to design the usage of both location tracking and proofing. We consider that location tracking enables the automated selection of vehicles for inspection and location proofing certificates the occurrence of an inspection.

Chapter 4

Implementation

A STOP prototype was implemented in order to assess and evaluate the feasibility and performance of the proposed system. The prototype implements the STOP architecture previously described with two mobile applications, a server and database system, as illustrated in Figure 4.1. Software was developed for both the Presentation and Logic tiers and an already existent database cluster was used to keep persistent records of the system, complementing the Data tier. Bluetooth was chosen for short-range communication.

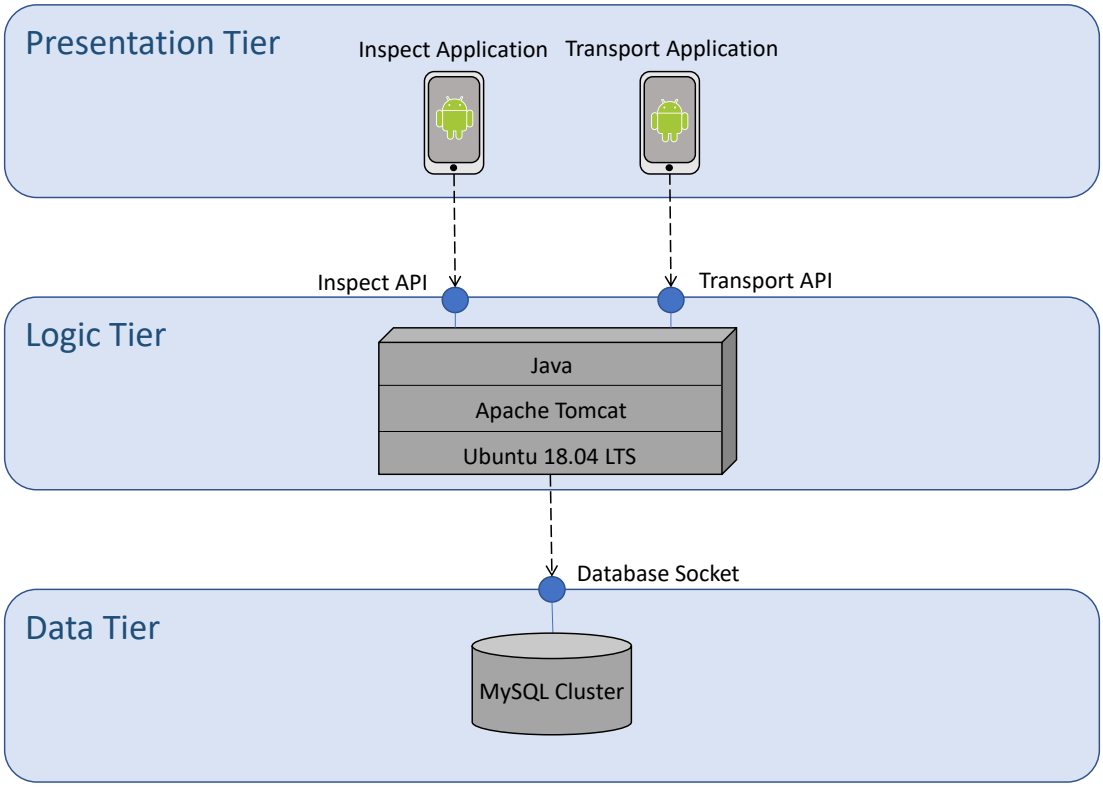

Figure 4.1: STOP Prototype Architecture

4.1 Software Structure

The software developed for the STOP Prototype contains one shared library for the mobile applications and Central Ledger and one library for mobile applications, as shown in Figure 4.2. The STOP Shared Library contains methods for encryption and objects used by all entities such as *Location Proof* and *Trip* objects. The STOP App Library contains code for actions such as location retrieval or Bluetooth communication. All code was written in Java and the build automation tool used was *Gradle*[1].

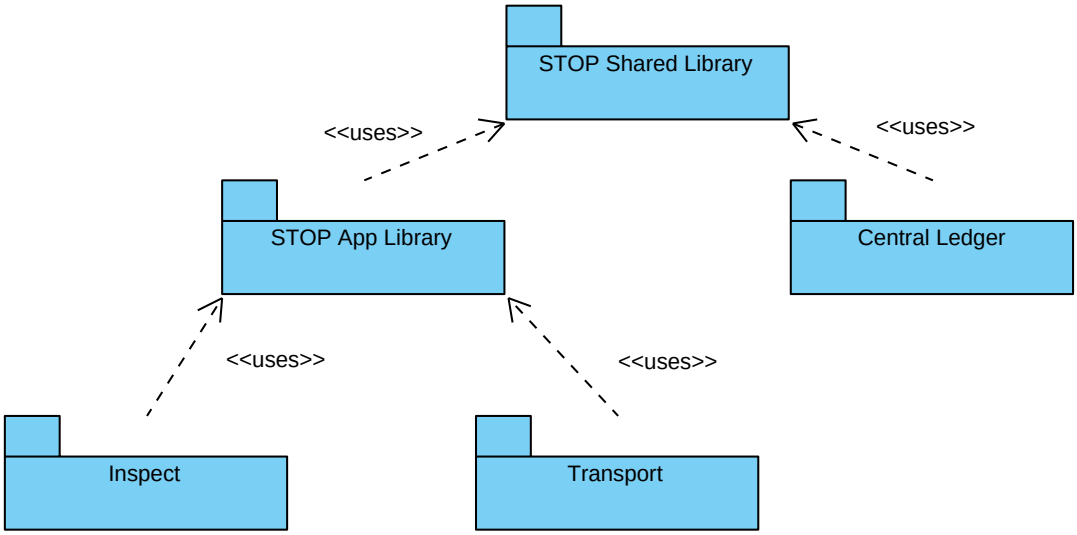

Figure 4.2: STOP Project Structure

4.2 Mobile Applications

Both *Inspect* and *Transport* applications were implemented for the Android platform. We chose this platform as it represents most of the smartphone market[2] and there is considerable documentation regarding the implementation of the desired functionalities. As support libraries, the Gson library was used to serialize Java objects to JSON objects and vice-versa, for communication with the Central Ledger, and the OsMoDroid library was used to display maps from OpenStreetMap in both applications. Upon developing the mobile applications, we discovered that, at operating system level, Android devices do not support encrypted Bluetooth communication with unpaired devices[3]. Originally, when designing the architecture of the system, we considered that this was possible to do. Therefore, as pairing devices requires user interaction and additional time, we successfully implemented encrypted Bluetooth connection at applica-

[1] `https://gradle.org/`
[2] `https://www.gartner.com/newsroom/id/3876865`
[3] `https://developer.android.com/reference/android/bluetooth/BluetoothDevice.html`

tion level with unpaired devices and proceeded to adapt the inspection protocol, as described in Section 3.3.3. We use *hybrid encryption*, where a message contains the object encrypted with a random AES symmetric key and the key is encrypted with the RSA public key of the receiver. The object is sent along with the encrypted key. The receiver decrypts the AES key with its RSA private key and retrieves the message key. The message is also signed with the private key of the sender to allow the receiver to check the integrity of the received message.

Additionally we considered registering the Media Access Control (MAC) address of the Bluetooth interface of each device to identify devices in the inspection interaction. However it is not possible in recent versions of Android to retrieve the Bluetooth MAC address of the device running the application because of security policies. Therefore we decided to use pseudonyms generated by the Central Ledger as transient Bluetooth device names, as described in Section 3.3.3.

Android devices provide the *Google Play services location* Application Programming Interface (API)[4], which allows to program constant location retrieval at a specified interval. For evaluation purposes, we defined the interval at one second and requested the *PRIORITY_HIGH_ACCURACY* value to get the most accurate location positions possible. Documentation of this feature states that the location updates may be slower or faster than the desired interval, due to battery optimization or poor connectivity. Additionally, the Transport application does not allow the user to start the trip while a location point cannot be retrieved. This prevents that a trip can be started in difficult initial location retrieving situations, such as in a moving vehicle. The initialization of the location retrieving process improves when the device is stationary, as explained in Section 2.1.

Figure 4.3 shows examples of the User Interface (UI) of the applications created for the STOP system prototype.

4.3 Central Ledger Implementation

As the STOP architecture considers that communication with the Central Ledger is done through a Representational State Transfer (REST)ful API endpoint, we used the Central Ledger *OpenAPI* description mentioned in Section 3.2.1. The API accepts and returns data in JavaScript Object Notation (JSON) format. Additionally, we used the *Swagger Editor* to generate documentation and initial code from this description. The *Jersey* and *FasterXML* Jackson frameworks were used to implement the API of Central Ledger and to serialize JSON objects to Java objects and vice-versa, respectively.

[4] `https://developer.android.com/training/location`

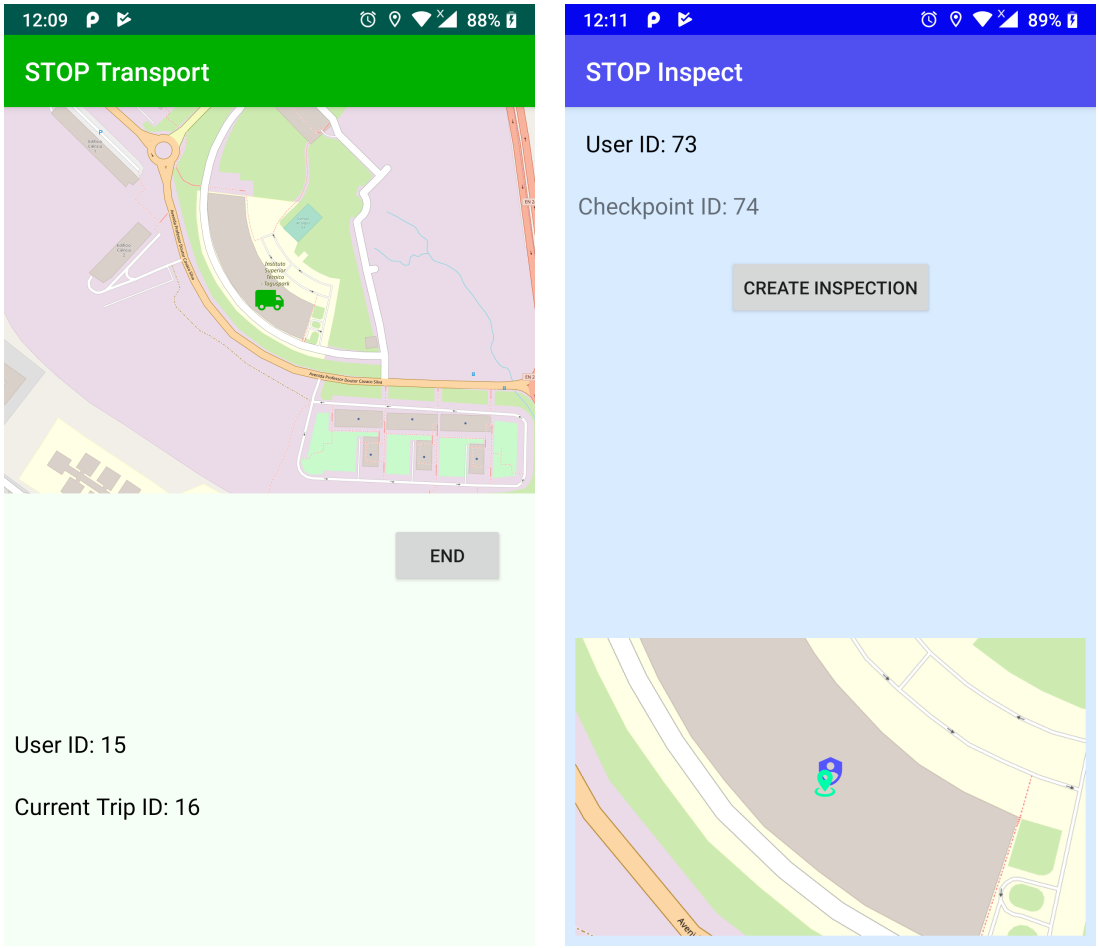

Figure 4.3: Screenshots of the Transport and Inspect applications, respectively

To use the desired Database Management System (DBMS), we used the *Hibernate* library to manage the connection and transactions. Additionally, the developed code performs data reading or writing through one single class, *DataHolder*. This allows programming more connections to other DBMSs or other software without substantial code change.

The Central Ledger implementation was deployed on an Apache Tomcat application server instance. A virtual machine was instantiated in a private cloud provider with 4 vCPUs, 8GB of RAM and 40GB of hard disk size, running the Ubuntu 18.04 LTS operating system.

As mentioned, we used an already existent and maintained *MySQL* DBMS cluster, which provides data replication. The Central Ledger server and the DBMS cluster were hosted in the same data center.

4.4 STOP System Parameters

As described in the architecture, STOP contains parameters that are defined upon setup of the Central Ledger. In this prototype, we considered that trip details were submitted in the Transport application. As mentioned, we set the location retrieval rate of the applications in periods of 1 second. Additionally the Transport application would check if the vehicle was selected for inspection after sending a location point.

Regarding the selection of vehicles, we defined the inspection selection range at 500 meters and did not set a maximum number of inspections per vehicle. For vehicle selection rule, we first consider the vehicles which the last reported location point is within the inspection selection range. We then retrieve the fifth recent previous location point and calculate its distance to the checkpoint. If the previous distance is greater than the current distance of the vehicle, then we assume the vehicle is heading towards the checkpoint. Additionally if the distance between both is lower than 10 meters, we considered the vehicle is stopped. Therefore in both cases, the vehicle is considered for selection and the system will randomly select one from this set. In sum, the defined rule takes into account the direction of moving vehicles inside the inspection selection range and also stopped vehicles in this perimeter.

4.5 Summary

The STOP system prototype was implemented with two Android mobile applications, a Java server application and a database management system. Several frameworks and APIs were used in order to implement the location retrieval, Bluetooth communication, web service and data persistence aspects of the system. The prototype implementation was evaluated in detail, as

described in the next Chapter.

Chapter 5

Evaluation

The STOP prototype was evaluated and we present and discuss the results in this Chapter. As described, the system relies on the location retrieved from mobile devices and provides the possibility of setting parameters related with the inspection scenario. Additionally the system presumes that both mobile devices can maintain a Bluetooth connection during the complete inspection process. Therefore, with a practical evaluation, it is possible to assess if the assumptions made while designing the system are correct. Also it is possible to assess what are the parameters that best improve the system operation. Considering the goals presented in Section 3.1.1, the evaluation focused on the following subjects:

- Are the location coordinates retrieved from Android mobile devices accurate enough for the STOP system procedures?
- What are the ideal STOP parameters for the selection of vehicles for inspection?
- Is the designed inspection interaction protocol suitable for Bluetooth communication in an inspection scenario?

5.1 Location Accuracy

As the system uses the latest reported location from the on-board device of a vehicle, it is important to determine if mobile devices are capable of retrieving accurate location points. We set out two different courses, *I* and *II*, tracked with the STOP Transport application. Course I was done using a mobile device inside of a automobile. Course II was done with 3 groups of two users, each one with a mobile device and each group traveling in a different bus. Having the users traveling course II in groups of two allowed us to assess possible discrepancies between devices performing the same exact route. In Table 5.1 we show the mix of different types of

Category	Minimum speed (Km/h)	Maximum speed (Km/h)
Highway	90	120
Mixed	51	89
City	0	50

Table 5.1: Road section categories

road, with different speed limits. We consider that a vehicle moving at 90Km/h or more is on an highway section and a vehicle moving at a speed inferior to 51Km/h is on a city environment. The mixed sections exist between cities and are often chosen as transportation inspection sites by authorities.

As shown by Figures 5.1 and 5.2, Course I is primarily a highway course with occasional mixed and city sections and Course II is a city course.

Figure 5.1: Course I used for location accuracy experiments

In Table 5.2, we show the different devices used and how they were used by the teams of volunteers. We had a mixed set of recent high-end devices and older low-end ones. This allowed us to have a sample of what kind of devices would be used in a real scenario. Additionally we arranged users B through G in groups of two in order to detect possible discrepancies between the reported location points of devices moving together.

User	Course	Vehicle	Device	Android Version	Location Rate (s)
A	I	Car	Nokia 8 (2017)	9	1
B	II	Bus 1	Huawei Mate 10 (2017)	8	1
C	II	Bus 1	LG G5 (2016)	6	1
D	II	Bus 2	Nokia 5 (2017)	9	1
E	II	Bus 2	Huawei Y5II (2016)	5.1	1
F	II	Bus 3	Sony Xperia E5 (2016)	6	1
G	II	Bus 3	Xiaomi Mi 5 (2016)	7	1

Table 5.2: Mobile devices used

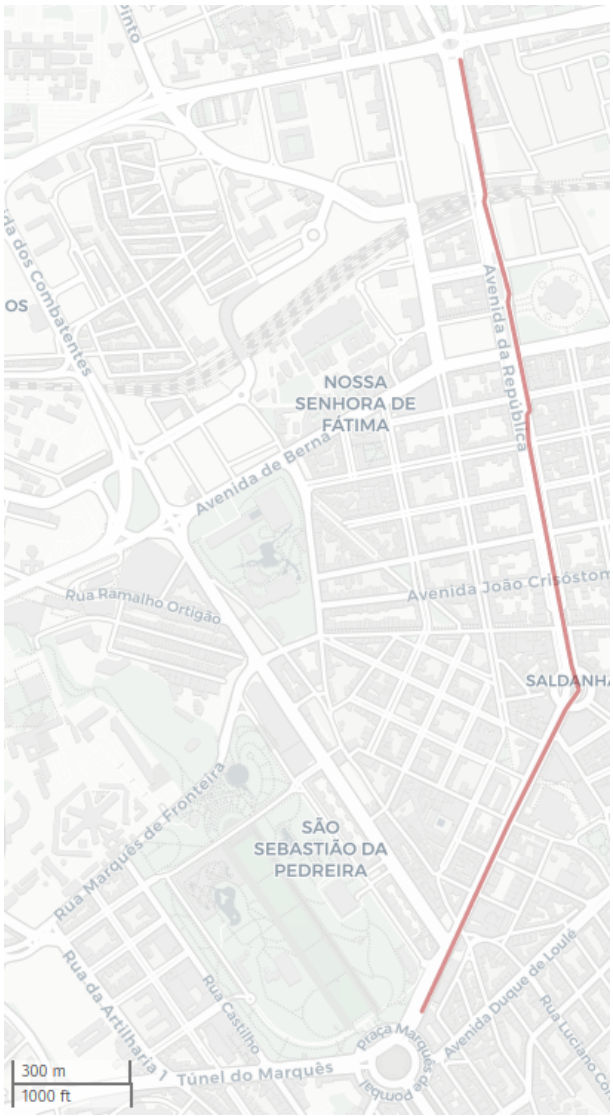

Figure 5.2: Course II used for location accuracy experiments

5.1.1 Visual Analysis of the Reported Courses

Upon visualizing the reported location points throughout the different courses, overall location points are close to the real trajectory but it is possible to detect some anomalies. Appendix A contains the visual representation of the recorded trajectories of all users. Course I contains a section inside of a tunnel where the mobile device of User A did not report any location point, as shown by Figure 5.3. In this figure, the blue line represents the real course and the red dots represent the reported location points. We can clearly see that there were moments where no location point was reported inside of the tunnel.

Course II has tall buildings in its surroundings which is known to affect Global Positioning System (GPS) signal. Upon visualizing the reported courses of users B through G, we noticed moments where the reported location coordinates were in buildings. Although we could not

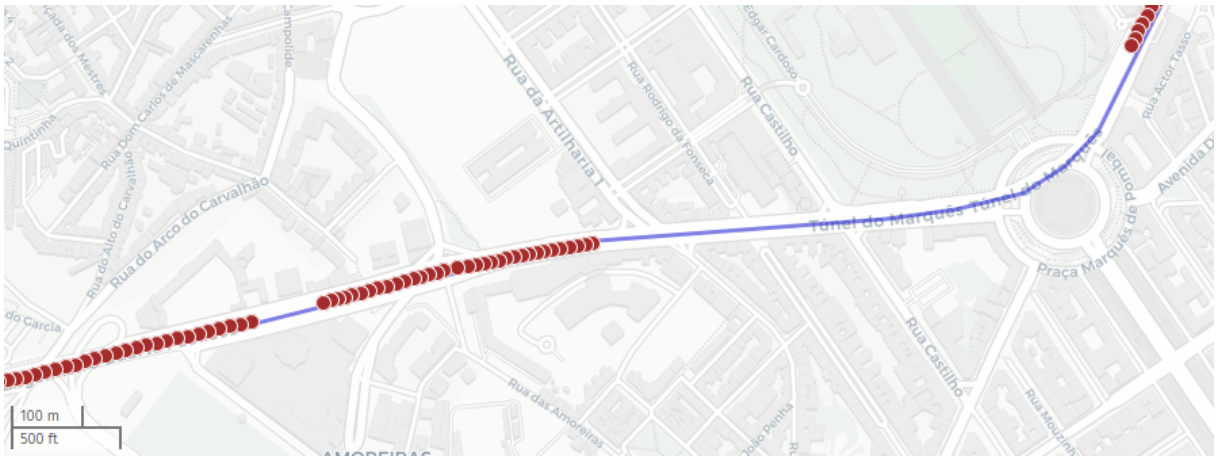

Figure 5.3: Issues inside of a tunnel with User A

confirm it during the experiment, we suspect that the devices might have detected known Service Set Identifier (SSID)s and Basic Service Set Identifier (BSSID)s of Wi-Fi networks in these buildings, as Android also uses Wi-Fi fingerprint for location retrieval. With a poor GPS connectivity, the device might have calculated its positions inside of the building, taking into account the Wi-Fi networks detected. Figure 5.4 shows examples of this behavior with users B, C and D. The lines represent the real course, we can see in the three examples that some of the reported location points, represented by the dots, were reported in buildings, with a considerable distance from the real trajectory.

Figure 5.4: Issues with building surroundings with users B, C and D

5.1.2 Accuracy Offset

Although visual analysis helps recognizing and understanding some issues, it does not gives us the overall accuracy levels of the reported location points. Therefore we have performed calculations on the retrieved location information of the devices. We first transformed the

location coordinates stored in our database into GeoJSON format, a geospatial data format based on JavaScript Object Notation (JSON)[1]. Finally we computed the distance of each reported point to the real trajectory using the *Turf.js* framework with *Node.js*.

Additionally we included the Android location accuracy value that represents the estimated horizontal accuracy of the location, radial, in meters. The documentation[2] states that the developers consider horizontal accuracy as the radius of 68% confidence. This means that there is a 68% probability that the distance in meters between the reported and the true location is lower or equal to the accuracy value returned. We want to assess if this value is correct, as it could be used by the system to detect a non-accurate location point.

Table 5.3 shows the obtained results. Column *No. Points* represents the number of location points retrieved per user and column *Average distance* represents the average distance to the real trajectory of all reported points by user in meters. Regarding the comparison with the accuracy values of Android, column *Average reported accuracy* shows the average value returned for all points by user and column *Correct accuracy estimation* represents the percentage of points which the distance between the reported and the true location was in fact lower or equal to the accuracy value returned.

User	**No. Points**	**Average distance (m)**	**Average reported accuracy (m)**	**Correct accuracy estimation**
A	1244	4,637432751	4,339750804	0,487942122
B	1673	5,574245725	4,049834429	0,566646742
C	832	7,319386648	4,284322115	0,260817308
D	1375	8,188775068	6,133682909	0,471272727
E	1376	8,935287117	4,775788462	0,210755814
F	1820	18,968411020	3,391062500	0,173626374
G	1885	7,507671515	7,578453050	0,576127321

Table 5.3: Location retrieval accuracy results

The average distance of user A from the reported to the real trajectory is lower than 5 meters, which we consider tolerable as the vehicle was mainly traveling between 90Km/h and 120Km/h and the city sections were not surrounded by tall buildings and did not include narrow roads. With the rest of the users, we conclude that accuracy in a complete city environment is not as good as in a highway. Vehicle speeds are lower but the average distance was higher. All users of course II, except user F, had an average distance to the real trajectory between 5 and 9 meters. User F reported that his device may have a GPS malfunction because previous usages of navigation applications showed incorrect location positions. We conclude that this malfunction

[1] `https://tools.ietf.org/html/rfc7946`

[2] `https://developer.android.com/reference/android/location/Location.html#getAccuracy()`

justifies the substantial average distance to the real trajectory, as user F traveled always with user G and this user had an overall average similar to the other users.

This accuracy assessment allows us to determine where the optimal location for a inspection site is. Inspectors should assess if the area inside the selected inspection selection range is not surrounded by tall buildings and does not include narrow roads. To our knowledge, heavy road vehicle inspections often occur in locations that fulfill this requirement, as most of these vehicles do not travel in a constant city environment.

Regarding the Android accuracy estimation, from our results, only one user had an overall correct accuracy estimation. However after analyzing some points in particular that had substantial distances from the real trajectory, the returned accuracy values were also high, showing that the device also detected that the reported points were not accurate. Therefore the system could set a threshold for the accuracy value returned by Android, 10 meters for example, and use it to detect possible non-accurate points. Nevertheless as results show, this value must be used as an approximate value that can help to detect non-accurate values.

5.2 Vehicle Selection

Location tracking of the on-board devices allows the system to select vehicles for inspection near their current location. We consider that the parameters defined in our architecture and by the Authority user should be evaluated as they influence the selection procedure. Having the real-time position of each vehicle and an optimal selection rule and range will allow the system to select vehicles that will be able to stop at the checkpoint, without any detours or delays. First we will present our assessment on the frequency of the retrieval of location points. As vehicles will be traveling at different speeds and we want to have an efficient application, we want to assess if a fixed rate should be implemented or not, taking into consideration that a higher location retrieval rate requires more processing from the mobile device and Central Ledger. Secondly we assess if the selection rule and the value chosen in our prototype for the inspection selection range are viable for every inspection that can be made.

5.2.1 Location Retrieval Frequency

We consider the location retrieval period a parameter of the system that needs to be defined experimentally, considering different types of routes. The highest location retrieval rate possible will ensure the system has the most recent location of each vehicle, however it will demand more processing from the components. Before assessing this parameter, we wanted to confirm if the location retrieval rates defined in the Android implementation were in fact being fulfilled. We

defined the location retrieval rate at 1 second, as upon reading the documentation of the location retrieval feature of Android[3], we consider this value as the fastest interval suitable for our use case, considering the different speeds vehicles travel at, the accuracy of the location points and device performance. We also assumed that if Android devices could fulfill our fastest location retrieval rates, then they could fulfill slower rates. Table 5.4 shows the results of this evaluation. Column *Retrieval rate set* represents the location rate set in seconds, column *Average retrieval rate* represents the average retrieval rate of all points by user and column *Retrieval rate not fulfilled* shows the percentage of points that did not fulfill the retrieval rate set. For all users, less than 5% of the retrieved points were retrieved in more than 1 second.

User	**No. Points**	**Retrieval rate set (s)**	**Average retrieval rate (s)**	**Retrieval rate not fulfilled**
A	1244	1	1,059	0,001609010
B	1673	1	1,092	0,031100478
C	832	1	1,951	0,034897714
D	1375	1	1,073	0,014556041
E	1376	1	1,067	0,020363636
F	1820	1	1,049	0,014843321
G	1885	1	1,038	0,010615711

Table 5.4: Location retrieval intervals performed

Figure 5.5 illustrates the reported location retrieval rates. The horizontal axis represents the number of the location point and the vertical axis represents the duration that the location point took to be retrieved in seconds. Upon visualizing the results, most of the points are in the exact 1 second mark, however there are a few location points that took much more time to be retrieved. This showcases why the percentage of points that have not fulfilled the set rate is minimal and the average rate is above one second. We presume that a higher location retrieval interval occurs when the GPS signal is not satisfactory, the device cannot use mobile data or the device is doing battery optimization.

Results show that it is possible to have a one second retrieval rate, therefore we conclude that we can rely on the location retrieval rate defined on Android systems. However as mentioned, having a one second retrieval rate would create a considerable demand from the device and Central Ledger, despite guaranteeing that the system would have the most possible up-to-date location. We suggest that the location retrieval rate should be variable considering the speed of the vehicle. The device would constantly change its location retrieval rate to adapt to the speed at which the vehicle is moving. Speed can be calculated with the already retrieved points or

[3] `https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest`

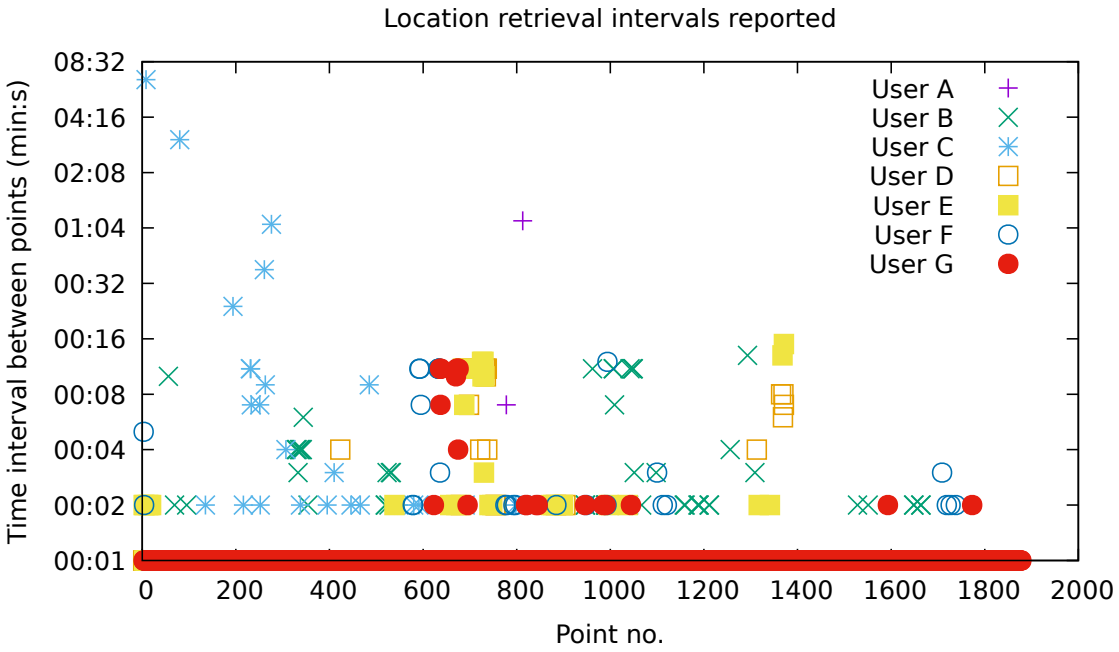

Figure 5.5: Location retrieval intervals reported

with an Android method to get speed from a location point[4]. A possible function to calculate the retrieval rate according to the speed could be the following example:

$$f(x) = \begin{cases} -\frac{4}{120}x + 5, & \text{if } 0 \leq x \leq 120 \\ 1, & \text{otherwise} \end{cases}$$

The x variable is speed in Km/h and the returned value from the function is the retrieval rate in seconds. This will allow better battery optimization and lower mobile data consumption as the device may not need to retrieve and send location points every second in certain situations. The retrieval rate would be proportionate to the speed, having one and five seconds as the maximum and minimal location retrieval rate, respectively. If the vehicle is stopped, the device only retrieves points every 5 seconds, as the traveled distance is minimal. If the vehicle is traveling at 120Km/h or higher, then the system needs faster location updates, therefore the one second rate is set.

5.2.2 Selection Rule

We performed inspection selection tests simultaneously with users B through G. Two Inspect users, I1 and I2, were at one checkpoint each. The distance between the two checkpoints was higher than the defined inspection selection range of 500 meters. The six Transport users started

[4] `https://developer.android.com/reference/android/location/Location#getSpeed()`

the course and the Inspect users were at the corresponding checkpoint requesting an inspection every minute until the request was fulfilled. Inspection protocols would be made with the two devices side by side.

User B was the first to be selected for an inspection with inspector I1. The inspection protocol was done with no issues. User B continued the trip and was selected to be inspected by inspector I2. Again, the inspection protocol had no problems. However after this inspection, inspector I2 requested another inspection and the system chose user B again, which had just been inspected and was already on route. The inspector had to wait for the user as he had to return back to the checkpoint. We consider this as a system malfunction. Meanwhile user F was selected to be inspected by inspector I1, which occurred normally.

The issue with user B occurred because the user was stopped due to traffic near the checkpoint, therefore he was eligible for selection due to the defined rule in the prototype. Clearly the rule was not correct. We consider the implemented parameter for stopped vehicles should not be included.

Additional improvements could be done to the vehicle selection rule. One improvement that could prevent situations where a vehicle has already been inspected or is too close to the inspection site to be notified on time is to establish a minimum selection range. Vehicles too close to the checkpoint would not be considered for inspection. This value should be variable considering the speed of the vehicles near the checkpoint. A minimum selection range should consider the time it takes for a driver to acknowledge a selection, plus the average distance the vehicle is traveling per second. We provide an example for calculating such range. Our inspection experiments show that the average time between a vehicle being selected and the device of such vehicle retrieving such notification was 511.5 milliseconds. As explained in Section 4.4, every Transport device would check the selection for inspection after sending a location point, therefore, as shown by Table 5.4, the used devices checked the selection every two seconds at most on average. In an inspection site at a highway section, vehicles typically travel between 90Km/h and 120Km/h, which is 25 meters per second and 33.3 meters per second respectively. Considering a driver would take 10 seconds to react to the notification and to acknowledge the inspection site, and adding the time between selection and retrieving the notification, it would take approximately 11 seconds for a driver to be notified. During this time, the vehicle would travel 366.3 meters at most so a considerable minimum selection range would be 400 meters in a inspection site at a highway section. This range would prevent drivers from acknowledging inspections after driving past them.

Another possible malfunction with the current selection rule is the selection of a vehicle inside

the range where the driver would have to significantly change its route to meet the inspector. In this scenario, the system would have to implement a route calculation procedure to ensure that selected vehicles could easily reach the checkpoint. If the vehicle could not reach the inspection site in a defined time interval or without a route change, then it would not be considered for inspection. This would also allow the system to present an estimated time of arrival to the inspector waiting for the selected vehicle. Additionally, it could help the system to estimate the road where a vehicle is if the device is reporting location points outside of roads, as demonstrated in Section 5.1.1. The route planning feature could also have in consideration the destination of the transportation, present in the electronic Freight Transport Information (eFTI), to check if the planned transportation route would go through the checkpoint.

5.3 Bluetooth Inspection Interaction

We simulated an inspection area with a metal container similar to ones that carry goods in transportation vehicles. Figure 5.6 shows the used container, which size is industry standard, 5,9m x 2,4m x 2,4m.

Figure 5.6: Standard sized metal container used for Bluetooth evaluation

A Samsung Galaxy S9 device running Android 8 was used as the Transport device and a Nokia 8 device running Android 9 was used as the Inspect device. Both devices have Bluetooth 4.0. We positioned the Transport device in front of the container and proceeded to request an inspection in the Inspect device. The Transport device was selected.

In a typical inspection scenario, an inspector might move around the container and our

architecture considers that a Bluetooth connection is maintained during this procedure. However a metal container might interfere with the Bluetooth connection. Therefore we performed several movements around the container to test if the connection was maintained.

The inspector was able to walk around the container and approve the inspection near the Carrier user. This procedure was done successfully 3 times. This did not happen when the inspector would stop for more than 5 seconds behind the container, the connection would be lost. Therefore we conclude that the Bluetooth inspection protocol cannot consider that a Bluetooth connection is fully maintained during an inspection process.

A possible change to the protocol would be to divide it in two parts. In the current protocol, the Inspect device waits for the connection of the Transport device to send the proof request. However in this possible improvement, the socket is closed and the Transport device awaits for the connection of the Inspect device. After ending the inspection procedure, the inspector heads towards the driver and approves the inspection to send the proof. The Inspect device establishes the connection and sends the object. Although this implies two connection setups, it would work in this scenario as the metal container will not interfere if the two devices are close together during all the Bluetooth connections.

5.4 Discussion

We evaluated important features of Android devices used for our prototype, specifically location retrieval and Bluetooth communication. Regarding location retrieval, we have presented data to assess the accuracy of the location points. We concluded that in a highway course location points are accurate however, inside tunnels for example, devices cannot simply retrieve location information. In a city course, we concluded that GPS signal strength varies and the device may report location points outside of roads for example. Regarding the location retrieval rate, we found the results to be satisfactory as the Android devices were able to report most of the location points at the defined location rate. We suggest a variable location retrieval rate for better optimization of the device processing and battery.

Upon testing the initial selection rules implemented, we detected some anomalies. Therefore we proposed that the selection rule should be composed of the following parameters: maximum and minimum inspection selection range and a estimated time of arrival with a route planning procedure. This would allow vehicles to be notified on time and guarantee that a selected vehicle would not have to change its route to reach the checkpoint.

During the experiments with Bluetooth transmission around the cargo container, we found that the designed protocol needed to be changed to better suit an inspection scenario. We

proposed the protocol should be divided in two phases, therefore in two separate Bluetooth connections. Initially the Inspect device receives a connection from the Transport device to receive a proof request and the connection is terminated. After the inspection procedure, the Transport device would receive a new connection from the Inspect device to receive the inspection proof.

Chapter 6

Conclusion

In this dissertation we presented STOP, a road transportation inspection support system using mobile devices. It aims to assist inspection procedures by automating vehicle selection, electronic Freight Transport Information (eFTI) retrieval and inspection outcome submission. Inspectors are able to receive information regarding a soon-to-be inspected vehicle, prepare the inspection procedure and submit outcome reports with no usage of paper documents. Additionally, transportation companies can present proofs that the freight was thoroughly inspected and reduce stop time during inspections. The usage of off-the-shelf smartphones allows hardware costs to be lower, as devices already purchased for another means can be used.

We first identified the current reported issues and improvements of the road transportation sector. This gave us an overview of what could be improved and also integrated with new government approaches. We chose the inspection procedure of transportation vehicles as the aspect that we aim to provide a support system. After analyzing systems already used by carriers, we concluded that due to their proprietary aspect, it was not possible to use them easily. We therefore researched the possibility of the usage of mobile devices as enterprises already are integrating mobile applications in their internal processes. Research showed issues regarding the accuracy and integrity of the Global Positioning System (GPS) aspect of mobile devices. Therefore we presented works regarding location certification with mobile devices, which is a relevant aspect for our work.

6.1 Achievements

STOP uses the location retrieval, mobile data and short-range communication features of mobile devices. The system considers the entities typically involved in the transportation of goods: Authority, Company, Carrier and Inspector. The architecture is composed of a central server

and two mobile applications. The Authority is responsible for setting the Central Ledger with the legal requirements of a road inspection, the Company and Carrier users use the Transport application and the Inspector uses the Inspect application. The Company registers upcoming transportations in the system and the Carrier uses the application while transporting the reported goods. The Inspector sets up an inspection location, called checkpoint, and proceeds to use the Inspect application to notify close vehicles for inspection. The driver of the selected vehicle is notified of the checkpoint through the Transport application. After the inspection is concluded, the Inspect device creates a location proof to certify an inspection was conducted and sends it to the device of the driver and to the Central Ledger. We provided a detailed description of this process and the protocols used. Every event of the transportation is registered in a Location Chain that contains a combination of location points and proofs. Every location chain item contains the signature of the previous item to guarantee the item ordering and prevent tampering.

A prototype was implemented using Android devices. The implementation was evaluated to assess the viability of the usage of such devices for inspection scenarios and the performance of this system. The retrieved dataset and the evaluation results allowed us to verify location retrieval and short-range communication aspects of Android devices. Additionally we evaluated procedures and protocols of the system such as inspection selection rules and the inspection protocol to create a location proof. The retrieved results allowed us to approve aspects of the location retrieval of mobile devices but also detect anomalies in some scenarios. Also we suggested improvements of the vehicle selection rule and the inspection communication protocol.

Overall we believe this work presents valid assessment on the use of mobile devices to support government procedures and also on the combination of location tracking with location certification. Despite some inaccuracy being detected in location retrieval, we believe newer mobile devices will have improvements on this feature and retrieved location points will be more accurate.

6.2 Future Work

Several improvements can be done to this solution. Initially, it would be very beneficial having the feedback of a domain specialist about STOP. This would allow further validation of the requirements and this would improve the system to better integrate with the current inspection procedures.

The system could benefit from analytical features to inform authorities and inspectors of useful information and anomalies, using the retrieved data from the Transport users. Some

examples are indicating congested traffic areas in real-time and ideal inspection areas.

A possible attack from a Carrier user is to stop the vehicle and modify the content of the transportation when notified for inspection. To prevent this situation, the system could use the route planning feature, presented in Section 5.2.2, to detect a longsome arrival to the checkpoint and also implement a mix of planned and surprise inspections. In a surprise inspection, the inspector would receive the transportation information beforehand and would have the real-time position of the vehicle, but the Carrier user would not be notified. The inspector would order the vehicle to stop when it would be driving through the checkpoint.

Regarding the location retrieval aspect, if the integration with the already in-use fleet management system was possible, then location points from these systems could be used to report the route of the vehicle. The mobile application would continue to be used for inspection selection notification and to create location proofs. This would reduce the dependability on the mobile device, specially on the reported location information. Nevertheless such integration could be difficult to accomplish as these systems may not have a Application Programming Interface (API) to retrieve information and governments and enterprises would have to establish private networks to securely communicate this information.

If governments identified the necessity of proving the traveled route of each transportation, then vehicle-to-vehicle location proofing could be implemented to have witness-based proofs. Other transportation vehicles on the road could testify when and where they have "seen" the inspected vehicle. This interaction could occur in road lanes when traveling at a stable speed, for example, in a highway lane, or at stop lights or parking lots, for example. Colluding prevention mechanisms would have to be implemented to make sure the witnesses are reliable, just as is done in other location proofing works.

Additionally, a distributed and decentralized STOP system could be implemented. In this scenario, each enterprise would participate in a network where Transport and Inspect devices would submit location points and proofs. The network of computers would validate submitted information and it could be recorded on a trusted government database. This feature would increase the complexity of the system as technical requirements are higher and also the privacy of the data would be crucial to prevent companies to identify sensitive data from others, but could provide further transparency to the inspection processes.

Bibliography

[1] Agência Portuguesa do Ambiente. e-GAR. URL https://vimeo.com/245032348.

[2] Agência Portuguesa do Ambiente. e-GAR: Guias eletrónicas de resíduos, 2018. URL https://apambiente.pt/_zdata/DESTAQUES/2018/ApresentacaoPERSU2020/E-GAR.pdf.

[3] Agência Portuguesa do Ambiente. Resíduos - Enquadramento, 2019. URL https://apoiosiliamb.apambiente.pt/content/enquadramento-eGar?language=pt-pt.

[4] Android Developers. Optimize location for battery, 2019. URL https://developer.android.com/guide/topics/location/battery.

[5] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal. Gps: location-tracking technology. *Computer*, 35:92–94, 2002.

[6] N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stopsignature schemes without trees. In *Advances in Cryptology - EUROCRYPT 1997*, pages 480–494, 1997.

[7] J. Benaloh and M. Mare. One-way accumulators: A decentralizedalternative to digital signatures. In *Advances in Cryptology - EUROCRYPT 1993*, pages 274–285.

[8] J. Camenisch and A. Lysyanskaya. Dynamic accumulators andapplication to efficient revocation of anonymous credentials. In *Advances in Cryptology - CRYPTO 2002*, pages 61–76, 2002.

[9] E. S. Canlar, M. Conti, B. Crispo, and R. Di Pietro. CREPUSCOLO: a Collusion Resistant Privacy Preserving Location Verification System. In *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*, 2013.

[10] Cartrack. Cartrack: Como funcionam os sistemas de localização de viaturas por GPS?, 2019. URL https://www.cartrack.pt/localizacao-gps-viaturas/.

[11] CBS New York. N.J. Man In A Jam, After Illegal GPS Device Interferes With Newark Liberty Operations, 2013. URL https://newyork.cbslocal.com/2013/08/09/n-j-man-

in-a-jam-after-illegal-gps-device-interferes-with-newark-liberty-operations/.

[12] Council of the EU. Easier use of digital information for freight transport – Council agrees on its position , 2019. URL https://www.consilium.europa.eu/en/press/press-releases/2019/06/06/easier-use-of-digital-information-for-freight-transport-council-agrees-on-its-position/.

[13] Council of the European Union. Proposal for a Regulation of the European Parliament and of the Council on electronic freight transport information - General Approach, 2019. URL http://data.consilium.europa.eu/doc/document/ST-9181-2019-INIT/en/pdf.

[14] eMarketer. Maps and Navigation Apps: Discovery, Exploration Features Open Up Ad Opportunities, 2018. URL https://www.emarketer.com/content/maps-and-navigation-apps.

[15] J. Ferreira and M. L. Pardal. Witness-based location proofs for mobile devices. In *17th IEEE International Symposium on Network Computing and Applications (NCA)*, 11 2018.

[16] GPS.gov. Gps space segment, 2019. URL https://www.gps.gov/systems/gps/space/.

[17] K. Hill. Jamming GPS Signals Is Illegal, Dangerous, Cheap, and Easy, 2017. URL https://gizmodo.com/jamming-gps-signals-is-illegal-dangerous-cheap-and-e-1796778955.

[18] T. Humphreys. Statement on the vulnerability of civil unmanned aerialvehicles and other systems to civil gps spoofing. Technical report, The University of Texas at Austin, 2012.

[19] M. Hynes, B. Miller, and M. Barrett. GPS Tracker, 2003.

[20] Inosat. Inosat: Como Funciona, 2019. URL https://www.inosat.pt/saiba-como-funciona-inofrota/.

[21] M. Juang. A new kind of auto insurance technology can lead to lower premiums, but it tracks your every move, 2018. URL https://www.cnbc.com/2018/10/05/new-kind-of-auto-insurance-can-be-cheaper-but-tracks-your-every-move.html.

[22] R. Khan, S. Zawoad, M. M. Haque, and R. Hasan. Otit: towards secure provenance modeling for location proofs. In *ASIA CCS '14 Proceedings of the 9th ACM symposium on Information, computer and communications security* , 2014.

[23] H. Krawczyk, K. G. Paterson, and H. Wee. On the security of the tls protocol: A systematic analysis. In *CRYPTO 2013: Advances in Cryptology*, pages 429–448, 2013.

[24] A. Loten. Life on the Road Gets a Little Easier as Truckers Adopt Digital Technology, 2019. URL https://www.wsj.com/articles/life-on-the-road-gets-a-little-easier-as-truckers-adopt-digital-technology-11559727001.

[25] S. Narain, A. Ranganathan, and G. Noubir. Security of gps/ins based on-road location tracking systems. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[26] M. Niestadt. Electronic freight transport information, 2019. URL http://www.europarl.europa.eu/RegData/etudes/BRIE/2018/630263/EPRS_BRI(2018)630263_EN.pdf.

[27] H. Onishi, K. Yoshida, and T. Kato. Gnss vulnerabilities and vehicle applications. In *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016.

[28] K.-W. Park. T-Box: Tamper-resistant vehicle data collection system for a networked digital tachograph. In *International Conference on ICT for Smart Society*, 6 2013.

[29] Rádio e Televisão de Portugal. Operação da ASAE nas estradas fiscaliza transportes de mercadorias, 2018. URL https://www.rtp.pt/noticias/economia/operacao-da-asae-nas-estradas-fiscaliza-transportes-de-mercadorias_v1099919.

[30] M. Remac. Electronic documents for freight transport, 2018. URL http://www.europarl.europa.eu/RegData/etudes/BRIE/2018/615673/EPRS_BRI(2018)615673_EN.pdf.

[31] Ridester. Inside the Ridesharing Revolution: 2018 Edition, 2018. URL https://www.ridester.com/2018-rideshare-infographic/.

[32] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In ACM, editor, *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 9, 2009.

[33] J. E. Siegel. Design, Development, and Validation of a Remotely Reconfigurable Vehicle Telemetry System for Consumer and Government Applications. Master's thesis, Massachusetts Institute of Technology, 2011.

[34] The University of Texas at Austin. UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea, 2013. URL https://news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/.

[35] United Nations. Additonal Protocol to the Convention on the Convention on the Contract for the International Carriage of Goods by Road (CMR) concerning the Electronic Consigment Note, 2008. URL https://www.unece.org/fileadmin/DAM/trans/conventn/e-CMRe.pdf.

[36] United Nations Economic Commission for Europe (UNECE). UN Transport Agreements and Conventions, 2019. URL https://www.unece.org/trans/maps/un-transport-agreements-and-conventions-27.html.

[37] M. van Leijen. Electronic freight document should not be the new ERTMS, 2018. URL https://www.railfreight.com/policy/2018/09/11/electronic-freight-document-should-not-be-the-new-ertms/.

[38] Waze. How does Waze work?, 2019. URL https://support.google.com/waze/answer/6078702?hl=en&ref_topic=9022747.

[39] K. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang. A Practical GPS Location Spoofing Attack in Road Navigation Scenario. In *ACM Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2017.

[40] Z. Zhu and G. Cao. Applaus: A privacy-preserving location proof updating system for location-based services. In *IEEE Conference on Computer Communications (INFOCOM) 2011*, 2011.

Appendix A

Road Courses Used for Testing

In this appendix we show the visual representations of the location evaluation made with the STOP prototype. In the following figures, the lines represent the real trajectory and the dots represent the reported location points.

Figure A.1: Reported course of User A

Figure A.2: Reported courses of User B and C

Figure A.3: Reported courses of User D and E

Figure A.4: Reported courses of User F and G