STOP: Secure Transport lOcation Proofs for vehicle inspections

Henrique Figueiredo dos Santos
hfigueiredosantos@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

October 2019

Abstract
An effort is being made by authorities around Europe and worldwide to digitize the processes that support the transportation of freight. 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.

We propose *STOP* - a road transportation vehicle inspection support system. This system uses mobile devices and a central server and allows inspectors to select and notify vehicles for inspection while retrieving the needed information to prepare the inspection procedure beforehand. A STOP prototype was implemented in the Google Android platform 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 what are the best parameters for the system.
Keywords: Smart vehicle inspections, Mobile applications, Location proofs, Location tracking, Digital transformation in road transport

1. Introduction

Currently there is a strong focus in implementing the digitalization of freight transport support processes. Around the world, there are reports of issues in the transportation sector, as paper documents are still used in freight transport activities and there is lack of a legal framework requiring authorities to accept eFTI [18, 21]. Several approaches and proposals have been presented in order to adopt several procedures to mandate the implementation of such digitalization [7, 6, 25]. 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.

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 [21, 18, 26]. This can bring several positives outcomes, such as decreased environmental impact, less longsome bureaucratic procedures and significant savings.

The inspection of road transportation [20] is one of the scenarios that could improve with digitalization. At 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. Therefore 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.

2. Background and Related Work

We have identified location as the main subject to be researched. As we were not able to find works with similar purposes, we focused on research about location systems and their use cases, as a means to

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

achieve our system requirements and goals. The location reporting of each vehicle enables the selection of vehicles and the consecutive preparation of inspections. Therefore the proposed system needs reliable location reporting.

2.1. GPS-based Location Systems and Applications

The Global Positioning System (GPS) is composed by a set of 31 operational satellites that emit radio signals that a GPS receiver can use to determine its position on Earth [2, 10]. 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.

Location Tracking Systems A GPS tracker is a device that enables real time position tracking of attached objects [13]. This device continuously retrieves its location by retrieving satellite signals from GPS. Currently transportation companies use *fleet management systems* that receive and gather data from the trackers inside vehicles to present real time information of the vehicles to the users [14, 4]. These solutions allow companies to monitor their fleet, ensuring secure transportation and reporting the delivery to a client as it happens. The device transmits the collected information through Global System for Mobile Communications (GSM) cellular network to the servers of the provider, which is presented through a web portal or computer software.

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 [8, 22]. 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 [1].

Navigation applications have also been used in the transportation sector [16]. 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.

Security Despite being widely used, GPS is not considered fully secure [19, 17]. 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 [11], therefore an attacker can easily purchase such devices. It is then possible to deceive mobile devices running road navigation applications [27], air drones [12], ships [24] and working vehicles [5].

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 [23]. The authors considered that a component of an existent wireless infrastructure such as Wi-Fi Access Points and cellular towers can issue metadata containing location information. Nearby mobile devices can then use this information to prove their location. A device can therefore request a location proof from the infrastructure and this proof can be sent to applications with the intent of proving the location of the mobile device. There have been several systems that allow the creation of location proofs, namely, Saroiu and Wolman's work, APPLAUS, CREPUSCOLO and SureThing.

Zhu and Cao proposed a location proof system called APPLAUS using only Bluetooth enabled mobile devices [28], using five entities: *Prover*, the mobile device who collects proofs from neighbors, *Witnesses*, untrusted mobile devices that generate location proofs, *Location Proof Server*, to store proofs, *Certificate Authority*, to store and validate public keys, and *Verifier*, that verifies submitted proofs. The system does not use an existent wireless infrastructure. It uses pseudonyms for each Prover and Witness to prevent device tracking. A Prover broadcasts through Bluetooth a location proof request. If it is received and accepted, a witness creates a proof, signs the proof with its private key and the proof is encrypted with the public key of the location proof server, to guarantee it is only decrypted by the server. This proof is sent to the Prover, who then sends it to the location proof server. The system may ask the Prover to obtain a threshold number of proofs from Witness nodes, becoming more difficult for an attacker to have the number of devices requested to successfully create a false proof. Validation is performed by a Verifier with access to the location proof server.

Canlar et al.[3] created CREPUSCOLO to address both the *neighbor-based* type of proof-based solutions, where nearby mobile devices create proofs, and the *infrastructure-based* type, where lo-

cation proofs are acquired from trusted infrastructure elements, such as Wi-Fi Access Points. The system uses the same entities of APPLAUS with the addition of the *Token Provider*, a trusted entity placed at a strategic location that generates a proof, called *token*, that may contain an object, such as a picture from a surveillance camera, that proves the device was at that location. Location proofs are exchanged and created like in APPLAUS, with the addition of a nonce in the proof request and in the associated location proof, to prevent replay attacks. The Token Provider is used to mitigate attacks where one device may broadcast messages from another device located at a different site and therefore witnesses may create proofs of the prover located at a different place.

SureThing [9] aims to provide correct location proofs to other applications and services, indoors or outdoors, using as motivation improving the APPLAUS and CREPUSCOLO works. It uses multiple entities similar to the ones in the two previous works presented, *Prover*, *Witness*, *Verifier* and *Certification Authority*, and it also uses geographical coordinates, Wi-Fi fingerprinting and Bluetooth beacons as location proof techniques. There are two types of witnesses. The *Witness* can be similar to the Witness entity found in APPLAUS and CREPUSCOLO, called *Mobile Witness*, or to the Token Provider entity in CREPUSCOLO, called *Master Witness*, trusted by the system. The *Verifier* is the central entity of the system who validates and stores all submitted location proofs. Ferreira and Pardal introduced two methods for collusion avoidance, to prevent colluding devices to create incorrect location proofs. The *Witness Redundancy* mechanism forces the Prover to gather proofs from more than one Witness and chooses the number of witnesses according to the level of service possible. Each proof has a different trust value according to the number of witnesses used. *Witness Decay* ensures that if a Prover is getting proofs from the same Witness, they gradually become less valuable and the Verifier will not validate the location if the Prover can not gather proofs with enough value.

3. STOP

We present a road transportation inspection support solution named STOP - 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. The system is composed of a central server and two mobile applications. The two mobile applications submit and retrieve information from the central server. Companies are able to report upcoming transportations in the system. 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.

STOP 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 Section 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.

3.1. 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.

Goals 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. 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. 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;
- Viable usage of mobile devices.

Participants We consider that a transportation process starts with a company registering the freight transportation information 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* is the entity responsible for the rules for goods inspection in a given country and audits the system to ensure rules are being followed. It defines the user authentication and authorization policy, the required information associated with each vehicle and transport, including the sender and receiver, freight description and planned itinerary. It also sets the policy for selecting vehicles for inspection. The *Inspector* is the agent conducting an inspection at a checkpoint and it is trusted by the Authority for this task. The *Company* is the entity sending goods to another enterprise or individual and it registers the trip details to comply with the rules defined by the Authority. The *Carrier* is the entity actually transporting the goods and is represented by the driver of the vehicle. 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.

The system uses *pseudonyms* instead of the real identities of the participating entities as it does not need this information to operate. The additional information, such as the location coordinates, is stored as required by the Authority that can audit its correct use.

Operation We have defined three components for the STOP system, further detailed in Section 3.2. 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.

The Company registers an upcoming transportation in the Transport application with the parameters previously defined by the Authority. 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.

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*, a perimeter from inspection sites where all vehicles inside are considered for random selection. 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 transportation information while the vehicle arrives and the Inspect application displays the current location of the vehicle.

The driver will drive 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.

3.2. Solution Architecture

The STOP system is structured in three tiers: Presentation, Logic and Data tiers, as shown by Figure 1. This allows for integration of new components

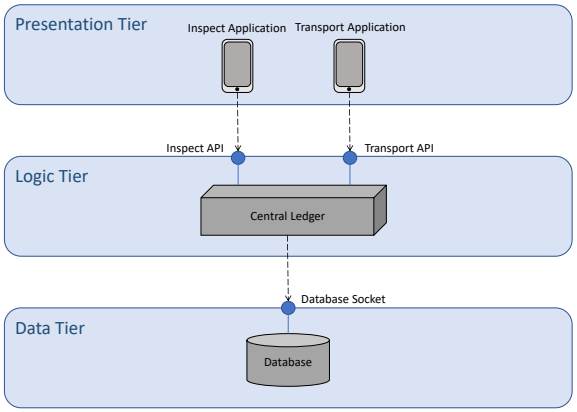

Figure 1: STOP Architecture

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. The *Central Ledger* is a central server that receives data of transportations and inspections. 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. Inspect operations allow creating, showing and ending checkpoints and inspections. A detailed description of the interface was done in *OpenAPI* 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.

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. The device running this application must have an active Internet connection during the transportation process. The application uses the Trip operations of the Central Ledger API.

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*. 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 item was modified or is 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. When the transportation ends, the Central Ledger has the complete *Location Chain* of the trip.

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. The application uses the Inspect operations of the Central Ledger API. After a vehicle is selected, the application presents the respective transportation information for the inspector to analyze while the vehicle 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. The proof contains pseudonyms of the *Transport* and *Inspect* devices, a trip identifier, 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.

Interactions 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.

The *remote* communication between the applications and the Central Ledger is done through the provided REST API web service via cellular network. This API uses standard HTTP over TLS [2] to protect the messages [15].

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

The Transport and Inspect communicate in *proximity*, through short-range Bluetooth 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.

3.3. Security Architecture

Each user generates a pair of *RSA* public and private cryptographic keys in their 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.

At a inspection scenario, 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. When the vehicle arrives to the checkpoint, the Transport application starts searching for the device announcing as device name the pseudonym of the device 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 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 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. 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.

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

4. Implementation

We chose Android devices as the mobile device platform as they represent most of the smartphone market [3] and their lower cost and high availability allows for testing with different devices from different manufacturers. Additionally, Android provides the *Google Play services location* API [4], which combines GPS, Wi-Fi and cell network information [5], to retrieve location information. We chose Ubuntu Server 18.04 LTS as the operating system of the server.

The applications were mainly developed in the Java programming language. As support libraries, the Gson library was used to serialize Java objects to JSON objects, for communication with the Central Ledger, and the OsMoDroid library was used to display maps from OpenStreetMap in both applications. The Central Ledger code was also developed in Java, as it shares some of the code modules with the mobile applications. The Central Ledger interface was specified in OpenAPI format and used the Swagger Editor tool to generate code to use as basis for the implementation of the Central Ledger. The Jersey and FasterXML Jackson frameworks were used to implement the RESTful API of Central Ledger and to serialize received JSON objects to Java objects, respectively. The program was deployed in an Apache Tomcat application server instance.

5. Evaluation

The evaluation conducted on our prototype 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 done with the STOP Transport application with several users. Upon visualizing the reported location points throughout the different courses, it is possible to detect some anomalies, but overall location points are close to the real trajectory. One of the performed courses contains a section inside of a tunnel and the mobile device that performed this course did not report any location point in this section, as shown by Figure 2.

Figure 2: Issues inside of a tunnel

User	No. Points	Average distance (m)
A	1244	4,637432751
B	1673	5,574245725
C	832	7,319386648
D	1375	8,188775068
E	1376	8,935287117
F	1820	18,968411020
G	1885	7,507671515

Table 1: Location retrieval accuracy results

Another performed course has tall buildings in its surroundings which is known to affect GPS signal. Upon visualizing the several reported user trajectories in this course, we noticed moments where the location coordinates reported were in buildings. Although we cannot confirm it, we suspect, as Android also uses Wi-Fi fingerprint for location retrieval, that the devices might have detected known SSIDs and BSSIDs of Wi-Fi networks in these buildings. With a poor GPS connectivity, the devices might have calculated their positions inside of the building, taking into account the Wi-Fi networks detected.

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. Table 1 shows the average distance between the reported and the exact trajectories of each user.

User A performed a course that was primarily highway courses with occasional city sections, while the rest of the users performed the same city course. The average distance of user A 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 of the course 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

[3] https://www.gartner.com/newsroom/id/3876865

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

[5] https://developer.android.com/guide/topics/location/battery

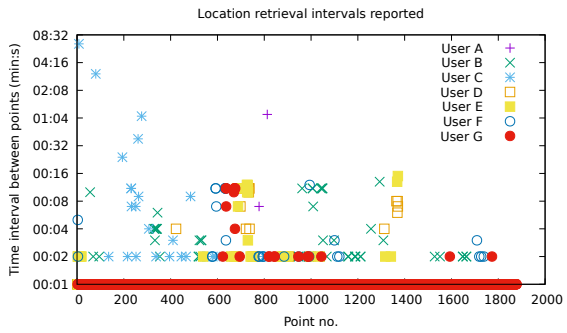

Figure 3: Location retrieval intervals reported

but the average distance was higher. All users of this course, except user GF, 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 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 location that fulfills this requirement, as most of these vehicles do not travel in a constant city environment.

5.2. Vehicle Selection

We consider that the parameters defined in our architecture and by the Authority user should be evaluated as they influence the selection procedure. As vehicles will be traveling at different speeds and we want to have an efficient application, we want to assess if a fixed location retrieval 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. 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. Figure 3 illustrates the reported location retrieval rates. The horizontal axis represents the number of the reported location point and the vertical axis represents the time interval the location point took to be retrieved.

For all users, which had a 1 second rate set, there were some points with a substantial interval, however most of the points are in the exact 1 second mark. This showcases why the average rate is above one second but the percentage of points that have not fulfilled the set rate is minimal. We presume that a substantial location retrieval interval occurs when the GPS signal is not satisfactory, the device cannot use mobile data or the device is optimizing the battery consumption.

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 with an Android method to get speed from a location point. [6]

Considering that the defined location rate could be fulfilled in the prototype, we assessed 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. We performed inspection selection tests simultaneously with 6 users. Two Inspect users were at one checkpoint each and the distance between the two checkpoints was higher than the defined inspection selection range of 500 meters. The six Transport users started 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 performed with the two devices side-by-side.

Out of the inspections performed, there was an occasion where a user that had just been inspected and was already on route was selected for inspection. This meant that the inspector had to wait for the user to return back to the checkpoint. The issue 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. We consider the implemented parameter for stopped vehicles should not be included. Additionally one improvement that could prevent this situation is to establish a minimum selection range. Vehicles too close to the checkpoint would not be considered for inspection and there would not be any risk of a vehicle being selected and not being able to stop on time. This value should be variable because of the velocity of the vehicles near the

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

checkpoint.

The rest of the inspections performed did not have any anomalies.

5.3. Bluetooth Inspection Interaction

We simulated an inspection area with a metal container similar to ones that carry goods in transportation vehicles. 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, while the inspector moves to perform the inspection.

A possible change to the protocol would be to divide it in two phases. 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 after the proof request is received 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 device optimization.

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.

We found that the designed Bluetooth protocol was not suitable in 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.

6. Conclusions

We presented STOP, a road transportation vehicle inspection support 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 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 several Android devices.

Acknowledgements

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

References

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

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

[3] 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.

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

[5] 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/.

[6] 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/.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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/.

[25] 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.

[26] 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/.

[27] 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.

[28] 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.