University of Twente

Master Thesis

Sousveillance on Intelligent Transportation Systems

Author:
Djurre Broekhuis

Supervisors:
Prof. Dr. Frank Kargl
Dr. Jonathan Petit
Michael Feiri, Dipl.-Inf.

December 2014

To Inti. . .

“ How long do you want these messages to remain secret?”
“I want them to remain secret for as long as men are capable of evil.”

Neal Stephenson, Cryptonomicon

“ In general you could not assume that you were much safer in the country than in London. There were no telescreens, of course, but there was always the danger of concealed microphones by which your voice might be picked up and recognized; besides, it was not easy to make a journey by yourself without attracting attention.”

George Orwell, 1984

UNIVERSITY OF TWENTE

Abstract

Faculty of EEMCS
Services, Cybersecurity and Safety Research Group

Master of Science

Sousveillance on Intelligent Transportation Systems

by Djurre Broekhuis

Intelligent transportation systems (ITSs) are an upcoming technology that allow vehicles and road-side infrastructure to communicate to increase traffic efficiency and safety. One part of such systems is cooperative awareness, where vehicles continually broadcast messages containing their location. These messages can be received by anyone, and can jeopardize location privacy. In this thesis we research how feasible it is to track a vehicle in an ITS in the presence of a mid-sized attacker, an attacker that has partial network coverage but can choose which parts to cover. We conduct an empirical study on the campus of the University of Twente by deploying ITS hardware on a small scale. We determine that road intersections are likely targets for an attacker to eavesdrop, and propose a graph based approach to determine which intersections an attacker should cover. We then analyse tracking feasibility using a route-based and a zone-based approach, considering both our empirical results and a theoretical expanded scale. Based on these results, we perform a cost analysis to give an indication of the financial resources an attacker needs to track a vehicle. We then look at pseudonyms as a mitigation strategy, and evaluate different pseudonym change strategies with different privacy metrics. We find that tracking a vehicle in the presence of a mid-sized attacker is feasible if such an attacker has sufficient resources to cover multiple intersections. We conclude that whilst pseudonyms cannot completely mitigate tracking, they do have a positive effect on location privacy and can increase the resources that an attacker requires to track a vehicle.

Acknowledgements

This thesis presents the work I have done at the Services, Cybersecurity and Safety research group at the University of Twente during a six month period. In this time, I worked with various people who contributed their time and effort to my research. First and foremost, this thesis could not have been completed in its current form without the ideas and advice of my supervisors at the University of Twente, Frank Kargl, Jonathan Petit and Michael Feiri. Beyond being supervisors, I would also like to thank them for involving me in the PRESERVE project, which allowed me to work with ITS hardware first-hand and enabled me to deploy this hardware for my experiments. Furthermore, I would like to express my gratitude to Geert-Jan Laanstra for his happily shared expertise and assistance, and the University of Twente security department for allowing me to fill their patrol vehicle with various equipment. My respect and gratitude also goes out to all the Kerckhoffs students and teaching staff that I have met, worked and learned with in the past two years. Their enthusiasm for all matters security related was a constant source of inspiration. Finally, I would like to thank my family, my girlfriend, and my friends for their continued support, not only during my thesis, but during the entirety of my education.

Contents

List of Figures

List of Tables

Abbreviations

BSM	**B**asic **S**afety **M**essage
CAM	**C**ooperative **A**wareness **M**essage
DENM	**D**ecentralized **E**nvironmental **N**otification **M**essage
DSRC	**D**edicated **S**hort **R**ange **C**ommunications
EMLR	**E**xpanded **M**ost **L**ikely **R**oute
EMLZ	**E**xpanded **M**ost **L**ikely **Z**one
GA	**G**lobal **A**ttacker
ICA	**I**ntersection **C**ollision **A**voidance
ITS	**I**ntelligent **T**ransportation **S**ystem
LA	**L**ocal **A**ttacker
LBS	**L**ocation **B**ased **S**ervices
LIDR	**L**inear **I**nterpolation-**D**ead **R**eckoning
MA	**M**id-**S**ized **A**ttacker
MTT	**M**aximum **T**racking **T**ime
MHB	**M**ulti-**H**op **B**roadcast
MLR	**M**ost **L**ikely **R**oute
MLZ	**M**ost **L**ikely **Z**one
OBU	**O**n-**B**oard **U**nit
OSM	**O**pen**S**treet**M**ap
PER	**P**acket **E**rror **R**ate
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
RSU	**R**oad-**S**ide **U**nit
SCAM	**S**implified **C**ooperative **A**wareness **M**essage
SHB	**S**ingle-**H**op **B**roadcast
TTFF	**T**ime **T**o **F**irst **F**ix
VANET	**V**ehicular **A**d-hoc **NET**work
QoS	**Q**uality **o**f **S**ervice

Chapter 1

Introduction

Modern vehicles are becoming increasingly equipped with a multitude of sensors that allow them to gather data on their surroundings. Vehicles may, for example, collect information about the temperature, road conditions or the distance to other objects and vehicles. Along with these sensors, vehicles are also starting to become equipped with wireless communication systems that allow them to communicate with other vehicles and infrastructure and set up Vehicular Ad-Hoc Networks (VANETs). Combining these two features allows for cooperative awareness and the development of advanced applications. These networked, context-aware vehicular networks along with their supporting infrastructure are often called Intelligent Transportation Systems (ITSs).

ITS applications can significantly improve driver safety and comfort, for example by providing warnings on road dangers or traffic jams, or automatically braking a vehicle when a collision seems likely. At the same time, vehicles collecting and sharing data about themselves and their surroundings gives rise to privacy issues. Many envisioned ITS applications rely on vehicles knowing the position of both themselves and their neighbours. Therefore, one sort of data that are periodically broadcast as part of cooperative safety applications are real-time location and trajectory beacons, a feature that most likely cannot be turned off. Broadcasting these data may jeopardise the location privacy of drivers by allowing them to be tracked.

On the one hand, tracking may be of particular interest to criminals when we consider certain classes of vehicles, such as police vehicles or money transports. For example, if burglars could track patrolling police vehicles they can wait until all police vehicles are outside of a certain area before attempting a robbery, which would increase the response

time before the police can be at the crime scene to intervene.

On the other hand, the deployment of ITSs also puts radio networking equipment into the hands of the car owners. In an age where surveillance on the general public seems to have become common place, ITSs may allow for a role reversal where the general public can record the activities of those usually doing the surveillance. This type of recording by the general public is called sousveillance, and the general setup can be seen in the cartoon in Figure 1.1. In an ITS where all cars are equipped with networking equipment, anyone is able to eavesdrop on messages from equipment in government or police vehicles, and use this to try to track them.

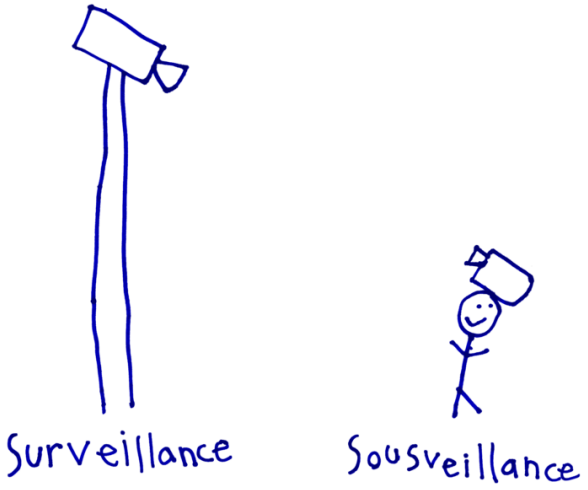

FIGURE 1.1: Difference between surveillance and sousveillance [1]

In this thesis we investigate empirically how feasible it is to track vehicles in an intelligent transportation system, by deploying ITS equipment on the campus of the University of Twente. Using data from this real-world experiment, we analyse different tracking methods that an attacker can employ. We subsequently investigate a theoretical expanded scale of the experiment, and describe tracking feasibility in terms of attackers of various levels of resources. By determining the requirements and resources of an attacker we give a cost analysis, giving us a realistic overview of how likely these attacks on privacy might be in reality. Finally we look at what can be done to mitigate tracking, looking at pseudonyms as a promising mitigation strategy. We describe the pros and cons of pseudonyms and to what extent they are effective in the context of our experimental data. We conclude that even though pseudonyms cannot eliminate the risk of tracking completely, they can still form an important line of defence. Through this thesis we hope to shed light on the complexities of location privacy in vehicular networks, and

more importantly, to raise awareness of the need to ensure such privacy in all upcoming ITSs.

The rest of this document is organised as follows: Chapter 2 gives a description of the system model, describing what components constitute an ITS, the security requirements of the system, and the classes of adversaries that we consider. Chapter 3 describes the objectives of our research, and states our research questions. Chapter 4 puts our research into context by examining the related work. Chapter 5 describes how the experiment was set up, and what decisions an attacker needs to make to track vehicles. Chapter 6 describes how the experimental data was processed and analysed, and how this data can actually be used to track a vehicle. Additionally, this chapter looks at what the effects are if a larger scale is considered, and gives a cost analysis. Chapter 7 looks at how tracking can be mitigated using pseudonyms, and evaluates the effectiveness of pseudonyms using different privacy metrics. Chapter 8 gives an overview of how we answered our research questions, and what future work remains to be done. Finally, it also gives our overall conclusions and final words.

Chapter 2

System Model

2.1 System Architecture

We consider a VANET consisting of both vehicles and supporting road-side infrastructure. An example of such a set up can be seen in Figure 2.1.

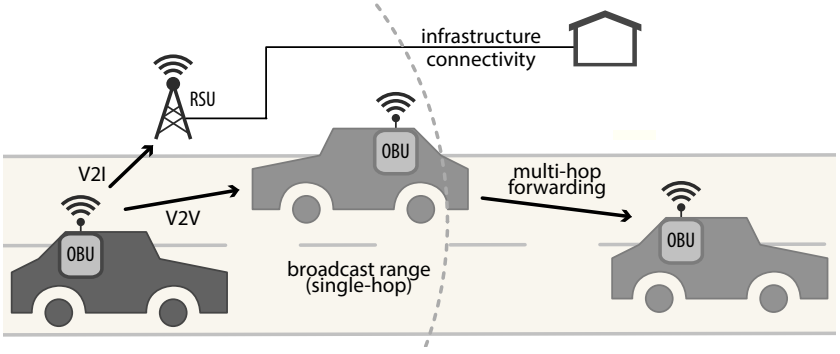

FIGURE 2.1: Typical ITS setup

To allow vehicles in a VANET to send and receive messages, they are equipped with a station called an On-Board Unit (OBU). An OBU typically consists of a car computer with networking hardware. An OBU can collect diverse sensor information such as vehicle trajectory data or road conditions, and process and send these data. Apart from the OBUs in the vehicles, there is also static infrastructure to improve data dissemination and to provide connectivity with back end systems. These static infrastructure stations consist of Road-Side Units (RSUs), which are similar to OBUs except that they are fixed in place and typically have additional network access.

Both OBUs and RSUs can send different types of messages to any other stations that are in range. The ETSI ITS standard defines two different types of facility layer messages

that vehicles can transmit, namely Cooperative Awareness Messages [2] and Decentralized Environmental Notification Messages [3].

Decentralized Environmental Notification Messages (DENMs) enable vehicles to send asynchronous warning notifications to vehicles, for example when there has been an accident or there are hazardous road conditions. DENMs are delivered to vehicles in the area affected by an event. Messages are forwarded using multi-hop broadcasts (MHB), where vehicles and RSUs may forward messages so that they reach the appropriate vehicles. DENMs are only sent when there has been a noteworthy event, and typically require reliable packet delivery. Cooperative Awareness Messages (CAMs) support vehicular safety and traffic efficiency. Their main purpose is to allow applications to know about the status of a vehicle or RSU. The ETSI standard specifies that these messages should be broadcast with a frequency of 1-10Hz [2]. CAMs are broadcast only to the immediate neighbourhood of a vehicles, and as such are single-hop broadcasts (SHB). A typical CAM includes the unencrypted latitude and longitude of a vehicle, its trajectory, a timestamp and an identifier.

In order to send and receive these messages, standardized equipment and protocols must be used that are suitable for vehicular environments. Due to high node mobility and short intervals of direct connectivity, VANETs have unique network requirements. IEEE 802.11p has been defined as a standard to take into account these requirements specifically for vehicular networks. 802.11p is an amendment to the IEEE 802.11 standard that allows for low overhead and quick connection setup, which is achieved by discarding all authentication processes [4]. To enable ITS applications, ETSI has allocated 30MHz in the 5.9GHz frequency band. Within this band, 802.11p can use channels of 10MHz bandwidth to send and receive data.

Security Requirements

The very same features that make ITSs useful may also be abused by attackers. For example, warning vehicles of hazardous road conditions is one of the envisaged applications of VANETs. However, this functionality may also be abused by an attacker purposefully reporting incorrect conditions. This could cause a vehicle to brake unnecessarily, which in turn could lead to accidents. Vehicles could also try to masquerade as other vehicle to try to escape liability in the case of an accident. These simple examples already indicate that in order to be able to use VANETs reliably, the communications

amongst vehicles and infrastructure must be secured. Papadimitratos et al. and Raya and Hubaux identify the following basic security requirements that VANETs need to satisfy for secure safety messaging [5] [6]:

- *Authentication and Integrity*: When a vehicle receives a message, it should corroborate that the sender is a legitimate vehicle in the network and that the message has not been changed. If this is not done, the receiver may react to incorrect messages which could cause a hazardous situation. Thus messages need to be checked for authenticity and integrity, for example by including signatures and certificates in the messages. Note that as 802.11p does not include any of the MAC layer authentication features that are present in standard 802.11, this needs to be implemented at higher layers such as the network layer.
- *Data Consistency*: Message legitimacy can also be analysed by looking at data consistency. For example, if the same legitimate vehicle sends two contradicting messages, then the receiver can question the legitimacy of these messages.
- *Availability*: Vehicles rely on network availability to receive safety messages. For example, if the network is not available during a hazardous scenario, then an accident may occur. Thus the network needs to be resilient against both network congestion and denial of service attacks.
- *Non-repudiation*: Senders should not be able to deny that they have transmitted any specific message, as which vehicle sends which message may be important when investigating a traffic accident.
- *Privacy*: VANETs should protect the personal and private information of its users. This means that this information should not be disclosed directly, but it should also not be possible to make inferences that reveal private data.
- *Real-time Constraints*: The high node mobility and short connectivity intervals of VANETs mean that there are real-time network limitations. Thus strict time constraints need to be adhered to, to ensure that vehicles receive the required messages.

Apart from the above, Schaub et al. and Bibmeyer et al. identify another security requirement, namely accountability [7] [8]:

- *Accountability*: When vehicle misbehaviour is detected or when there is an accident, it is desirable that authorities are still able to identify which vehicle was the culprit. This should be possible even when the identifier of a vehicle is not directly evident to protect location privacy.

In our system model we concern ourselves primarily with privacy, and in particular location privacy. However, privacy may also have a direct effect on other requirements. For example, privacy may be ensured by making all messages completely anonymous, but this contradicts the requirements of authentication and accountability. We will see later that non-persistent identifiers called pseudonyms are often suggested to provide location privacy, while still allowing for authentication and accountability.

Attacker

We have seen that CAMs are periodically sent by vehicles, and that they contain unencrypted location beacons. This means that an attacker with 802.11p equipment can receive these CAMs and possibly track any vehicle. In our system model, we consider an attacker using such equipment as sniffing stations. These sniffing stations can be deployed to areas to allow eavesdropping on any location beacons within range. A sniffing station is similar to an OBU or RSU, consisting of hardware which allows it to receive and process 802.11p packets. The main difference is that a sniffing station is not physically constrained to one place or vehicle, the attacker is free to place the station where he/she wants. The number of sniffing station that an attacker can have is limited by the available resources of the attacker. We will look more closely at these resources in chapter 6. An attacker can use these sniffing stations to eavesdrop on the positions beacons of any vehicles that are in range of a station. The next section describes which characteristics we consider to constitute an attacker.

2.2 Attacker Model

Similar to Raya and Hubaux, we identify 5 different properties that describe an attacker in our system model [6]:

- *Scope*: The scope is the area over which the attacker can eavesdrop. On one end of the spectrum is the *global attacker*, which has complete coverage and can eavesdrop on any message that has been transmitted in the network. On the other end of the spectrum is the *local attacker*. This is an attacker that can only cover one small area. In between these two extremes, we introduce the *mid-sized attacker*, which can cover any number of different local areas, without obtaining complete network coverage.

- *Passive/Active*: A passive attacker is only capable of receiving and processing any packets that it receives, whereas an active attack can also inject packets into the network.
- *Internal/External*: An internal attacker possesses keys and credentials that make it a legitimate participant of the system, whereas an external attacker does not.
- *Honest/Dishonest*: An honest attacker complies with the implemented protocols, whilst a dishonest attacker can deviate from them.
- *Tracking Period*: The tracking period defines over what period an attacker tries to link location samples and track a vehicle. We distinguish between the following:
  - *Short-term tracking* means that an attacker tries to link consecutive location samples occurring in a time frame of a couple of seconds. Given multiple location samples of different vehicles, the attacker tries to link the location samples to the specific vehicles that sent them.
  - *Mid-term tracking* means that an attacker tries to link position samples from a single trip. A vehicle trip is the entire time period from when a vehicle start a journey until it ends, and can be in the order of a couple of minutes to a couple of hours.
  - *Long-term tracking* means that not only does an attacker try to link consecutive location samples, but it is also tries to link different sets of location samples from different trips. Long-term tracking can cover a time period of over one day. For example, the attacker tries to identify that a police vehicle that was tracked in a certain area one day is the same vehicle that passes through that area the next day or a couple of days later.

We do not consider all the described attacker properties. Firstly, we do not consider active attackers. CAMs are broadcast without any interaction from other vehicles, and so actively injecting packets into the network is not necessary. For the same reason we also do not distinguish between honest and dishonest attackers, as for CAMs there are no communication protocols to comply to. We also do not make a distinction between internal and external attackers because the CAMs that we eavesdrop on are not encrypted, and can be read by both internal and external attackers. Finally we are not interested in short-term tracking, as we want to investigate where a vehicle is in a certain areas, hence covering a single trip to multiple trips.

Taking into account these limitations to the attacker model, this leaves us with three different attacker types: the Global Attacker (GA), Mid-sized Attacker (MA) and Local Attacker (LA). An LA is however in effect no different than an attacker physically

following a target to track it. A GA on the other hand can observe the entire area and is just as hard to defend against. A more interesting and more realistic scenario is that of the MA: an attacker that has partial coverage of the entire network, but is capable of choosing which parts of the network area it covers. How much an MA can cover is limited by the resources that the attacker has, and so how many sniffing stations it can deploy. Thus the attacker model for this research consists of an MA, with varying levels of available resources.

Chapter 3

Objectives & Research Questions

We consider an MA that has distributed yet limited coverage of a vehicular network, using sniffing stations to eavesdrop on CAMs containing position beacons. However, the positions of these sniffing stations will affect the coverage, and with it the strength, of an adversary. There are a number of variables that come into play when considering sniffing station placement. One aspect that may have a significant effect is how high the sniffing station is placed. A sniffing station at ground level most likely has a smaller coverage area than one that is placed higher. This leads us to our first research question:

- How does the vertical positioning of a sniffing station affect its coverage area?

A second matter is where to place the sniffing stations in the network so that an attacker can maximize network coverage at minimum cost. This problem is not all that dissimilar to the problem of determining where RSUs should be placed when deploying ITSs. The models that are used to determine the placement of RSUs may be adaptable to improve the coverage of an MA. This gives the following research question:

- How can an attacker determine where sniffing stations should be deployed?

An MA by definition does not have full network coverage. However, full network coverage may not be necessary to still allow for tracking a vehicle in the entire network. An attacker could use street-level knowledge to place the sniffing stations in areas that give the most information to assist in tracking. Intersections are often proposed as the optimal place to position sniffing stations, so an attacker could use knowledge of where intersections are to decide where to place the stations. However, real-world tests on these placement strategies are still lacking, leading to the research question:

- Is an MA that uses street level knowledge to only eavesdrop on intersections capable of mid-term and long-term vehicle tracking in a real-world scenario?

Next, we consider the effectiveness of mitigation strategies. Pseudonyms are most commonly proposed as (part of) mitigation strategies to increase unlinkability and so reduce the risk of tracking. However, the effectiveness of pseudonyms have mostly been studied in theoretical contexts and in the presence of a global attacker. To our knowledge, there have not been any privacy studies with actual 802.11p hardware and mid-sized attackers. To measure the effectiveness of pseudonym changes, there have been many proposed location privacy metrics. Real world experimentation would give an opportunity to validate pseudonyms in the context of these privacy metrics. This gives rise to the research question:

- How effective are pseudonyms as a strategy to mitigate tracking and how can this be measured?

Finally we look at the strength of an MA. We define the strength as the amount of resources an attacker has, and so the extent of the network that it can cover. The strength of an MA can thus lie anywhere between the LA and the GA. The results of the previous research questions can potentially be used to model a relation between the strength of an MA and its capabilities. This gives our final research question:

- What is the relation between the resources of an attacker and its tracking capabilities? How can this be modelled?

Chapter 4

Related Work

4.1 Related Work

Tracking mobile nodes has been an active area of research as mobile nodes themselves become more and more ubiquitous. In this section we give an overview of related work on tracking in VANETs and other domains, as well as mitigation techniques.

4.1.1 General Tracking

Mobile phones are the most prevalent type of mobile nodes and because of this it has been a popular topic for mobile tracking. Drane et al. examined the ability to derive position information from GSM signals, and analysed which features of GSM signals are relevant for positioning a mobile phone [9]. They identified propagation time, time difference of arrival, angle of arrival and carrier phase as different positioning techniques that can be used to determine the position of a phone. However, propagation time and time difference of arrival require accurately synchronised clocks between mobile phones and base stations.
They also defined two different types of positioning, namely mobile-based positioning and network-based positioning. Mobile-based positioning is where a mobile phone uses the signals transmitted by different base stations to determine its position. Network-based positioning on the other hand, uses signals transmitted by the mobile phone and received by the base stations to perform the positioning. A hybrid approach is also possible, which takes aspects from both of these methods.

Cell ID

Another way of positioning mobile phones is by Cell ID [10][11]. A Cell ID is a unique number used to identify each base transceiver station in a GSM network, and base transceiver stations continually broadcast their Cell ID. As a mobile phone continually receives these broadcast messages, it can approximate its position using the known geographical coordinates of the base transceiver station. However, as the distance that may be covered is large, the accuracy of this method is limited. Experimental results give an average accuracy of 500 metres in urban environments [10].

Signal Strength

Tracking mobile phones is also possible using the signal strength of static base stations. Chen et al. used the signal strength of GSM signals to estimate the location of a mobile device [12]. They analysed three different positioning algorithms in a real-world scenario. The first method uses the Cell ID of base transceiver stations which have a known location. By weighting the received signals with the received signal strength, an estimation of the location of the mobile phone can be made. A second algorithm that they analysed uses fingerprinting of received signal strengths. First a training phase takes place, where signal strengths from all base transceiver stations are recorded for all locations. A mobile phone can then search this index of radio fingerprints and locations, and choose the k fingerprints with the lowest Euclidean distance from the current radio fingerprint. The location of the device is then estimated as the average of the locations of these best k matches. A final positioning algorithm that they analysed uses a radio propagation model and Markov localisation. This method is similar to radio fingerprinting, but instead of a training phase, the fingerprints are created by using an abstract model of the signal environment. A sensor model is built to predict the signal strength at each location, and then a Bayesian particle filter is used to determine the likelihood of measurements and so estimate the mobile phone's true location.
They found that in a high fingerprint density area, the basic fingerprinting algorithm is most effective, with an average error of 94 metres. In lower density areas, the modelled fingerprint method works the best, with an average error of 196 metres.

One downside of the basic fingerprinting approach is that it is a deterministic process, where it is assumed that the signal strength does not change over time. Ibrahim and Youssef improved on these techniques by taking a probabilistic approach that they called CellSense [13]. Instead of taking an instantaneous reading of signal strengths at each

location during the training phase, a signal strength histogram is built up over time. As this significantly increases the training overhead, a grid-based approach is taken where a histogram is built up for each grid area instead of for each location. The location of a mobile phone can then be estimated by calculating the average location of the k most probable locations given the observed signals strengths. With these improvements, they found the average accuracy to be 30 metres and 105 metres for urban and rural areas respectively. Under the same urban conditions, this represents a 23.8% increase over the basic fingerprinting technique and a 157.1% increase over the modelled approach. In rural areas the improvements are 197.5% and 86.4% respectively.

Due to the limited accuracy, the above positioning methods are not suitable for vehicular networks. Safety applications in particular require sufficient accuracy to distinguish cars, and so this accuracy needs to be in the order of a couple of meters. For this reason, GPS is used in ITSs to establish vehicle positions with high accuracy.

Other Domains

Apart from tracking mobile phones, the above techniques have also been applied to other domains. Oka et al. used received signal strength measurements for tracking targets in wireless sensor networks [14]. As opposed to signal strength tracking in GSM networks, the target that is to be tracked sends out signals, and the signal strength is measured at the receivers. Time-of-flight measurements for localisation are used by the Cricket [15] and Active Bat [16] systems. Again there is a difference in whether the mobile node or the infrastructure performs the positioning measurements. In the Cricket system, a passive mobile device measures the time-of-flight from infrastructure transmitters. For the Active Bat system this is vice versa, with the mobile device sending out signals to a grid of static receivers. Both systems require line of sight between the transmitters and receivers, and thus require sufficient infrastructure for full coverage. This line of sight requirement means that this method of localisation is not suitable for vehicular networks.

4.1.2 General Privacy Issues & Mitigation

Being able to track a mobile node opens up the way for many different location based applications. However, these capabilities combined with the increasing ubiquity of trackable mobile devices raises legitimate privacy concerns. As such, there has also been work

done on identifying and analysing these privacy issues.

Location Privacy

Duckham and Kulik defined location privacy as a special type of information privacy which concerns the claim of individuals to determine for themselves when, how and to what extent location information is communicated to others [17]. They also identified three negative effects associated with a failure to protect location privacy, namely location-based spam, personal well-being and safety, and intrusive inferences. The latter is most relevant to the issue of tracking, as being able to identify at which times a person is at which locations allows for inferences of, for example, a person's political views, state of health or personal preferences [18].
Furthermore, they identified four different strategies for protecting location privacy. Firstly, regulatory strategies include rules, laws and fair information practices that allow people to control their location information. Secondly, privacy policies are trust-based mechanisms that rely on implementing parties to adhere to these. However, policies are not privacy enforcing, and are vulnerable to malicious behaviour. Third is anonymity, which dissociates information about an individual from that individual's actual location. A final strategy that they proposed was obfuscation. Here, the quality of information about a person's location is degraded to protect that person's location privacy. Regulatory strategies and privacy policies fall out of the scope of our research, and as such, we focus on technical solutions such as anonymity, pseudonymity and obfuscation.

Anonymisation

As mentioned above, anonymisation is offered as a potential solution to tracking, and in particular to intrusive inferences as described by Duckham and Kulik [17]. However, even when all identifiers are removed, anonymised location samples are not sufficient to mitigate tracking as there is a high correlation between successive location samples. There are well-established techniques to link consecutive location samples to create trajectories, and even to link these to individual people [19]. For example, Gruteser and Hoh used multi-target tracking to accurately link completely anonymised GPS location samples from 3 different people, and went on to successfully demonstrate the same attack on GPS data from 5 different people [20] [21]. Thus, naive anonymisation is not sufficient to solve the location privacy problem. Moreover, anonymisation conflicts with

various security requirements. For example, we have seen in Chapter 2 that accountability is an important requirement for ITSs. If all messages are completely anonymous however, then accountability is not possible. Pseudonyms aim to solve these problems by allowing an individual to be anonymous whilst keeping a persistent identity.

Perturbation

Gruteser proposed to increase unlinkability between consecutive location samples by a perturbation algorithm that aims to mitigate the problem of trajectory tracking [21]. They investigated a mechanism that prevents an adversary from tracking a complete individual path by introducing tolerable errors into location samples. In their setup, location samples are first sent to an anonymisation server, which acts as a proxy and forwards the data to Location Based Services (LBSs). These LBSs can then use the anonymised location samples. Thus they consider the privacy problem after transmitting these samples to an untrusted third party application service. However, to ensure that these services are still useful to a user, they aim to increase the level of confusion while still enabling statistical location-based applications.
The key idea underlying their solution is the concept of *path confusion*. Every time two nodes come into close proximity, the location samples of both nodes are perturbed so that there is a chance that the adversary confuses the two tracks. This is achieved when two nodes travel parallel to each other for a short segment; the location samples are perturbed so that it seems as if the paths cross. After this, it is harder for an adversary to distinguish which node is which from the location samples. There are however also a number of drawbacks to their proposed solution. Firstly, they formulate perturbations as a constrained non-linear optimisation problem, which results in a computationally complex system that is not feasible for deployment in a real-time information systems with large numbers of users. Secondly, adequate privacy only achievable if user density is sufficiently high. Unfortunately, perturbation is not suitable for location samples in CAMs, because vehicular safety applications rely on the position information that is broadcast being as accurate as possible.

CliqueCloak

Another solution to prevent tracking is by spatiotemporal cloaking; location samples are obfuscated in location and/or time to make it harder for an adversary to track an individual. The downside of this kind of obfuscation is that users generally obtain coarser

results from location based services, which means that additional local filtering is required, which in turn results in higher computational and network costs. Furthermore, temporal cloaking may increase network delays which may lead to a lower perceived quality of service (QoS) [22].
A framework for allowing a user to specify a level of cloaking according to the preferred level of privacy was proposed by Gedik [22]. In this framework, a user can specify per message the minimum level of required privacy as measured by the users k-anonymity, which indicates that the user is not distinguishable from k-1 other users [23]. Along with this, the user can also specify the preferred spatial and temporal tolerances as a set of anonymisation constraints. Before location samples are sent to an LBS, they first go through a message perturbation engine which performs the anonymisation and cloaking according to these anonymisation constraints. To determine which other messages need be considered and how much they need to be cloaked, they model the anonymisation constraints as a constraint graph. Two messages are connected in the constraint graph if they are sent by different mobile nodes and their spatiotemporal points coincide, taking into account the specified spatiotemporal tolerances. They then translated the problem into the problem of finding cliques that satisfy certain conditions in the constraint graph. As such, they called their system *CliqueCloak*. Similar to the perturbation method described above, VANET safety applications rely on accurate spatiotemporal samples and so CliqueCloak is not suitable in this case.

Mix-zones

A different way to make it harder for an adversary to determine which node is which is by using mix-zones [24]. A mix-zone is analogous to a mix-network as originated in the work of Chaum [25]. In a mix network, a mix-node collects n equal length messages, adds padding, reorders them by some metric and forwards them in the new random order, giving unlinkability between incoming and outgoing messages. A mix-zone, on the other hand, considers a Euclidean space without spatial constraints [26]. A set of k users enter in some order and change identifiers (or pseudonyms). No users leave before all users are in the zone, and they spend random time inside before exiting in different order. Assuming that inside a mix-zone the location cannot be tracked, this gives unlinkability between the old and new pseudonyms. A mix-zone works in a similar way for mobile nodes. A mix-zone is an area in which a mobile node does not request any location based information, and thus does not need to send its location. Assuming that a user changes to a new pseudonym on entering a mix-zone, applications that see a user emerging from the mix-zone cannot distinguish that user from any other who was in the mix-zone at

the same time. Thus nodes going into a mix-zone cannot be linked with those coming out of it. Mix-zones can be suitable for VANETs as well, as described in the next section.

4.1.3 Mitigation in VANETs

Mix-zones have also been proposed as solutions for tracking in vehicular networks. However, there are a number of issues that complicate the situation for mobile nodes and for VANETs in particular. Firstly, vehicles will not spend a random time inside a mix-zone, there is a correlation between ingress and egress times. Secondly, there is also a correlation between where vehicles enter and exit the mix-zone due to the spatial constraints of the roads themselves. This means that the transition probability is not uniform, but constrained by limited trajectory paths and speeds of travel. A node may enter a mix-zone with a known and predictable trajectory, which leaks information that may make it easier to link egress events with ingress events.

One simple way to model mix-zones in VANETs is to define any area that is not observed by an adversary as a mix-zone, as was done by Buttyán et al. [27]. Of course, it is almost impossible to detect which areas are and which are not covered by an observer. Freudiger et al. proposed to force the establishment of mix-zones at appropriate places in VANETs to achieve location privacy in the presence of randomly changing identifiers and a global passive observer [28]. The effectiveness of mix-zones depends heavily on the density of vehicles and the unpredictability of their whereabouts. Therefore they suggested to establish mix-zones at vehicle intersections, which generally have a high density of vehicles that change direction.

CMIX

Vehicular mix-zone were also proposed in the CMIX protocol of Freudiger et al. [28]. In their CMIX protocol, all legitimate vehicles in a mix-zone get a symmetric key from a RSU. Key forwarding is used to ensure that the vehicles already possess this key the moment they enter the mix-zone, which is essential for safety application. Once in the mix-zone, all messages are encrypted with this key, meaning that a global observer can no longer see the content of messages and the location information contained within them, resulting in unlinkability between vehicles entering and subsequently exiting mix-zones. Keys are updated when the mix-zone is empty. Unfortunately, this protocol does not protect location privacy from internal attackers. Any legitimate member of the

network can place a vehicle at one or more mix-zones and obtain the keys and decrypt the encrypted messages. The system also requires an authentication mechanism to ensure that only legitimate users can obtain the key. Furthermore, a GA can observe the ingress and egress of vehicles in mix-zones and get a probability distribution of possible mappings, which gives some information that may still make tracking possible. With an adversary that knows only the set of vehicles entering or exiting a mix-zone, the level of privacy is only dependent on the number of vehicles in the mix-zone. With a stronger adversary that also knows trajectory and timing information the level of privacy also depends on the delay characteristics of the intersection and the vehicle trajectories. To somewhat alleviate this problem, they propose using several mix-zones in a chain to create a mix-network. They show with simple simulations that unlinkability of individual mix-zones is generally low, but this can be greatly improved by chaining mix-zones. However, the performance of their system is heavily dependent on vehicle density, as less congestion can make vehicles easier to track. The protocol also assumes that all vehicles participate in the anonymisation process.

MobiMix

MobiMix aims to solve some of the shortcomings of the CMIX protocol, by taking into account the spatial constraints and limitations of the road network, the timing of vehicles entering and exiting a mix-zone, and the transitioning probability in terms of movement trajectories [26]. This prevents *timing attacks*, which rely on the correlation between ingress and egress times to decreases the anonymity set size as well as *transition attacks*, which estimate probability of each possible turn at an intersection.

To achieve this Palanisamy and Liu proposed to construct the mix-zone using different techniques [26]. The basic technique is the 'naive rectangular' method, where the mix-zone is a regular rectangle around an intersection. With this technique, all users in the mix-zone at the same time are in the same anonymity set. The CMIX protocol described above resembles this technique most closely, as all vehicles within range of an RSU are considered to be in the same mix-zone. The main downside of this method is that at the moment a vehicle enters the mix-zone, some vehicles in its anonymity set may already have been in the mix-zone for a much longer time, and thus are more likely to leave earlier.
A second mix-zone construction technique tries to solve this issue and is called 'time window bounded rectangular'. This is similar to the naive rectangular approach, but when a vehicle enters the mix-zone the anonymity set is assumed to include only those

vehicles that enter within a certain time window of that event. The size of this time window is based on the characteristics of the road junction. However, even taking this into account, information may still be leaked by differences in speed distributions (for example due to different road classes) which could lead to timing attacks.
A third mix-zone construction technique is 'time window bounded shifted rectangular'. This is similar to the method above, but it is not centred around a junction. Instead it is shifted so that it takes the same time from all directions to reach the centre of the junction assuming all vehicles travel at a certain mean speed. Thus it takes into account the speed characteristics of the road network. The downside of this method is that it does not perform well when vehicles deviate from the mean speed.
The last mix-zone construction technique that they proposed is the 'time window bounded non-rectangular' approach. This approach is again similar to the previous one, but now mix-zones start from the centre of the junction and only cover the outgoing road sections. The length of the mix-zone on each segment is based on the mean speed of the segment, the chosen time window and the desired level of privacy. They found that this last mix-zone construction is the most effective and immune to the timing attacks that are possible with the other techniques. However, it does require a mix-zone length of a couple of hundred metres on each outgoing segment.

Pseudonyms

Pseudonyms have also been offered as possible solution in VANETs to increase unlinkability between location samples. A pseudonym is an abstract identifier that a vehicle can use to communicate. Most theoretical models on pseudonyms originate from the work of Chaum [25]. Since then, a lot of work has been done to determine how pseudonyms can be used to mitigate the privacy problems of mobile node tracking. However, using a single abstract identifier still allows linking of consecutive location samples to each other and through this even to an individual. For example, Gruteser and Alrabady analysed one week of pseudonymised GPS traces from drivers in Detroit, and their home-finding algorithm was able to find plausible home locations for 85% of the drivers [29]. Krumm used pseudonymised GPS traces to determine the location of a driver's home with a median accuracy of 61 metres [30]. Using a reverse white pages lookup, they were able to correctly identify the correct home address of a driver 13% of the time and their names 5% of the time. To decrease this sort of linkability, pseudonyms can be changed periodically. Note that pseudonyms need to be changed on all communication stack levels to make sure that location samples cannot be linked by a persistent identifier.

Simple Pseudonym Change

How and when to change pseudonyms is still an open research challenge, and there have been many different proposed pseudonym change strategies [31]. Wiedersheim et al. analysed the effectiveness of simple change strategies, where a pseudonym is changed every message or every few seconds [32]. They considered a GA that can receive all beacon messages that are sent in the network. By using multiple hypothesis tracking and Kalman filtering on a large quantity of pseudonymous position samples, they tried to connect those samples to location profiles or tracks. Using simulations, they found that even when changing the pseudonym every message and sending a beacon every second, tracking is largely successful. However, it is unlikely that pseudonyms can be changed this often as vehicles will probably only have a limited number of pseudonyms and thus cannot change pseudonyms every message [33].
Increasing the time that a vehicle uses one pseudonym increases that chance of tracking success even more, and they find that with 20% of vehicles sending a beacon every second and changing pseudonyms every 10 seconds, a vehicle can be tracked almost 100% of the time. Thus it seems that simple pseudonym change strategies are not sufficient to ensure location privacy in the presence of a GA.

Swing & Swap

More complex pseudonym change strategies were proposed by Li et al. [34]. They devised two different pseudonym change strategies called Swing and Swap. Swing enables nodes to independently initiate and loosely synchronise pseudonym updates, whereas Swap is an extension of Swing which allows nodes to exchange identifiers. Both approaches are user-centric in that nodes can independently determine when and where to change pseudonyms to increase their location privacy, whereas with other solutions such as mix-zones this can only happen at fixed locations.

Swing improves location privacy because asynchronous location updates limit the location privacy provided by each update [17]. By initiating synchronised updates at opportune locations and times, the size of the anonymity set can be increased and tracking may be mitigated. The anonymity set of a node includes nodes that update their identifiers along with the initiating node and appear in the reachable area of the target. Swing works by a node first initiating a pseudonym change. This node then monitors the channel to ensure that the neighbourhood size is at least 1, and if it is it broadcasts a pseudonym change message. Other nodes may receive this message and choose to update

their pseudonyms as well, giving loosely synchronised updates within the neighbourhood of the initiating node. After changing its pseudonym, each node enters a random silent period where it no longer broadcasts any messages. In order to prevent an adversary from using the predictability of node movement to correlate node positions, pseudonym updates are only performed when changing direction and/or speed. Note that not all neighbouring nodes have to change their pseudonym when receiving an update message, as some may already be at their desired level of anonymity. As such, Swing does not account for neighbours that do not update their identifiers and which may decrease the size of the anonymity set.

Swap builds on Swing, but instead of nodes always updating their pseudonyms, they swap pseudonyms with probability of 0.5 and then enter the random silent period. With this method, neighbours of the target contribute to its anonymity set despite not updating identifiers, as long as they change their velocity and broadcast only during a specific interval in the exchange process. Nodes have an incentive to cooperate as they are provided with enhanced privacy enhancement whilst conserving the number of pseudonyms that they have. Swap may have a larger and more uniformly distributed anonymity set, but it does come with additional protocol overhead caused by the actual swapping of the pseudonyms and the additional identity management that is required. Using simulations, Li et al. found that Swap outperforms Swing, and that both are better than random pseudonym updates when it comes to location privacy. They also found that location privacy increases when the silent period is longer. Unfortunately, both Swing and Swap assume that a node can estimate when and where a trajectory change can occur, and Swap is only possible with extra infrastructure for identity management.

Mix Contexts

Gerlach and Guttler proposed a different method of synchronising pseudonyms changes to increase location privacy [35]. They introduced the concept of mix-contexts, where vehicles use context information such as the number of neighbours, their direction and speed to decide whether or not to change pseudonyms. Thus nodes cooperatively identify good opportunities to change pseudonyms, based on when the context allows for at least a certain amount of anonymity (for example, when a certain number of vehicles in range are travelling in a similar direction). After changing its pseudonym, a vehicle assesses whether the change was successful based on how many other nodes changed at the same time.

Silent Cascade

Another pseudonym change strategy is Silent Cascade, as proposed by Huang et al. [36]. Silent Cascade tries to use pseudonyms to achieve unlinkability between location samples without violating a user's QoS requirements. This method builds on silent periods, which increase privacy at the cost of losing communication time. Silent Cascade enhances location privacy while reducing this QoS degradation. Silent Cascade works with two states. In the active state, a node uses one specific pseudonym as a communication identifier. In the silent state, a node is not allowed to disclose either its old or new pseudonym, and thus is equivalent to a silent period. A silent cascade is then defined as a duration of time where a node switches between the silent state and active state periodically. Thus a node switches its operation mode from active state to silent state after each pseudonym update. After staying in silent state for certain period of time, the node switches back to active state so that it can communicate normally. Afterwards, the station iteratively switches its operation mode between active state and silent state. Each time the node enters the silent state, it introduces ambiguity into the time and place when the pseudonym change occurred. The maximum amount of time that a node can stay in either of these states it determined by the QoS requirements. In effect, this create a chain of mix-zones as described in [27]. Silent Cascade adds an additional trade-off parameter to basic silent period. Whereas with basic silent periods there is a trade-off between anonymity and QoS, Silent Cascade adds a third parameter in the form of the silent cascade delay, which allows it to ensure a user required level of QoS.

CARAVAN

The CARAVAN scheme attempts decrease linkability between location samples by increasing the length of the silent period [37][38]. Sampigethaya et al. propose to do this by allowing vehicles to form groups, where vehicles are defined to be in a group if each group member can receive the broadcasts of every other group member. Then, since vehicles in a group move relative to each other and have on average the same velocity, the group can be seen as a single large vehicle represented by the group leader. The group leader then communicates on behalf of all vehicles in the group, and the other vehicles can extend their silent period for as long as they are a member of the group. Group members can also use the group leader as a proxy to increase unlinkability. Unfortunately they only analyse their scheme with a freeway model and a simple street model. This may not capture the true mobility of vehicles, the dynamic nature of which can adversely affect the formation and membership of groups.

Pseudonym Trade-offs

With the many different proposed pseudonym change strategies, it is important to consider what trade-offs come with introducing pseudonyms into an ITS. Lefevre et al. investigated the effects of pseudonym changes strategies on an intersection collision avoidance (ICA) system [39]. To do this, they simulated an intersection and analysed the effects of three different pseudonym strategies: fixed-id, baseline and adaptive. In the fixed-id case there are no pseudonym changes at all, but a vehicle uses one long term pseudonym. In the baseline case pseudonyms are changed every 120 seconds and each change is followed by a random period between 0 and 13 seconds. The current SAE J2735 standard for Dedicated Short Range Communications (DSRC) proposes a silent period of 50 to 250 metres or 3 to 13 seconds, whichever comes first [40]. Finally they proposed the adaptive strategy, which is similar to the baseline strategy except that a pseudonym change is only authorised if it does not affect the safety application. They analysed the effectiveness of these three strategies using the rate of missed accident interventions, the rate of avoided collisions, and the rate of failed interventions. They found that silent periods longer than 2 seconds strongly affect ICA applications, and that the adaptive approach only authorised average of 10 percent of pseudonym changes when the silent period was larger than 2 seconds. This indicates that whilst pseudonym changes and silent periods may be beneficial for location privacy, they may also have an impact on the main functionalities of an intelligent transportation system.

Pseudonym Effectiveness

To determine how effective changing pseudonyms are, Buttyán et al. defined a model that allows this to be studied [27]. To do this, they defined all areas that are unobserved by an adversary as a mix-zone. As vehicles do not know when they are in mix-zone, pseudonyms are constantly changed. They assumed that this rate of change is high enough that pseudonyms are changed at least once per mix-zone. Under this simplifying assumption they simulated vehicles in a part of Budapest covering 59 road junctions, attempting to give a relationship between strength of the adversary and level of location privacy achieved by changing pseudonyms. The adversary strength was varied by eavesdropping on the k busiest junctions, with an eavesdropping range of 50 metres. Different traffic densities were simulated and then they quantified the success of the adversary by calculating the number of successful tracking attempts. Tracking was considered a success when a vehicle entering a mix-zone was correctly linked to a vehicle exiting that mix-zone. Linking was done using a basic dead reckoning approach where the probability of linking the correct vehicle was based on the speed and distance covered in the

mix-zone. They found that tracking success was independent of vehicle density, and that tracking was successful 60% of the time with only a few tens of eavesdroppers.

Cloaking

Apart from pseudonyms, cloaking has also been proposed in the context of vehicular networks. Gruteser and Grunwald proposed both spatial and temporal cloaking [41]. For spatial cloaking, they proposed a quadtree based algorithm that decreases the location accuracy until the anonymity set is as large as required by an anonymity parameter. To allow for more accurate spatial accuracy, they also proposed temporal cloaking, where information requests are delayed until at least a certain number of vehicles have been in an area. They simulated their cloaking algorithms with vehicles in a road network, and found that spatial accuracy quickly decreases with an increase in the required anonymity set size. However, they consider 125 metres to be sufficient accuracy, which for modern ITSs is not the case.

As a final note on privacy strategies in vehicular networks, Schaub et al. gave a good overview of the privacy requirements in vehicular communications systems, and categorised the possible solutions [42]. Petit et al. focussed on pseudonyms in particular, and gave an extensive overview of the current state of the art of this research area, as well as proposing a pseudonym lifecycle [31]. A classification of attacks on privacy solutions was given by Wernke et al. [43].

4.1.4 RSU Placement

One of the main parameters in tracking vehicles in vehicular networks is the placement and density of sniffing stations to maximise coverage. This problem is analogous to the problem of RSU placement in ITSs which also aim to maximise coverage at minimal costs, and is strongly related to the range of an RSU.

Road Position

The first matter to decide on when placing RSUs is where they should be placed on the road. Trullols et al. simulated realistic vehicular mobility over a simple road topology.

They measured the number of vehicles that came in range of an RSU and the time that they were in range, with RSUs located at different positions on the road [44]. They found that placing RSUs at intersections performed better than placing them in the middle of road sections between intersections, independent of the reception/transmission range of the RSU. They then modelled the problem of which intersections to place RSUs at using two different methods, under the assumption of intermittent RSU coverage. First they modelled the problem as a Maximum Coverage Problem, maximising the number of vehicles that come in range of an RSU at least once. Secondly they also modelled the case in which the duration that a vehicle is in contact with an RSU had an impact on the dissemination of information, and so aimed to maximise both the number of contacted vehicles as well as contact times. As both of these problems are NP-hard, they proposed heuristic algorithms as a solution. They found that simple heuristic algorithms can give near optimal performance, but that this can only be achieved when the characteristics of vehicular mobility in the covered area are known.

Although placing RSUs at intersections seems to result in the best connectivity, Kafsi et al. note this does not decrease the proportion of vehicles that are isolated from the network [45]. Isolated vehicles are more likely to be in the middle of road sections or at entry points to a road. Thus RSUs placed at intersections will not be in range of these vehicles and they suggest placing RSUs in the middle of road sections if the aim is to benefit these vehicles.

For our research we will consider intersections as a possibility to place sniffing stations. However, we will also consider the effect of height in sniffing station coverage, as potentially a sniffing station placed high up but away from an intersection might provide for better coverage than a sniffing station placed lower but on an intersection.

Density Based

Barrachina and Garrido proposed a density-based approach to placing RSUs, where more RSUs are deployed where there is a higher vehicle density [46]. This approach aims to maximise performance in notifying emergency services of an accident whilst minimising deployment costs. They simulated a section of Madrid to compare 3 different RSU deployment strategies. The minimum cost strategy only deploys RSUs where there is already existing infrastructure such as network connections to do so. The uniform mesh approach deploys RSU's over the area with a uniform distribution. Finally, in their

density based approach RSUs are deployed with a density that is inversely proportional to the expected traffic density. They found that the density based approach performed better than the uniform mesh approach with higher vehicle densities, but that uniform mesh performs better at low densities. They also found that the density-based approach requires 2 RSUs per km^2 for 100 percent notification with low equipment density.

Minimising Costs

Liang et al. proposed a framework for optimal deployment and configuration of RSUs to minimise financial costs [47]. In their framework, each RSU can individually select configuration settings based on the surrounding environment, the traffic density, the network connection offerings and the overall cost. They then formulated the problem of where to place the RSUs and how to configure them as an optimisation problem to find a trade-off between network coverage and cost. The framework also takes into account the effects of buildings on propagation and the effect of road topology on RSU antenna configurations. Using a path loss formula, they estimated an RSU's range to be 243-309 metres with an omni-directional antenna, with transmission powers ranging from -10dBm to -6dBm.

Data Dissemination

Lochert et al. looked at data dissemination in terms of the number of vehicles that participate in the vehicular network, called the equipment density [48]. They then simulated traffic in a city to compare three different RSU placement strategies: random placement, placement at areas with the most successful information transfers and placement in areas with high traffic density. They found that placing RSUs at areas with high traffic density is the best for data dissemination. They concluded that RSUs are necessary in the initial VANET rollout phase, as they can then compensate for low equipment densities.

Kone et al. also looked at data dissemination in vehicular networks to examine RSU density in a highway scenario [49]. They measured data dissemination by looking at the effective dissemination range and the reception percentage of messages, with the goal of determining the minimum required RSU density to support vehicular applications. They concluded that with an equipment density of 10%, data lifetime can be 15 seconds

with RSUs 100 kilometres apart. Unfortunately, they only consider delay-tolerant applications, and the large distance between RSUs is mainly due to multi-hop broadcasts where each vehicle forwards messages to other vehicle to increase data dissemination range. This is not the case for location samples in safety messages as these are only interesting for neighbouring vehicles.

Further Improvements

Rashidi et al. looked at the trade-offs between the size of the gaps between RSUs and other system parameters such as the data delivery ratio [50]. They considered the case where cars buffer their data when they are not in range of an RSU, so that data is collected over a time period and then bundled. They concluded that packet delay and packet loss are directly influenced by RSU placement, and suggest 1 RSU every 5 kilometres to ensure s 95% delivery ratio. However, they limit their study to a highway environment and only consider delay-tolerant applications which focus on reliability of data delivery.

Lee and Kim designed an RSU placement scheme to reduce the disconnection interval of vehicles as well as the overlap of RSUs in a city [51]. They took into account road network organisation and real-life vehicle movement data. Their placement scheme considered all intersections as candidate positions. In their scheme, all candidate positions were tried and the number of vehicles in range were counted. The candidate positions were then sorted by the number of vehicles in range. Then, starting from the top, each candidate position was considered and all other candidate positions were removed if they fell within range of this position, taking into account a certain amount of overlap. They simulated this placement scheme with a road network of Jeju city, and found that with 1000 RSUs with a range of 300 metres, connectivity was 72.5% and the disconnection time below 10 seconds. They also found that each 100 metres extra range increased the connectivity by +17%.

Lochert et al. proposed to use genetic algorithms to determine the best RSU placement for a cooperative traffic management system. Along with data dissemination they also look at data aggregation [52]. They used a hierarchical structure, where the farther away a region is, the coarser the traffic information will be. For many different candidate positions for RSUs it is infeasible to try all different placement combinations and

see which result in the best performance. For this reason they proposed a genetic algorithm that starts off with a random set of RSU placements and then creates a new generation of placements based on the largest decrease of travel time resulting from the traffic information aggregated by the RSUs. However, in all their simulations they only use an average equipment density of 25%.

Unfortunately, the above approaches on RSU placement seem to all only cover the situation of multi-hop broadcasts. Location samples on the other hand are only interesting for vehicles in the immediate vicinity, and so CAMs are sent as single-hop broadcasts. In the next chapter we propose a graph-based approach for an attacker to determine sniffing station placement to maximize the available information that can be used for tracking.

Chapter 5

Experimental Setup

One of our research questions asks if an MA that uses street-level knowledge is capable of tracking a vehicle in a real-world scenario. Such a real-world scenario requires at least two elements, namely a vehicle to track and an attacker with sniffing stations whose aim it is to track this vehicle. Apart from these, there are also a number of parameters that define the scenario, such as the geographic domain where the attacker wishes to track the vehicle and the resources available to the attacker (which determines the number of sniffing stations that this attacker can deploy).

In the introduction we identified that one of the use-cases in which an attacker may want to track a vehicle is when this vehicle is a patrolling police car. Knowing when a police car is where can be desirable information for a criminal. To emulate this use case, we attempted to track the vehicle used by the security team of the University of Twente. This vehicle is used to patrol the university campus and therefore is a similar target. This also solved the issue of which geographic area to track a vehicle in. As the campus security vehicle mostly drives on the university campus, this was used as our *tracking domain*. Apart from the vehicle, we also needed to emulate an attacker with sniffing stations. Similar to a real attacker, we had limited resources in this regard. More specifically, we had available two sniffing stations which we could deploy on the university campus to track the security vehicle.

In this chapter we describe the set-up of the experiment which we conducted, looking at tracking the campus security vehicle in the area of the campus of the University of Twente. We describe the ITS hardware that we deployed and the messages that this hardware sent and received. Finally we describe how elevation and different types of antennas affect coverage and how we determined where to place our sniffing stations.

5.1 Hardware

To deploy our 802.11p systems on the university campus, we required two different types of hardware. The first type was the hardware that could be equipped in a vehicle. This hardware needed to be shock proof and able to be powered through the 12V connection of the car. Furthermore, it needed some way to be securely installed in a vehicle without moving around too much. The second type of hardware was that of the sniffing stations. As these were static stations, they did not need to be as rugged as the sending station. This hardware had to be relatively small and easy to install in any location. Having an internet connection was not a strict requirement but it was preferable to allow for remote monitoring of stations. An attacker could also use an internet connected sniffing station to see which vehicles are in range of which sniffing stations, remotely and in real-time. Finally, both types of station needed to have 802.11p connectivity and some form of local storage to store log data.

5.1.1 Sniffing Station

For the sniffing stations we used the Cohda Wireless MK3 platform, which we call the Cohda Box. This is a small, ARM based computer running Linux. With its built-in 802.11p radio, the Cohda Box allows two antennas to be connected for 802.11p connectivity. Apart from this it also includes an Ethernet port for internet access. Due to the limited local storage on the device, we added a 2GB SD car to store logs on. A photo of the Cohda Box can be seen in figure 5.1. We had access to two Cohda Boxes for this experiment, allowing us to deploy two sniffing station in the same configuration. For the antennas, we had a choice between low-gain and a high-gain antennas. The low-gain antennas were a MobileMark MRM3-5500, with a gain of 2.5 dBi. The high-gain antennas were Smarteq V09/54 antennas with a 9 dBi gain. Both of these covered the frequency range required for 802.11p. We decided to use the high-gain antennas for our sniffing stations. Section 5.3.1 describes our experimentation with both types of antennas and our reasons for choosing the high-gain antenna for this experiment.

5.1.2 Sending Station

As a transmitting station for inside the vehicle, we used a Nexcom VTC6201, which we call the Nexcom Box. This is a fanless in-vehicle computer with an Intel Atom D510 processor. It was expanded with a Unex CM10-HI Mini-PCI module for 802.11 a/b/g connectivity. Using custom drivers this also allowed for 802.11p connectivity. This module allows for the connection of two antennas and it has an SMA connector for a

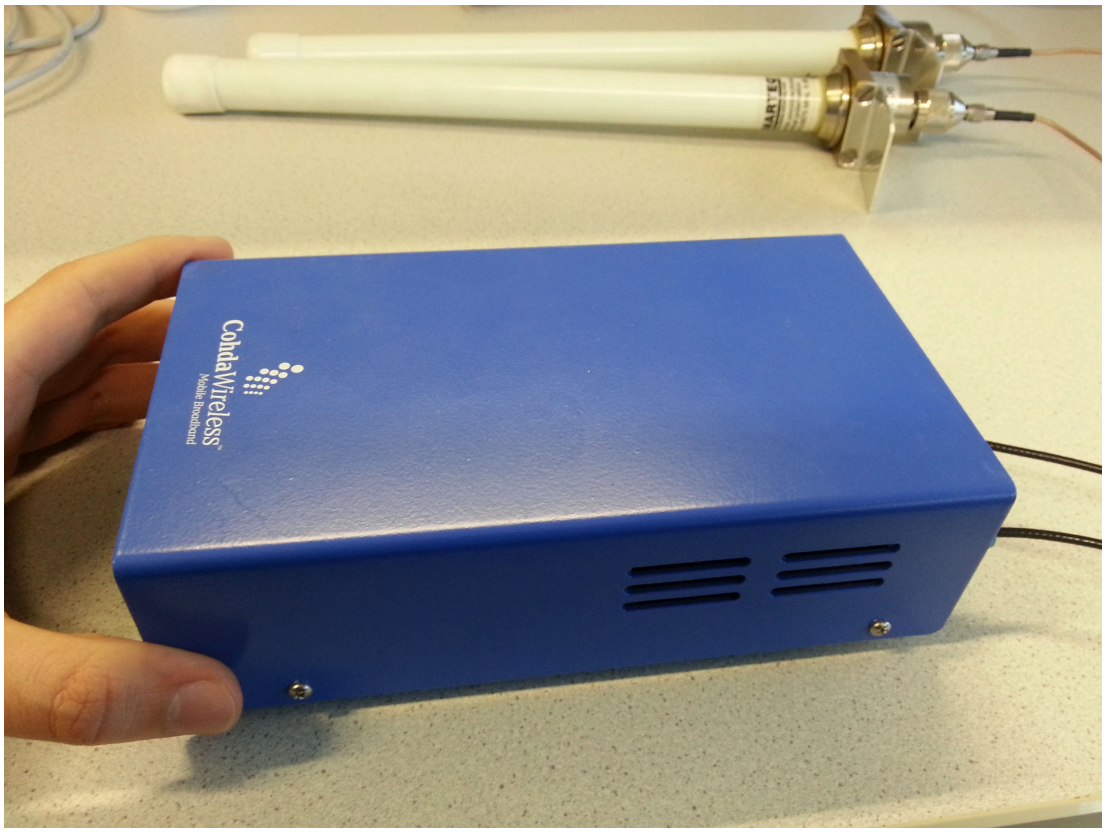

FIGURE 5.1: The Cohda Box used as a sniffing station

GPS module. Ubuntu 12.04 was installed on the built-in SSD. To transmit messages, we used two MobileMark ECOM9-5500 antennas. These are high-gain 9dBi antennas, with a magnetic mount so that they can easily be fixed to the roof of a vehicle. These antennas cover a frequency range of 5.0-6.0 GHz and thus are suited for 802.11p.

The Nexcom Box was powered by the 12V connector of a vehicle. However, this meant that as soon as the vehicle turned off, the power to the computer would be cut and it abruptly turned off as well. To prevent this from happening, we added a battery buffer and a battery charger. The complete set-up can be seen in figure 5.2. The in-vehicle computer can be seen on the right hand side, whereas the battery charger and battery itself can be seen on the left. All equipment was screwed onto a mounting board which could be placed securely in the trunk of the vehicle.

5.1.3 Power buffer

The battery acted as a power buffer for the Nexcom Box. When the ignition of the vehicle was turned on, it charged the battery. The battery in turn provided power to the Nexcom Box. When the ignition was turned off, the charger stopped charging the battery, but the Nexcom Box remained on, still powered by the battery. If the ignition

FIGURE 5.2: The battery, battery charger and Nexcom in-vehicle computer

was not turned on again within 3 minutes, the computer performed a clean shutdown. If the ignition was turned on again within this 3 minute window, the shutdown procedure was aborted, and the computer remained on. The reason for the time window was two-fold. Firstly, it allowed the computer enough time to perform a clean shutdown when the vehicle's ignition was turned off. Secondly, it prevented the computer from having to reboot when the ignition was off for only a short period of time (for example if the engine was turned off when waiting for green light, or if the vehicle stalled). Such a reboot would take time where messages could not be sent. Furthermore it would also cause the GPS module to lose its position fix, and re-obtaining such a fix can take a considerable amount of time, as we will see in section 6.1.1. Thus filtering out unnecessary reboots reduced the time when the vehicle sent messages without a GPS fix.

5.2 Simplified Cooperative Awareness Messages

To track vehicles, we assumed an attacker eavesdropped on the radio messages that the vehicle transmitted. Therefore it was important to define what these messages were. In the current ETSI ITS standards, cooperative awareness messages and their formats are defined, to support vehicular safety and traffic efficiency. These messages are

transmitted at a frequency of 1-10Hz, and can include information from the trajectory of a vehicle, to ambient air temperature and whether alarm lights are in use on emergency vehicles. For our purposes however, standard CAMs consisted of much more information than we needed; for tracking vehicles, a lot of the values transmitted in CAMs were inconsequential. Most important for tracking was information such as the exact location, speed and heading of a vehicles. For this reason, we introduce the Simplified Cooperative Awareness Message (SCAM), containing only the information that is relevant for vehicle tracking. The information described in these messages is a subset of ETSI standardized CAMs, and so any analyses based on SCAMs are also valid in the case of CAMs.

A SCAM is a 56 byte message, which is transmitted with a frequency of 10Hz. The structure of a SCAM can be seen in figure 5.3. A SCAM includes a unique identifier for each station as well as a sequence number that is reset to 0 every time the transmitting station starts up. It also includes a time of generation timestamp with microsecond resolution. As SCAMs are sent every 0.1 seconds, this timestamp uniquely identifies each message. Furthermore, the message includes the vehicle's GPS fix, which consists of its location, the speed and the bearing of the vehicle. Finally there is some extra information on the reliability of this fix. A full description of the SCAM fields can be seen in table 5.1.

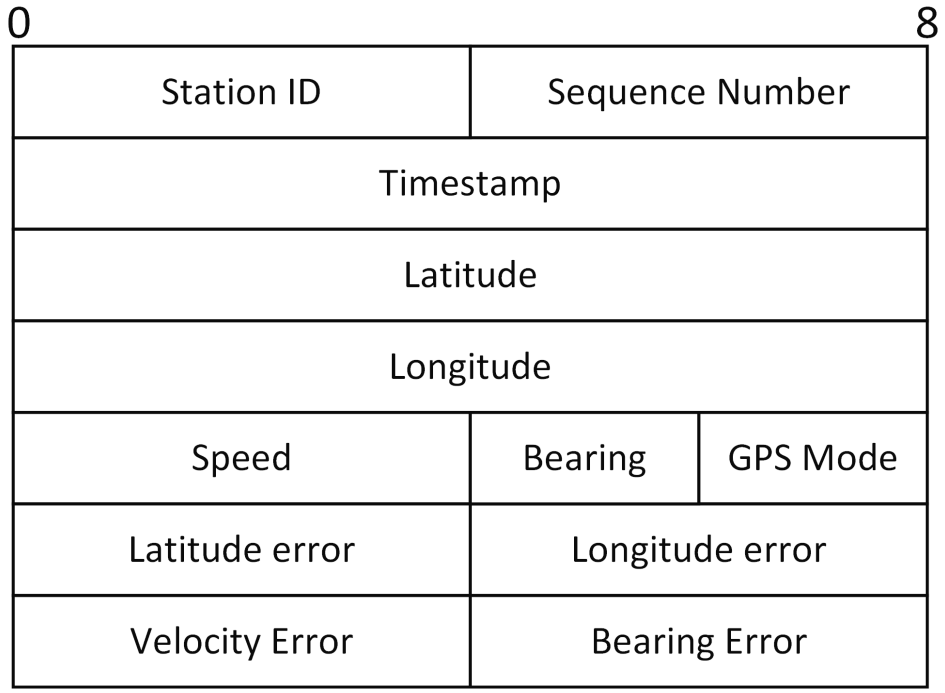

FIGURE 5.3: The format of a SCAM

SCAMs are generated by a sending station, where the location data in a SCAM is collected from a GPS receiver. SCAMs can be received by sniffing stations that are in range of a sending station. In order to analyse the sent and received SCAMs offline,

Field Name	Length	Description
Station ID	4	Unique number identifying the station
Sequence Number	4	The sequence number of a packet in a trip. Sequence numbers start at 0 when the station boots, and resets when the station turns off
Timestamp	8	The microsecond precision timestamp when the message was sent. Bytes 0-3 are the number of seconds that have passed since the Unix epoch, bytes 4-7 the number of microseconds in that second
Latitude	8	The latitude of the location in decimal degrees, using the WGS84 datum. Bytes 0-3 are the whole number part, bytes 5-8 are the fractional part
Longitude	8	The longitude of the location in decimal degrees, using the WGS84 datum. Bytes 0-3 are the whole number part, bytes 5-8 are the fractional part
Speed	4	The speed of the station, in meters per second
Bearing	4	The bearing of the station, in degrees from north
GPS Mode	2	The fix mode of the GPS. 0 = No mode value seen yet 1 = No fix 2 = 2D fix 3 = 3D fix
Latitude Error	4	Latitude error estimate in meters, 95% confidence. Present if mode is 2 or 3 and dilution of precision values can be calculated from the satellite view.
Longitude Error	4	Longitude error estimate in meters, 95% confidence. Present if mode is 2 or 3 and dilution of precision values can be calculated from the satellite view.
Velocity Error	4	Velocity error estimate in meters per second, 95% confidence.
Bearing Error	4	Bearing error estimate in degrees, 95% confidence.

TABLE 5.1: Description of SCAM fields

all SCAMs in our experiment were logged to a file. For the sending stations, these logs represented our ground truth, the actual position of the vehicle. The logs collected by the sniffing stations then represented the knowledge that the attacker gained by eavesdropping on these messages. Due to the limited range sniffing range, this is a subset of the ground truth. This is the information which an attacker could use to track a vehicle.

5.3 Preliminary Testing

5.3.1 Antenna Gain

One important aspect of the hardware was the antennas that were used to send and receive messages. In particular, different antennas can have a different *gain*. The gain of an antenna describes how well this antenna converts input power into radio waves headed in a certain direction. Antenna gain is specified as the peak gain stated relative to an isotropic radiator, or an antenna that radiates equally in all directions. Thus a high gain antenna will radiate strongly in one direction, but due to conservation of energy this means that it will radiate less strongly in other directions. A low-gain antenna on the other hand will not radiate exceptionally strongly in one single direction, but makes up for it by radiating more uniformly in all directions. One way to visualize this antenna gain relative to direction is with an elevation radiation pattern diagram. The elevation pattern of a low-gain antenna can be seen in figure 5.4 on the left. The pattern for a high gain antenna can be seen in the same figure on the right. These graphs show the gain of an antenna relative to peak gain at 0 dB, at different vertical angles. Note that these are still omnidirectional antennas, and they radiate equal power in all directions perpendicular to the antenna, with the power varying only with the angle to the axis.

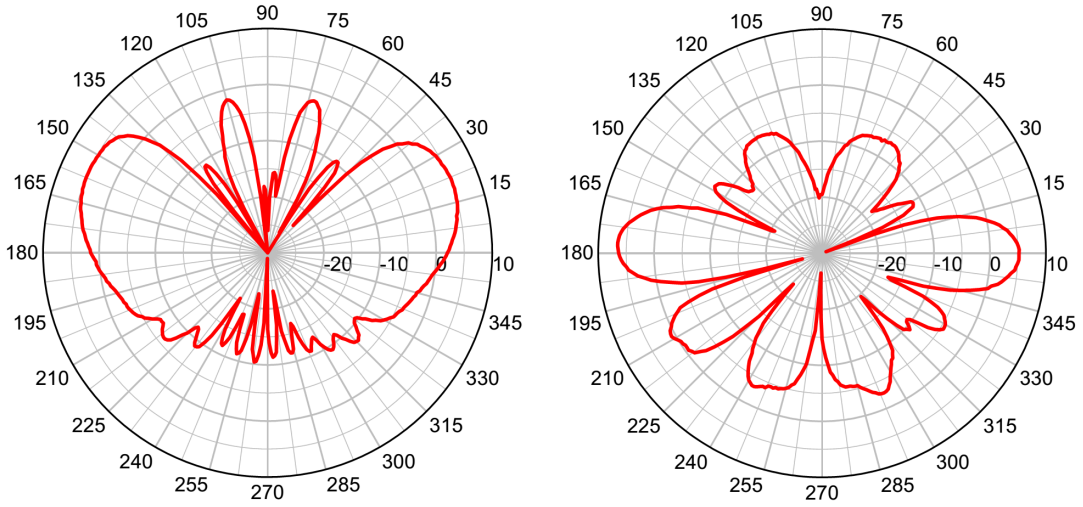

FIGURE 5.4: Elevation radiation patterns of a low-gain (left) and high-gain (right) antenna

As we can see, the high-gain antenna has a large peak at 0° and 180°. This means that the antenna radiates most strongly in a horizontal plane, whereas it does not radiate as strongly in other directions. In contrast, the low-gain antenna is not influenced as much by the angle, and this is shown by the larger lobe at each side of the graph. It is important to note that these diagrams show the gain relative to the peak gain for that antenna model only, and so cannot be used to compare radiation power between

different models directly. For example, the peak in the high-gain radiation pattern represents 9 dBi compared to an isotropic radiator, whereas for the low-gain radiation pattern this is 2.5 dBi. In general this means that such a high-gain antenna will radiate much more strongly at the peak angle than a low-gain antenna. In the case of our antennas this meant that the 9 dBi antenna would radiate $10^{\frac{9-2.5}{10}} = 4.22$ as strongly in its peak direction as the 2.5 dBi antenna. One final note is that up to now we have only mentioned antenna gain in terms of how strongly an antenna radiates in a certain direction. However, the gain also characterizes antennas receiving signals. In that case, the gain expresses how well an antenna can turn radio signals into an electrical signal relative to the signals direction of arrival. Thus instead of describing radiation in a certain direction, it describes reception in that direction.

The choice of which antenna to use depends largely on the usage scenario and the environment in which it is to be placed, as we discuss below where we look at the effect elevation has on an antenna's reception.

5.3.2 Elevation

One parameter that may have a big influence on the reception range of a sniffing station is the height at which it is placed. The higher an antenna is, the less likely it is that there are line-of-sight obstructions between the transmitter on the vehicle and the receiver. Due to the high frequency of 802.11p, any obstruction will significantly reduce signal strengths, and so it is desirable to have as few obstructions as possible. Given this information, it might seem logical to place the receiver as high as possible. However, the higher an antenna is placed, the larger the distance between sender and receiver. This increased distance will cause a decrease in signal strength.

For an attacker it is beneficial to cover as large an area as possible, as this gives more information that can be used to track vehicles. If an attacker has multiple sniffing stations, then the gaps of no reception between sniffing stations will be smaller if the range of these stations is made as large as possible. This in turn means less time where an attacker needs to guess where a vehicle is and more time where an attacker knows exactly where a vehicle is by eavesdropping SCAMs.

For this reason, we carried out an experiment to determine what effect height has on the reception of SCAMs. For this elevation experiment, the sending station was placed at ground level. The height of the sniffing station was then varied by moving it up and down the emergency stairs of an 8-storey building, as seen in figure 5.5. This staircase was on the outside of the building and so gave an unobstructed, line-of-sight view to the sending station below. For each floor, the horizontal distance between the sending

FIGURE 5.5: The building used to perform the elevation experiment

station and the sniffing station was varied in steps of 5 meters, from 0 meters up to 25 meters. Then for each combination of height and horizontal distance, the sending station transmitted for 100 seconds, thus sending 1000 SCAMs. The receiving station then recorded the packet error rate (PER), which is the percentage of packets that

were successfully received. Additionally, the received signal strength indicator (RSSI) was recorded for each packet. The experiment was conducted at first with high-gain antennas for the sniffing station, and then repeated with the low-gain antennas. The resulting packet error rates, averaged over the different horizontal distances, can be seen in figure 5.6.

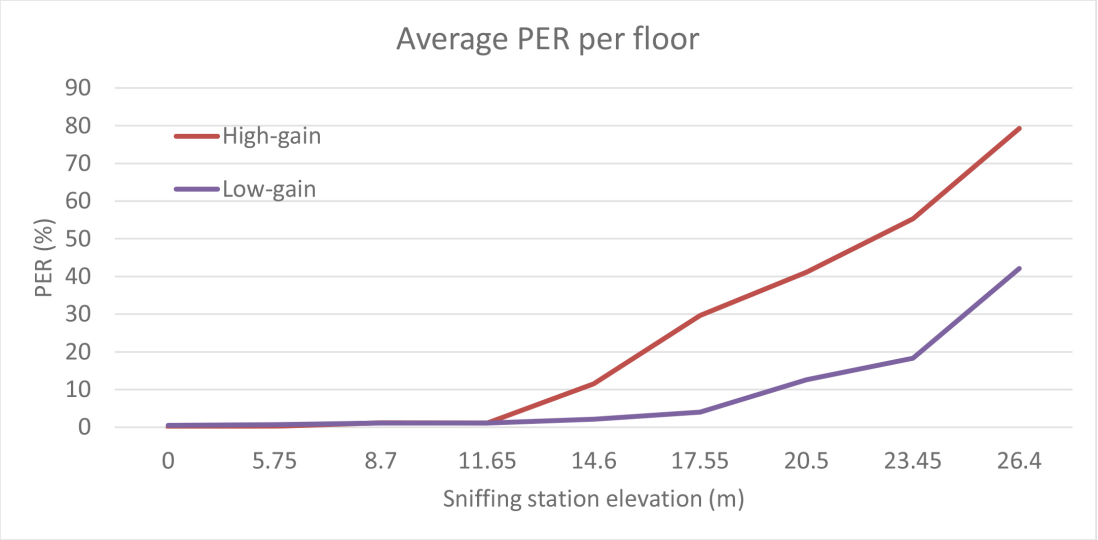

FIGURE 5.6: Average PER per floor for high-gain and low-gain antennas

It can be seen that the higher the sniffing station is placed, the greater the PER becomes. This is largely due to the reception pattern of the antennas as discussed in section 5.3.1; at higher elevation, the angle between the transmitter and the receiver becomes greater, and the reception sensitivity goes down. For the low gain antennas, we see that the antenna is much less sensitive to the angle, and continually performs with a lower PER at high elevations. However, for low elevation the low-gain antennas have a much shorter range. For example, at the maximum distance of 25 meters and at ground level, the high gain antennas had a PER of 0%, whereas the low-gain antennas already had a PER of 0.5%. Due to the lower reception sensitivity of the low-gain antennas, this difference would only increase as the distance at ground level was increased.

It is also important to realize that, due to the high message frequency, a high PER does not necessarily mean that it is impossible to track the vehicle. For example, with a PER of 80%, the probability that a sniffing station will receive at one least packet in a second (i.e. with 10 messages sent) can be calculated with a binomial trial:

$$
\begin{aligned}
P(X \geq 1) &= 1 - \binom{a}{b} p^k q^{n-k} \\
&= 1 - \binom{10}{0} \cdot 0.2^0 \cdot 0.8^{10} \\
&= 1 - 0.107 \\
&= 0.893
\end{aligned}
$$

This means that even when receiving only 20% of all transmitted packets (a PER of 80%) there is still a probability of 0.893 that a sniffing station will receive at least one packet per second.

Apart from the PER, we also looked at the RSSI of the two different antennas. This can be seen in figure 5.7

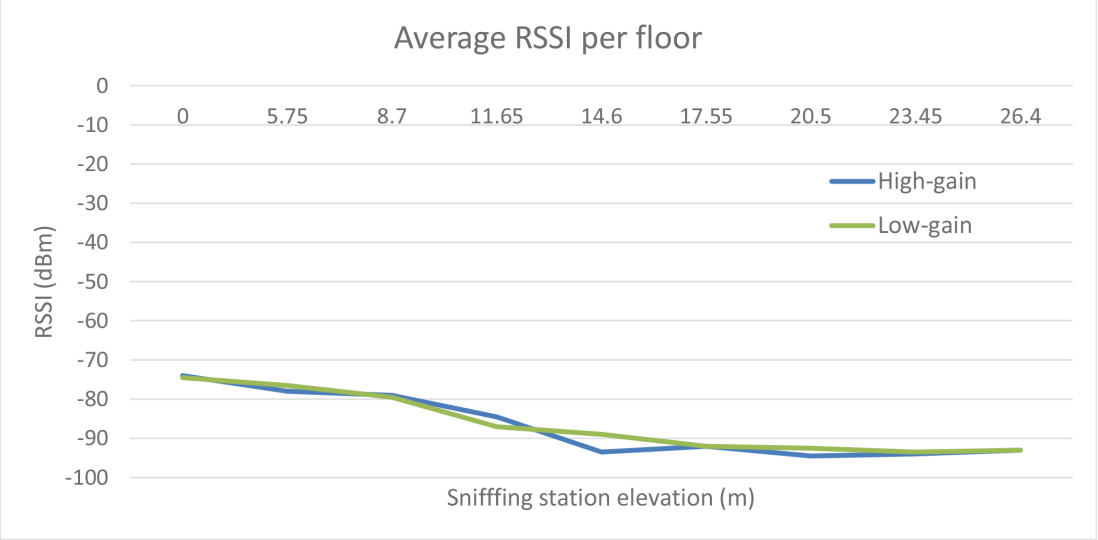

FIGURE 5.7: Average RSSI per floor for high-gain and low-gain antennas

Here again it can be seen that the RSSI drops as the elevation of the sniffing station increases. It is however also evident that the RSSI was not a good indicator of the probability of a sniffing station receiving a packet. For example, for the high gain-antenna, the RSSI stayed at about -94 dBm when the receiver was higher than 14.6 meters. As the sniffing station was placed higher however, the PER ranged from around 10% to 80% whilst the RSSI stayed approximately the same.

	Low	High
Close	High-gain	Low-gain
Far	High-gain	High-gain

TABLE 5.2: Types of antennas to use for different situations

So where should an attacker place a sniffing station to maximize reception range? This depends on the environment where the sniffing station is to be deployed. Table 5.2 shows what kind of antenna is best to use, depending on if the antenna must be placed high or low, and far away or close to an intersection. If the environment is such that the sniffing station can only be placed relatively high and close to an intersection then it is advisable to use low-gain antennas. Using a high-gain antenna in this situation results in a large probability that messages may be missed due to the large angle between the antenna and the station. On the other hand, if the environment allows for an unobstructed close view of the coverage area at a low elevation, than it is advisable to use high-gain antennas as this maximizes reception. For observing an intersection that is far away, it is always best to use a high-gain antenna, as the large distance will always give a relatively small angle. The only caveat here is that a low positioned antenna that is far away may have more obstructions, but of course this is dependent on the exact usage environment. Thus for optimal sniffing results, antenna elevation would need to be evaluated on a case by case basis. The question then is not whether or not elevation has an effect on reception range and coverage area, but rather whether or not the environment supports or opposes placement of a specific type of antenna at a specific elevation.

5.4 Sniffing Station Placement

The next question is how to determine where the sniffing stations should be placed geographically. Intersections have been proposed as a good location to maximize the number of vehicles that come within range [44] [51]. An additional advantage of observing intersections is that the turn-off that a vehicle takes at an intersection in large part determines its route until it reaches the next intersection, where it can turn again. Of course if there are unobserved intersections in between, then there is always the chance that the vehicle takes a turn onto a different road before the next observed intersection is reached, but it does allow some inferences. For example, if an attacker can observe two consecutive intersections, and a vehicle is observed in range of the first intersection and then in range of the second intersection within some time frame, then it can reasonably conclude what road the vehicle took without needing complete reception coverage over this road. This means that an MA might be able to successfully track a vehicle by using its limited coverage and street-level knowledge of the road network between coverage areas. A second advantage of intersections is that they often give a wide unobstructed view of the roads connecting to the intersection from different directions.

5.4.1 Graphing the Road Network

An MA by definition has some limitations to its resources and the number of sniffing stations it can deploy. This means that an MA also needs to choose which intersections to cover, and which to leave as uncovered gaps. To determine which intersections on the University of Twente campus may give the most information that can be used to track a vehicle, we represented key intersections and interconnecting roads as a graph. Intersections are represented by vertices in this graph, and interconnecting roads as edges. The resulting graph for the campus can be seen in figure 5.8.

FIGURE 5.8: Turning intersections into a graph

The graph gives an abstracted view of the road network, and not all roads and intersections are included. For example, when there are two routes between two adjacent intersections (and there are no roads leading from these roads to other intersections) then this is represented by a single edge. Thus the graph gives a high-level idea of which intersections connect to which other intersections, without being concerned with

the smaller details of the actual road network. An intersection covered by a sniffing station can be represented by removing the corresponding vertex from the graph. The remaining graph then represents where a vehicle can travel freely without being in range of an attacker.

We utilized this graph to determine which intersections to cover, and defined a number of criteria to help with this. Firstly, an intersection with a large number of connecting roads gives information on all of these roads. With the speed and bearing of a vehicle, an attacker can know exactly which road a vehicle came from and is going to, allowing an attacker to infer a vehicle's position over this larger number of roads. In the graph this translates to vertices with a large degree (i.e. an intersection with a large number of connecting roads), giving more information than vertices with a low degree. Thus, sniffing station placement should focus on those vertices that have the largest degree in the graph.

A second criterion we looked at were the so called articulation points of a graph. An articulation point in a graph is a vertex that when removed will completely partition the graph into different biconnected components. This is useful for an attacker, because if this vertex is covered, then the attacker will always know in which biconnected component of the graph a vehicle is. In other words, there would be no route that a vehicle could take from one biconnected component to another biconnected component, without passing through the intersection that the attacker observes. This allows an attacker to narrow down the position of a vehicle to a certain section of the area within which it wants to track this vehicle.

A third and final criterion is that it is beneficial to cover the busiest intersections, as vehicles are more likely to pass by these, resulting in a larger total time that a vehicle is in range and thus more information that can be used to track a vehicle.

5.4.2 Determining Placement

Using these criteria, we can determine which intersections were good candidates for sniffing stations in our coverage domain of the university campus. Looking at the graph, we can see that there is a single vertex that has a degree larger than the others, namely the vertex labelled 'A' which has a degree of 5. This is also the main entrance to the university and thus a busy intersection, complying to the third criterion as well. Therefore this intersection was chosen as the location to place one of our sniffing stations. For the placement of the second sniffing station, we identified articulated points. The vertex labelled 'B' was not an articulated point in and of itself. However, having decided that we would cover intersection A, we could remove the corresponding vertex from the

graph. This gave the situation as shown in figure 5.9(a), and in this situation intersection B did become an articulated point. This meant that covering both intersections A and B split the entire graph into two different biconnected components where vehicles could travel without crossing an intersection with a sniffing station, as shown in figure 5.9(b). Furthermore, these two biconnected components consisted of the residential part of the campus (the western biconnected component) and the university part of the campus (the eastern biconnected component). Vehicles thus could not travel from one section to the other without being observed by a sniffing station, giving the attacker insights which could compromise privacy.

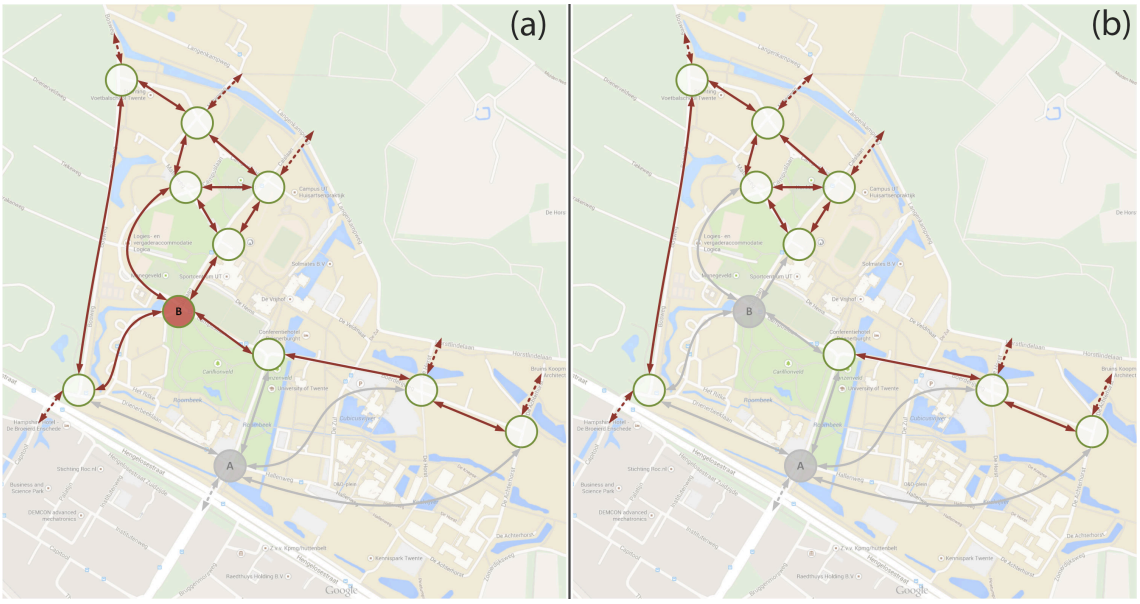

FIGURE 5.9: Intersection graph after covering (a) vertex A and (b) vertices A and B

Having decided which intersections to cover, we also needed to determine where at the intersections to place them. To give as good a coverage of the intersections as possible, the sniffing stations needed to be placed close to the intersections. They also needed to be placed somewhere where they were protected from the elements and preferably with an internet connection to allow for remote log retrieval and checking the operational status. This limited the placement to buildings that were near the relevant intersections. For our intersections this led to a simple choice, as there was only one building near each of the intersections that could be used.

A final choice then was where in the building to place the stations. We already saw from the height experiment that the optimal location for the antennas of a sniffing station was near ground level, so that the angle between the sending vehicle and the receiving antenna is as small as possible. One caveat here is that line-of-sight must not be compromised. Given these criteria, we decided to place the sniffing station at a ground floor office at intersection A, as can be seen in figure 5.10.

FIGURE 5.10: Sniffing station placement at intersection A

This position gave an unobstructed view of the intersection, as the roads of the intersection are higher than the parking lot in front of the building, and the antennas of the sending station on top of the car gives enough clearance over any parked vehicles. The distance between the sniffing station and the centre of the intersection was approximately 75 meters. The sniffing station at intersection B was placed by a window on the first floor, as the ground floor windows did not give an unobstructed view of the intersection. The distance between the intersection and the sniffing station in this case was approximately 110 meters. The placement of this sniffing station can be seen in figure 5.11.

One important thing to note is that both sniffing stations did not actually have a completely unobstructed view; due to the indoor placement, there is always a window between the transmitting and receiving station. We found windows typically give a signal attenuation of approximately 2-3 dB. However, due to the relatively small distance between the intersections and the sniffing stations this would still allow sufficient coverage of the intersection itself, and a part of the connecting roads, to determine which road a vehicle came from and went to.

FIGURE 5.11: Sniffing station placement at intersection B

Chapter 6

Experimental Results

The transmitting station in the vehicle and the two sniffing stations were deployed for a total of 16 full days. During this time, approximately 300MB of SCAM data were collected on all stations combined. The vehicle logged all transmitted SCAMs representing our ground truth. The sniffing stations logged all eavesdropped SCAMs, representing our observed data. This is the data that an attacker then had available to track a vehicle.

In this chapter we describe the collected data and which steps were necessary to clean up and process the data prior to analysis. From the cleaned and processed data, we propose a most likely path analysis to determine which road the vehicle took, and then a zone-based analysis to determine in which region the vehicle was. Next, we expand these analyses to take into account attackers with different levels of resources. Finally we give a cost analysis to determine what resources an attacker would need to improve its tracking capabilities, and discuss expanding the scale to an even larger geographic area.

6.1 Collected Data

To get an idea of what the data looked like, we first inspected the collected logs. In the 16 days that the experiment ran, the vehicle took 411 trips, and transmitted 2,734,691 SCAMs in a total time of approximately 76 hours. The logs from the sniffing stations on the other hand contained just over 68,542 eavesdropped SCAMs. This meant that the sniffing stations managed to pick up messages from the vehicle for a total time of approximately 1.9 hours, and that only 2.5% of all transmitted messages were eavesdropped. On average the vehicle drove for 4.75 hours per day, of which 7.1 minutes within range of

a sniffing station. Although this percentage seems low, we will see in Chapter 6 that it still allows for some degree of tracking. To examine the mobility pattern of the vehicle, we looked at the transmission log. We first looked what time of day trips typically took place. This can be seen in figure 6.1, which shows a histogram of trip departure times. We can see that the vehicle was utilized relatively evenly throughout the day and night. We also looked at how long the vehicle's trips took, a histogram of which can be seen in figure 6.2. The vast majority of trips were relatively short, with most trip falling between 300-400 seconds. There were no trips less than 3 minutes, as the battery buffer ensured that even if the ignition was turned on and immediately turned off again, the transmitter would keep running for 3 minutes. The average trip length was 655 seconds, though the battery buffer causes a slight bias. Discarding any trips that lasted exactly 3 minutes, the average trip length was 677 seconds. From these statistics, we can already see that the campus security vehicle does not follow the mobility pattern of a regular passenger vehicle, which will typically be used for longer trips. We will see later that the unusual mobility pattern of the security vehicle affects our tracking analyses.

Unfortunately, the collected data was not of sufficient quality to be used out of the box, and needed to be cleaned up and processed first. In the rest of this section we describe data quality and completeness issues that we encountered, and how we solved them.

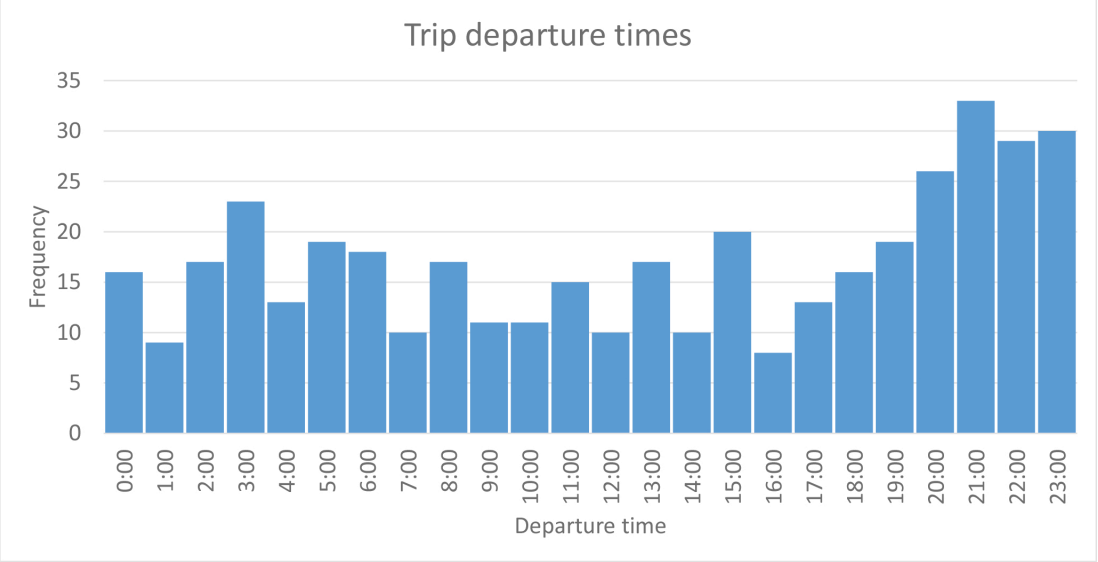

FIGURE 6.1: Trip departure times

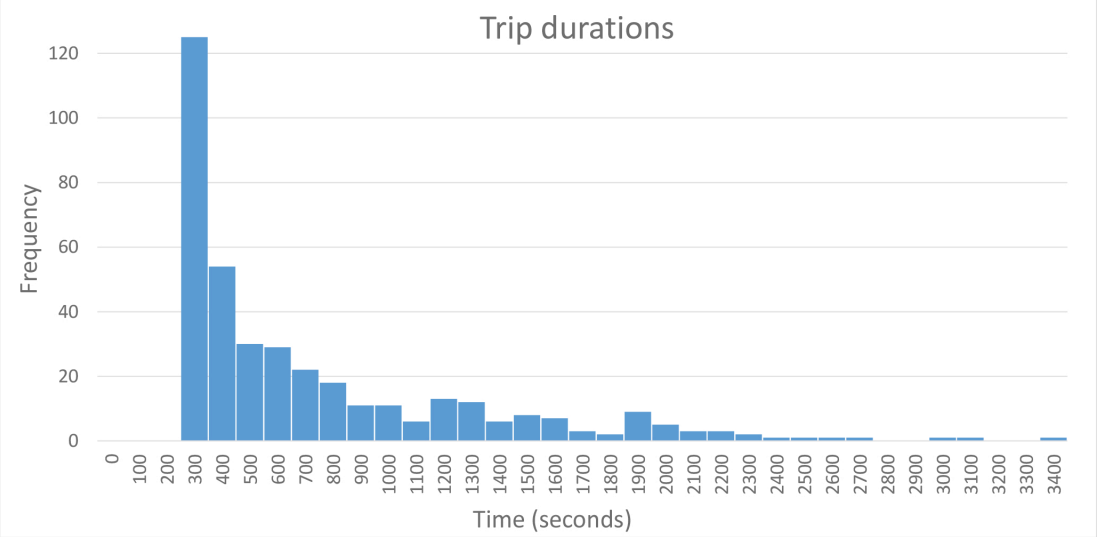

FIGURE 6.2: Trip durations

6.1.1 Data Clean-up

GPS Issues

One problem with consumer grade GPS receivers is that it can take some time before a first location fix is established, called the time to first fix (TTFF). This is due to the fact that before a receiver can obtain a position fix, it needs to receive almanac and ephemeris data. The former describes the status and information of all satellites, and allows the receiver to determine which satellites are in view. The latter describes the precise orbital data of a satellite and allows the receiver to determine its position. The TTFF largely depends on whether the receiver already has some valid almanac or ephemeris data. In our case, the OBU completely shut down when the vehicle's ignition was turned off, meaning that every time the OBU booted, it needed to re-obtain fresh satellite data. The TTFF for a cold start (starting without valid almanac or ephemeris data) depends on the exact environmental conditions, but due to the transmission rate of GPS data signals (50 bps) and the size of ephemeris data (900 bits), this takes at least 18 seconds. This is however an absolute best case scenario, and in practice the TTFF takes quite a bit longer. For example, Lehtinen et al. found an average TTFF of around 33 seconds for various consumer grade GPS receivers starting from a cold start [53]. After obtaining the first fix, it takes even longer for the error of the fixes to settle. For this reason, Lehtinen et al. advise to wait at least 30 seconds after the first fix has been obtained to ensure high accuracy position data. This time combined with the TTFF, meant that it could take more than a minute after the vehicle started until there were fixes that could be useful for tracking. In our case, this delay could comprise of a significant portion of the vehicle's trip. We saw in figure 6.2 that most trips taken

by the campus security were short, with the largest number of trips taking between the 300 and 400 seconds. For a trip that takes 300 seconds, a one minute delay until a useful fix already consisted of 20% of the total trip time. To solve this, we could have asked the driver to wait until a good fix was obtained before starting a trip, however this was not done as it would have been a burden to the drivers.

Apart from the initial delay, we also encountered additional GPS fix problems. Inspecting the log files from the vehicle, we found that there were some long trips where no fix was obtained at all. Furthermore, there were some trips where a large part of the trip there would be no fix, except for a few short periods of around 10 to 30 seconds where there was a fix. This happened on multiple occasions on open roads, disbarring direct environmental influences. We did not encounter such issues while testing the equipment outside of a moving vehicle, and perhaps can be attributed to a faulty or low quality GPS receiver. For our analyses, SCAMs without position information were not useful for tracking. As such, any SCAMs without a GPS fix were discarded and removed from both the transmission and reception logs.

Removing Uninformative Messages

The next step in cleaning up the log files was to identify any SCAMs that did not give any useful tracking information and prune them from the log files. Firstly, there were trips where the vehicle did not actually drive, but the ignition was turned on causing SCAMs to be sent. These trips were identified in two different ways, firstly by looking at any period of time in which SCAMs were sent but the position information indicated that the vehicle did not move, and secondly by filtering out any messages that indicated that the vehicle was travelling slower than a certain speed threshold. All these non-driving messages were not interesting to track and so they were pruned from the logs.

After cleaning the log files of both the vehicle and the sniffing stations as described above, we obtained a significantly smaller set of results. For the vehicle's log, we started with 2734691 transmitted SCAMs. Cleaning this up removed 53.56% of these, leaving us with 1270016 SCAMs. This represented about 38.24 hours of useful driving data. For the sniffing stations, we eavesdropped on 68542 SCAMs. After cleaning, 40254 eavesdropped SCAMs remained, a reduction of 41.27%. Of these remaining messages, 18293 were received at intersection A at the main entrance of the university, and 21961 were received at intersection B. Our eavesdropped messages then consisted of 3.17% of all transmitted SCAMs, and covered about 1.1 hours of vehicle driving time.

6.1.2 Data Processing

Closing Small Gaps

After cleaning up the data, we processed it to remove small gaps. A gap is any period of time where there is no location information. Large gaps may occur when the vehicle is out of range of any sniffing stations, and we will later see how tracking can be used to infer where the vehicle was in those gaps. Small gaps, on the other hand, may occur when not all messages are received due to obstructions or when there is general packet loss due to a low received signal strength. However, if these gaps were short enough, this packet loss did not necessarily lead to loss of information. The GPS receiver that we used established a new fix every 1 second, but the SCAM frequency was 10Hz. This meant that 10 SCAMs were sent with the exact same location, speed and heading before a new location fix was established. If any of these 10 messages were received, then the missed messages could be recreated and the attacker knew where the vehicle was during that second. Thus gaps that were shorter than 1 second did not result in loss of information. Of course, the vehicle itself did not stay in the same position in this second. To solve this, we used linear interpolation to infer the vehicle's actual position. With linear interpolation, a line is drawn between a position sample from one second and the subsequent position sample from the next second. The vehicle is then assumed to be somewhere on that line, depending on the time. Thus at the start the vehicle is assumed to be at the first position sample, and then one position sample later (1/10th of a second), the vehicle is assumed to be 1/10th of the way along that line. This continues until after one second, the vehicle is at the location of the next position sample.

The situation becomes more difficult if a gap lasts for more than 1 second, as this does result in loss of information. To close these gaps, the attacker needs to guess where the vehicle would be in that gap. Of course the larger the gap, the larger the error in this prediction. Therefore we focused on closing relatively small gaps. We investigated two ways to reconstruct the position beacons of the vehicle between two received messages with a gap between them that was larger than 1 second. The first of these was linear interpolation, in the same way that we used it to smooth out repeated position samples. The second method that we looked at was dead reckoning. Dead reckoning considers the last known speed and heading of the vehicle, and then extrapolates that into the future. Thus it looks at where the vehicle would have been, had it continued at the same speed and in the same direction. Whereas linear interpolation requires a next known position to interpolate to, dead reckoning does not. The downside is that dead reckoning also does not use the knowledge of the next known position sample, and so the dead reckoning prediction may have a large error by the time of the next location sample. Figure 6.3

shows this error. For every location sample in our ground truth, we calculated how long it took until the error between the prediction and the actual location was larger than the maximum error shown on the x-axis. The y-axis shows the average of these times, using all possible location samples in our ground truth as a starting point. The error bars show one standard error. We can see that the average tracking time increases by about 1 second for every extra meter of error. For a maximum error of 9 meters, dead reckoning can track a vehicle for approximately 8.5 seconds, averaged over the entire ground truth. The large standard error, especially for longer tracking times, is due to the large difference in dead reckoning performance in different situations. For example, tracking time can be long when the vehicle travels in a straight line, but a lot shorter when the vehicle turns a corner.

FIGURE 6.3: Dead reckoning tracking time

Linear interpolation is better in this respect as by the time the prediction gets to the next known location sample, the actual location is known and the predicted error is reduced to 0 again. However, linear interpolation does not take into account the vehicle's speed and heading (except insofar as it is implicit in the next known position sample), which may give a larger error in the beginning.

For this reason we propose LIDR (Linear Interpolation - Dead Reckoning) weighted averaging, which takes into account both the starting speed and heading of the vehicle as it leaves the location sample, and the information from the next known location sample. It tries to reconstruct the missing position samples between two known samples by taking a weighted average position between the positions predicted by dead reckoning and by linear interpolation. The weights change linearly with the time since the vehicle

left the first position sample. For the first predicted position sample, the weights take 100% of the dead reckoning prediction, and 0% of the linear interpolation prediction. By the time of the next known sample, these weights are reversed. Thus LIDR weighted averaging is given by the following formula:

$$P_n = W_{Dn} \cdot P_{Dn} + W_{Ln} \cdot P_{Ln}$$

Where n is the nth predicted sample (for example, to close a gap of 3.2 seconds, there are 31 position samples to reconstruct). P_n is then the position of the nth predicted sample, P_{Dn} is the nth position prediction by dead reckoning and P_{Ln} is the nth position predicted by linear interpolation. W_{Dn} and W_{Ln} are the weights and are given by:

$$W_{Ln} = \frac{t_i - t_1}{(t_2 - t_1)}$$
$$W_{Dn} = 1 - W_{Ln}$$

Where t_1 and t_2 are the times of the known samples before the gap and after the gap respectively, and t_i is the time of the sample that is being predicted.

To compare how well these methods work, figure 6.4 shows the predicted paths of a vehicle turning a corner for all three approaches. The figure shows the prediction of a vehicle's path during a 4 second gap. The blue markers indicate the actual SCAMs as transmitted by the vehicle. The yellow markers show the prediction according to dead reckoning. We can see that this method predicted the path the vehicle would have taken if it had not changed its speed or heading, and it did not predict the vehicle taking the corner. The green markers indicate the predictions by using linear interpolation from the start of the gap to the end of the gap. As we can see, this predicted that the vehicle was somewhere on a straight line between these points. Finally, the red line gives the LIDR weighted averaging prediction. Here the predictions started off by taking into account the vehicle's speed and heading and then slowly converged to the next known location. As we can see, this method gave a much more accurate prediction of the vehicle's position than the other two methods.

Of course, a large gap covering a corner is only one specific scenario. To determine the average accuracy of the three methods, we evaluated them for our entire ground truth. For each location sample in our transmission logs, we discarded the following n messages to give a gap of size $n/10$ seconds, up to a maximum gap size of 8 seconds. For each sample as a starting position and for each gaps size, we then calculated the average error

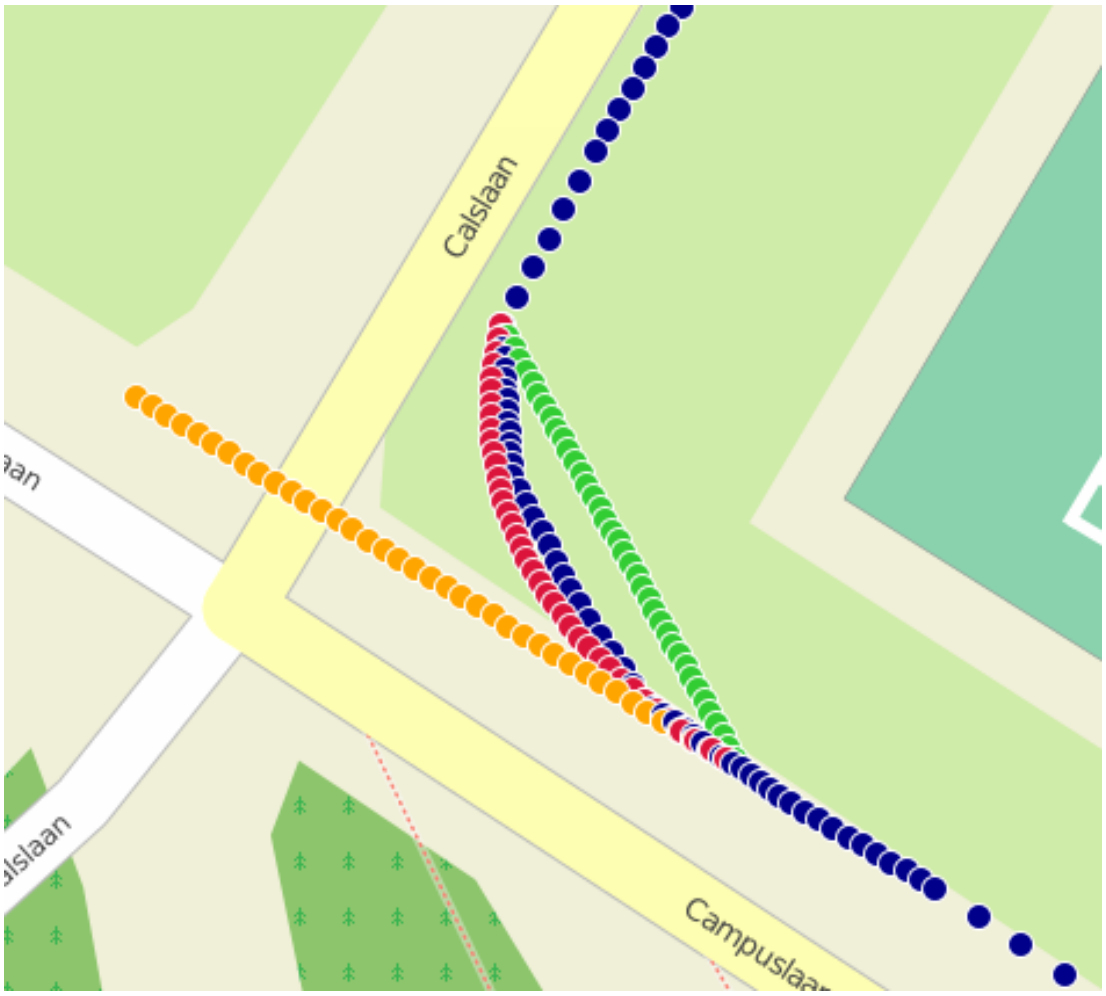

FIGURE 6.4: Predicted paths of different prediction methods

between the predicted location samples and the actual location samples. We did this for all three prediction methods, the results of which can be seen in figure 6.5.

Dead reckoning gives the worst performance, mainly due to the fact that it does not take into account information about the next known point. This means that the larger the gap, the more the prediction will diverge from the actual location. Linear interpolation performs marginally the best when the gaps are small. Once the gaps become larger than about 2 seconds, LIDR weighted averaging generates more accurate vehicle positions. Especially as the gaps become even larger, LIDR weighted averaging performs much better than the other two methods. For example, if we have a gap of 8 seconds, the average LIDR error is 3.3 meters, whereas for linear interpolation this is 9.5 meters and for dead reckoning it is 21.5 meters. On average, the linear interpolation error is approximately 35% of the dead reckoning error, and the LIDR error is approximately 50% of the linear interpolation error. As such, we consider LIDR weighted averaging a valid approach to close small gaps, and this method was used to close any small gaps in our eavesdropped messages.

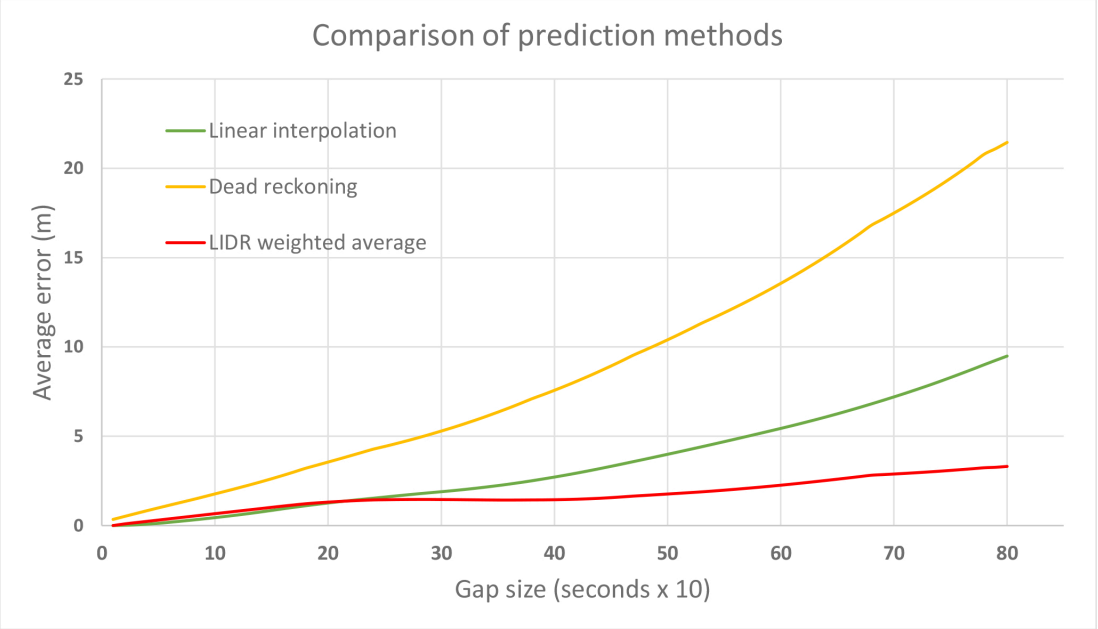

FIGURE 6.5: Comparison of prediction performance

Although dead reckoning did not seem to give very accurate results, it is still useful in situations where there is no next known close point to interpolate to. Especially on straight roads where a vehicle drives at a relatively constant speed, dead reckoning can be used effectively. Moreover, dead reckoning is the only method of the three which can extend a predicted path beyond the last known location.

The methods described above can be extended in a number of ways to improve their accuracy. One way to do this is by speed matching. For example, if the speed limit of a road changes inside a gap, the prediction algorithms could take these speeds into account to improve prediction accuracy. Another way to improve the location accuracy is to use map-matching, where a road map to match position samples to the closest matching road. This way the predicted positions that do not follow the road can be moved onto the road to give more accurate results. Furthermore, map matching can be useful to compensate for GPS error. For example, we can see in figure 6.5 that the vehicle locations are offset from the road on the map. Map matching could get rid of this offset and ensure that the GPS locations follow the roads exactly. For our purposes however, our samples were accurate enough to identify ingress and egress events at intersections, and as such we decided not to use these extra methods to process our results any further.

6.2 Tracking the Vehicle

The resulting ground truth vehicle locations after the above cleaning and processing took place can be seen in figure 6.6. The blue lines show data from the vehicle's logs,

which is everywhere the vehicle travelled. The red lines show the processed eavesdropped messages, based on the GPS position information in the received SCAMs. We can see that our sniffing stations managed to fully cover the two intended intersections, and that the sniffing station at intersection A actually managed to cover two intersections.

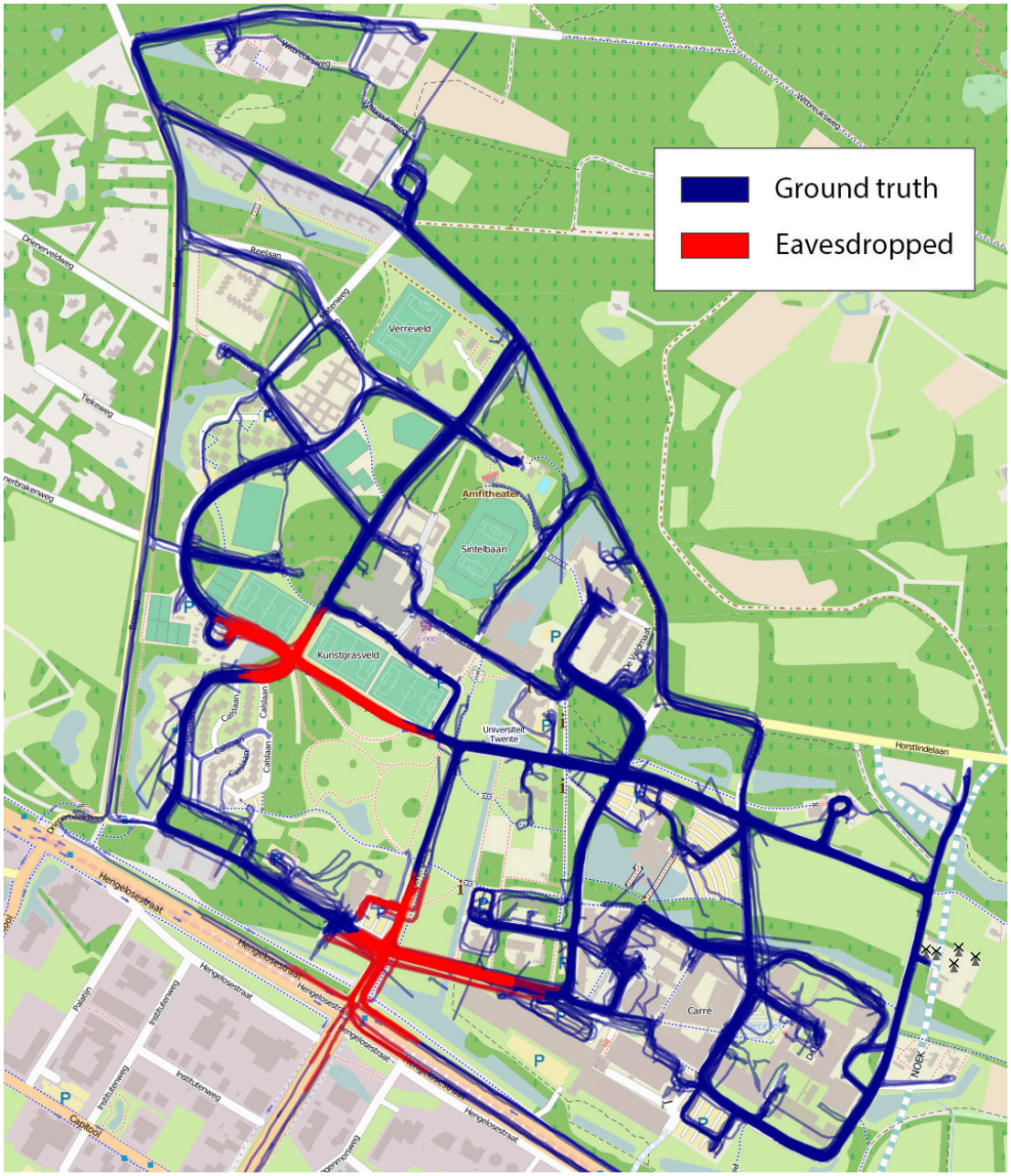

FIGURE 6.6: Overview of all actual and eavesdropped vehicle locations

Although figure 6.6 does give an indication where the vehicle went, it does not show which roads were used most often. For this reason we constructed a heatmap from the ground truth showing how frequently SCAMs were transmitted from each location. This can be seen in figure 6.7, where the darker the shade of blue the more often a road was used, with the red areas representing roads that were used the most. From this heatmap

it is evident that intersection A at the main entrance to the university was actually not the most used intersection as we hypothesized when deciding where to place the sniffing stations.

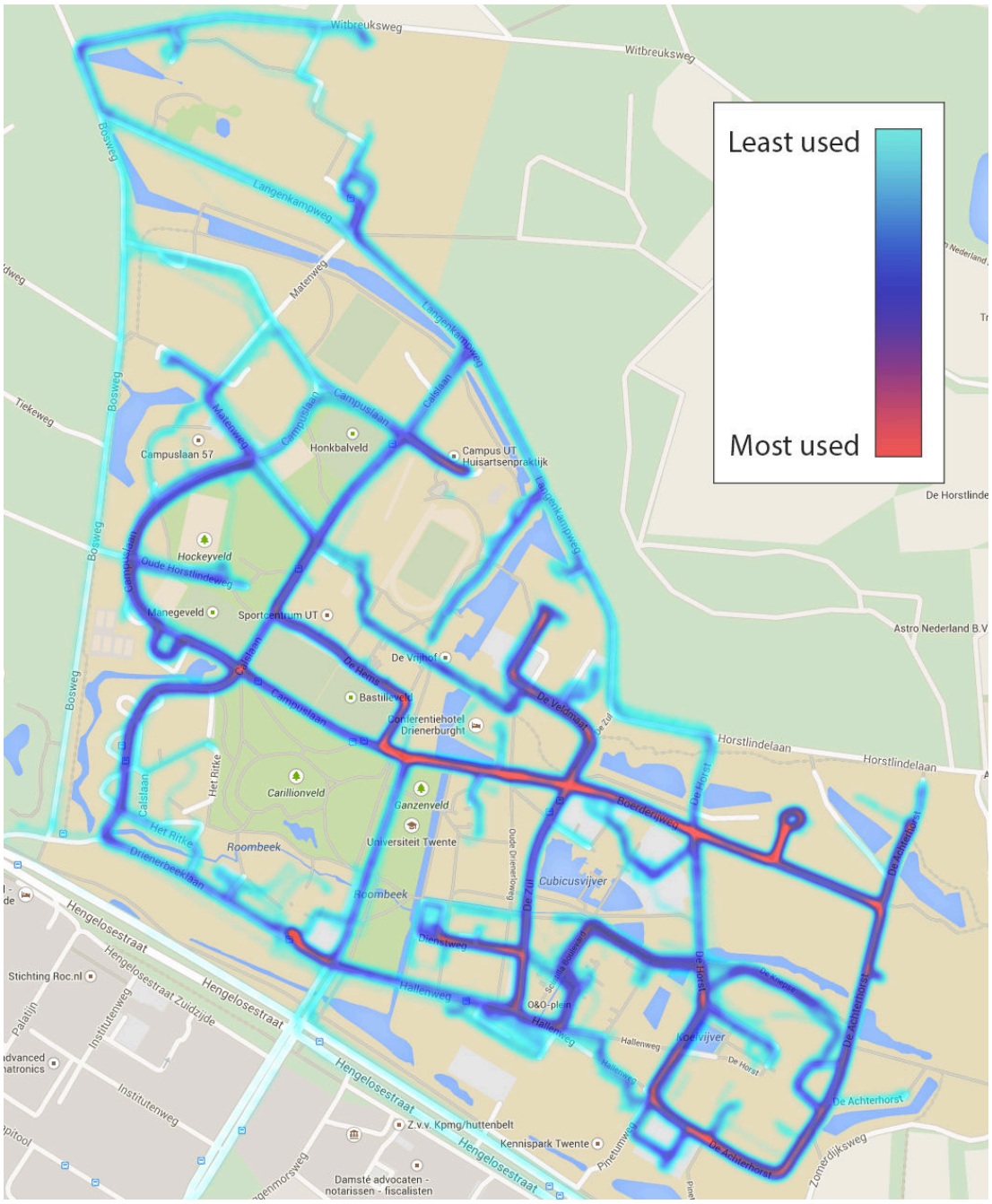

FIGURE 6.7: Heatmap of vehicle locations

The roads that the vehicle used also did not match our assumptions in chapter 5 (figure 5.8), as there were a number of roads used which did not match our graph. These roads can be split into two categories. The first are all the roads that we did not consider in our graph. Abstracting away some small roads was a necessary result of creating our graph, but the topmost east-west road was used much more than we anticipated and

should have been considered in our graph. Apart from these roads, the vehicle also took a number of roads which ordinary vehicles would either not be likely to take, or not be able to take at all due to mobile barriers such as sinkable bollards. As the campus security vehicle can control these barriers and use these roads, this opened up additional intersections which we did not consider. Thus the vehicle did not follow a mobility pattern similar to what we might expect from a normal vehicle.

Despite these issues, we could still use the data to evaluate how well a vehicle could be tracked. In the rest of this section, we evaluate two different tracking methods using data from our sniffing stations, investigating zone-level and road-level tracking performance. Note that although we only considered a single vehicle, our analyses can easily be extended to multiple vehicles, as unique identifiers in (S)CAMs allow an attacker to easily distinguish between different vehicles.

6.2.1 Most Likely Route

We have shown that we can observe vehicle locations at the intersections themselves, so then the next step was to determine what we could infer about the location of the vehicle in the large gaps between these intersections using only the information from the sniffing stations. One possibility is to try to determine the most likely route (MLR) that a vehicle took based on discrete observations at different intersections. To do this we first identified ingress and egress event pairs for our two observed intersections, where the vehicle is observed leaving one intersection and entering the other some time later. By observing these events, an attacker can infer which roads the vehicle was on prior to an ingress event, or after an egress event. It is evident that the vehicle will stay on this road until it reaches the first adjacent intersection or turn-off, as there will be no opportunities to leave the road until that time. Thus this knowledge already allows an attacker to perform road-level tracking for all road sections connecting to observed intersections. The downside is that this road-level tracking only lasts until the next intersection. However, it may also be possible to use observations from multiple intersections to track a vehicle beyond the first adjacent intersection. To achieve this, we identified direct routes between our two observed intersections and their distances. By approximating the average speed of a vehicle on these roads, we could then calculate how long a vehicle should take to travel between two intersections along the different routes. Finally by looking at a combination of the egress direction and actual time difference, we determined the MLR that the vehicle took.

The routes that we identified between our intersections can be seen in figure 6.8. We did not consider egress from the north or west of intersection B, or the south of intersection

A, as these did not lead to direct routes between the two intersections. Route 1 was 780 meters long, route 2 was 600 meters, and route 3 was 1200 meters long. Assuming an average speed of 10 m/s (36 km/h), it would take the vehicle 78, 60 and 120 seconds to cover these distances respectively. Now, for example, if we observed the vehicle leaving intersection B to the east, and arriving at intersection A around 80 seconds later, we could conclude that the vehicle took route 2. If this time difference was closer to 120 seconds, then it is more likely that the vehicle took route 3.

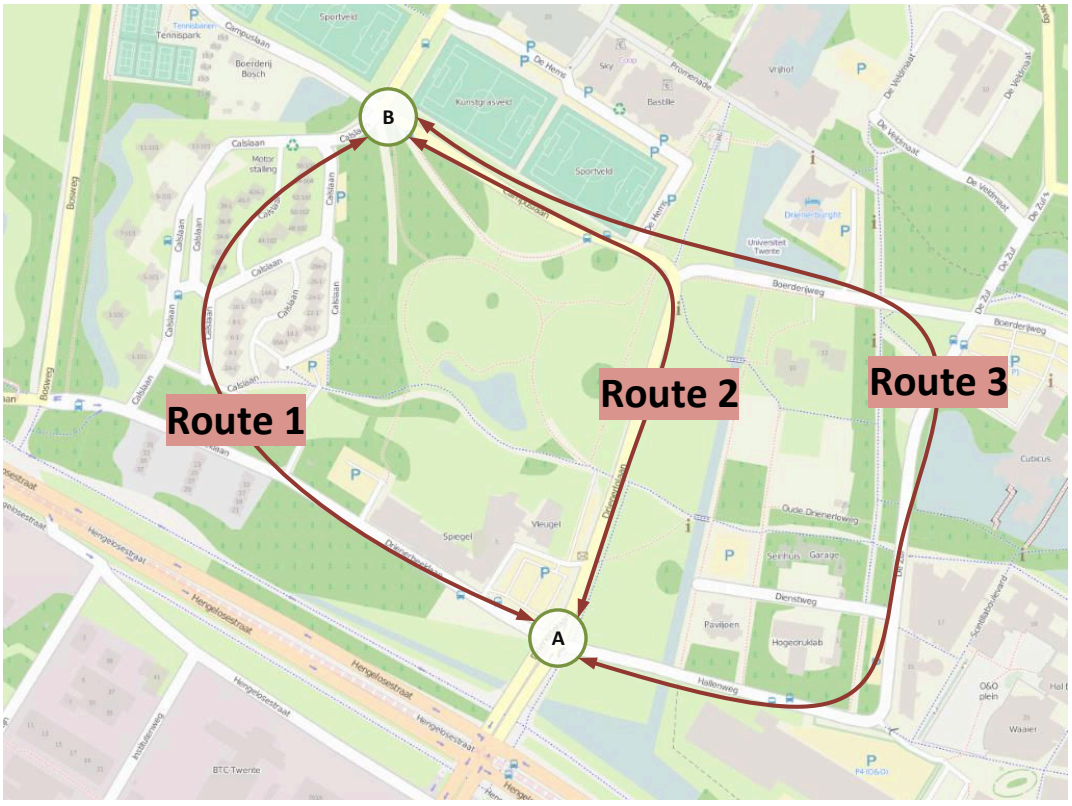

FIGURE 6.8: The routes used to determine the MLR

From our sniffing logs, we identified a total of 86 event pairs from one intersection to the other intersection. However, the time between these pairs ranged from a few tens of seconds to more than a day. Of course, the longer the time between events, the more likely that the vehicle took a route other than the direct routes we identified. For this reason, we discarded any event pairs with a time difference that was more than 240 seconds, leaving 13 remaining pairs. For these remaining event pairs, we matched the time difference to the closest expected time as we calculated above, taking into account the egress directions. Finally we compared our predicted MLR with the actual route the vehicle took. The results of this can be seen in table 6.1.

Our predictions for the 11 shortest events were correct. However, when the time between intersection events became longer, the vehicle often did not take the direct route, and

Event #	Time taken (s)	Egress intersection	Egress Direction	Assumed route	Actual route	Correct
1	38	A	North	2	2	Yes
2	49	B	East	2	2	Yes
3	55	B	East	2	2	Yes
4	65	A	West	1	1	Yes
5	79	B	East	2	2	Yes
6	85	B	East	2	2	Yes
7	90	B	South	1	1	Yes
8	91	A	West	1	1	Yes
9	93	A	West	1	1	Yes
10	109	B	South	1	1	Yes
11	133	A	East	3	3	Yes
12	201	A	North	2	1	No
13	240	A	North	2	1	No

TABLE 6.1: Most likely route predictions and results

predicting the MLR becomes less feasible. As we can see, the maximum time that we successfully tracked the vehicle was 133 seconds, which is close to the maximum expected route time of 120 seconds. Beyond this time, our predictions were no longer correct. Thus when a vehicle takes close to the expected time to cover a route, determining the MLR can be used to track the vehicle. Note that these results are correct without taking into account the ingress direction, which would provide even more information for an attacker. The only caveat is that when there are multiple direct routes with the same connecting roads and the same length, the attacker will not be able to distinguish between them.

There were however a number of problems with this approach. The first problem was the prediction accuracy due to the non-standard mobility pattern of the vehicle. Whereas a regular vehicle would be likely to follow the speed limit of the road it is driving on, the campus security vehicle in our experiment tended to vary its speed more often because it was patrolling. This speed variability meant that the inter-event timings varied as well, and a shorter route between two intersections could take longer to traverse than a longer one. Secondly, there were only a few pairs of egress events at one intersection and a subsequent ingress event at another intersection with only a small time between them. This may again have be due to the vehicle's mobility pattern; short trips along with routes that a normal vehicle would/could not take, leading to a lower chance that the vehicle crosses both intersection. This gave only a few event pairs with a small time difference between them. As it is only from these that we can make any realistic predictions about the MLR, this allowed us to track the vehicle on only a small number of occasions. Also note that the MLR tracking time is heavily dependent on the underlying

road network. Therefore, the tracking time values given above are not directly applicable when considering a different pair of intersections and cannot be generalized as such.

For these reasons, more MLRs are hard to establish without observing more intersections. As such, we do not consider our MLR approach with two covered intersections sufficient for tracking a vehicle on the campus.

6.2.2 Most Likely Zone

Although predicting MLR with two observed intersections was not feasible, it might be possible to track the vehicle at a higher level of abstraction. In chapter 5 we already based the locations where to place the sniffing stations on splitting the entire campus into the residential side on the west, and the university side to the east of these intersections. By observing the ingress and egress events at our observed intersections, we may be able to say something about which region the vehicle was in at what time. Thus instead of a most likely route, we determine the most likely zone (MLZ).

The first step was to identify exactly what constituted the residential zone and what constituted the university zone, which can be seen in figure 6.9. We limited the zones to only the campus, and discarded any trips that took place off campus. The red area was the residential zone, and covered all the student housing. The blue area was the university zone, and covered all the university buildings.

The next step was to determine how observed intersection events translated to zone predictions. For this we took into account both ingress and egress events at the two intersections. If an ingress or egress event was observed, the values in table 6.2 were used to determine what zone the vehicle was in. For egress events we assumed that the vehicle stayed within this zone until the next observed event. For ingress events, we assumed the vehicle was in the corresponding zone since the last observed event. However, if the vehicle travelled from one zone to the other unobserved, then it was possible to record an egress event into one zone, and then some time later observe an ingress event from another zone, giving conflicting information. To solve this, we divided the time between these observations into two equal parts, and assigned the egress zone to the first part and the subsequent ingress zone to the second.

	Intersection A	**Intersection B**
North	university	residential
East	university	university
South	university	residential
West	residential	residential

TABLE 6.2: Translation of intersection events to zones

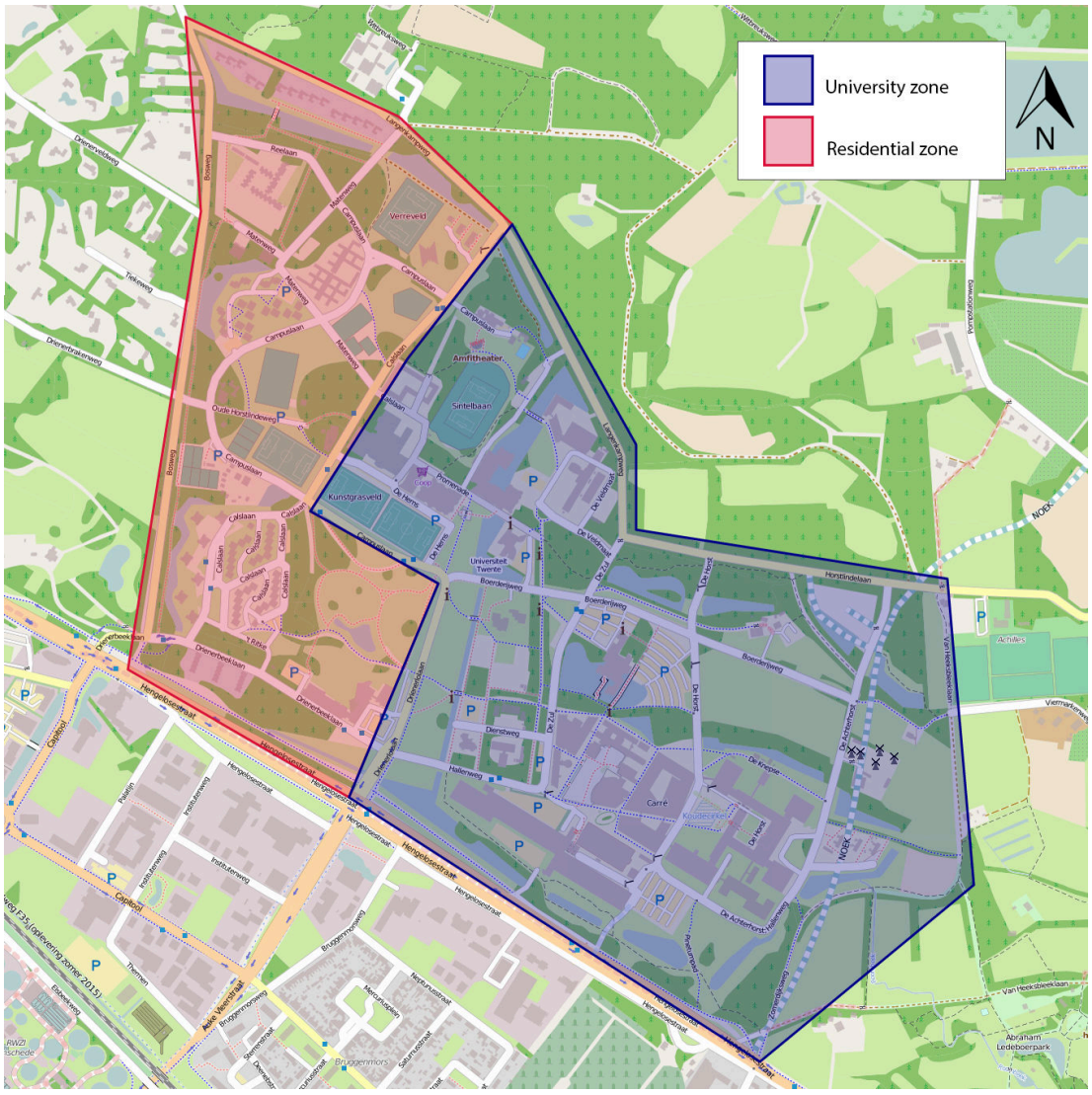

FIGURE 6.9: Splitting the campus into two zones

We then categorized all sent messages from the vehicle logs into these two zones as well. This gave 39% of all messages transmitted from within the residential zone and 61% transmitted from within the university zone. Finally we compared our zone predictions with the actual zone that the vehicle was in, giving an indication of how well accurate our predictions were. The prediction accuracy was calculated as the average percentage of correct predictions for the two zones. With only the information from intersection A, our prediction accuracy was 61.12%, with only information from intersection B this was 67.49%. Combining information from both intersections gave a prediction accuracy of 72.82%.

Considering that random guessing would give a prediction rate of 50%, these results are not a significant improvement. One problem was that the vehicle took a number of non-standard routes where normal vehicles would not be able to come, resulting in

intersections between zones that we did not cover. To investigate the extent to which these non-standard routes negatively affected our prediction accuracy, we discarded all routes that passed through non-standard areas and performed the same calculations. This gave a prediction accuracy of 70.57%, 65.76% and 79.26% for information from intersection A, intersection B, and both intersections respectively. These results are summarized in table 6.3.

	All routes	**Standard routes only**
Intersection A	61.12%	70.57%
Intersection B	67.49%	65.76%
Both	72.82%	79.26%

TABLE 6.3: Prediction accuracy for MLZ predictions

Even without taking into account these non-standard routes, there were still many incorrect predictions. There are two explanations which can account for this. Firstly, our GPS issues meant that the vehicle could cross observed intersections without a GPS fix, thus missing the transition from one zone to the other. Finally, the most northern road was used more than we anticipated, and so our covered intersections did not partition the graph as we assumed when determining where to place the sniffing stations. This meant that the vehicle frequently moved from one zone to the other unobserved, negatively affecting our prediction accuracy.

The core of this last issue is the same as with the MLR approach, namely the limited number of observed intersections. As such, we expect that covering more intersections would result in better tracking performance. In the next section we investigate whether or not this is the case, and how our tracking methods perform when more intersections are taken into account.

6.3 Expanding the Scale

For our experiment, we were limited to using two sniffing stations. However, an attacker may have more resources, and so may be able to deploy more than two. Of course, deploying more sniffing stations is only advantageous to an attacker if it improves how well vehicles can be tracked. With additional sniffing stations to deploy, an attacker also needs to determine not only which intersections to cover, but also where to place the sniffing stations at these intersections. In this section, we expand the tracking scale, and re-evaluate the same two tracking methods from the previous section, this time assuming an attacker with various levels of resources capable of covering more than two intersections. Next we describe tools that an attacker can use to help determine where to place sniffing stations at intersections. Finally, we investigate how an attacker's

resources that influence its tracking capabilities can be translated to financial costs, and we look at what resources are needed to track on an even larger scale.

6.3.1 Expanded MLZ

We first looked at expanding the scale of the MLZ approach, assuming an attacker that had the resources to cover more than 2 intersections. In order to emulate such a scenario, we assumed that the attacker could observe (at least) a 35 meter radius around the centre of an observed intersection. Given the fact that a sniffing station could also be weatherproofed, an attacker could place a station close to an intersection even when there are no buildings nearby. Such a sniffing station could be battery powered, as we found that our sniffing stations could run for more than 24 hours using the battery shown in figure 5.2. Thus, with a practical signal range of around 300 meters, a 35m coverage area around an intersection is a conservative assumption, and an attacker should be able to cover most intersections in this manner. The SCAMs from within the 35 meter area were then added to our set of eavesdropped messages. The information from these messages was sufficient to determine the speed and trajectory of the vehicle crossing an intersection, and so identify ingress and egress events and directions.

In our original scenario with two sniffing stations, we concluded that the prediction accuracy was negatively affected by the fact that the vehicle could transition from one zone to the other unobserved, which in turn was due to non-observed intersections on the zone boundary. Thus the first step in improving this prediction accuracy was to determine where these unobserved zone transitions could occur. We identified three such intersection (in addition to our original two that we actually eavesdropped on), which can be seen in figure 6.10. Intersections 1 and 2 correspond to the two intersections which we covered in our real-world experiment. The other 3 intersections were then areas where the vehicle could transition between zones unobserved in our original scenario.

Having identified these extra intersections, the next step was to determine the prediction accuracy if the attacker had observed them. Thus we calculated the prediction accuracy of all different combinations of observed intersections. This prediction accuracy was calculated using the same method as in section 6.2.2, where we predicted the zone according to observed intersection events and then compared this to the actual zone as given by the ground truth. The prediction accuracy was then the percentage of correctly predicted location samples. The results of this can be seen in table 6.4, which shows the prediction accuracy for each number and each combination of observed intersections. For example, row "1-3-5" means that the attacker covered 3 intersections (1, 3, and 5), which gave a prediction accuracy of 77.44%.

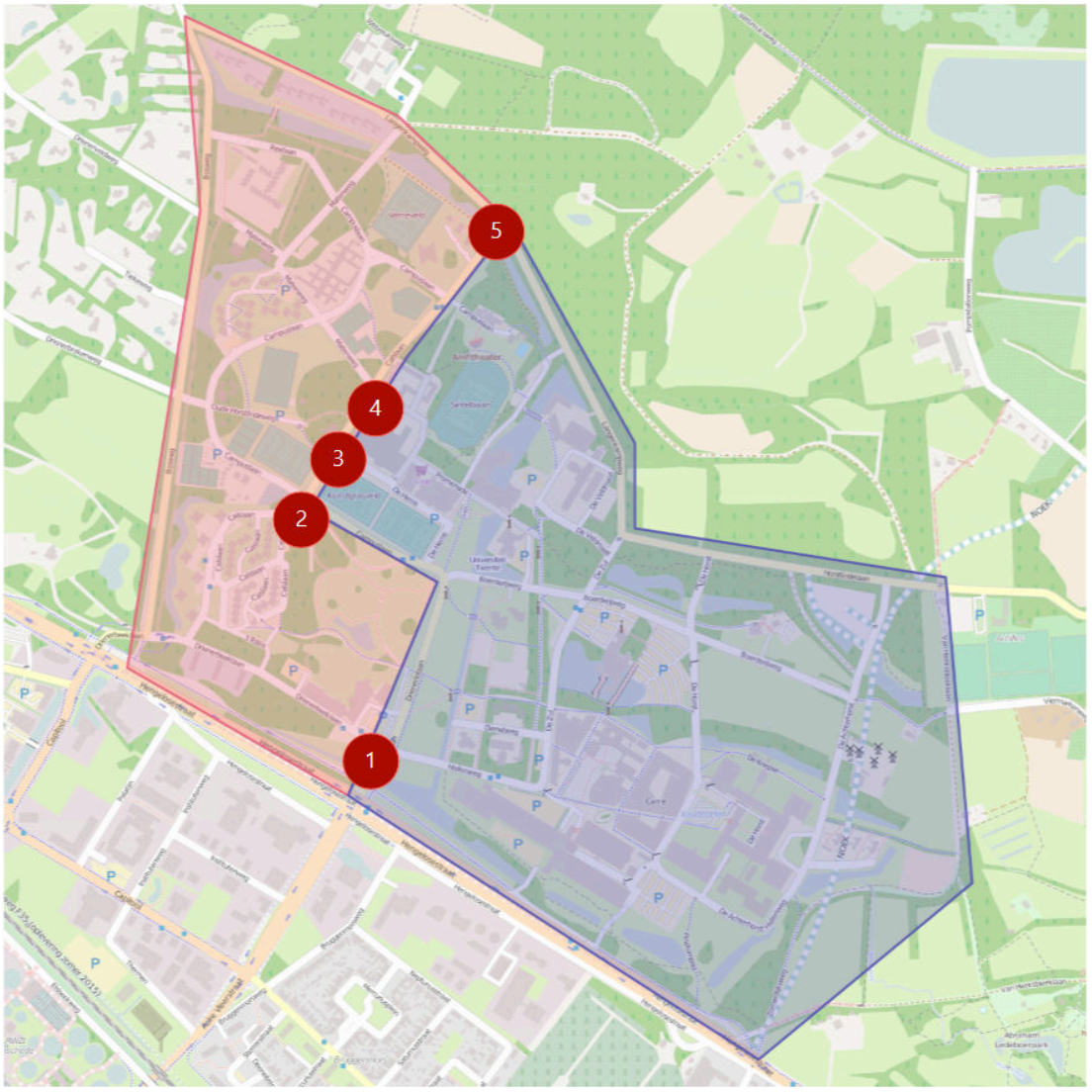

FIGURE 6.10: Identifying additional intersections between zones

As expected, covering only a single intersection gave the worst prediction accuracy, whereas covering all 5 identified intersection lead to the best prediction accuracy of 95.28%. The remaining incorrect predictions could be attributed to zone transitions where the vehicle did not have a GPS fix, as this meant that the sniffing station could not eavesdrop on the vehicle's trajectory and identify ingress and egress events. We can also see that for each additional sniffing station, the average prediction accuracy increases by approximately 8.5%. This could be useful for an attacker to determine the trade-off between the costs of an additional sniffing station and the desired prediction accuracy. In section 6.3.5 we describe in more detail how the number of sniffing stations translates into actual financial costs.

From these results we can conclude that, given sufficient resources, an attacker can collect the information required to accurately predict a vehicle's most likely zone.

# of intersections	1		2		3		4		5	
	1	61.12%	1-2	72.82%	1-2-3	81.40%	1-2-3-4	84.26%	1-2-3-4-5	95.28%
	2	67.49%	1-3	73.42%	1-2-4	78.96%	1-2-3-5	89.51%		
	3	54.85%	1-4	67.41%	1-2-5	81.53%	1-2-4-5	86.41%		
	4	52.53%	1-5	69.98%	1-3-4	73.15%	1-3-4-5	86.58%		
	5	58.10%	2-3	73.32%	1-3-5	77.44%	2-3-4-5	87.29%		
			2-4	71.76%	1-4-5	74.33%				
			2-5	78.62%	2-3-4	77.38%				
			3-4	61.44%	2-3-5	83.74%				
			3-5	67.66%	2-4-5	82.09%				
			4-5	59.10%	3-4-5	72.50%				
average		58.82%		69.55%		78.25%		86.81%		95.28%

TABLE 6.4: Expanded MLZ prediction accuracy for all intersection combinations

6.3.2 Expanded MLR

In section 6.2.1, we found that determining the MLR of a vehicle was difficult as we were limited by the number of observed intersections. Therefore we investigated what would happen if an attacker had more resources for this approach as well. Whereas for the expanded MLZ approach we only needed to identify intersections where unobserved zone transitions could occur, to determine the expanded MLR we needed to identify all large intersections within the tracking domain. We found 21 such intersections which can be seen, along with their connecting routes, in figure 6.11.

Note that even when all intersections in a tracking domain are observed, we are still dealing with a mid-sized attacker and not a global attacker which could eavesdrop on all messages in the tracking domain; due to the limited radio range the areas between intersections are considered unobserved. As with the expanded MLZ approach, we assumed that an attacker could observe the 35 meters surrounding the centre of each intersection. The attacker could then (by eavesdropping on the SCAMs) fully track a vehicle within this intersection area, but additionally it could also infer which roads connecting to the intersection the vehicle was on. More specifically, when observing an egress event, the attacker can infer the vehicle location up to the next intersection on this road. After this point the vehicle has the possibility to change roads, and if this intersection is not observed, then the vehicle can no longer be tracked. By observing an intersection, the MLR of a vehicle is fully known until the vehicle has the opportunity to change its route at the next unobserved intersection. Vice versa, when observing an ingress event, an attacker could infer the vehicle's past location up to the last intersection that it crossed.

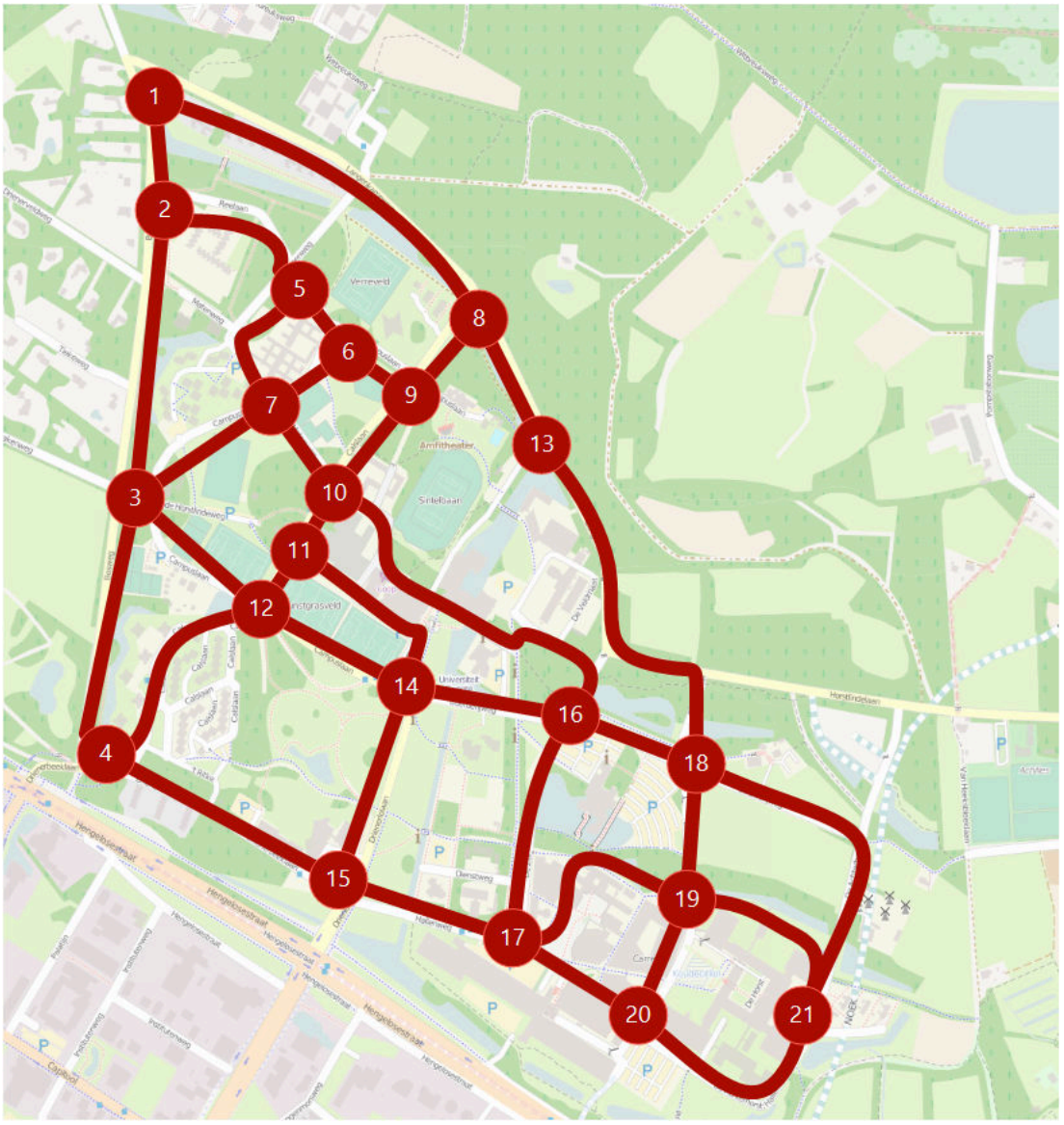

FIGURE 6.11: All identified intersections for the expanded MLR approach

With these assumptions, we calculated the percentage of all SCAMs where the attacker knew either exactly where the vehicle was (when it was in range of a sniffing station), or exactly on which road section the vehicle was (by inference as described above). We calculated this for every number of covered intersections between 1 and the maximum of 21. Furthermore we also considered every combination of intersections that an attacker could cover. This gave a total of $\sum_{i=0}^{21} \binom{21}{i} = 2097152$ different combinations of intersections that an attacker of various resources could cover. The results of these calculations can be seen in figure 6.12.

This figure shows both the maximum and mean tracking percentage for all combinations of n intersections. The maximum tracking percentage is the maximum that an attacker can obtain with n intersection, out of all possible combinations of this number

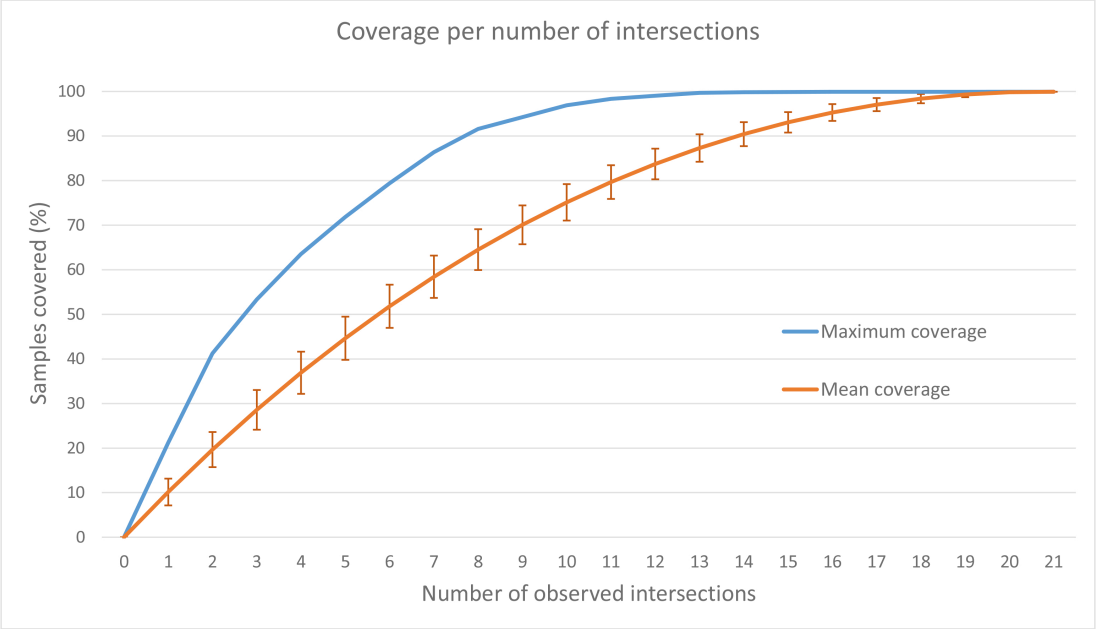

FIGURE 6.12: Expanded MLR tracking percentage for all intersection combinations

intersections. The combination of observed intersections that leads to this maximum is the optimal combination. The mean tracking percentage is the mean out of all these combinations, with the error bars indicating one standard deviation error. As we can see, the maximum tracking performance quickly increases as more intersections are observed. For example, to achieve a tracking rate of 90%, only 8 intersections need to be covered. The optimal combination for 8 intersections which lead to the maximum tracking percentage were intersections 3, 4, 8, 10, 14, 17, 18 and 21. The situation when these 8 intersections are observed is shown in figure 6.13, where the green routes and intersections represent areas where we can fully track a vehicle on a road-level, and the red routes and intersections are where the vehicle can move freely without being tracked.

We can see that there are a number of isolated red intersections. These are intersections where the attacker can infer the vehicle's location on all roads connecting to the intersection, but the intersection itself is not observed. However, as the roads extend to the centre of the intersection, these isolated red points do not indicate an area where the vehicle is safe from tracking. Apart from these isolated intersections, we can see that there are 8 road sections where the vehicle cannot be tracked on a road-level. Zone-level tracking is still possible however, as an attacker can infer that when the vehicle enters a red zone that it remains in this zone until it is observed again. Moreover, within these zones the MLR might still be inferred by using the timing approach that we used in section 6.2.1. We saw that if the period that a vehicle was unobserved was relatively short and matched the expected period, an attacker could predict which road was taken. Furthermore, the timing approach only incorrectly predicted routes if there were indirect

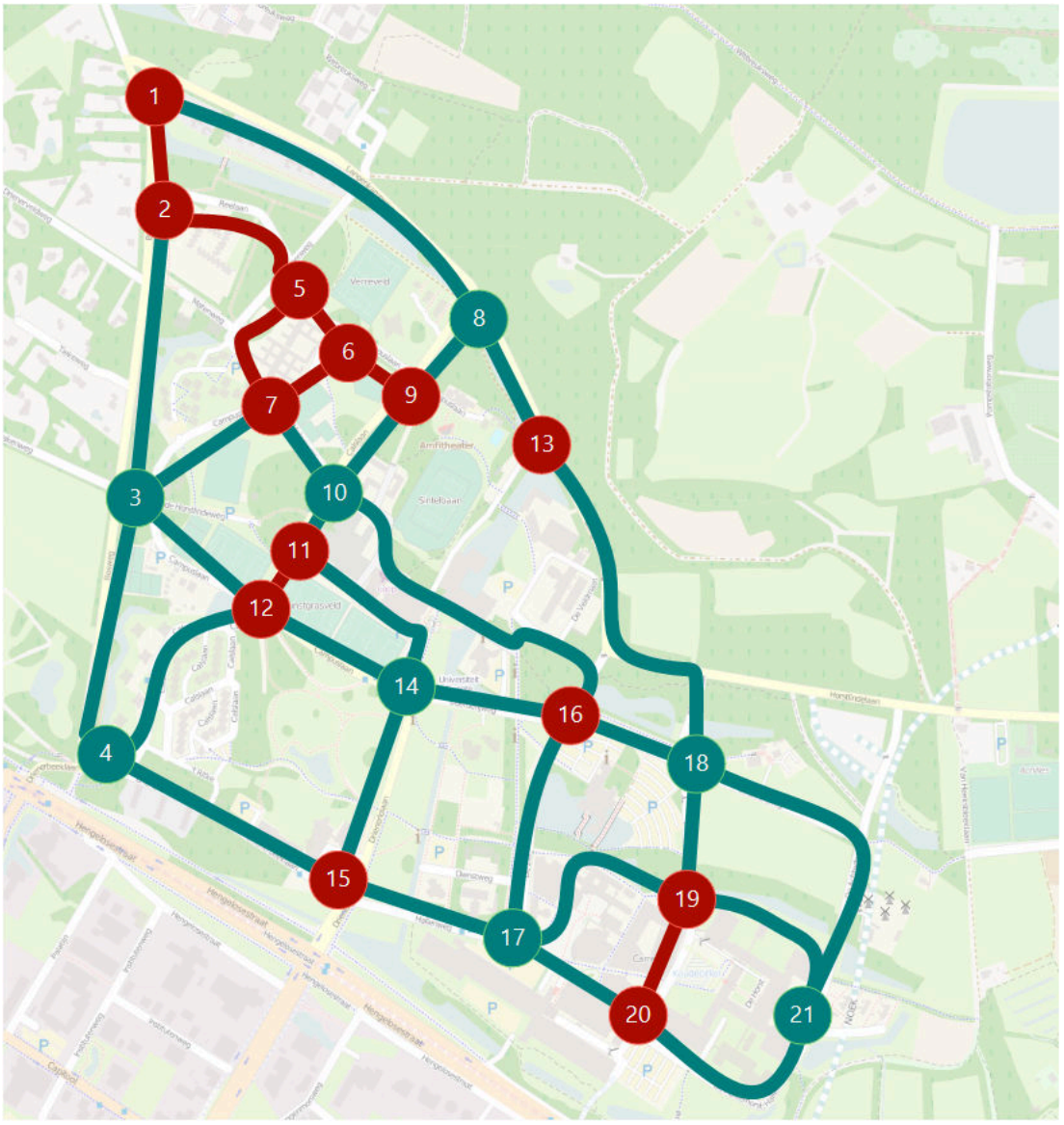

FIGURE 6.13: Expanded MLR optimal coverage for 8 intersections

alternatives. In the case with 8 observed intersections, the routes that are unobserved are all short and direct, except for the small loop between intersections 5, 6, and 7. Thus this approach is feasible to improve tracking performance, especially if the vehicle drives with a standard mobility pattern, following the speed limit.

However, even without timings there is still some information that an attacker can gain on unobserved zones. For example, if a vehicle is observed leaving intersection 17 towards intersection 20, and some time later is observed entering intersection 18 from the south, then the vehicle must have taken the red route between intersection 19 and 20, as there was no other unobserved way to get there. In fact, the only time when an attacker will not know what road, or set of connecting roads, the vehicle is on is when there is a loop in a red zone, such as between intersections 5, 6 and 7 in figure 6.13. However,

as we found the results without including a timing analysis already sufficient for vehicle tracking, we did not consider these extra methods any further.

Finally, it may also not be of interest to an attacker to be able to track a vehicle on the road-level inside the entire tracking domain. For some areas in the tracking domain, zone-level knowledge of a vehicle may suffice, whereas in other areas road-level tracking is required. For example, a burglar in the residential zone will be interested in the exact road the security vehicle is on in this zone, but will not care about where the exact road when it is in the university area, just that it is in this zone. Thus an attacker could create a denser network of sniffing stations where more tracking information is needed, and for other areas where only zone-level tracking is necessary, the sniffing stations would only need to be placed on the zone boundaries.

6.3.3 Real-time Tracking

It is important to note that up to now we have used data collected in the past to retrospectively track a vehicle, instead of tracking in real-time. Whether or not real-time tracking is desirable depends on the incentives of the attacker, and whether the actions of an attacker based on someone's location data are reactive or proactive. For example, if the attacker is a burglar that wants to know whether or not there is a police patrol nearby, then the attacker decides proactively whether or not it is safe to break in. In this case, it is desirable for the attacker to have real-time information on the location of all police vehicles. On the other hand, the attacker may want to infer some personal information based on someone's location, for example for directed advertising. In this case, the attacker first aims to learn from past locations of the target and then take action reactively. To do this, real-time location information is not necessary, and analysing the location data retrospectively at a later point in time is sufficient. In our model both types of tracking are possible. We have shown that retrospective tracking can be done by collecting location data on the sniffing stations and then retrieving and analysing this data at a later point in time. Real-time tracking is however also possible. By enabling remote access to the sniffing stations, for example by equipping them with a 3G connection, an attacker can retrieve eavesdropped messages as they are received. More importantly, all the tracking methods that we have described do not require future information to track a vehicle at any point in time. Thus these methods are just as valid for real-time tracking as they are for retrospective tracking.

6.3.4 Predicting Coverage

One open issue is how an attacker can know in advance to what extent an intersection can be covered. This is valuable information to know up front, as an attacker may have limited opportunities to install sniffing stations, especially considering that it may be a time consuming effort. Moreover, installing a sniffing station and then discovering later that it could not cover the required intersection would be a waste of resources. Thus it is useful for an attacker to be able to predict what the coverage of a sniffing station will be before actually deploying it. To achieve this, the attacker needs to know not only the exact characteristics of the antenna and how the radio signals will behave, but also what the environment looks like. Whilst it may be possible to investigate this manually for a small number of intersections, it becomes very time consuming if more intersections need to be investigated. Fortunately, there are tools that can simulate this and allow an attacker to determine sniffing station coverage beforehand without actually installing them.

We validated one such tool to investigate if it was feasible to accurately predict what our sniffing stations should have observed, and compared that to what they actually observed. The tool that we evaluated was called GEMV^2 (Geometry-based Efficient propagation Model for V2V communication) [54]. It utilizes a radio signal propagation model to estimate how the radio signals would behave. However, as 802.11p is very sensitive to obstructions, this is not sufficient. Any obstructions such as vehicles and groups of trees need to be taken into account to get a good indication of which areas a sniffing station could observe and which it could not. Getting accurate data on which obstructions there are is a time consuming process, as buildings and trees would need to be modelled in 3D before a propagation model could take them into account. To solve this problem, the simulation uses data from OpenStreetMap, which for many areas includes data on building outlines and, to a more limited extent, areas of vegetation. However, a building outline does not represent an actual 3D building. Therefore the outlines are extruded upwards to generate 3D building models. As the exact architecture of a building will not significantly affect radio propagation, this is enough to determine large scale obstructions that can influence 802.11p signals.

To validate the simulation, we imported OpenStreetMap map data of the university campus, as well as the transmitted SCAMs from our vehicle. We then input the appropriate parameters according to the specifications of our antennas, and ran the simulation. A screenshot of a section of the results can be seen in figure 6.14.

This figure shows a 3D model of the building where we placed our sniffing station, at intersection B in figure 5.8. The wide red lines show the path of the vehicle to the

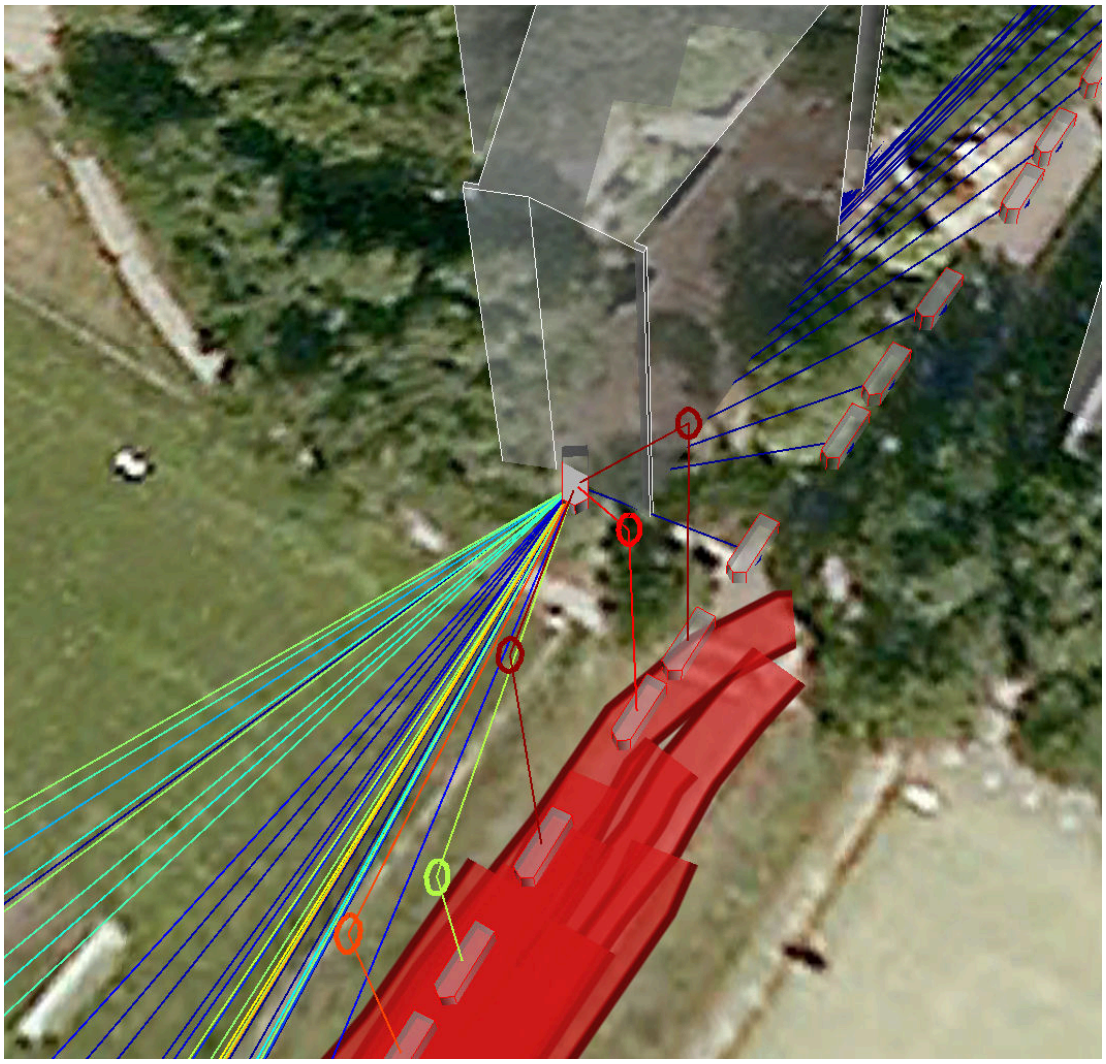

FIGURE 6.14: Propagation model showing signals blocked by buildings

extent that it was received by the sniffing station. The thin coloured lines show the expected signal strengths according to the propagation model in relation to the vehicle's location, with the redder the line, the higher the signal strengths were. The dark blue lines indicate the lowest signal strengths, where the sniffing stations would not be able to receive any messages. Finally we see small 3D polygons. The polygon next to the building indicates our sniffing station location, and the others are the vehicle position as transmitted in the SCAMs. We can clearly see that the wide red lines where the sniffing station actually received signals matched where the simulation predicted that there would be higher signal strengths. After the vehicle passed the building, we can see that the propagation model predicts that the signal strength will be extremely low, and indeed the sniffing station did not receive any messages from beyond this point.

In conclusion, we found that the propagation model gave a good indication of what can be expected from a sniffing station in terms of intersection coverage, and thus that it is

a useful tool for attackers to plan sniffing station placement.

6.3.5 Cost Analysis

In the previous sections we have seen that tracking a vehicle becomes feasible when sufficient intersections are observed. However, the more intersections an attacker wants to cover, the more resources are needed. These resources consist of time to install hardware, computational resources, the knowledge to analyse eavesdropped messages, and finally financial resources to purchase sniffing stations. We have seen that it is quite feasible to install sniffing stations that can observe intersections, and that the computational resources and knowledge needed to track a vehicle are easily obtained. The main limiting factor is then the financial costs to purchase sufficient sniffing stations to cover the area where an attacker wishes to track a vehicle.

We found that the cheapest solution for a sniffing station including antennas came to approximately €500. To fully track a vehicle on the campus, an attacker would need to place a sniffing station at all 21 intersections identified in section 6.3.2. This would mean a financial cost of 21× €500 = €10500. However, we saw in figure 6.12 that not all intersection may need to be covered to track a vehicle for a large period of time, and adding more intersection results in diminishing returns. For example to track a vehicle for more than 90% of the time, only 8 sniffing stations were needed. This means that for €4000, an attacker could almost completely track a vehicle on the campus. This is well within the range of even a small attacker with few resources.

We can also expand the scale, by looking at the area that an attacker can cover. The total area of the university campus that we considered as a tracking domain was approximately 1.75 km^2. Assuming the 21 sniffing stations required to cover this intersection, this would mean that an attacker would require approximately 12 sniffing stations per 1 km^2 that an attacker wants to cover. Given a sniffing station cost of €500, this results in €6,000 per km^2. If we extrapolate this to the entire city of Enschede with a total area of approximately 143 km^2, an attacker would require 1716 sniffing stations, and an attacker would need €858,000 to fully cover the entire city. This is however only if the attacker does not consider the road network and assumes that all intersections need to be covered. If an attacker does not require 100% coverage and uses the road network as we did in section 5.4.1, significantly less financial resources would be required. For example, if we assume that an attacker only needs to 8 out of every 21 intersections, then the costs of covering the whole city already reduce to €326,857. We must however be careful when extrapolating in this manner. The university campus is a large and

relatively isolated road network, and so it likely has a different intersection density and type of road network than the rest of the city.

Another important aspect to consider is that the above assumes a sniffing station price of €500. As the technology is relatively new, a large part of these costs cover research and development of the hardware. Thus we expect the prices to drop significantly in the coming years, as more competition comes to market, and the production quantity increases to match increased demand from ITSs being deployed. Beyond this, it is likely that in the future other devices will be able to receive ITS messages as well. Especially considering that 802.11p is just a modification of 802.11a, there is already talk of bringing 802.11p functionality to mobile phones [55]. Another option would be to use an inexpensive computer such as a Raspberry Pi with an 802.11a dongle and a driver patch to support 802.11p. This would already bring the costs of a single sniffing station down to approximately €50. Following the example above of an attacker that wants to cover the most of Enschede using road network knowledge, using this hardware would bring the total costs down to €32,686. The similarity of 802.11p to 802.11a also opens up other attack vectors. For example, if an attacker can compromise a large number of 802.11a routers found in homes and patch them to receive 802.11p messages as well, it could be possible to quickly create a sniffing station botnet covering a considerable geographic area. Finally, the United States Department of Transportation has announced intention to mandate DSRC (the United States standards for 802.11p) in all new vehicles in the near future. This means that in the future it may be possible to obtain sniffing stations with a trip to the junk yard.

Considering the above, it is clear that currently the ITS market is dynamic, and care must be taken when drawing conclusions about tracking feasibility given current hardware prices. All the described developments will very likely drive down the costs for any potential attacker, which in turn means that sniffing stations could soon become a ubiquitous threat.

6.3.6 Further Expansion

Up to now, we have mostly discussed tracking vehicles on the university campus. This however only represents a small geographic region. An attacker may want to consider a much larger tracking domain, but the number of intersections to consider when taking a larger domain can increase exponentially as the domain grows. Furthermore, to do any graph-based analyses, the road network would need to be converted into graph representation (to get an idea of how large these graphs may be, the Stanford Large Network Dataset Collection provides a graph representation of the road network of the

state of California with 1,965,206 nodes and 5,533,214 edges [56]). For the university campus generating this graph was still a trivial exercise to do manually, but for a larger area this could quickly become infeasible. Fortunately, there are already graphs available for most major road networks. For example, navigation software already uses a graph based representation of the road network for routing calculations.

It is however also important to remember that not all intersections need to be observed to be able to track a vehicle for a large portion of time. We saw in our experiment that with a domain consisting of 21 intersections, we only needed to observe 8 of them to give a tracking rate of over 90%. Moreover, the number of sniffing station also depends on where the attacker wishes to track vehicles on a road level, and where zone-level tracking is sufficient. As zone-level tracking requires fewer observed intersections than road-level tracking, this would reduce the required resources even further.

When many intersections need to be observed to track a vehicle, it is important for the attacker to make optimal use of the available resources. In figure 6.13 we identified the optimal combination of 8 intersections to track our vehicle, but we used the established ground truth of all the vehicle's locations to determine this optimal configuration. An attacker deploying sniffing station may not have the luxury of having this data beforehand. The attacker may however still be able to guess what the optimal combination would be. To do this, first a graph of the road network needs to be established, as we did in section 5.4.1. The edges in the graph then need to be weighted, according to how much tracking information the attacker expects to get from the road that the edge represent. This could be for example based on how busy the road is expected to be, or the length of the road section until the next intersection. The latter is easily determined by existing maps, and the former could be collected by using a real-time traffic service, such as the service provided by Google Maps. Finding the optimal combination then comes down to maximizing the total weights of all observed intersections. However, if two adjacent intersections are selected, the road between them will be counted twice even though this will not give an attacker any extra information on the vehicle's location. Taking this into account, finding the optimal combination is then equivalent to the maximum-weighted independent set problem for weighted undirected graphs. Unfortunately this problem is NP-complete and an exact solution will not be feasible for large graphs, but approximations will allow an attacker to reduce wasting resources.

We can also make some inferences if we simplify the road network. For example if we assume the road network to be a square grid (a common layout in the USA), we get the graph shown in figure 6.15. To get 100% coverage on such a graph, an attacker needs to observe the maximum independent set. For a grid this is easy to determine; it is a checker pattern, as shown by the blue nodes in the figure. The number of intersections

to cover in a grid with x intersections on a side can be generalized to $(\frac{1}{2} \cdot x)^2 = \frac{1}{4} \cdot x^2$. This means that an attacker needs to only observe one quarter of all intersections in a grid to fully track a vehicle on the road-level.

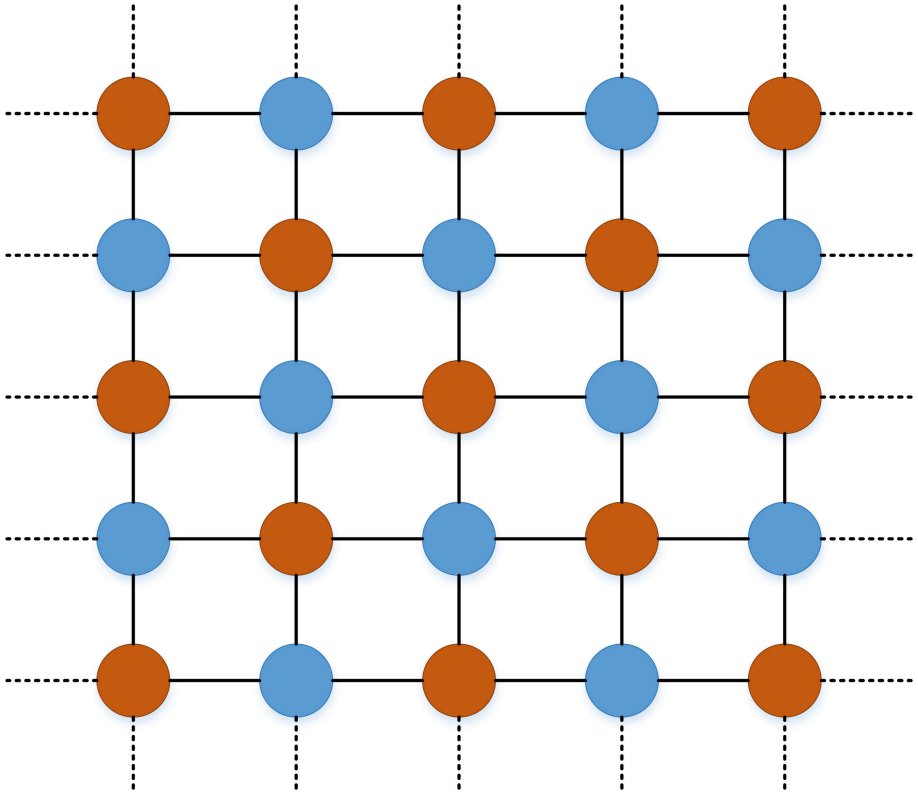

FIGURE 6.15: A grid plan road network

Finally we present a practical example: Lee and Kim determined that for Jeju city, a city in South Korea covering 977.8 km^2, only 1000 of the 17000 intersections needed to be covered to give a connectivity of 72.4%, assuming a range of 300 meters [51]. This also resulted in an average disconnection interval of less than 10 seconds. Thus with 1000 intersections, vehicles could be tracked precisely for 72.4% of the time. Road-level inferences would make the tracking rate even higher, especially given the short disconnection interval. Given our sniffing station cost of €500, this means that an attacker would need to invest €500,000 to cover all these intersections. This makes it infeasible for small (in terms of available resources) attackers, but for larger attackers it could be worthwhile, especially as it would allow this attacker to track every single ITS equipped vehicle in an entire city for a large portion of the time. Especially as the hardware prices are likely to drop in the near future, we consider this a likely threat if ITSs in the current form are deployed.

In this chapter, we have shown that with two observed intersections is it difficult for an MA to track a vehicle on a road-level, and to a lesser extent on a zone-level. However, when we expanded the scale and considered an attacker with the resources to cover more

intersections, it became clear that tracking is definitely feasible, especially if an attacker has the resources to cover a large portion of all intersections. We also found that the costs of covering intersections are still relatively high at the moment but, given the rise of ITS technologies, are likely to drop significantly in the near future. Taken together this means that when ITSs are deployed it is not merely possible that attackers will attempt to track vehicles, but is likely that it will happen. The question that then arises is what can be done to prevent it? In the next chapter we will look at pseudonyms, a technique that is often proposed as a tracking mitigation strategy.

Chapter 7

Mitigation

We have established that vehicle tracking in intelligent transportation systems is feasible for a mid-sized attacker with sufficient resources. However, we established this under the assumption that there was only single transmitting vehicle in the tracking domain. This meant that any received radio signals were certain to originate from that vehicle. Our tracking methods however do not lose their validity when tracking multiple vehicles with 802.11p transmitters. This is due to the underlying routing protocols that are required to deliver messages to specific vehicle, and due to applications needing to know which message came from which vehicle. These two factors mean that vehicles need to use unique identifiers when communicating with other vehicles and with road-side infrastructure. By looking at these identifiers, an attacker can identify which sender messages originated from and link these messages together to track any vehicle. It is this *linkability* that makes tracking possible. Linking consecutive location samples allows for short-term tracking, whereas linking together samples over an entire trip, and then linking these trips allows for mid-term and long-term tracking. The core of the problem is thus the persistent identifiers that allow an attacker to determine which vehicle any received messages originate from. It is also in these identifiers that we need to look for solutions to mitigate tracking.

In chapter 4 we already introduced pseudonyms as a tracking mitigation strategy. In the rest of this chapter, we first describe how pseudonyms can be used in different ways to mitigate tracking. We then describe different available privacy metrics that can measure how effective mitigation strategies are. Next, we investigate how effective pseudonyms are in the context of our experimental scenario by evaluating different pseudonym change strategies and privacy metrics. We then consider the same strategies for a larger geographic scale, and finally give a cost analysis describing how the use of pseudonyms affects the resources needed by an attacker to track a vehicle.

7.1 Pseudonyms and Pseudonym Change Strategies

A pseudonym is a unique identifier with which a vehicle can communicate to other vehicles and road-side infrastructure. The strength of pseudonyms however is that they are not fixed, but they can be changed. In fact, they are only effective when they are changed, if a vehicle uses one pseudonym for a long period of time the situation is not different than with the persistent identifier in our experiments. As opposed to complete anonymisation, pseudonyms can provide location privacy whilst still allowing authentication and accountability. In the context of VANETs, a pseudonym change should affect all public information that can be directly linked to a vehicle. This means that all identifiers on the communication stack, such as a vehicle's MAC or IP address, as well as any (public) keys that the vehicle uses for authentication should be changed. For tracking, the problem then becomes linking one pseudonym to another. If an attacker can do this, then all messages sent using the different pseudonyms can be linked as well. When and where a pseudonym is changed can however have a large effect on privacy and is not a trivial issue. Therefore many different pseudonyms change strategies have been proposed [31]. We discussed some of these in chapter 4, and below we give a short overview of the different categories of pseudonym change strategies.

The most effective pseudonym change strategy to mitigate tracking would be to simply use a different pseudonym for every transmitted message, as this would make every message completely anonymous. However even this method is only effective if there are multiple transmitting vehicles to confuse an attacker. For example, in our experiment, the security vehicle would have been trackable even if the messages were completely anonymous, because there was only one transmitting vehicle in our tracking domain. Thus any messages were certain to come from the vehicle we were trying to track. Another downside of this pseudonym change strategy is that it uses many different pseudonyms, as many as 10 per second in the case of (S)CAMs. Vehicles will, however, probably only have a limited number of pseudonyms available to them, and generating 10 new pseudonyms a second is then not feasible. This is especially true as access to pseudonyms needs to be limited so that vehicle cannot use multiple pseudonyms at the same time, as this would confuse ITS applications and open up the possibility of sybil attacks [57].

To reduce this problem, it is logical to use the same pseudonym for multiple messages before the pseudonym is changed. With a fixed time change strategy, a vehicle changes its pseudonym according to a fixed periodic schedule. However, such a predictable schedule may allow for consecutive pseudonyms to be linked, reducing their effectiveness. To solve this predictability, pseudonyms could also be changed after a random time period. An attacker would then not be able to predict when the next pseudonym change would take

place. However, this has a problem that is also present fixed period changes, namely that is likely that when a pseudonym change occurs, all neighbouring vehicles keep the same identity. An attacker would then see a single new identity appear with the same location trajectory as the old identity, and linking the old and new pseudonyms would be trivial. To solve this, a silent period where no location beacons are transmitted can be introduced. This would make it harder for an attacker to link a newly appeared pseudonym with an old one, as the locations and trajectories will no longer be the same. This could however have negative effects for any safety application, which rely on knowing where neighbouring vehicles are at all times (for example, entering a silent period will likely break any collision avoidance applications) [39].

However, even with silent periods, a pseudonym change is not effective unless other vehicles change pseudonyms in the same time period, as it is easy to link pseudonyms when exactly one pseudonym disappears and another one appears some time after that in the same observed area. Thus, to stop an attacker linking old and new pseudonyms, a pseudonym change should be spatially and temporally coordinated amongst different vehicles. Additionally, a vehicle cannot free-ride on the pseudonym change of other vehicles to achieve location privacy as its pseudonym can then still be tracked. In general this means the pseudonym changes are only effective if there are enough neighbours to confuse an attacker, and these neighbours need to change pseudonyms around the same time. The former can be solved by taking a density-based approach, where pseudonyms are only changed when the number of neighbours is larger than a certain threshold. The latter is a more complex problem, and requires a collaborative approach where vehicles agree to change pseudonyms at a certain time. To avoid linking location beacons, this collaborative change also needs to occur in a way that an attacker cannot observe the change. This can be achieved through silent periods (with the same problems as described above) or by only changing pseudonyms when the vehicles are in an area that where it is safe to, so called mix-zones. In effect all areas where the vehicle is not in range of a sniffing station can be considered a mix zone [27]. For example in our university campus scenario, any unobserved area between intersections could be considered a mix-zone and a good place to change pseudonyms. However, a vehicle does not know which areas are observed by an attacker and which are not, making it difficult for a vehicle to establish when it is safe to change pseudonyms.

In conclusion, changing pseudonyms is a complex problem. There is a trade-off between the privacy provided by pseudonyms messages and the security of applications that require accurate location data of other vehicles. The former requires messages to be as unlinkable as possible, whereas the latter relies on linkability for consistent predictions that make safety application possible. This problem also manifests when deciding where to change pseudonyms. Changing at intersections means that there are likely to be

many other vehicle with changing trajectories that can confuse an attacker. On the other hand, intersections are also the areas where applications like intersection collision avoidance rely on a lack of confusion. The situation is made even more complex by the fact that whilst pseudonym changes are most effective in mix zones where the vehicle is not observed, there is no way for the vehicle to know there these zones are. Despite these issues, pseudonyms seem to be a promising solution. Especially due to the fact that mitigation strategies that limit the quality or quantity of location samples may not be compatible with ITS safety applications (e.g. location obfuscation, silent periods), pseudonyms may be the only realistic solution to tracking mitigation.

7.2 Privacy Metrics

With the many different categories and types of pseudonym change strategies, it is important to be able to determine which are the most effective. To achieve this, we need some quantitative measure that would allow for a direct comparison between different strategies. These quantitative measures to establish and compare location privacy are called *privacy metrics*. In this section, we give an overview of which privacy metrics have been proposed in existing literature, and how useful they are for measuring privacy in our experimental scenario.

Privacy Loss Functions

In order to give a good indication of privacy, we look at vehicle-centric metrics, meaning that they describe the level of privacy for a vehicle instead of at a network level. One of the simplest of these privacy metrics is a privacy loss function [39][58]. As the name suggest, this is a function that describes the privacy that a vehicles loses as time passes. A privacy loss function typically takes as input the elapsed time, and the time since the last (successful) pseudonym change. Privacy loss is set to zero when the vehicle changes pseudonyms, and then slowly increases as time passes. How fast this loss of privacy happens depends on the assumed strength of the attacker. Existing privacy loss functions in literature however do not consider attackers with different levels of resources, and assume only a global attacker. This means that they do not correctly model the situation in the presence of a mid-sized attacker, as is the case in our experimental scenario. In section 7.3 we give a modified privacy loss function that does take this into account.

K-anonymity

Another privacy metric is k-anonymity, or anonymity set size [41]. With k-anonymity, a vehicle is indistinguishable from k-1 other vehicles. In the context of pseudonyms, k-anonymity is achieved if a vehicle changes pseudonyms collaboratively with k-1 other vehicles, after which an attacker would not be able to identify a target within this set of k vehicles. There are however a number of problems with this privacy metric. Firstly, even when k-anonymity is fulfilled, it is still possible to link location samples, as shown by Shokri et al. [59]. Furthermore, k-anonymity assumes that all vehicles in an anonymity set are equally likely to be the target, which is often not the case. Finally, k-anonymity does not take into account any prior knowledge that an attacker possesses. In the case of our experimental scenario, these shortcomings are evident. If a vehicle changes pseudonyms with k other vehicle at the same time, k-anonymity is fulfilled. However, if all these k vehicles are travelling in the opposite direction than the target, the attacker can easily determine the target's new pseudonym and there is no actual location privacy. The problem here is that not all vehicles are equally likely to be the target, as is assumed by k-anonymity.

There are a number of extensions to k-anonymity which aim to solve some of the problems. However, none of these take into account a non-uniform probability distribution when determining the target vehicle, and so we only discuss them briefly. Strong k-anonymity guarantees that the anonymity cluster stays the same over multiple queries, to stop set intersection attacks where an attacker can gain information by looking at the intersections of different anonymity sets [60]. Another extension is l-diversity, which measures diversity in locations that are distant enough from each other [61]. It guarantees that all vehicles in an anonymity set not only have different locations, but are also located distant enough from each other. Thus it only guarantees privacy if the location of a vehicle is unidentifiable from a set of l different physical locations. Extending l-diversity, t-closeness also specifies that the probability distribution within a cluster and the probability distribution over the entire set should be larger than a certain threshold t [62]. Finally, p-sensitivity puts a lower bound on the number of distinct confidential values within an anonymity set [63].

Entropy

Entropy is a privacy metric that aims to alleviate the problems with k-anonymity. In particular it aims to solve the fact that k-anonymity only considers a uniform probability distribution when determining the target vehicle. Entropy originates from information

theory, and is a measure of information gained by an attacker, considering the uncertainty in a random variable [64]. For location privacy, the entropy of the anonymity set can give a measure of the privacy gained by a pseudonym update [24]. Entropy measures the uncertainty of an attacker attempting to determine whether a vehicle is the target vehicle it is trying to track, and it is typically defined as:

$$H(X) = -\sum_{i=1}^{N} p_i \cdot log p_i$$

Where N is the number of vehicles in the mix-zone that change pseudonyms at the same time, and p_i is the probability that vehicle i is the target vehicle. Thus entropy is affected by two factors, namely the total number of vehicles in the mix-zone and the similarity of the distribution of a vehicle being the target vehicle to the uniform distribution. Entropy solves the problem of k-anonymity assuming that all vehicles in the anonymity set are equally likely to be the target; if one vehicle has a much higher probability of being the target than the other vehicles in the anonymity set, then entropy will be low. Although entropy depends on the absolute number of vehicles in the anonymity set, it can be normalized to the maximum achievable entropy. However, entropy does not consider whether locations are actually different. For example, if two cars are next to each other but are equally likely to be the target, then they will have a high entropy, whereas the actual level of location privacy will be low. Finally, determining the probability that a vehicle is the target vehicle depends on the tracking techniques of the attacker, and so it is difficult to determine from a vehicle's point of view.

Adversary Success Rate

Another privacy metric is the adversary success rate. For this metric, we first need to establish what the goal of the attacker is, and what is considered a success. One potential attacker goal is to minimize the distance between where the attacker thinks a vehicle is and where it actually is. This metric is then equivalent to establishing an attacker's estimation of distance error, as described by Gruteser [21]. However, they only analyse this metric in the context of multi-hypothesis tracking. Rebollo-Monedero et al. give a more in-depth analysis of an attacker's estimation error, and argue that most privacy metrics can be construed as specific cases of the attacker's estimation error [65]. In the case of our experimental scenario, we assume that the attacker's goal is to track a vehicle for as long as possible. The metric then becomes a measure of the maximum tracking time (or distance to confusion). Thus this metric measures for how long or far an attacker can track a vehicle before it gets confused, for example by pseudonyms. One

advantage of this metric is that it gives an easy to interpret quantitative value. The downside of this metric is that it does not consider an attacker drawing probabilistic conclusions to re-link pseudonyms at a later time. For example, if an attacker sees a pseudonym, and there is a high probability that it belongs to a vehicle seen earlier because the vehicle has the same physical characteristics, then the pseudonyms can be linked and this may not be properly reflected in the maximum tracking time.

Comparing Privacy Metrics

It is clear that there is a wide range of available privacy metrics. Some of them are more applicable to specific requirements, whereas other give a more general approach. However, at the moment of writing, there is no simple framework for comparing all the different metrics to compare which are more effective than others. Furthermore, some of the described metrics are quite abstract, and do not give an intuitive indication of what the actual level of privacy is. This makes it difficult to relate them directly to practical privacy implications. Especially if users are to be made aware of the privacy risks in ITSs, it is imperative that the level of privacy can also be expressed in an easy to grasp value. To achieve this, a framework that allows different privacy metrics to be compared and expressed in terms of each other would be beneficial.

Nevertheless, in the next section we use some of these privacy metrics to determine how well different pseudonym change strategies would have protected the vehicle from tracking in our experimental scenario.

7.3 Measuring Pseudonym Effectiveness

We know that pseudonyms are a possible solution to mitigate vehicle tracking, but we have not yet established how effective they are. We found in the previous chapter that without pseudonyms it is feasible to track a vehicle, as shown with data from our experimental scenario. In this scenario, the vehicle did not use pseudonyms and all messages were transmitted with a static identifier, which allowed the vehicle to be tracked for a large portion of the time that it spent driving. To determine what effect pseudonyms would have had on tracking feasibility in our experiment, we look at introducing them into our system model. However, pseudonym changes are only useful if there are more transmitting vehicles that also use different pseudonyms. We only had a single vehicle in the experiment, and so it was not possible to measure pseudonym effectiveness directly. Fortunately, it is possible to evaluate pseudonyms in a more general sense. By assuming that there are always vehicles that change pseudonyms collaboratively with

the target vehicle, it becomes possible to evaluate the best case scenario for pseudonyms. As pseudonym changes are then not dependent on probabilistic encounters with other vehicles, we can establish an upper bound on the effectiveness that pseudonyms would have had on our experimental tracking scenario.

In this section we investigate pseudonym effectiveness based on the ground truth of our experimental scenario. We investigate two different privacy metrics. Firstly we compare different pseudonyms change strategies relative to the maximum duration that an attacker can track a vehicle. Next we propose a modified privacy loss function that can describe the different sources of uncertainty in the presence of a mid-sized attacker.

7.3.1 Maximum Tracking Time

The first privacy metric that we consider is the maximum tracking time (MTT) that an attacker can track a vehicle. We saw in section 6.3.2 that if an attacker observed all intersections, it was possible to track a vehicle without pseudonyms on a road-level 100% of the time. We now look at the maximum tracking time with pseudonyms, which is the longest contiguous period of time that a pseudonym can be tracked on a road-level. There are two variables that affect the MTT, namely the pseudonym change strategy that is used and how many intersections an attacker has the resources to cover. We assume that there are no silent periods when changing pseudonyms at an intersection, as this is likely to negatively affect safety applications. Moreover, silent periods in the gaps in between intersections are superfluous, as an attacker cannot receive the messages anyway. As a consequence, pseudonym changes that occur within an observed area are not effective, as an attacker can easily link consecutive location samples, and thus link consecutive pseudonyms. Pseudonym changes are then only effective when they occur in unobserved areas, where the unobserved areas are considered to be mix-zones [27]. Assuming that all observed pseudonym changes can be linked together, the maximum tracking time is then maximum time that an attacker observes an arbitrary pseudonym and all pseudonyms that can be linked to it.

As described above, we only had a single vehicle in our experimental scenario, making a direct implementation of pseudonyms infeasible. However, we can evaluate a best case situation for users, to give an indication of what level of privacy pseudonyms can at best provide. To get this best case of the effect of pseudonyms, we assume that all pseudonym changes are perfect. A perfect pseudonym change is any change where there is enough generated uncertainty that an attacker cannot link the old pseudonym and the new pseudonym, and thus cannot conclude that they belong to the same vehicle.

To introduce these perfect pseudonym changes into our system, we make a distinction between two different types of changes, based on the type of linking that an attacker can do. Firstly, pseudonym changes can occur within a trip (intratrip changes). If an attacker can link all these pseudonyms, the vehicle can be tracked for the entire duration of the trip. On the other hand, pseudonyms can also be changed between trips (intertrip changes). For example when the vehicle changes pseudonyms every time it starts, but also when the vehicle exits the tracking domain in one trip and enters the tracking domain with a different pseudonym in another trip. With the former, it might be possible to link pseudonyms by inferring where the vehicle stopped between trips, but with the latter it is more difficult to link the pseudonyms. Note that pseudonym changes that happen in an observed area are always linkable when there is no silent period, and so we assume that only unobserved pseudonym changes are effective.

An attacker that can link both unobserved intratrip and unobserved intertrip pseudonym changes can fully track the vehicle in the tracking domain, and in this situation pseudonyms are not effective. The MTT is then equal to the entire duration that the vehicle travels in the tracking domain. On the other hand if an attacker can link intratrip pseudonym changes, but cannot link intertrip changes, then it can track a vehicle for the entire duration of any trip in the tracking domain. The MTT is then equal to the length of the longest trip, similar to what can be seen in figure 6.2 (though this figure shows all the ground truth trip lengths, so the trips covering only the tracking domain are slightly shorter).

Worst Case

In the worst case for an attacker, it cannot link intertrip nor intratrip pseudonym changes, except when they are observed. This situation gives a lower bound for the MTT. If we consider such an attacker, then the effect of pseudonyms can be seen in figure 7.1.

In this figure, we consider two different types of pseudonym change strategies. The first is when the vehicle only changes pseudonyms at the start of a trip, the least often a vehicle can change pseudonyms (except not using pseudonyms at all). As we assume that unobserved intertrip changes cannot be linked, an attacker can never track a vehicle for longer than the duration of a trip. The second strategy is a periodic change, with different pseudonym change periods. We include a vehicle changing pseudonyms every single message, a strategy that is not practical in real-world applications due to (cryptographic) overhead, but it does give the lowest MTT in this situation. We can see in the figure that the MTT decreases as the pseudonym change period becomes

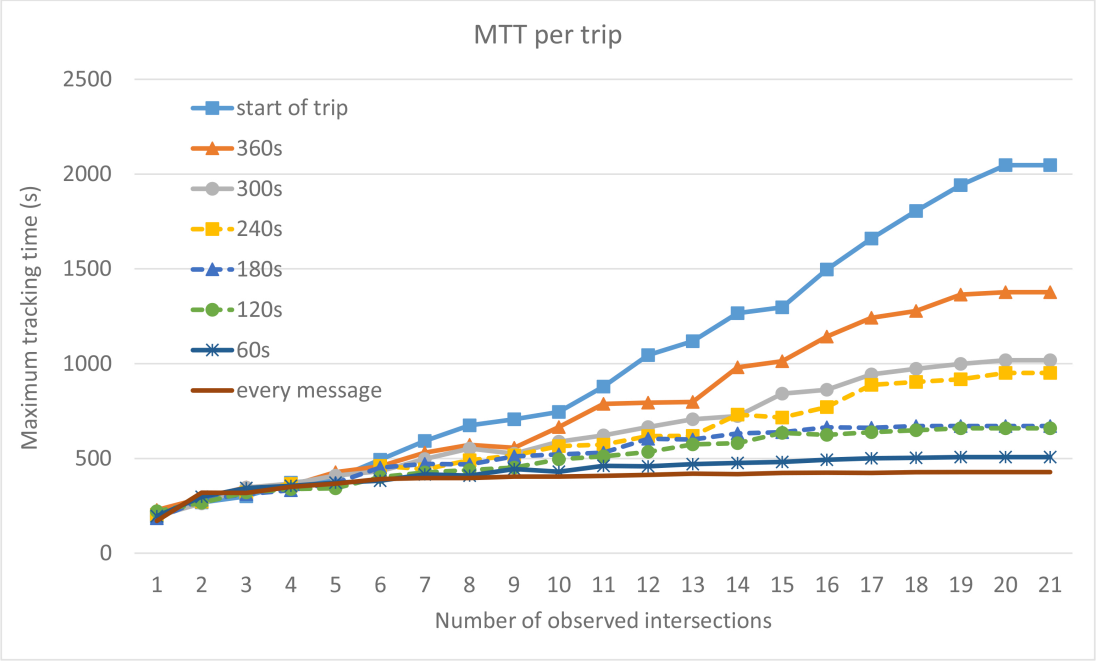

FIGURE 7.1: Maximum tracking time for unlinked trips

shorter. There are however diminishing returns, and there is only approximately a 10% difference in MTT between changing pseudonyms every message and changing every 60 seconds. This is due to the fact that when an attacker observes an egress event, it knows which road section that vehicle was on, independent of how often the vehicle changes pseudonyms on that road section. Finally, changing pseudonyms only at the beginning of a trip means that an attacker who covers all (relevant) intersections can potentially track the vehicle for the entirety of this trip.

In terms of the number of intersections that an attacker covers, we can see that for 5 or less covered intersections the MTT is approximately the same regardless of the pseudonym change strategy. The reason for this is that with few covered intersections, there is a larger probability that all of them are isolated, and there are no adjacent covered intersections that an attacker can leverage to observe a pseudonym for a longer period of time. With an attacker that covers between 6 and 12 intersections we see a gradual increase in MTT as the pseudonym change period becomes longer, and beyond this point the difference between the strategies starts to become larger. For example, between 6 and 12 covered intersections there is on average a 14% difference in MTT between the per trip pseudonym change strategy and changing every 360 seconds, but between 12 and 21 intersections this this difference is 39%. It is clear that with more than 5 covered intersections, a per trip pseudonym change is not a good strategy, as it leads to a high MTT. In fact, with an attacker that observes 21 intersections and a per trip pseudonym change strategy, we can see that the entirety of the longest trip of 2106 seconds can be tracked. Furthermore we see that beyond 12 covered intersections, and

with pseudonym change periods shorter than 300 seconds, adding more covered intersections does not lead to a higher MTT, a horizontal asymptote is reached. However, even with these strategies an attacker can already achieve an MTT of around 700 seconds, which is almost a third of the longest trip, without linking unobserved intertrip nor intratrip pseudonym changes at all. Finally, we see that covering 21 intersections does not give a higher MTT than covering 20 intersections, as the remaining uncovered intersection will be completely surrounded by covered intersections.

If a user knows that an attacker only has 5 or less covered intersections, then it may suffice to change pseudonyms only at the beginning of every trip, as changing more often will not influence the tracking capabilities of the attacker. Furthermore if, for example, a user wishes not to be tracked for more than 1000 seconds, it is clear that pseudonyms should be changed every 240 seconds or less. From an attacker's point of view, conclusions can be drawn based on the pseudonym change strategy that is used. For example, if an attacker knows that a vehicle is using a pseudonym change period of 180 seconds, it knows that deploying more than 12 intersections will be a waste of resources, as this will not influence the tracking performance.

Linking Intertrip Pseudonym Changes

Next we consider a situation where an attacker can link unobserved intertrip pseudonym changes. In our experimental scenario this is a realistic assumption, as the vehicle often parked at the same spot between trips at the campus security office. As an attacker can now track a vehicle for longer than the duration of a trip, we can now consider the use of periodic pseudonyms with a longer period than in figure 7.1. The situation for shorter pseudonym changes periods is then unchanged. For longer periods, we get the MTT as shown in figure 7.2.

In this figure, we see that the MTT increases, especially when there is a higher probability that pseudonym changes occur in observed areas due to more intersections being observed. Similar to the results above, we see the same MTT for $6/21 \approx 29\%$ or fewer covered intersections, and thus there is no significant impact of pseudonym change period on the MTT if only a few intersections are covered. Between 6 and 16 observed intersections we see a gradual increase in MTT, with each additional observed intersection adding on average 200 seconds to the MTT. When covering more than 16 intersection, each additional intersection adds on average more than 700 seconds to the MTT. Thus for an attacker it is advisable to cover at least $15/21 \approx 76\%$ of all intersections to get a high MTT. Of course, the desired MTT depends on the attacker. We assume an attacker that aims to maximize the MTT as much as possible, but different attackers

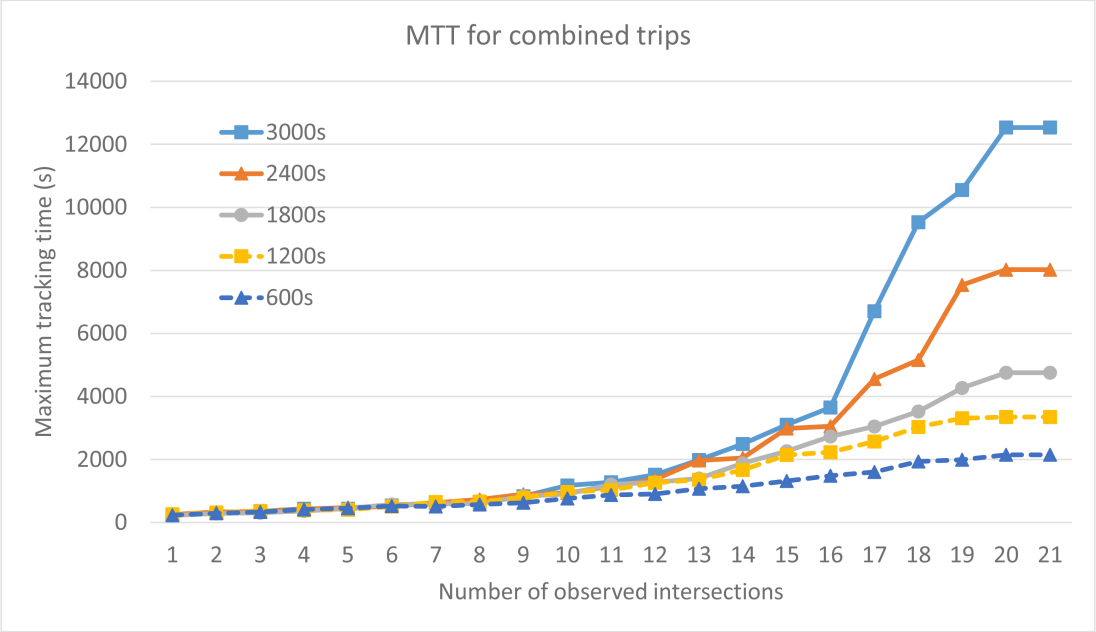

FIGURE 7.2: Maximum tracking time for combined trips

might be satisfied with a lower MTT depending on their goal, and thus can choose to cover fewer intersection. Regardless of an attacker's motives, we see that pseudonyms are effective, as an attacker needs to deploy more sniffing stations to get the same level of tracking when pseudonyms are changed more frequently. For example, to get the same MTT, an attacker needs to go from 13 to 19 covered intersections when a vehicle changes pseudonyms every 2500 seconds or 500 seconds respectively.

It also seems that with a large number of covered intersections, the increase in MTT does not increase linearly with the pseudonym change period, and observing a large number of intersections becomes increasingly effective as the vehicle changes pseudonyms more often. This could be solved by ensuring that a vehicle changes pseudonyms every time it is in the unobserved areas between covered intersections, so that a new pseudonym is obtained every mix-zone. However, it is impossible for a vehicle to know exactly where such pseudonym changes could happen unobserved, especially as the locations of sniffing stations will not be not known. Moreover, sniffing stations can in practice cover a large part of the roads connecting to intersections as well, as we saw in figure 6.6. Finally, it is important to note that these results are only for our specific scenario, and may well be different for different road networks. In section 7.4 we will look at the effectiveness of pseudonym changes over a larger geographic area, covering a different road network.

Best Pseudonym Change Strategy

From these results, it is logical to question whether we can determine what the optimal pseudonym periodic change strategy would be. It is clear that changing pseudonyms more often gives a lower MTT, and thus that a per trip pseudonym change strategy is not advisable. Changing pseudonyms every message would give the most privacy, but there is a trade-off between the costs of using pseudonyms and how often they are changed, especially as a vehicle will only have a limited number of pseudonyms to use at any one time. Thus, if a vehicle's pseudonyms run out, the same pseudonym will have to be used until new one can be requested, and using the same pseudonym for too long will increase the MTT. Thus a middle ground will have to be found that allows for pseudonym changes frequently enough to give a user sufficient privacy, but not so frequently that the costs of changing outweigh the benefits. The pseudonym change period that will eventually be standardised is currently still a matter of discussion but in the U.S., a pseudonym change period of 300 seconds has been proposed [66]. Given our results, this seems reasonable for this specific scenario. However, the absolute best pseudonym change strategy cannot be concluded from the above results. Another complicating factor concerning privacy is how we can quantify what sufficient privacy is. Different road users will have different privacy requirements, and even for a single user the requirements may change per trip. Furthermore, these requirements may change according to which areas it is more desirable for an attacker to track a vehicle. Thus the user experience and how users perceive their privacy level may have a significant impact on the pseudonym change strategy that is appropriate at any moment in time.

Re-identifying Vehicles

Looking at the MTT, it seems that an attacker is able to track vehicles for relatively long periods of time under the right conditions. However, the MTT measures the maximum time that an attacker can track an *arbitrary* pseudonym. When we consider an attacker that aims to track a specific vehicle, as in our scenario with the security vehicle, the situation becomes slightly different. To track a specific vehicle in the presence of multiple ITS equipped vehicles, there needs to be some form of initial identification to match this vehicle to its identifier. This needs to happen even in the absence of pseudonyms, as the attacker needs to know which static identifier belongs to the target vehicle. When a vehicle does change pseudonyms, this identification need to happen more often. Specifically, the vehicle needs to be re-identified for each pseudonym change which results in unlinkable pseudonyms. In the case of a mid-sized attacker in a geographically limited tracking domain, this would mean re-identification at least every

time the vehicle changed pseudonyms outside of this tracking domain, as well as for every sufficiently confusing pseudonym change within the tracking domain. Note however that in practice not all pseudonym changes will be sufficiently confusing. The effectiveness of a pseudonym change is heavily dependent on various external factors, such as the number and geographic location of neighbouring vehicles that change pseudonyms collaboratively. Especially if there is a low vehicle density or the unobserved time frame is relatively short, pseudonyms may be linked by attacks such as the timing attack we demonstrated in section 6.2.1. If pseudonyms are not sufficiently confusing and can be linked, then the vehicle will not need to be re-identified.

The risk of tracking is then relative to the capabilities of an attacker in linking pseudonyms by (physically) identifying the target vehicle. This identification can occur either manually by the attacker but it is also likely that this too can be automated, for example by radio fingerprinting, fingerprinting based on the contents of CAMs, or even with cameras and object recognition techniques from computer vision. These extra methods of re-identification will increase the resources needed by an attacker to track a vehicle. However, these techniques fall outside of the scope of this research and in section 8.2 we discuss some of them as a topic of future work.

In general we can see that the MTT depends heavily on the capabilities of an attacker in linking different pseudonyms, and that changing pseudonyms more often will reduce the MTT. However, it is also clear that pseudonyms do not reduce the risk of tracking completely. Indeed, we saw in figure 7.1 that an attacker covering all intersections was able to track a vehicle for an entire trip, even without linking unobserved pseudonym changes. This does not however mean that pseudonyms are not useful, as even though they do not make tracking infeasible, they do make it harder for an attacker. Especially when we consider an attacker that aims to track a specific vehicle, the necessity of re-establishing a vehicle's identity may increase the required resources so that it is no longer feasible for an attacker.

Up to now, we have assumed that a vehicle can change pseudonyms with an unknown number of other vehicles that will confuse an attacker. However, there is no possibility to define how this happens. Intuitively, a vehicle gains more privacy if it changes pseudonyms with many other vehicles collaboratively than if this happens with just a few vehicles. Taking this further, when there are no other vehicles to change pseudonyms with, as was the case in our experimental scenario, then there is actually no privacy at all as an attacker can conclude that all observed pseudonyms belong to that vehicle. MTT as a privacy metric cannot reflect this situation. To account for the level of uncertainty that arises from a pseudonym change we need to include entropy, as we will see below.

7.3.2 Including Entropy

Entropy allows us to account for probabilistic effects on privacy, and can be used to model the uncertainty that an attacker has in tracking a vehicle. In existing literature, entropy has largely been investigated in the presence of a global attacker, where entropy is gained through pseudonym changes. The entropy gained is the given by the formula shown in section 7.2. If we now consider a scenario where there is only one vehicle, then the entropy becomes:

$$\begin{aligned} H(X) &= -\sum_{i=1}^{N} p_i \cdot log p_i \\ &= -\sum_{i=1}^{1} 1 \cdot log 1 \\ &= 0 \end{aligned}$$

Thus we see that entropy correctly reflects the fact that no uncertainty (and thus privacy) is gained when there is only a single pseudonym changing vehicle. This is however still not entirely valid under our attacker model which assumes the presence of a mid-sized attacker. Under a global attacker, entropy is gained by changing pseudonyms, where the entropy describes the attacker's uncertainty in whether a vehicle is the vehicle that the attacker wishes to track. Under a mid-sized attacker however, there are additional sources of uncertainty, which make tracking more difficult. In fact, we can identify the following sources of uncertainty for an MA:

- Uncertainty about which vehicle is the target vehicle
- Uncertainty about the road section a vehicle is on
- Uncertainty about the exact location of a vehicle on a road section

The first source of entropy can be gained by collaboratively changing pseudonyms with other vehicles on the same road section, as we described in section 7.2. The second source of uncertainty is the different routes that a vehicle can take between intersections. This uncertainty is not present if the attacker observes adjacent intersections, as then there is only a single possible route. This uncertainty is however present as soon as a vehicle crosses an unobserved intersection. The final source of uncertainty is the estimation error that an attacker makes on the travel time of a target between intersections. We saw in section 6.2.1 that the longer the route between intersections, the more difficult

it becomes to accurately predict where the target vehicle is. Thus the longer a vehicle drives without being observed, the greater this uncertainty is.

These three different types of uncertainty can be combined to give an indication of the level of privacy that a vehicle has in the presence of a mid-sized attacker. However, as this privacy level also depends on the time since a vehicle is last observed by an attacker, it cannot be clearly be represented as a snapshot value. For this reason we propose a privacy function that combines all these sources of uncertainty over time, in the presence of a mid-sized attacker.

7.3.3 Hybrid Privacy Flux Function

The level of privacy that a vehicle has is not static. When a vehicle drives through an area with only few observed intersections and a high traffic density, it is already intuitive that it experiences a higher level of privacy than when the same vehicle travels through an area with many observed intersections. As a vehicle can transition between these two situations while it is moving, we can state that privacy is something dynamic that changes over time. Thus it makes sense not to look at a single value of privacy for a vehicle, but to define a function that describes how the level of privacy changes over time. Such functions are typically called location privacy loss functions, and model how a vehicle loses and gains privacy over time. For example, Freudiger et al. proposed a loss function that describes the amount of privacy lost in relation to the time, the time since the last pseudonym change and the duration of the silent period [58]. In this model, privacy loss is set to zero after a pseudonym change and during the subsequent silent period. After this, the level of privacy loss increases according to a sensitivity parameter λ, which models the tracking power of the attacker. Privacy loss can increase to a set maximum, which is dependent on the number of vehicles involved in the last pseudonym change.

Whilst this privacy loss function does give a good indication of the level of privacy that a vehicle has due to pseudonym changes, it does this under the assumption of a global attacker. This means that outside of silent periods, the level of location privacy is always decreasing. A mid-sized attacker is however weaker, and thus privacy does not always decrease. In fact, privacy will only decrease when a vehicle is within an observed area. Outside of these observed areas the level of privacy will stay the same, or may even increase due to other sources of uncertainty. Thus the privacy loss function described above is not directly applicable to our experimental scenario.

To account for these limitations, we propose an adjusted privacy loss function that takes into consideration the extra variables that a mid-sized attacker introduces. As this

privacy loss function takes into consideration factors that can both increase and decrease the level of privacy, and that there can be multiple different sources of uncertainty, we call it the hybrid privacy flux function. Our hybrid privacy flux function takes into account the three sources of uncertainty as described in section 7.3.2. Thus a vehicle can gain privacy through any of these three sources. However, a vehicle can also lose privacy. In our model, this occurs when a vehicle encounters an observed intersection. Thus when a vehicle encounters many observed intersections, its overall level of privacy will be lower than when it encounters many unobserved intersections, which confirms our intuition. To establish when a vehicle gains or loses privacy, we partition our ground truth into distinct sets of events, based on the current state of the vehicle. At any discrete time t, a vehicle is either in an observed area, or an unobserved area, and within an unobserved area it is either changing pseudonyms, crossing an unobserved intersection or just driving. Thus we define four sets of events, namely T_{upc} for all samples when a vehicle changes pseudonyms unobserved, T_{ui} when a vehicle crosses an unobserved intersection, T_{urs} when it drives on an unobserved road, and finally T_{obs} when the vehicle is observed by a sniffing station. Our hybrid privacy flux function then becomes:

$$P_{pnm}(t) = \begin{cases} max(P_{pnm}(t-1) - \sum_{i=1}^{N_{veh}} p_i \cdot logp_i, P_{pmax}) & \text{if } t \in T_{upc} \\ 0 & \text{if } t \in T_{obs} \end{cases}$$

$$P_{int}(t) = \begin{cases} max(P_{int}(t-1) - \sum_{j=1}^{N_{road}} p_j \cdot logp_j, P_{rmax}) & \text{if } t \in T_{ui} \\ 0 & \text{if } t \in T_{obs} \end{cases}$$

$$P_{road}(t) = \begin{cases} max(P_{road}(t-1) + \lambda(t_{last} - t), P_{dmax}) & \text{if } t \in T_{urs} \\ 0 & \text{if } t \in T_{obs} \end{cases}$$

$$P(t) = P_{pnm}(t) + P_{int}(t) + P_{road}(t)$$

This function gives $P(t)$, a measure of privacy of the pseudonym used by a vehicle at time t. Privacy is gained according to the sum of the three sources of privacy. The first way a vehicle can gain privacy is similar to the privacy loss function given by [58] and is when an unobserved pseudonym change occurs at time t ($t \in T_{upc}$). This is given by $P_{pnm}(t)$, and the new level of pseudonym privacy is then the previous level, modified by the amount of entropy gained by the pseudonym change. The amount of entropy gain is dependent on N_{veh}, the number of vehicles that collaborated with the pseudonym change, and p_i, the probability that each vehicle is the target vehicle. A

vehicle can also gain privacy by crossing an unobserved intersection at time t ($t \in T_{ui}$). The level of intersection privacy is given by $P_{int}(t)$, and is the previous level modified by the amount of entropy gained by the crossing the intersection. The amount of entropy gain is dependent on N_{road}, the number of roads the vehicle can take at the intersection and p_j, the probability of taking each road. Note that these two events can happen at the same time, and both sources of entropy will be added to $P(t)$, the total privacy level at time t. The final way to gain privacy is due to the uncertainty of the exact location of the vehicle when it drives on an unobserved road section at time t ($t \in T_{urs}$). The level of privacy gained is dependent on t_{last}, which is the last time the vehicle was observed. Thus the longer the vehicle has driven since the last observation, the more privacy is gained. How fast the level of privacy increases is dependent on the sensitivity parameter λ, which models how well the attacker can predict the vehicle's exact position on any given road section. The total level of privacy $P(t)$ is then the sum of these three privacy sources when the vehicle is not observed at time t ($t \notin T_{obs}$), and a vehicle loses all location privacy when it is observed ($t \in T_{obs}$), as the attacker now knows the location of the observed pseudonym.

For all the methods to increase privacy, we also see that they are limited by a maximum value. In the case of pseudonym changes, the maximum level of gained entropy is limited by the number of vehicles in the tracking domain, as it is not possible for an attacker to be confused between more vehicles than are present. This maximum can then be given by $P_{pmax} = log(N_{tv})$, where N_{tv} is the total number of vehicles that an attacker can be confused by in the tracking domain. The same is true for the entropy gained by crossing unobserved intersections. In this case, the attacker cannot confuse the road an attacker is on between more than the total number of roads in the tracking domain and is given by $P_{rmax} = log(N_{tr})$, where N_{tr} is the total number roads in the tracking domain. Finally, the uncertainty that an attacker has in the exact location on a road of a vehicle cannot exceed the length of the longest road, P_{dmax}. The range of values for $P(t)$ is then from a minimum of 0 to a maximum of $P_{pmax} + P_{rmax} + P_{dmax}$. In section 8.2 we suggest how the assumption that privacy reduces to 0 when a vehicle observed might be relaxed, as well as how the model can be extended to include predictions on the value of p_j.

We can now use this loss function to evaluate the level of privacy that the vehicle had in our experimental scenario. In the original scenario, the vehicle did not use pseudonyms. This means that $P_{pmax} = 0$ and the vehicle could not gain any privacy through pseudonym changes alone. However, the vehicle could still gain some privacy through crossing unobserved intersections. In chapter 6 this was seen in the fact that the attacker did not have a 100% tracking accuracy, especially if not all intersections were observed. The maximum level of intersection privacy that an attacker could obtain

was $P_{rmax} = log(N_{tr}) = log(35)$, as we considered 35 roads as in figure 6.11. The level of entropy that a vehicle gained by passing an unobserved intersection depended on the number of roads leaving an intersection, and the probability that the vehicle took each road. For example, an intersection where a vehicle almost always takes the same road will give less entropy than an intersection where the vehicle is equally likely to take each road. To get this information, we could use traffic services such as Google Maps, and by looking at traffic over some period of time these probabilities could be constructed. However, we already had this data available for our vehicle in our ground truth, and so we used this to establish the probabilities of each road being taken. For each intersection ingress event, we calculated how often each road at that intersection was taken. This gave us the probability p_j in the loss function, and allowed us to calculate the entropy gained for each unobserved intersection that the vehicle could pass.

An example of these entropies can be seen in table 7.1 and table 7.2, which show the entropies gained for each ingress direction at the two intersections where we placed sniffing stations in our experimental scenario (intersections 12 and 15 in figure 6.13). The tables show the entropy gained for each ingress direction and each corresponding egress direction, identified by the intersection number that the vehicle came from or went to respectively. For each ingress direction, the number of times that a certain egress direction was taken is shown, as well as what percentage this was of all egress events. In the last column we show the entropy gained for each ingress direction. We can see in the first row of table 7.2 that when there is an approximately equal egress direction likelihood, the entropy is high. On the other hand, we see in the last row of this table that if one egress direction is much more likely than the others, that the entropy is low. Finally we see that having more possible egress directions such as in table 7.1 leads to a higher overall entropy than when there are fewer possibilities as in table 7.2. This is due to the fact that when there are more possibilities of egress direction, the confusion for the attacker about which road the vehicle is on is greater as well.

Note that calculating privacy level assumes that it is known which intersections are observed and which are not. In a real-world scenario this is however known only to the attacker, which would mean that it is not possible for a driver to estimate the current level of intersection privacy. One way to solve this would be to use the expected level of resources of an attacker. For example, if it is expected that an attacker has sufficient resources to cover only 8 out of 21 intersections, then the user can assume that each intersection is observed with a probability of $\frac{8}{21}$. Thus by calculating the entropy that can be gained by the intersection and multiplying by this fraction, a user-centric indication of the privacy level can be obtained.

ingress	egress	count	percentage	entropy
3	4	2	11.11%	1.35
	14	10	55.56%	
	11	6	33.33%	
4	3	7	31.82%	1.53
	11	5	22.73%	
	14	10	45.45%	
11	3	4	26.67%	1.53
	4	4	26.67%	
	14	7	46.67%	
14	3	13	61.90%	1.27
	4	2	9.52%	
	44	6	28.57%	

TABLE 7.1: Entropy gained per direction for intersection 15

ingress	egress	count	percentage	entropy
4	14	5	45.45%	0.99
	17	6	54.56%	
14	4	18	62.07%	0.96
	17	11	37.93%	
17	4	19	82.61%	0.67
	14	4	17.39%	

TABLE 7.2: Entropy gained per direction for intersection 12

The vehicle could also gain some privacy by the attacker not knowing exactly where on the road a vehicle was. However, if we consider an attacker that is only interested in road-level tracking, then this does not give any privacy at all. An attacker that is only interested in which road a vehicle is on will not be interested where on that road the vehicles is. As we assume such an attacker this source of uncertainty is negligible, and we do not take it into account. With these assumptions, we can show how the privacy level of a vehicle changes over time. Figure 7.3 shows the privacy level of the vehicle in our experimental scenario over a period of 15 minutes. In this graph, the vehicle uses a pseudonym change period of 300 seconds and the attacker covers 8 of the 21 intersections. We can see that at $t0$ the privacy level is 0. A short time after this, the vehicles starts to gain privacy, through a combination of pseudonym changes and crossing unobserved intersections. At $t1$, the maximum privacy level from crossing intersections (P_{int}) has reached its maximum (P_{rmax}), and the vehicle can only gain privacy by changing pseudonyms. This happens twice between $t1$ and $t2$, and at $t2$ the vehicle attains the maximum level of privacy achievable in this scenario ($P_{pmax}+P_{rmax}$). Finally, at $t3$ the vehicle comes within range of a sniffing station, and the privacy level is reduced back to 0.

To compare different pseudonym change strategies, we need to determine the overall

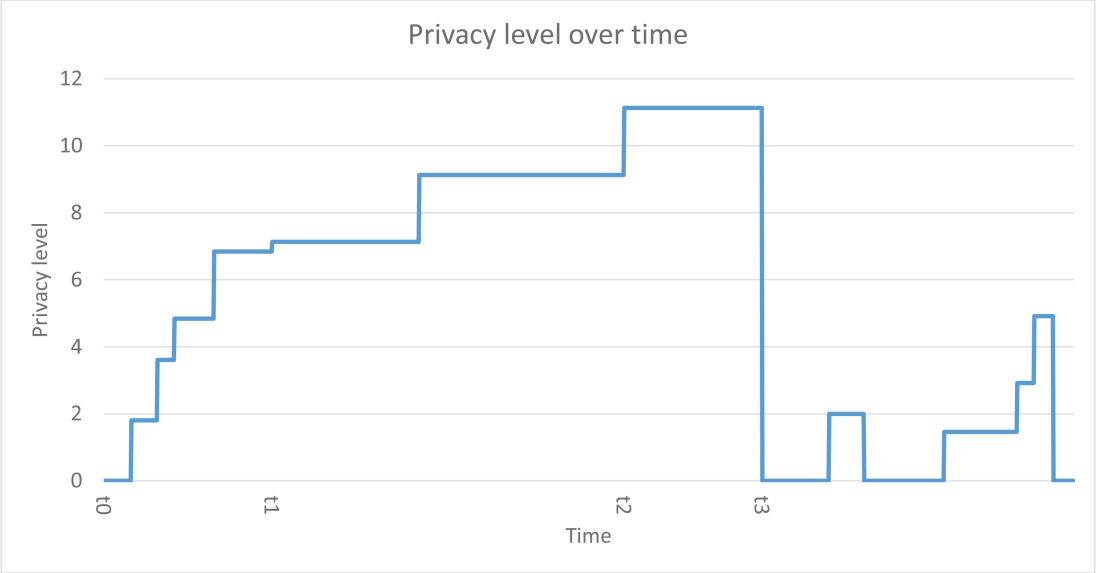

FIGURE 7.3: Privacy level change over a period of 15 minutes

privacy level that a certain change strategy results in. To do this, we calculated the mean privacy level over all location samples. The normalized privacy level of the vehicle in our experimental scenario can be seen in figure 7.4. This figure shows the privacy level of our vehicle for different numbers of observed intersections.

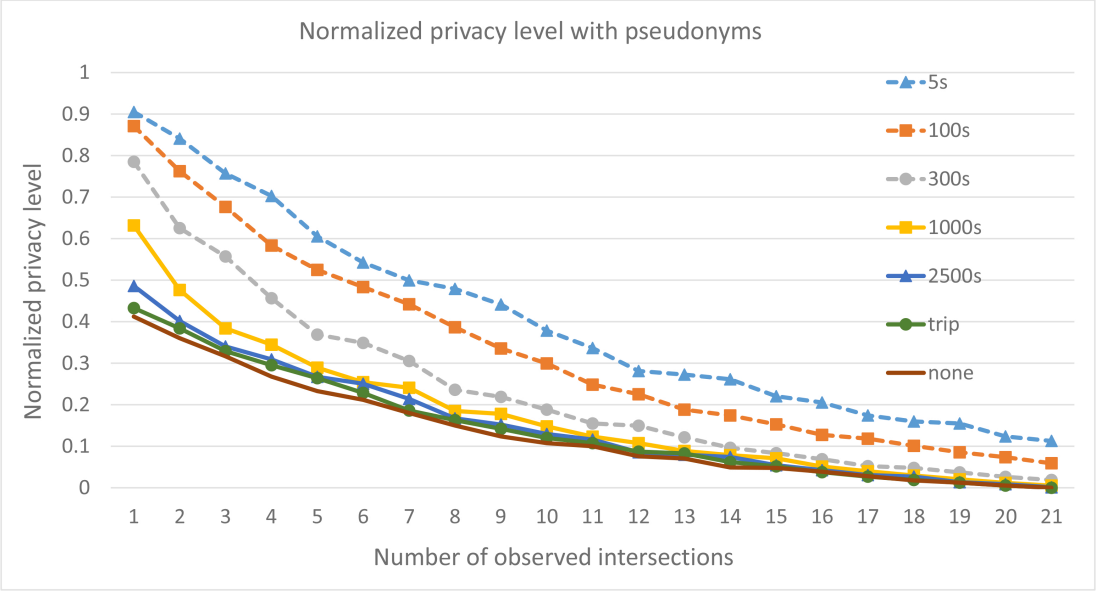

FIGURE 7.4: Privacy level for different pseudonym change strategies

We first look at a situation where no pseudonyms are used, which can be seen in the curve labelled 'none'. As we can see, even without pseudonyms, there is some level of privacy due to unobserved intersections, and the larger the number of observed intersections, the lower the privacy level. If the attacker covers no intersections the vehicle

has approximately 40% of the maximum privacy level, but if the attacker covers all intersections the privacy level is 0.

We now consider what happens when we add pseudonyms. As in section 7.3.1, we take into account two different pseudonym change strategies, namely per trip and periodic. With pseudonyms, a vehicle can gain entropy as the attacker may confuse the target vehicles with all the vehicles that change pseudonyms on the same road at the same time. As we did in section 7.3.1, we assume perfect pseudonym changes to give a best case on the amount of privacy that pseudonyms provide. The factor that influences this privacy level is the number of vehicles that change pseudonyms at the same time, and so for each change $p_i = \frac{1}{N_{veh}}$. Furthermore, we know that the total amount of privacy gained by pseudonyms is limited by the total number of vehicles present in the tracking domain. To analyse pseudonyms in our experimental scenario, we need to make assumptions about these variables. We base these assumptions on what we consider to be realistic for our university campus scenario. Firstly, we assume that every time a pseudonym is changed, there are 3 other vehicles that change pseudonyms at the same time. Furthermore, we assume that there are 100 vehicles in the tracking domain, giving $P_{pmax} = log(N_{tv}) = log(100)$. With these assumptions, the effect of pseudonyms on the mean level of privacy in our experimental scenario can also be seen in figure 7.4.

Again we can see that the level of privacy decreases as the number of observed intersections increases, and that changing pseudonyms more often gives a higher level of privacy. Furthermore, changing pseudonyms more often seem to be most effective when there are only a few observed intersection. When the number of observed intersections increases, the pseudonym change strategy that is used seems to have less influence on the privacy level. In fact, for a pseudonym change period of 2500 seconds, the privacy level is just marginally better than not using pseudonyms at all. This is due to the fact that with many observed intersections, the chance is high that a vehicle will come across such an intersection quickly, and all gained privacy will be lost. Thus with more observed intersections, the larger the influence of intersections on the privacy level, and with fewer observed intersections the more pseudonym changes affect the privacy level. For short pseudonym periods we do get a higher privacy level, even with many observed intersections.

We can use this graph to get an indication of what the effects of pseudonyms are on the resource level of the attacker. For example, to get the same privacy level, the attacker needs to cover only 1 intersection in the case of no pseudonyms, but needs to cover 10 intersections with a pseudonym change period of 5 seconds. Thus introducing pseudonyms increases the resources that an attacker needs to commit to be able track a vehicle to the same extent. The mean level of privacy does however not tell the entire

story. Intersections cover a geographic area in the tracking domain, and thus there will be some areas where there is a higher level of privacy and some areas where there will be a lower level of privacy. Intuitively, areas where there is a higher density of observed intersection will result in a lower privacy level, but also the probability of which roads are taken will affect this level. To get a visual indication of the privacy level of a vehicle, we mapmatched all location samples in our ground truth the roads as shown in figure 6.11. We then split all the roads into 5 meter segments, and calculated the mean privacy level per segment. Each segment was then assigned a colour depending on the mean privacy of this segment. This resulted in a heatmap that could be calculated for any combination of observed intersections. The privacy heatmap of our experimental scenario with two observed intersections, and a pseudonym change period of 300 seconds, can be seen in figure 7.5.

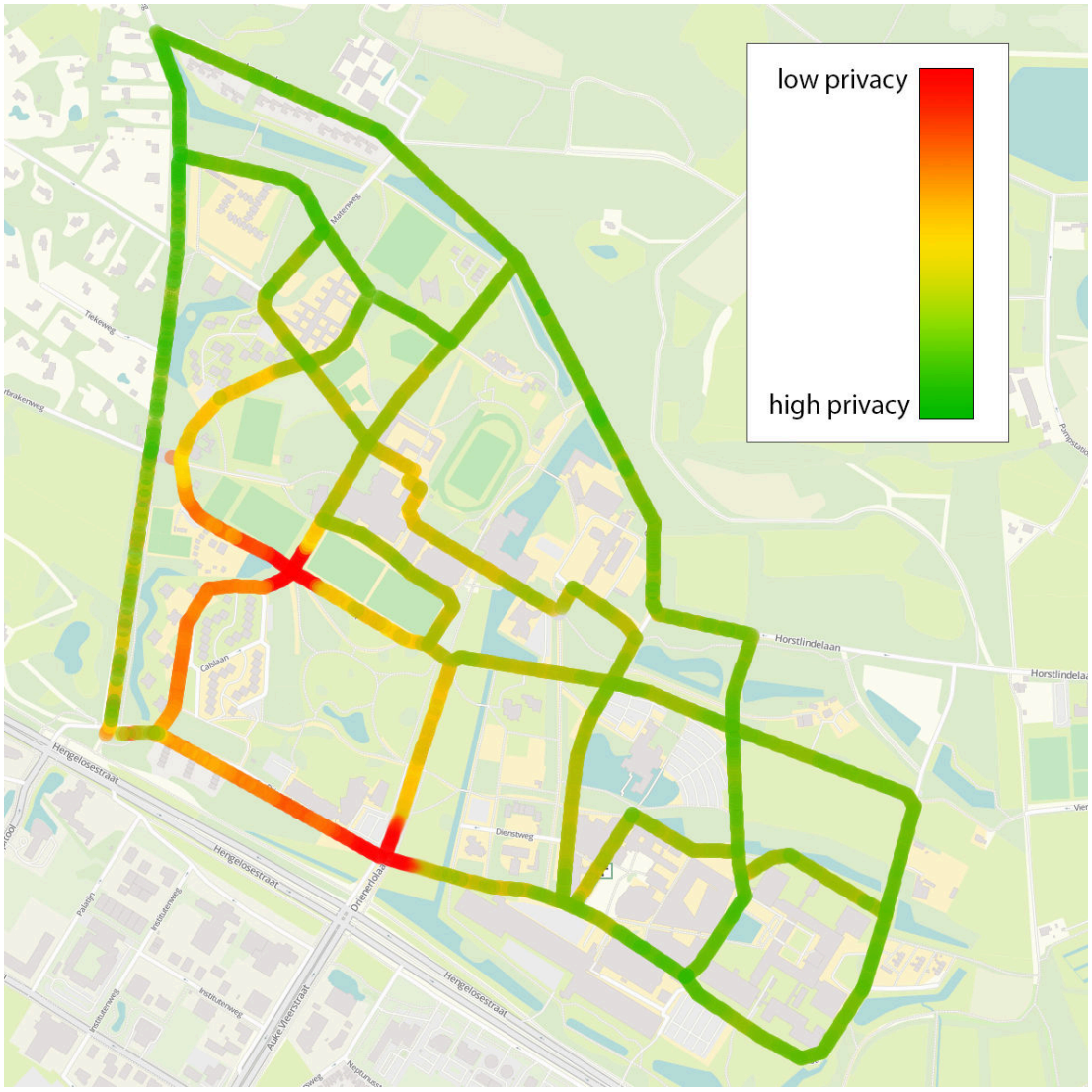

FIGURE 7.5: Privacy heatmap for an attacker covering two intersections

We can see that the vehicle has a high level of privacy for most of the university campus.

In the observed intersections themselves there is a low privacy level, as this is where the vehicle is actually eavesdropped on. Furthermore, we see that on the roads between the two intersections the privacy level is low to medium. In section 6.3.2 we established that with more observed intersection it was easier for an attacker to track the vehicle. This can be shown with our privacy flux function as well. If we assume a pseudonym change period of 300 seconds and that the attacker covers the same 8 intersections as shown in figure 6.13, we get the heatmap as shown in figure 7.6.

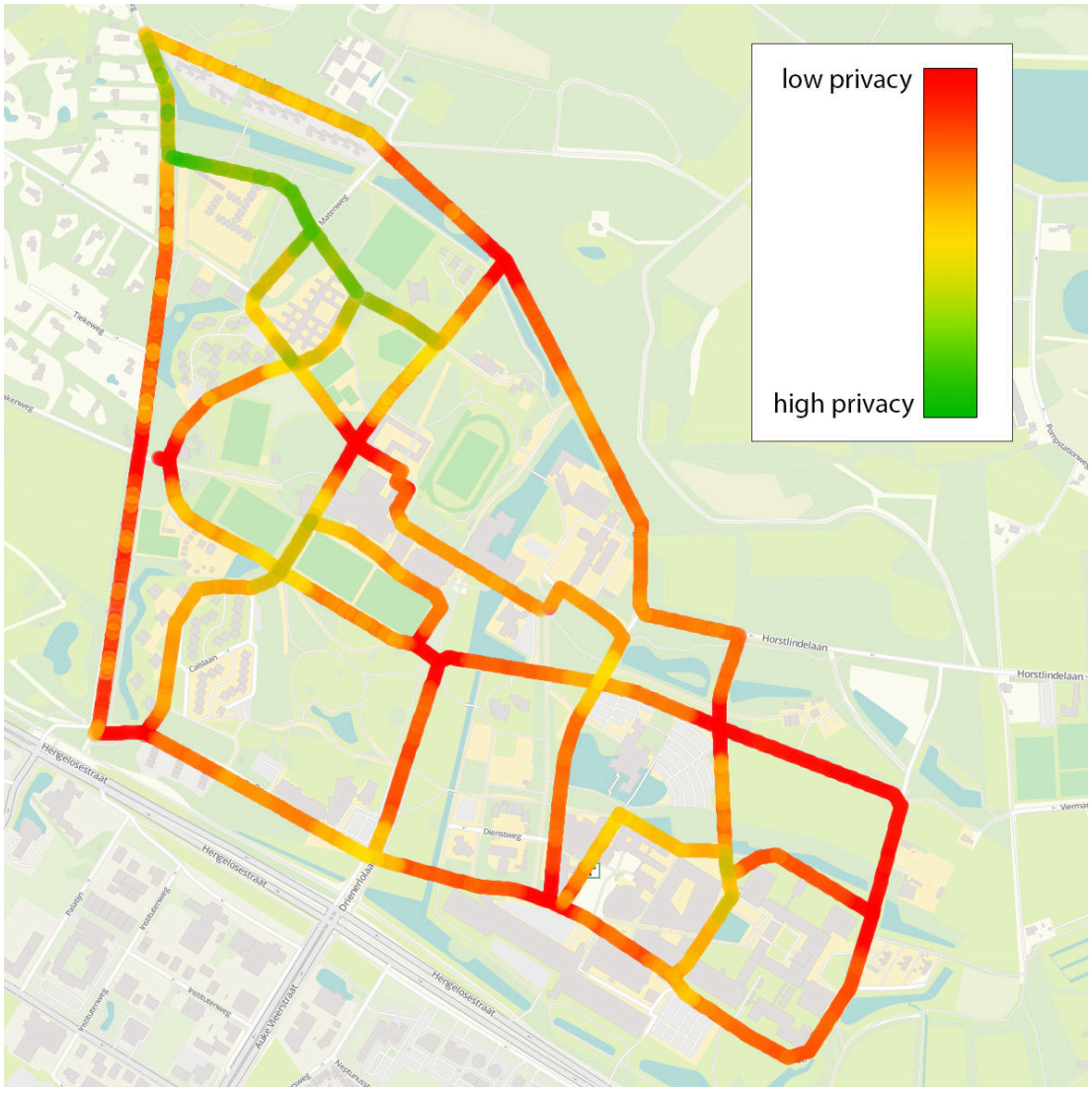

FIGURE 7.6: Privacy heatmap for an attacker covering eight intersections

In this figure, we can see that there is an overall low level of privacy. Only around unobserved intersection is the privacy level somewhat higher, with privacy being the highest around the cluster of unobserved intersections in the north. By using a heatmap to visualize our hybrid privacy flux function, it is possible to get an intuitive understanding of how the placement and quantity of sniffing stations affects the privacy level of a vehicle.

7.4 Expanding the Scale

Up to now, we have only considered the effects of pseudonyms in our limited university campus scenario. This scenario had a number of limitations in that it covered a small area and the trips taken by the vehicle in our experiment were relatively short. Furthermore, our results so far are only relevant for the specific road network of our experiment. In order to determine if our results are also applicable for different scenarios, it would be useful to analyse the same privacy metrics on a larger geographic scale. However, the vehicle in our scenario was mostly limited to the campus, and so our collected data were not useful for this. Fortunately, the United Stated the Federal Highway Administration shares data of past ITS research projects through its Research Data Exchange. One of the available datasets is one month of Basic Safety Messages (BSMs), the United States equivalent of CAMs, for transit buses in the city of Orlando, Florida [67]. Whilst buses may not be a likely target for attackers, we assume that they take similar routes to commuters driving between home and work, and thus are still useful to evaluate tracking and privacy. As the messages in the dataset contain for a large part the same information as CAMs, and the buses cover a significantly larger geographic area than our security vehicle, the dataset is well suited to analyse the same privacy metrics and pseudonym change strategies as above on a larger scale.

7.4.1 Identifying Intersections

The dataset for Orlando buses that we used consisted of all BSMs transmitted over a period of just less than a month. These BSMs contained, among other data, the GPS location of the transmitting vehicle, its bearing and the time elapsed since the start of a trip. Furthermore, the BSMs were transmitted with a frequency of 10Hz. The BSMs were thus very similar to the (S)CAMs that we used in our experimental scenario. Due to this similarity, we processed the data in the same manner as described in section 6.1. This left us with approximately 4.2 full days of driving data.

Despite the above similarities, there was one limitation to the data. In our scenario, we had sniffing stations at intersections receiving the messages sent from the vehicle. With the Orlando dataset, there were only messages transmitted from the vehicle itself and no data on when and where these messages were received. Thus, as with our expanded analyses for our experiment, we had to assume that an attacker could cover arbitrary intersections. This, however, meant identifying all intersections in the tracking domain. For our small tracking domain of the university campus, this was trivial to do manually, but for the entire city of Orlando this was not feasible. We therefore used

OpenStreetMap (OSM) data to attempt to automatically identify intersections in the tracking domain.

Firstly, an OSM datafile for the entire city of Orlando was downloaded, and subsequently filtered to contain only the roads. Each road in this remaining dataset consisted of a large number of nodes that described the different sections of the road. This also meant that any nodes that were part two or more roads represented roads crossing at that node, and thus represented an intersection. However, due to how roads are represented, the identified intersections required some processing. Firstly, multiple intersections were often identified close to each other. This occurred, for example, when a two-lane road crossed another two-lane road. Whilst in reality this was only a single intersection, our method identified 4 intersections at these points, one for each pair of crossing road sections. To solve this, all close intersections were grouped together to represent a single intersection. The algorithm identified 1023 intersections in the entire OSM dataset, and after grouping close intersections, 623 remained. Unfortunately, the automatic detection did not work perfectly, and sometime spurious intersections were identified, or existing intersections were not detected. This problem was mainly due to the quality of the OSM dataset. Furthermore, intersections were identified when a small road section left a road, for example at bus stops. These issues were fixed these manually as best we could. Despite the need for these manual adjustments, the process of identifying intersections with OSM data was considerably faster than manually identifying all intersections in the entire tracking domain.

As a last step, we removed all intersections that the vehicles did not pass through, as these would have no effect on privacy. We were left with 327 intersections that the vehicle passed through in the tracking domain. An overview of this tracking domain, along with all the intersections that we considered, can be seen in figure 7.7. The red lines depict everywhere the vehicles transmitted beacons, and the green dots represent the identified intersections. We can see that the vehicles covered a long and narrow area within the city. The total area of the tracking domain was approximately 48 km^2, giving an intersection density of 6.8 intersections per km^2. In comparison, our campus scenario had 21 intersections in an area of approximately 1.75 km^2, giving 12 intersections per km^2. Thus on average, the intersection density in Orlando was approximately half of what it was in our experimental scenario.

To see which roads within this tracking domain were used the most, a heatmap of the vehicle locations in the tracking domain can be seen in figure 7.8. Here we can see that the southern section of the tracking domain was used much more than the northern part. Furthermore, we can see some clear bus routes which were used most. Whilst the northern section was not have been used much, it is a residential zone and may still

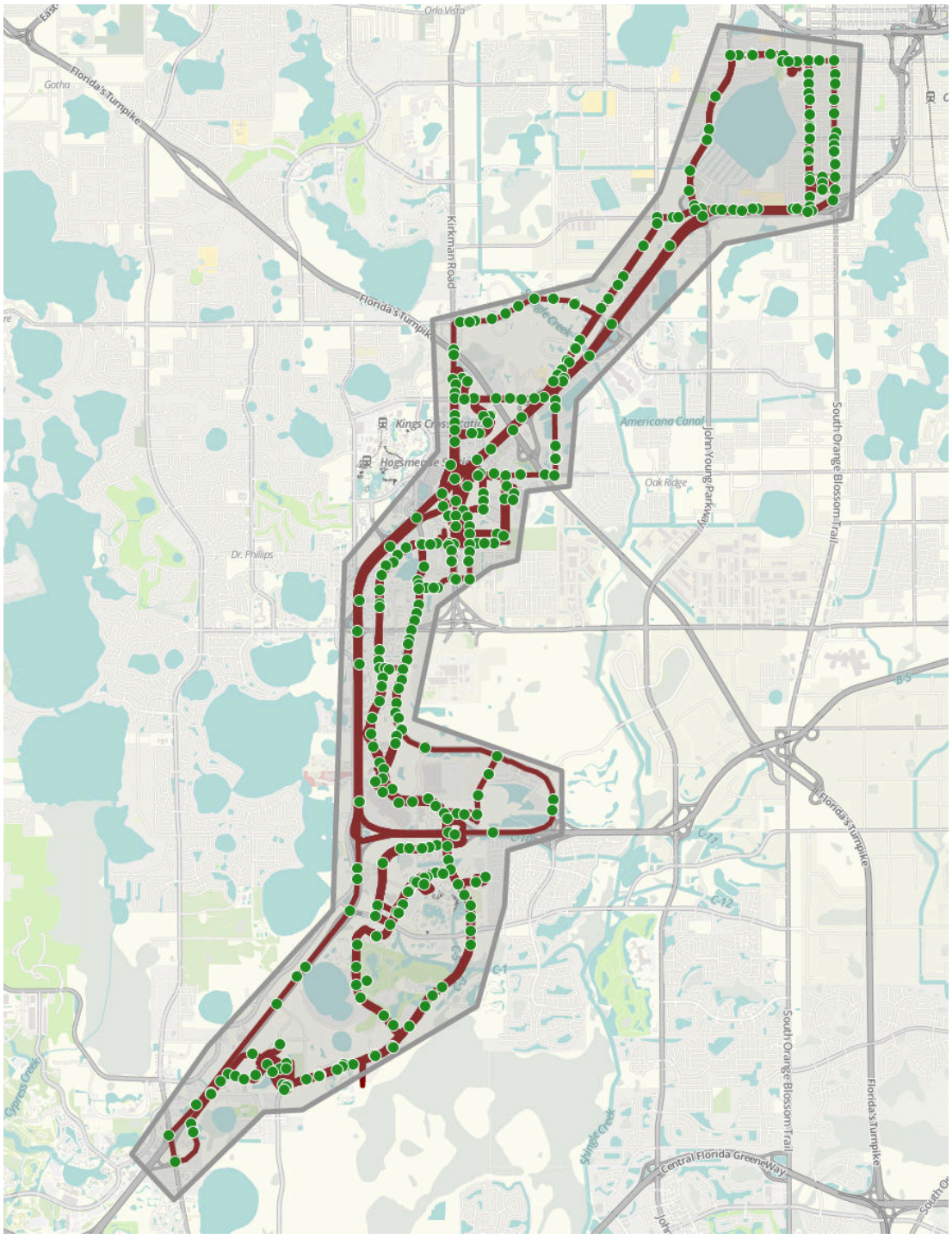

FIGURE 7.7: Map of the Orlando tracking domain and its intersections

be interesting to an attacker, depending on its goals. For example, an attacker trying to infer where someone lives could choose to deploy more sniffing stations in this area, despite it not being used as much by the vehicles.

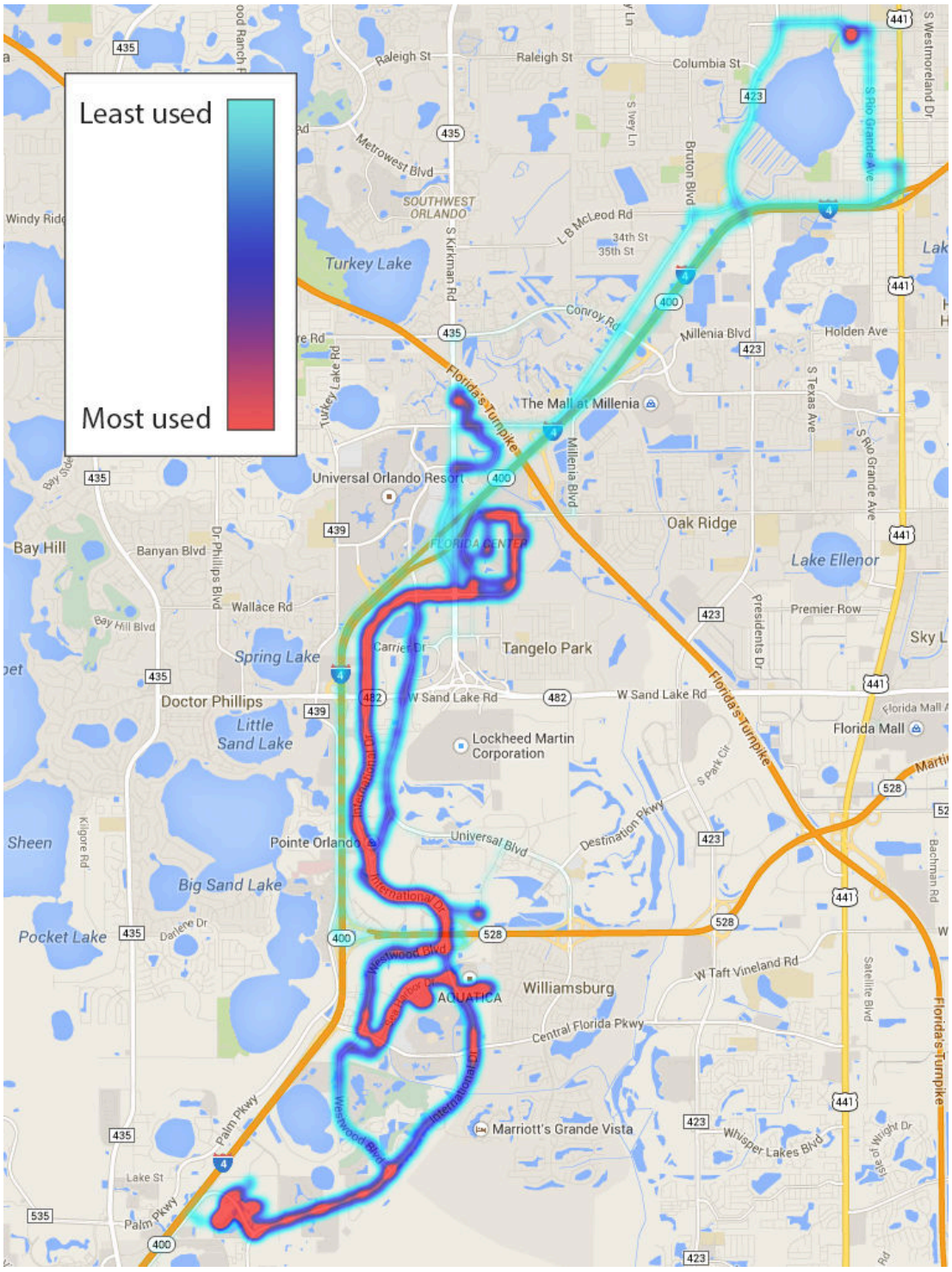

FIGURE 7.8: Heatmap of vehicle locations in Orlando

7.4.2 Pseudonym Effectiveness

We can now use this data to analyse pseudonym effectiveness on a larger scale than in our experimental scenario. Similar to section 7.3 we investigated different pseudonym change strategies, using both the MTT and the output of our privacy flux function to

quantify the level of privacy. For our experimental scenario, we measured these for all different combinations of each number of intersections. This was possible as the number of intersections was small, and thus there were not that many combinations to consider. However, for the Orlando tracking domain, we had 327 intersections, and this was no longer feasible. For example, if an attacker covers 10 of these intersections, there are $\binom{327}{10} = 3.35 \times 10^{18}$ different combinations to consider. As this is computationally infeasible in a reasonable amount of time, we randomly sampled 100 different combinations for each number of covered intersections, and took the mean result from these selected combinations.

Similar to figure 7.1, we calculated the MTT for different pseudonym change strategies, considering an attacker that could not link intertrip pseudonym changes. The results of this can be seen in figure 7.9. As we can see, these results are similar to what we saw in the experimental scenario. When an attacker covers fewer than 90, or approximately 28%, of all intersections, there is not much difference between pseudonym change strategies. It is only when an attacker covers more than a third of all intersections that we start to see a difference. One difference that we do see between the experimental scenario and the Orlando scenario is that the latter has a higher absolute MTT. This is because the trip durations of the vehicles in the Orlando scenario were longer than in our experimental scenario, with a maximum trip length of 7618 seconds and an average trip length of 3697 seconds.

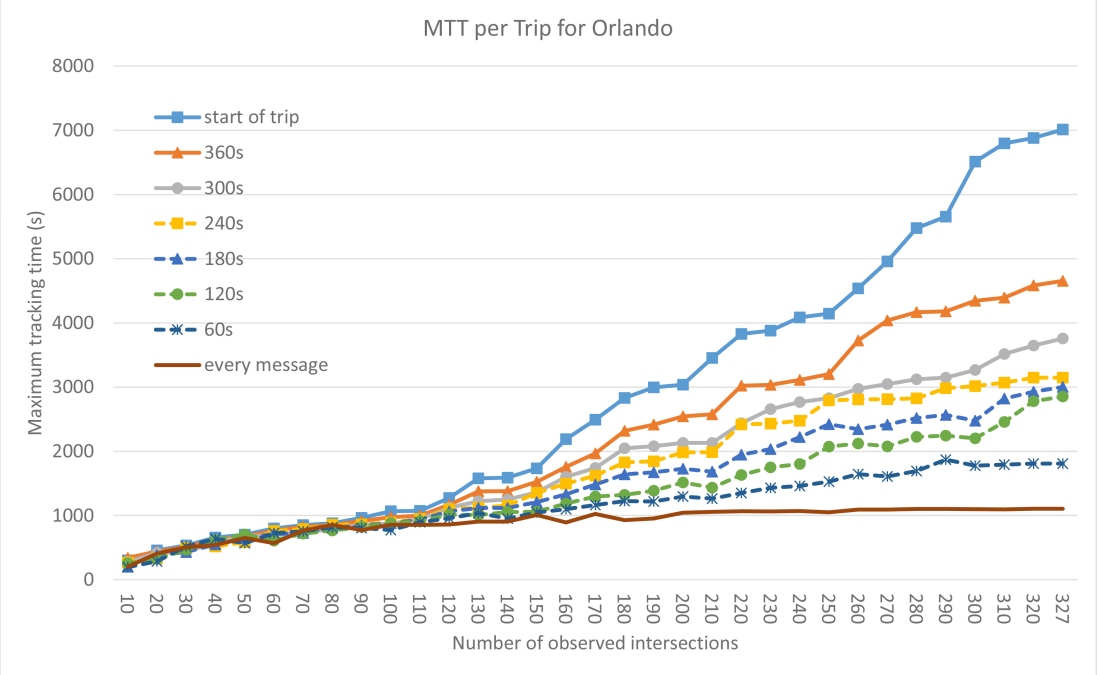

FIGURE 7.9: Maximum tracking time for (unlinked) trips in Orlando scenario

Similar to figure 7.2, we also calculated the MTT for different pseudonym change strategies considering an attacker that could link intertrip pseudonym changes. Again we considered the situation where an attacker could link intertrip changes, the results of which can be seen in figure 7.10.

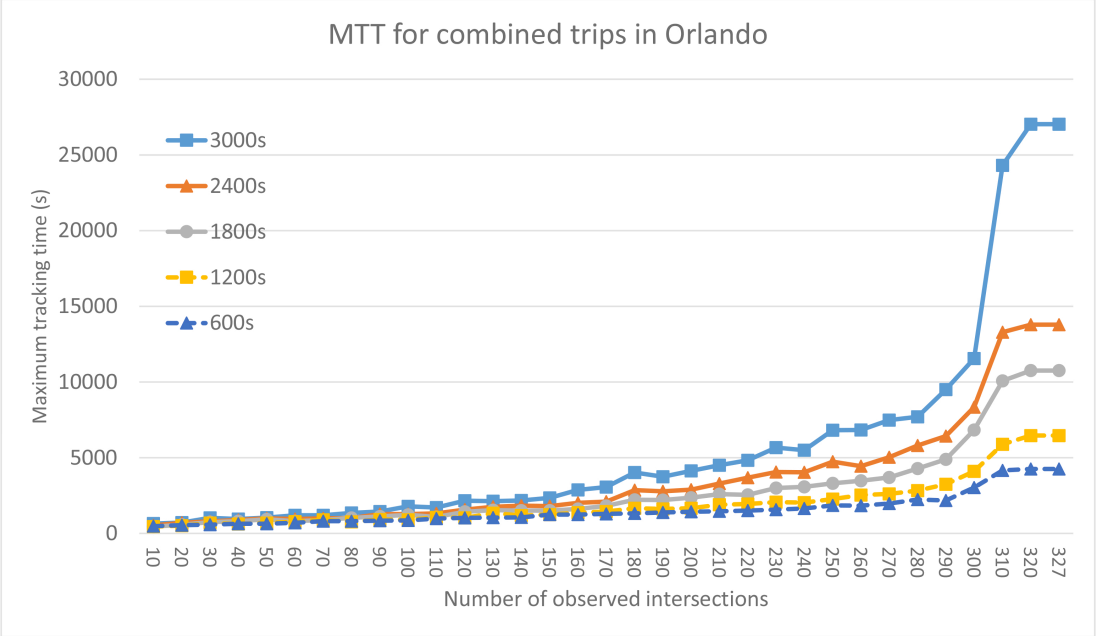

FIGURE 7.10: Maximum tracking time for combined trips in Orlando scenario

We can see that the overall trend is very similar to what we saw in figure 7.2. For an attacker that covers fewer than $90/327 \approx 28\%$ of all intersections, there is not a big difference between the different pseudonym change strategies. Between 90 and 270 covered intersections there seems to be a gradual increase in MTT, and covering more than $260/327 \approx 80\%$ of all intersections gives a more significant effect. For our experimental scenario, we found that these values were approximately 29% and 76%, and so these changes occur at around the same point. The only real difference between the two scenarios seems to be the absolute MTT. For example, in our experimental scenario, the MTT for a vehicle changing pseudonyms every 3000 seconds was approximately 12500 seconds, whereas for the Orlando scenario this value doubled to approximately 26000 seconds. In fact, the MTT seems to be approximately double for all the pseudonym change periods that we considered. One possible reason for this is that the intersection density in our experimental scenario is double that of the Orlando scenario. As the distance between observed and unobserved intersections increases, the time until an attacker can be confused on a road-level increases as well. Our results seem to suggest that this is a linear relationship. However, more scenarios would need to be examined before this could be concluded. Note that as we take 100 random combinations for each number of intersections, we cannot draw fine-grained conclusions on the exact effects of

the different pseudonym change strategies. Despite this, the overall trend of a higher number of observed intersections leading to a larger effect on the MTT, is valid.

There is however one problem with these results, especially at lower numbers of covered intersections. Due to randomly sampling 100 different combinations for each number of intersections, there is a high probability that the optimal combination that gives the highest MTT is not among them. For example, if we assume an attacker that can cover only 10 out of the 327 intersections, the 100 random combinations will likely consist of intersections that are relatively far apart. These isolated intersections will however not lead to the true MTT. By covering intersections that are far apart, there is a high probability that the attacker will be confused by pseudonym changes or unobserved intersection between the observed intersections. The MTT will be higher if all these intersections are adjacent, as then there is a smaller chance that the attacker will be confused in the time that the vehicle travels between observed intersections. The true MTT is only achievable with the optimal combination. Unfortunately it is not feasible for us, or for the attacker, to compute this optimal combination for our data, as this would require exhaustively testing every single possible combination.

We can however give a small practical example, to get an idea of the true MTT. We consider an attacker that has the resources to cover 10 intersections. In practice, an attacker is not likely to pick a random selection of intersections to cover, and will probably choose 10 adjacent intersection in an area where it can gain useful information. For our example, we consider that the attacker wishes to cover the highway in the west of the Orlando tracking domain seen in figure 7.7. A highway may be an attractive target for an attacker, as where a vehicle enters and exits a highway gives a good indication about which part of the city a vehicle comes from and goes to. Covering 10 adjacent intersection on this highway and assuming a pseudonym change period of 300 seconds, we get an MTT of 637 seconds. In figure 7.9 this value was 279 seconds, and so we can see that in this case the true MTT is about twice as high as the average MTT from random combinations.

Apart from the MTT, we can also look at the average privacy level based on our hybrid privacy flux function. In our experimental scenario we assumed that every time the vehicle changed pseudonyms, there were 3 other neighbouring vehicles that could change at the same time. This seemed realistic for the university campus, as there is usually a low traffic density. For the Orlando scenario on the other hand, the vehicles drove on well used public roads, and so it is likely that there was a higher traffic density. For this reason, we assumed that with every pseudonym change, there were 25 vehicles that changed at the same time. As with the MTT results above, we used a random sampling of 100 different combinations per number of observed intersections, to decrease the

computation time. The normalized privacy level under this assumption can be seen in figure 7.11.

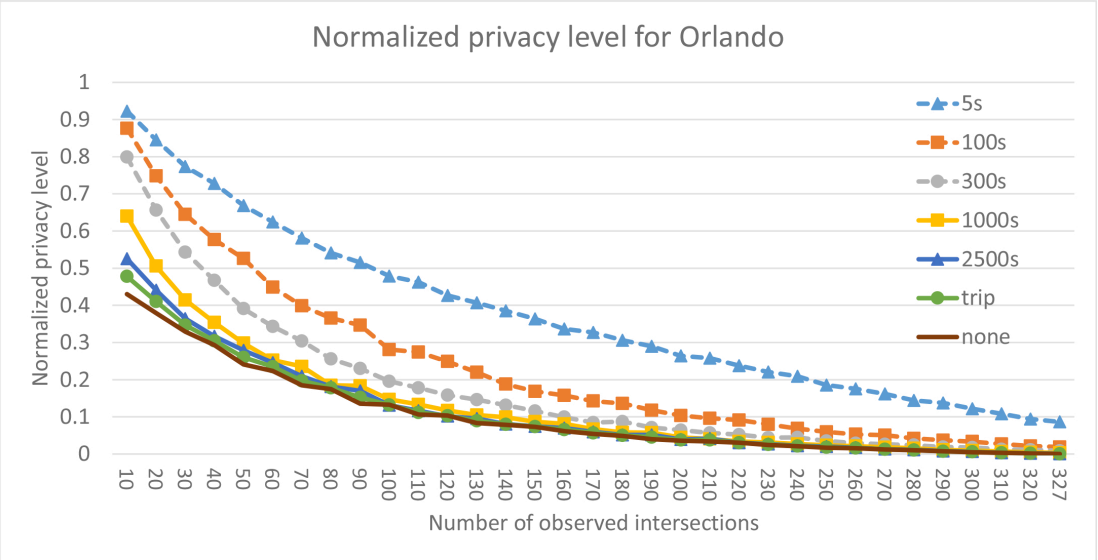

FIGURE 7.11: Privacy level for Orlando scenario

In this case, we see almost exactly the same results as with our experimental scenario in figure 7.4. We see that changing pseudonyms more often, consistently gives a greater level of privacy. We also see that when the majority of the intersections are covered, the exact pseudonym change period does not have a large effect on the privacy level. Note that as there is a larger number of roads and vehicles in the tracking domain, the maximum absolute privacy level in this scenario is higher than in our experimental scenario. However, as we look at the normalized pseudonym change, we do not see the effects of this, and can compare the two results directly. On average, the privacy level of changing pseudonym every 100 seconds is 50.9% of the privacy level when changing every 100 seconds. The difference going from 100 seconds to 300 seconds is 34.5%, and for consecutively longer periods this difference continues becoming smaller until it is almost 0 when comparing the per trip pseudonym change period with not changing pseudonyms at all. However, whilst the overall trends are correct, we again need to be careful drawing conclusion on the exact data points as this is the average of a random sampling of 100 combinations of intersections.

In conclusion, we see that we get very similar results in the Orlando scenario as we got with our experimental scenario, despite covering a much larger geographic area and a completely different road network. These similar patterns suggest that the results of our experimental scenario are useful for other scenarios as well. Especially for the MTT, there seems to be a direct relation between the intersection density and the MTT. However, this would need to be investigated further before any conclusions can be drawn.

7.5 Cost Analysis

In section 6.3.5, we described what the financial costs are for an attacker in terms of the number of observed intersections, and thus the number of sniffing stations that are deployed. We now consider how pseudonyms affect the costs for an attacker. Again, we assume a sniffing stations cost of €500, with the caveat that this is likely to drop significantly in the near future. The main effect that pseudonyms have on the financial resources required by an attacker, is that they increase the number of sniffing stations that an attacker needs to deploy to get the same tracking capabilities as without pseudonyms. The extent of the impact is dependent on the objectives of the attacker. For example, if an attacker wishes to achieve an MTT of 1000 seconds as shown in figure 7.1, it would need to deploy $12 \times 500 =$ €6000 worth of sniffing stations, assuming the vehicle changed pseudonyms once per trip. However, if the vehicle starts to change pseudonyms every 300 seconds, the same attacker would require 19 covered intersections with a total cost of €9500. Thus an attacker would need to spend 58% more to get the same level of tracking. The same situation is evident when we look at the normalized privacy level in figure 7.4. In this case, using no pseudonyms and covering 8 intersections leads to a normalized privacy level of 0.16. To get the same privacy level with a vehicle that uses pseudonyms and changes every 300 seconds, the attacker would need to cover 12 intersections. Again this increases the cost from €4000 to €6000, an increase of 50%. The situation is also the same for the Orlando scenario and an attacker that can link intertrip changes, as seen in figure 7.10. In this case, an attacker that wants to achieve a MTT of at least 5000 seconds, would need to cover around 220 intersection in the entire tracking domain if the vehicle changed pseudonyms every 2500 seconds. For the attacker this would entail a financial cost of $220 \times 220 =$ €110,000. If the vehicle then changed pseudonyms every 1000 seconds, the attacker would need to cover almost 320 different intersections which would cost €155,000, an increase of 45%. However, for the Orlando scenario, we must take care to conclude on the costs for an attacker, as the results do not contain optimal combinations of observed intersections. If an attacker carefully selects a set of intersections, the true MTT will be higher, and the costs to track to the same level will be lower. Furthermore we must also remain aware of the likelihood that the hardware costs will be much lower in the near future. If we assume a cheap sniffing station such as a Raspberry Pi with a cost of €50, the entire tracking domain in Orlando could be covered for a total of $50 \times 327 =$€16,350.

The above are extreme examples from our results, and often the increases in costs are less than this. However, the use of pseudonyms consistently make it more difficult for an attacker to track a vehicle to the same level as when no pseudonyms are used, and so it is always advisable to use some form of pseudonyms. Apart from only increasing

the number of sniffing stations required to track a vehicle to the same level, using pseudonyms may also force an attacker to change its tracking methods. For example, take a vehicle that changes pseudonyms every 300 seconds and an attacker requires a MTT of 6000 seconds as shown in figure 7.2. In this case, we see that this required MTT can never be reached, even if an attacker covers all intersections. The attacker would then need to deploy sniffing stations at additional points, for example at road sections instead of intersections. Alternatively, an attacker could be forced to use different tracking methods altogether. Both of these changes would require additional resources from an attacker.

7.6 Pseudonym Considerations

Using two different privacy metrics and different pseudonym change strategies, we found that there are two main factors that influence the level of privacy of a vehicle. The first is the number of intersections that the attacker observes, with a greater number of observed intersection leading to an overall lower level of privacy. The second is the pseudonym change strategy, where changing pseudonyms more often, and with more vehicles collaboratively, gives a greater level of privacy. There are however a number of problems that come with the introduction of pseudonyms.

One problem is that pseudonyms are only effective if a vehicle changes collaboratively with other neighbouring vehicles. However, it is impossible to guarantee that there will actually be neighbouring vehicles to collaborate with. This is especially true in areas with low traffic density. For example, on the university campus or in suburban areas, it is relatively unlikely that a vehicle has sufficient neighbours that are travelling in the same direction and geographically close enough that changing pseudonyms will sufficiently confuse an attacker. This situation is even worse in the initial roll-out phase for ITSs when the penetration rate is still low, as even in areas with a high traffic density there may only be a few ITS equipped vehicles to change pseudonyms with.

Furthermore, including pseudonyms in a system is not as trivial as merely changing an identifier. All linkable information needs to be changed, including public keys and certificates to ensure authentication and accountability. This means that pseudonym distribution needs a cryptographically secure infrastructure which may entail considerable overhead and limit how many pseudonyms can be distributed. It is also actually desirable to limit access to pseudonyms, to prevent vehicles from using multiple pseudonyms at the same time. We saw that changing pseudonyms more often will increase privacy, but in practice it will probably not be possible to change pseudonyms too often. Finally

there are also side-effect of pseudonyms that need to be taken into account, such as the negative effects on routing performance [33].

It is clear that pseudonyms cannot completely mitigate tracking in the presence of an MA. Despite this, we still consider pseudonyms to be a valid mitigation technique. Especially in high traffic density areas, there will be sufficient opportunities for a vehicle to collaboratively change pseudonyms to confuse attackers and make it harder to execute timing attacks. Furthermore, pseudonyms increase the resources that an attacker requires to track a vehicle. Not only does an attacker require more sniffing station to track a vehicle than if there are no pseudonym changes, successful changes also require re-identification of a target vehicle to track it over a longer period of time. Thus we conclude that it is not the goal of pseudonyms to mitigate tracking completely, but in fact to make it sufficiently difficult and resource intensive for an attacker to track a vehicle that the costs start to outweigh the incentives.

Chapter 8

Discussion & Conclusion

8.1 Discussion & Conclusion

In this thesis we investigated sousveillance on intelligent transportation systems, examining how feasible it is to track vehicles using public information as well as ITS hardware that anyone can easily acquire. More specifically, we investigated how easily a vehicle could be tracked in the presence of a mid-sized attacker, an attacker that has only partial coverage of a road network, but is capable of choosing which parts of the networks it covers. We investigated this by deploying actual ITS hardware as sniffing stations and in a vehicle, so that we could perform an empirical, real-world experiment. In this chapter we give an overview of our initial research questions and how we answered them. We then discuss our work and describe what important areas of research remain before the questions can fully be answered. We then conclude this thesis with our final words of advice and concern.

8.1.1 Research Questions & Overview

In chapter 3 we defined five research questions that we aimed to answer. Below we will shortly discuss the answers to these questions and how we obtained them.

Our first research question was how the vertical positioning of a sniffing station affected its coverage. To answer this, we set up an 802.11p transmitter at ground level, and measured the packet error rate at a receiver at different heights. We investigated high-gain and low-gain antennas and found that the type of antenna and its corresponding elevation radiation pattern had a significant effect on packet reception at different elevations. We also found that antenna elevation can have a significant impact on sniffing station coverage when placed higher than a few meters from the ground, but that using

the correct type of antenna was most important to ensure good coverage. Contrary to our expectations, placing an antenna at a higher elevation was not always better for coverage. Based on this, we concluded that antenna type and placement needs to be evaluated on a case by case basis, as these factors are determined by the environment where a sniffing station is to be placed.

Having looked at antenna placement at a given location, our next research question was how an attacker can determine where sniffing stations should be placed in the road network. We identified road intersections as good locations as they often give a good view of the road network and give good vehicle coverage. More importantly, intersections largely define what road any given vehicle will be driving on until the next intersection. The next problem was then to identify how an attacker with limited resources could decide which intersections to cover. To answer this, we represented the road network as a graph, with intersections as vertices and the roads between them as edges. We identified three criteria to determine which intersections were best to cover. Firstly, vertices with a large degree represented intersections where information could be collected on the most roads and, similar to this, the busiest intersections would give the most vehicles in range. Finally, we identified articulation points, vertices that when removed partition the graph, as good candidate locations.

The next research question was to determine if a mid-sized attacker that eavesdropped on intersections was capable of tracking a vehicle in a real-world scenario. To answer this, we set up an empirical experiment on the campus of the University of Twente. We first identified which intersections to cover based on the criteria above. Based on the graph of our tracking domain, we identified two intersections that we needed to cover, and deployed sniffing stations at these locations on the university campus. Next we installed a transmitting station in the vehicle of the campus security department. Based on data collected over a period of 16 days, we investigated how this data could be used to track the vehicle. We first described how the data could be cleaned up prior to analysis, and investigated the use of linear interpolation and dead reckoning as methods to predict vehicle positions and close small gaps in trip data. We then proposed a combination of the two called LIDR weighted averaging. This method was evaluated and subsequently used to process the data. Using this process data, we examined two different tracking methods, namely a most likely route based approach for road-level tracking and a most likely zone based approach for zone-level tracking. We concluded that tracking the vehicle was only marginally possible with only two sniffing stations, and that the situation was made more difficult due to a low quality GPS receiver and the non-standard mobility pattern of the vehicle.

We then continued our analyses with the ground truth data from our vehicle, to answer the research question of what the relation is between the resources available to an attacker and its tracking capabilities. We answered this by identifying an additional 19 intersections that could be observed, and then assumed that the attacker had resources to deploy more than two sniffing stations. From this, we investigated what the effect of such an attacker was on road-level and zone-level tracking. We found that increasing the number of sniffing stations made tracking much more feasible, and the only 8 out of 21 intersections were needed to give a road-level tracking rate of over 90%. Additionally, we found that only 5 intersections needed to be observed to correctly predict the zone the vehicle was in for over 95% of the time. From these results we concluded that it is definitely feasible for an MA to track a vehicle if it has the resources to cover sufficient intersections. To give a better indication of the financial resources required for an attacker, we also gave a cost analysis. We found that with a current market price of €500 for a sniffing stations, covering the university campus is well within the reach of a dedicated attacker. Finally, we concluded that hardware prices are likely to drop significantly in the near future, and that this development makes tracking an attractive target for attackers when ITSs are deployed.

Our final research question was to determine how effective pseudonyms are as a strategy to mitigate tracking and how this could be measured. We established what the different pseudonym change strategies are and which privacy metrics exist to evaluate these pseudonym change strategies, as well as to evaluate the use of pseudonyms as a whole. We then used the maximum tracking time as a metric to evaluate the effectiveness of two different pseudonym change strategies. After that we looked at using a privacy loss function as a privacy metric, but found that existing loss functions have only been proposed in the presence of a global attacker and not for mid-sized attackers. To solve this, we proposed a privacy function that takes into account different sources of uncertainty that cause a vehicle to gain or lose privacy in the presence of a mid-sized attacker. This hybrid privacy flux function was then used to evaluate different pseudonym change strategies, and we found that pseudonyms are most useful when changed relatively often. We also used the flux function to give a visual representation of privacy in our tracking domain, allowing a quick, intuitive grasp of the privacy level that a vehicle has at any location in the tracking domain. We then used beaconing data from transit buses in Orlando to evaluate the same pseudonym change strategies and metrics in a larger geographic domain, and gave a cost analysis to determine what the effects are of pseudonym on the financial resources of an attacker. We concluded that pseudonyms cannot completely mitigate tracking, but that they are useful as they can increase the resources that an attacker needs to track a vehicle to the same extent as when no pseudonyms are used.

8.1.2 Discussion

In this thesis we investigated tracking feasibility and mitigation in the presence of a mid-sized attacker with the data from our real-world experiment. There were however a number of limitations to our experiment that could have been improved. Firstly, the GPS receiver that we used in the vehicle did not work as expected during the experiment. There were considerable periods of time where the GPS receiver did not have a reliable location fix or even no fix at all. This meant that we had to discard a large number of messages as they were not useful to us without location data. Even though this could have been easily solved by using a high quality GPS receiver, we only noticed the problems after the experiment had already completed and it was too late to change receivers. A better GPS receiver could also have given more accurate data, as some modern receivers have a polling rate of 10Hz, instead of the 1Hz of our receiver.

Another problem that we had was the non-standard mobility pattern of the vehicle used in the experiment. As the vehicle belonged to the campus security department, it was able to drive in areas that we had not considered at the start of the experiment. This had a negative effect on the graph based approach that we used to determine sniffing station placement. Additionally, the vehicle did not have the speed profile of a normal passenger vehicle, and due to patrols it often slowed down at unexpected places. Whilst this mobility pattern did give realistic results for the specific use case of tracking a security vehicle, it is not directly comparable to that of a standard passenger vehicle. As a passenger vehicle tends to follow relatively predictable routes at predictable speeds, it may well be that tracking these kinds of vehicles is easier than what was possible with our data.

Our experimental results were also affected by the limited number of sniffing and transmitting stations that we were able to deploy. In terms of sniffing stations, we found that two were not sufficient to track a vehicle. We analysed the tracking on a larger scale by assuming that an attacker could in theory receive packets within a 35 meter radius of an intersection. In reality, the exact coverage depends heavily on the environment. Thus more accurate results could have been achieved if we had been able to deploy additional sniffing stations. In terms of transmitting stations, we only had a single vehicle that transmitted SCAMs in our experiment. This limitation was most evident when analysing pseudonym effectiveness. As pseudonym changes are only effective when done collaboratively with other vehicles, we had to make assumptions about the presence of these vehicles. Whilst assuming perfect pseudonym changes did allow us to get an upper bound of the effectiveness of pseudonym changes, more accurate and realistic results could have been achieved if we had been able to deploy more ITS equipped vehicles in our experiment. Having said this, increasing the scale in such a manner means a larger

time investment as well. As such it would not have been possible in the limited scope of a master thesis, and so we suggest an expanded scale experiment as future work in the next section.

Looking at measuring privacy, we used the MTT as a privacy metric to evaluate the effectiveness of pseudonyms in different scenarios. However, we must be careful when using these metrics that we understand what they show, as they may not fully capture the true level of privacy of a vehicle. For example, consider a vehicle that takes a trip lasting 1000 seconds, and that it can be tracked for the whole trip except for the 500th second. The MTT in this case will be 500 seconds, even though there is only a single second when the vehicle cannot be tracked (our privacy flux function would fare better in this case, as the average level of privacy would be very low). Whilst this is an extreme example, it does indicate that care must be taken when interpreting the results of privacy metrics, and that we must understand their underlying characteristics and idiosyncrasies. In the next section we stress the need for further research into good privacy metrics.

Despite the above, we have shown that a mid-sized attacker with sufficient resources is capable road-level tracking in our tracking domain. Furthermore, we have shown that whilst pseudonyms cannot eradicate location privacy problems in intelligent transportation systems, they can still be a useful mitigation strategy. Even though pseudonyms may not be effective in certain situations such as low traffic density, they still impose an extra barrier to increase the technological and financial resources that an attacker requires to track a vehicle. This extra barrier is important, as the financial resources that an attacker requires to track a vehicle, in combination with the dropping hardware costs, make the presence of attackers increasingly likely as ITSs are deployed. This means that mid-sized attackers are a realistic threat to ITSs in the near future. As road users we need to be especially aware of this threat. In the not too distant future, we will likely not have a choice whether or not our vehicles will be ITS equipped, and whether or not our vehicles continually broadcast our location. As these future systems are being developed now, now is also the time to ensure that location privacy is implemented by design instead of as an afterthought.

8.2 Future Work

We have shown that tracking in the presence of an MA is feasible and how pseudonyms can be used to make it more difficult for an MA to track a vehicle for long periods of time. However, we also identified a number of improvements to the experiment that would have allowed for more results or more accurate results. Furthermore, there are a

number of ideas that fell outside of the scope of our research. In this section we give an overview of how we believe our research efforts can be extended as well as give salient new topics that will give a better indication of how feasible tracking really is, especially in the presence of an MA.

8.2.1 Experimentation

In our experimental scenario we only had a single vehicle to investigate tracking feasibility and the effect of pseudonyms. Even though this was sufficient to show pseudonym effectiveness, it did require us to make certain assumptions about the presence of neighbouring vehicles to collaboratively change pseudonyms with. More accurate results could be obtained if the experiment was repeated on a larger scale. By running the experiment with a larger number of ITS equipped vehicles, a real-world analysis of pseudonym effectiveness could be performed. Such an experiment would need to include sufficient ITS equipped vehicles to accurately analyse how effective collaborative pseudonym changes truly are in low and high density scenarios. Likewise, the number of sniffing stations could also be increased to get a more accurate view of an attacker with the resources to cover multiple intersections.

8.2.2 Tools

In section 6.3.4 we described a simulation tool that can be used by an attacker to determine beforehand what the expected coverage of an intersection will be. An attacker can use this to predict the coverage of a sniffing station at an intersection before it is actually deployed. We believe that automated tools such as these will significantly improve the effectiveness of an attacker in tracking vehicles, especially in the initial stages where they can facilitate an attacker in setting up a sniffing network. An example of tools that would have assisted an attacker in our scenario are automated methods to transform a road network into a graph that an attacker can use to plan sniffing station placement. With this, graph theoretic analysis to find articulated points and biconnected components could help an attacker to identify which combination of observed intersections would be most effective. Additionally, in section 7.4 we described how we used OSM data to automatically identify intersections. However, the resulting intersections still needed manual processing to remove spurious intersections and add missing intersections, and so there was still considerable room for improvement. As tracking starts with the initial step of deploying sniffing stations, we believe investigating the potential capabilities of these kinds tool to help attackers should be investigated.

Besides this, we also described the need for vehicle re-identification every time an attacker can no longer link pseudonyms to a previous identification. The capabilities and deployment costs of hardware and software to accomplish this would be a good area of research to investigate how feasible it is to subvert pseudonyms completely.

8.2.3 Tracking Improvements

Up to now, we have only looked at tracking by using the (S)CAM beacons that vehicles transmit periodically. However, there are many other sources of information that may improve tracking performance. For example, the messages that vehicles sent may contain additional identifying information apart from the (pseudonymous) identifier which we have already discussed. The CAM specification defines the *lightbarinuse* field to indicate when an emergency vehicle's emergency lights are turned on, and there are also fields called *stationLength* and *stationWidth*. If the contents of these fields can be used to distinguish a specific vehicle, then the advantages of pseudonym changes will be circumvented and they will not be effective.

There are also external data sources that an attacker may be able to use. For example, an attacker could use traffic data from public traffic sources such as Google Maps to improve vehicle tracking. Additionally, real-time navigation data could be used to predict which routes vehicles are going to take. Drivers are becoming increasing reliant on in-car navigation services. As anyone may be able to use these services and request the same navigational data, and attacker could use this information to continually predict which route a vehicle may take. This will become increasingly relevant when self-driving vehicles start to come in use, as these vehicles will always follow the routes prescribed by a navigation service.

With more and more data becoming available and ever advancing data mining methods to search through and process these data, it may be that it becomes possible for an attacker to link pseudonyms in ways we have not yet considered. More work needs to be done to analyse tracking feasibility holistically, taking into account all the different sources of information that an attacker may have access to.

8.2.4 Road Topology

In chapter 7 we calculated the MTT and privacy level for our experimental scenario, as well as for the larger Orlando scenario. Even though these two scenarios have completely different road topologies, the result of our privacy metrics were very similar. This suggests that the privacy of a vehicle is not dependent on the road topology of where

it drives. However, we only managed to evaluate two scenarios and it would be an interesting avenue of research to investigate if there are certain classes of road topologies that may give an inherent lower or higher degree of privacy. As road topologies are static, this information would be useful for drivers. For example, if a driver knows it is currently driving on a road topology that does not give a high degree of privacy, the decision could be made to compensate by changing pseudonyms more often. Thus, we suggest the effects of road topologies on tracking as an interesting area of future work.

8.2.5 Hybrid Privacy Flux Function

In section 7.3.3 we introduced our hybrid privacy flux function to measure the privacy of a vehicle in the presence of a mid-sized attacker. This is however a first version of this function, and we can envision a number of improvements and extensions that could improve it. Firstly, we made the simplifying but strong assumption that the privacy level of a vehicle drops to 0 as soon as it is observed. In reality, this may however not necessarily be the case. For example, receiving a single CAM from a vehicle at an intersection may give an attacker no information on its ingress or egress direction, and thus will not completely reduce the privacy level to 0. This decrease in privacy would be better modelled by a decay function. How this decay function could model privacy decay in the presence of a mid-sized attacker would be a useful area of research to improve the flux function. Secondly, the function could be extended using models describing driver behaviour. For example, Lefèvre et al. suggested a technique to determine probabilities of manoeuvres at intersections based on their topology and geographic characteristics [68]. Combining this model with the flux function might give a more accurate indication of the amount of privacy that can be gained by crossing an unobserved intersection, and would be an interesting way to improve the function.

8.2.6 Silent Periods

During our investigation of pseudonym effectiveness, we assumed that vehicle did not use silent periods after a pseudonym change, as this could interfere with safety applications. There may however be situations where silent periods can be used. For example, there may be intersections where the intersection topography is such that the chance of a collision is very low. It could also be a user decision whether to use silent periods or not. If a user crosses an intersection where there have historically been few accidents, the decrease in safety due to pseudonyms may be worth the increase in privacy. Thus an occasional silent period could potentially lead to a significantly higher privacy level. Another factor that could affect the safety and the privacy level is where the silent period

occurs, for example at intersection, or on the road sections approaching intersections. For these reasons, it would be useful to investigate the effects of silent periods on the privacy level in the presence of a mid-sized attacker.

8.2.7 Privacy Metrics and Mid-Sized Attackers

In chapter 7 we gave an overview of existing privacy metrics and used two different metrics to evaluate privacy in our experimental scenario. However, the discussed privacy metrics all have their respective downsides and they do not fully capture and represent the true level of privacy that a vehicle has. Good privacy metrics are not only important to evaluate different mitigation strategies, but they would also allow a driver to determine when it believes it does not have sufficient privacy and take appropriate mitigating actions. To enable drivers to make these kinds of decisions, privacy metrics should give an intuitive quantitative value. Perhaps a single privacy metric is not sufficient to get the entire picture of the privacy level of a vehicle, and we need to look at certain combinations of metrics that do. Furthermore, it is not always obvious what a sufficient level of privacy is, as this can vary by location and driver. The perceived level of privacy is a matter of user experience, and more work needs to be done to investigate the link between the actual privacy of a user and how the user experiences this privacy.

Another problem is that most existing literature on pseudonyms and their effectiveness often assume a global attacker. A global attacker is however not realistic in practice, and a more likely threat is the mid-sized attacker. For these reasons, more work needs to be done to develop good privacy metrics that allow for quantitative comparisons not only between different pseudonym change strategies, but also between different mitigation strategies altogether, in the presence of a mid-sized attacker.

8.3 Final Words

It is our hope that through this thesis the reader has gained an appreciation of the complexity and necessity of ensuring location privacy in intelligent transportation systems. Moreover, we wish to emphasize the necessity of taking steps to ensure privacy at an early development stage. Especially as ITSs are quickly approaching the day where they will be an always-on feature of every vehicle, it is imperative that we start to implement privacy enhancing technologies now, instead of as an afterthought when the systems are already deployed. In a time where mass surveillance by governments seems to have

become commonplace, ITSs may open up even more ways for our privacy to be compromised. If ITSs are not developed with this in mind, we will hand our location data on a silver platter to anyone who is listening in.

Bibliography

[1] Stephanie Mann. Sousveillance Cartoon. Retrieved on 07/04/2014. URL `http://en.wikipedia.org/wiki/File:SurSousVeillanceByStephanieMannAge6.png`.

[2] ETSI TS 102 637-2 V1.2.1 - Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical report, ETSI, . URL `http://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01.02.01_60/ts_10263702v010201p.pdf`.

[3] ETSI TS 102 637-3 V1.1.1 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. Technical report, ETSI, . URL `http://www.etsi.org/deliver/etsi_ts/102600_102699/10263703/01.01.01_60/ts_10263703v010101p.pdf`.

[4] IEEE Draft Standard for Amendment to Standard [for] Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-Specific requirements-Part II: Wireless LAN Medium Access Control (MAC) and Physical Lay. *IEEE Std P802.11p/D11.0 April 2010*, (June):1–35, 2010.

[5] Panagiotis Papadimitratos, Virgil Gligor, and Jean-Pierre Hubaux. Securing Vehicular Communications - Assumptions, Requirements, and Principles. In *Workshop on Embedded Security in Cars*, pages 5–14, 2006.

[6] M Raya and JP Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15:39–68, 2007. ISSN 0926-227X.

[7] F. Schaub, F. Kargl, Zhendong Ma Zhendong Ma, and M. Weber. V-Tokens for Conditional Pseudonymity in VANETs. *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, 2010.

[8] Norbert Bibmeyer, Jonathan Petit, and Kpatcha M. Bayarou. Copra: Conditional pseudonym resolution algorithm in VANETs. In *2013 10th Annual Conference on*

Wireless On-demand Network Systems and Services (WONS), pages 9–16. IEEE, March 2013. ISBN 978-1-4799-0749-6. doi: 10.1109/WONS.2013.6578314.

[9] C. Drane, M. Macnaughtan, and C. Scott. Positioning GSM telephones. *IEEE Communications Magazine*, 36(4), 1998. ISSN 0163-6804. doi: 10.1109/35.667413.

[10] E Trevisani and Andrea Vitaletti. Cell-ID Location Technique, Limits and Benefits: An Experimental Study. *Sixth IEEE Workshop on Mobile Computing Systems and Applications*, pages 51–60, 2004. doi: 10.1109/MCSA.2004.9.

[11] K Dufková, Michal Ficek, Lukáš Kencl, and Jakub Novák. Active GSM cell-id tracking: Where Did You Disappear? In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 7–12, 2008. ISBN 9781605581897.

[12] MY Chen, Timothy Sohn, and Dmitri Chmelev. Practical metropolitan-scale positioning for gsm phones. *UbiComp 2006: Ubiquitous Computing*, pages 225–242, 2006.

[13] Mohamed Ibrahim and Moustafa Youssef. CellSense: A Probabilistic RSSI-Based GSM Positioning System. *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, December 2010. doi: 10.1109/GLOCOM.2010.5683779.

[14] Anand Oka, Lutz Lampe, and Senior Member. Distributed Target Tracking Using Signal Strength Measurements by a Wireless Sensor Network. *IEEE Journal on Selected Areas in Communications*, 28(7):1006–1015, 2010.

[15] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom '00*, pages 32–43, New York, New York, USA, 2000. ACM Press. ISBN 1581131976. doi: 10.1145/345910.345917.

[16] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking - MobiCom '99*, pages 59–68, New York, New York, USA, 1999. ACM Press. ISBN 1581131429. doi: 10.1145/313451.313476.

[17] Matt Duckham and Lars Kulik. Location privacy and location-aware computing. *Dynamic & mobile GIS: investigating change in space and time 3*, 2006.

[18] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Grutese, Wade Trappe, and Ivan Seskar. Security and privacy vulnerabilities

of in-car wireless networks: A tire pressure monitoring system case study. *19th USENIX Security Symposium*, 39(4):11–13, 2010. ISSN 0040-5736. doi: 10.1177/004057368303900411.

[19] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979. ISSN 0018-9286. doi: 10.1109/TAC.1979.1102177.

[20] Marco Gruteser and Baik Hoh. On the anonymity of periodic location samples. In *Second International Conference, SPC 2005, Boppard, Germany, April 6-8, 2005. Proceedings*, volume 3450, pages 179–192, 2005.

[21] M. Gruteser. Protecting Location Privacy Through Path Confusion. *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 194–205, 2005. doi: 10.1109/SECURECOMM.2005.33.

[22] B. Gedik. Location Privacy in Mobile Systems: A Personalized Anonymization Model. *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 620–629, 2005. doi: 10.1109/ICDCS.2005.48.

[23] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and ...*, 10(5):1–14, 2002.

[24] AR Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, January 2003. ISSN 1536-1268. doi: 10.1109/MPRV.2003.1186725.

[25] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, February 1981. ISSN 00010782. doi: 10.1145/358549.358563.

[26] Balaji Palanisamy and Ling Liu. MobiMix: Protecting location privacy with mix-zones over road networks. *2011 IEEE 27th International Conference on Data Engineering*, pages 494–505, April 2011. doi: 10.1109/ICDE.2011.5767898.

[27] Levente Buttyán, Tamas Holczer, and Istvan Vajda. On the effectiveness of changing pseudonyms to provide location privacy in VANETs. *Lecture Notes in Computer Science*, 4572:129, 2007.

[28] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-Zones for Location Privacy in Vehicular Networks. *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007.

[29] M. Gruteser and A. Alrabady. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing*, 5(4):38–46, October 2006. ISSN 1536-1268. doi: 10.1109/MPRV.2006.69.

[30] John Krumm. Inference Attacks on Location Tracks. *Pervasive Computing*, 10 (Pervasive):127–143, 2007. doi: 10.1007/978-3-540-72037-9_8.

[31] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. Pseudonym Schemes in Vehicular Networks: A Survey. *IEEE Communications Surveys & Tutorials*, PP(99):1–1, 2014. ISSN 1553-877X. doi: 10.1109/COMST.2014.2345420.

[32] Bjorn Wiedersheim, Zhendong Ma, Frank Kargl, and Panos Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 176–183. IEEE, February 2010. ISBN 978-1-4244-6059-5. doi: 10.1109/WONS.2010.5437115.

[33] J Freudiger, M Raya, E.Z. Ma, F Kargl, and Others. Secure Vehicular Communication Systems: Design and Architecture. *IEEE Communications Magazine*, page 101, 2008.

[34] Mingyan Li, Krishna Sampigethaya, Leping Huang, and Radha Poovendran. Swing & swap: User-Centric Approaches Towards Maximizing Location Privacy. In *Proceedings of the 5th ACM workshop on Privacy in electronic society - WPES '06*, page 19, New York, New York, USA, 2006. ACM Press. ISBN 1595935568. doi: 10.1145/1179601.1179605.

[35] Matthias Gerlach and Felix Guttler. Privacy in VANETs using Changing Pseudonyms - Ideal and Real. In *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pages 2521–2525. IEEE, April 2007. ISBN 1-4244-0266-2. doi: 10.1109/VETECS.2007.519.

[36] Leping Huang, Hiroshi Yamane, Kanta Matsuura, and Kaoru Sezaki. Silent Cascade : Enhancing Location Privacy Without Communication QoS Degradation. In *Security in pervasive computing*, pages 165–180. Springer Berlin Heidelberg, 2006.

[37] Krishna Sampigethaya, Leping Huang, and Mingyan Li. CARAVAN: Providing location privacy for VANET. Technical report, Washington Univ., Seattle. Dept. of Electrical Engineering., 2005.

[38] Krishna Sampigethaya, Mingyan Li, Leping Huang, and Radha Poovendran. AMOEBA: Robust Location Privacy Scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, October 2007. ISSN 0733-8716. doi: 10.1109/JSAC.2007.071007.

[39] Stephanie Lefevre, Jonathan Petit, Ruzena Bajcsy, Christian Laugier, and Frank Kargl. Impact of V2X privacy strategies on Intersection Collision Avoidance systems. *2013 IEEE Vehicular Networking Conference*, pages 71–78, December 2013. doi: 10.1109/VNC.2013.6737592.

[40] SAE J 2735 - Dedicated Short Range Communications (DSRC) Message Set Dictionary. Technical report, SAE.

[41] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services - MobiSys '03*, pages 31–42, New York, New York, USA, 2003. ACM Press. doi: 10.1145/1066116.1189037.

[42] Florian Schaub, Zhendong Ma, and Frank Kargl. Privacy Requirements in Vehicular Communication Systems. *2009 International Conference on Computational Science and Engineering*, 3:139–145, 2009. doi: 10.1109/CSE.2009.135.

[43] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, November 2012. ISSN 1617-4909. doi: 10.1007/s00779-012-0633-z.

[44] O. Trullols, M. Fiore, C. Casetti, C.F. Chiasserini, and J.M. Barcelo Ordinas. Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications*, 33(4):432–442, March 2010. ISSN 01403664. doi: 10.1016/j.comcom.2009.11.021.

[45] Mohamed Kafsi, Panos Papadimitratos, Olivier Dousse, T. Alpcan, and J. P. Hubaux. VANET Connectivity Analysis. *IEEE AutoNet*, December 2008.

[46] Javier Barrachina and Piedad Garrido. D-RSU: A Density-Based Approach for Road Side Unit Deployment in Urban Scenarios. In *International Workshop on IPv6-based Vehicular Networks (Vehi6)*, pages 1–6, 2012. ISBN 9788469534724.

[47] Yingsi Liang, Hui Liu, and Dinesh Rajan. Optimal Placement and Configuration of Roadside Units in Vehicular Networks. *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*, pages 1–6, May 2012. doi: 10.1109/VETECS.2012.6240345.

[48] Christian Lochert, Bjorn Scheuermann, Murat Caliskan, and Martin Mauve. The feasibility of information dissemination in vehicular ad-hoc networks. In *2007 Fourth Annual Conference on Wireless on Demand Network Systems and Services*, pages 92–99. IEEE, January 2007. ISBN 1-4244-0860-1. doi: 10.1109/WONS.2007.340478.

[49] Vinod Kone, Haitao Zheng, Antony Rowstron, and Ben Y. Zhao. On infostation density of vehicular networks. *Proceedings of the 5th International ICST Conference on Wireless Internet*, 2010. doi: 10.4108/ICST.WICON2010.8533.

[50] Maryam Rashidi, Iulian Batros, Tatiana K. Madsen, Muhammad T. Riaz, and Thomas Paulin. Placement of Road Side Units for floating car data collection in highway scenario. *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pages 114–118, October 2012. doi: 10.1109/ICUMT.2012.6459649.

[51] Junghoon Lee and CM Kim. A roadside unit placement scheme for vehicular telematics networks. In *Advances in Computer Science and Information Technology*, pages 196–202. Springer Berlin Heidelberg, 2010.

[52] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. Data aggregation and roadside unit placement for a vanet traffic information system. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking - VANET '08*, page 58, New York, New York, USA, 2008. ACM Press. ISBN 9781605581910. doi: 10.1145/1410043.1410054.

[53] Mikko Lehtinen, Ari Happonen, and Jouni Ikonen. Accuracy and time to first fix using consumer-grade GPS receivers. In *2008 16th International Conference on Software, Telecommunications and Computer Networks*, pages 334–340. IEEE, 2008. ISBN 978-953-6114-97-9. doi: 10.1109/SOFTCOM.2008.4669506.

[54] Mate Boban, Joao Barros, and Ozan Tonguz. Geometry-Based Vehicle-to-Vehicle Channel Modeling for Large-Scale Simulation. *IEEE Transactions on Vehicular Technology*, PP:1–1, 2014. ISSN 0018-9545. doi: 10.1109/TVT.2014.2317803.

[55] Xinzhou Wu. Future Technology Trends for Vehicular Communication - Retrieved on 22/10/2014. URL `http://www.ieeevtc.org/wivec2014/plenary.php`.

[56] Stanford Large Network Dataset Collection. Retrieved on 01/10/2014. URL `http://snap.stanford.edu/data/#road`.

[57] Bo Yu, Cheng-Zhong Xu, and Bin Xiao. Detecting Sybil attacks in VANETs. *Journal of Parallel and Distributed Computing*, 73(6):746–756, June 2013. ISSN 07437315. doi: 10.1016/j.jpdc.2013.02.001.

[58] Julien Freudiger, Mohammad Hossein Manshaei, Jean-pierre Hubaux, and David C Parkes. On non-cooperative location privacy: A Game-Theoretic Analysis. In *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09*, page 324, New York, New York, USA, 2009. ACM Press. ISBN 9781605588940. doi: 10.1145/1653662.1653702.

[59] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-pierre Hubaux. Unraveling an old cloak. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society - WPES '10*, page 115, New York, New York, USA, 2010. ACM Press. ISBN 9781450300964. doi: 10.1145/1866919.1866936.

[60] Chengyang Zhang and Yan Huang. Cloaking locations for anonymous location based services: a hybrid approach. *GeoInformatica*, 13(2):159–182, April 2008. ISSN 1384-6175. doi: 10.1007/s10707-008-0047-2.

[61] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 237, New York, New York, USA, April 2008. ACM Press. ISBN 9781605580852. doi: 10.1145/1367497.1367531.

[62] Li Ninghui, Li Tiancheng, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings - International Conference on Data Engineering*, pages 106–115, 2007.

[63] Agusti Solanas, Francesc Sebé, and Josep Domingo-Ferrer. Micro-aggregation-based heuristics for p-sensitive k-anonymity. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society - PAIS '08*, page 61, New York, New York, USA, March 2008. ACM Press. ISBN 9781595939654. doi: 10.1145/1379287.1379300.

[64] Andrei Serjantov and George Danezis. Towards an Information Theoretic Metric for Anonymity. In *Proc. Workshop on Privacy Enhancing Technologies*, pages 41–53, 2002.

[65] David Rebollo-Monedero, Javier Parra-Arnau, Claudia Diaz, and Jordi Forné. On the measurement of privacy as an attacker's estimation error. *International Journal of Information Security*, 12(2):129–149, December 2012. ISSN 1615-5262. doi: 10.1007/s10207-012-0182-5.

[66] Walton Fehr. 5.9GHz DSRC Roadside Equipment Device Specification. Retrieved on 01-12-2014. URL `http://www.its.dot.gov/safety_pilot/pdf/T-10001-T2-05_RSE_Device_Design_Specification_v30.pdf`.

[67] FDOT Orlando ITS World Congress Reseaerch Data Exchange. Retrieved on 13-11-2014. URL `https://www.its-rde.net/showds?dataEnvironmentNumber=10003`.

[68] Stéphanie Lefèvre, Christian Laugier, and Javier Ibañez Guzmán. Exploiting map information for driver intention estimation at road intersections. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 583–588, 2011.