VEHICULAR SENSING NETWORKS

EFFICIENCY, SECURITY & PRIVACY

By

BIN ZAN

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

written under the direction of

Marco Gruteser

and approved by

New Brunswick, New Jersey

October, 2013

ABSTRACT OF THE DISSERTATION

Vehicular Sensing Networks

Efficiency, Security & Privacy

by Bin Zan

Dissertation Director:

Marco Gruteser

The increasing integration of sensors and wireless communication devices into highly mobile platforms such as automobiles makes vehicular sensing networks one of the most promising platforms for many applications. The performance of applications such as traffic reporting, environmental monitoring and distributed surveillance can be improved by using the new techniques developed in vehicular sensing networks. Several distinct features such as highly mobile and predictable movement patterns make vehicular sensing networks different from general computer networks. Therefore, unique solutions for vehicular sensing networks are necessary. In this thesis, we show our efforts on three aspects of vehicular sensing networks: efficiency, security and privacy.

The efficiency issue is most critical when there is no central infrastructure or when vehicle-to-infrastructure (V2I) communication bandwidth is a precious resource. Keep uploading every piece of sensor data to a remote server is obviously inefficient. Local data aggregation is required to reduce the communication cost and improve the efficiency. However, how and when to perform the aggregation is not trivial. In this work, a GeoCache concept and Boomerang anchoring protocol are proposed to address this issue.

Our work in security is focused on building secret keys for both vehicle-to-vehicle (V2V) and the V2I communication modes. Many applications require secure V2V and

V2I communications and two sets of secret keys need to be created independently. Based on the special characteristics existing in vehicular sensing networks, we develop two key agreement algorithms to achieve the target.

In many of the applications developed within vehicular sensing networks, GPS data has to be submitted to the central server continuously, which results in serious privacy violation. We have three contributions under the privacy domain. First, we develop a privacy preserving algorithm which protects user privacy without filtering too many location traces. Second, we study the possible privacy leakage due to the detailed location information available. At last, we extend the privacy system design under the assumption that there is no location proxy server and the information available is limited.

Acknowledgements

It would be impossible to write this doctoral thesis without the unwavering support and encouragement of many people from teachers to family to friends to associates around me, to only some of whom it is possible to mention particularly here.

First and foremost, I would like to thank my advisor, Prof. Marco Gruteser for his valuable guidance and patience, not to mention his advice and unsurpassed knowledge of pervasive wireless systems: location-aware networking, measurement, vehicular networks, and location privacy. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive. He is showing me excellent example of joy and enthusiasm for his research as a successful scientist and professor, even during my tough times in the Ph.D. pursuit.

This thesis would not have been possible without the help, support and the good advice, of my co-advisors, Prof. Yanyong Zhang and Prof. Xuegang Ban, They advise me on both an academic and a personal level, for which I am extremely grateful. It was my honor to have done research work with them.

I would like to give my special thanks to Prof. Dipankar Raychaudhur, our lab director who is the most admirable professor I ever met. He is also the first professor I met at WINLAB seven years ago. In his office, he introduced me the research life at WINLAB, his valuable advice and speech inspired me.

I will also thank Prof. Wade Trappe, Prof. Roy Yates, Prof. Predrag Spasojevic and many other professors for imparting their knowledge to me from the bottom of their hearts.

Furthermore, thanks to Ivan Seskar, the most important person in our lab. Without his help and support, I can't imagine how I would implement most part of my research work.

I would also like to express my gratitude to Prof. Fei Hu, colleagues Tingting Sun, Zhanbo Sun, Peng Hao who have worked together with me, thanks for their hard work and their contribution on all the projects we work as a team.

Above all, I would like to give thanks to my loved ones, my wife Juan Wang for her personal support throughout my entire Ph.D program; my parents for giving me unequivocal support; my beloved daughters, Ivy and Kayla, for their magical smiles to cheer me up all the time.

I would also like to give thanks to many other friends and colleagues I met in WINLAB. Shridatt Sugrim, Kuo-Chun Huang, Baik Hoh, Sanjit Kaul, Mesut Ali Ergin, Sangho Oh, Jing Lei, Liang Xiao, Shu Chen, Dan Zhang, Yao li, Kishore Ramachandran, Yu Zhang, Chenren Xu, Zhibin Wu, Zhuo Chen, Tianming Li, etc. During the last seven years, I received countless helps and valuable advice from my friends and also because of them my life in WINLAB is full of juice.

Thanks to all the staffs working in WINLAB, Noreen DeCarlo, Elaine Connors, etc. Their hardwork make research life much easy in WINLAB, in Rutgers.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The increasing integration of sensors and wireless communication devices into highly mobile platforms such as automobiles enables novel pervasive monitoring services that continuously sense the surrounding environment and report events of interest on a real-time basis. Today's higher-end car already carry a range of sensors, e.g., rain gauges, accelerometers, GPS, wheel rotation/traction sensors, and cameras, which can be used to report on a variety of road conditions such as potholes, obstacles, or slippery roads. The ability to timely communicate such events with road management authorities, who can issue warnings to following vehicles, can significantly improve maintenance efficiency and potentially reduce accidents. The automotive industry is currently defining the Dedicated Short Range Communication (DSRC) standards that will enable wireless transceivers in cars to communicate with other nearby vehicles and roadside infrastructure. At the same time, the incorporation of cellular communication systems, either through dedicated in-car transceivers or by interfacing with drivers' cell phones, is also being considered.

Due to the constraints of road, speed limits and commuting habits, vehicular sensing networks hold several notable features such as highly mobile and more predictable mobility patterns comparing to general ad hoc networks. Thus, the purpose of our work is to find specific solutions for this novel research area.

In this thesis, we tackled three major issues in vehicular sensing networks: Efficiency, Security and Privacy. The problem of efficiency is most important in a scenario such that there is no central infrastructure available or when the bandwidth of vehicle-to-infrastructure (V2I) communication is limited. For event trigger applications such as pothole detection, a vehicle must upload its sensing data to the server. However when

a large amount of sensing data are generated by every vehicle, it becomes inefficient to use V2I manner to upload every single piece of data. Thus, we propose a GeoCache concept and a Boomerang anchoring protocol to make the system more efficient. Our solutions for security are focused on establishing secret keys for both vehicle-to-vehicle (V2V) and the V2I communications. It is built up on the DSRC and 802.11p similar vehicular communication system, and the assumption that different secret keys should be used in different communication modes. We have three contributions under the privacy domain. The development of privacy preserving algorithm in vehicular sensing networks is based on the observation that some transportation applications need users to keep uploading not only sensing data but also GPS location data. It has been shown that continuous location information can be a big threat to a user's privacy. Current transportation applications usually do not take this factor into consideration. On the other hand, existing location privacy algorithms do not show enough attention to the performance of the transportation application. Therefore, our first contribution under privacy domain is to provide solutions that can achieve desired privacy level while the same time provides high quality data for applications. Our second effort is the opposite, in which we develop a new type of attack on privacy when detailed location trace is available. Our third contribution is to present a set of privacy system models under the assumption that there is no location proxy server and the information available for the privacy algorithm is limited. The following three sections provide an overview on each part: efficiency, security and privacy.

1.1 Efficiency

The Challenges: Vehicular sensing networks try to infer information of the environment from in-car sensors originally designed for other purposes. Achieving high detection accuracy can be expected to pose a challenge. We envision that these systems rely on sensing redundancy, by considering sensor readings obtained from mobiles that pass the same event location. This could be achieved through a naive centralized design, where each car communicates its readings with a server for further processing.

This design, however, will make the cellular channel a communication bottleneck, especially when the penetration rate, the percentage of participating cars, and the scale and sophistication of monitoring services increase. Further complicating the design is a fact that the cost of maintaining the server infrastructure (which can consist of networked servers) may also become significant. Fortunately, with increasing penetration rate, cars frequently enter each other's inter-vehicle communication range and the resulting communication capacity is comparatively large due to a high degree of spatial reuse. This observation suggests that distributed data processing and aggregation among vehicles through inter-vehicle communication can scale the system to higher penetration rates at low cost. The subsequential challenge is the definition of communication protocols and programming abstractions that can ease the implementation of efficient data aggregation for pervasive monitoring services.

Related Work: The presented challenge fundamentally differs from conventional static sensor networking, because the specific nodes participating in the aggregation process continuously change. While sensing systems with some node mobility have already been considered (e.g., [1]), the specific challenges of highly mobile networks with nodes moving and sensing along particular routes, frequent disconnections, and continuous changes in topology remain an open problem. This challenge is also distinctly different from other mobile ad-hoc networks or peer-2-peer networks, where random node pairs communicate (e.g., [2]). While these networks also experience frequent topology changes, in our system, communication is highly localized among nodes that have passed the same geographic location and thus it is beneficial to integrate location information into the networking protocols. Several projects specifically address on vehicular sensing [3–5]. However, none of them fully utilize the special characteristics of vehicular sensing networks and achieve similar performance as our solution does. CarTel [3], for example aims at exploring in-network computing on individual mobile units, as we do, but it does not use inter-vehicle communication, which in our project, is a main focus to enable distributed aggregation of sensor readings from multiple cars.

Proposed Solution: To address the challenge, we take inspiration from real life solutions. In reality, if an item was lost/found, we post a note around the spot where

it was lost/found. If we need information, we always go search the physical location where the event occurred. Similarly, in the truly anytime-anywhere mobile sensing, we advocate building a "directory" [6] at each event location by having one or more mobiles near the location carry information of the data collected in the location. We refer to the directory information as *GeoCache*[1] of the location, and the location as *anchor*. By having the geocache always carried by the node close to the anchor, we believe we can tie the data around the location where they are collected. Once the geocache for a location reaches a certain size, we can first consider compressing the information. Next we can utilize a "chaining" technique, so as to retain only the latest geocache entries around the anchor while leaving a link to the storage locations of older entries. Finally, we can delete the old entries or unimportant ones.

1.2 Security

The Challenges: It is expected that vehicular sensing networks can provide more effective applications to avoid accidents and traffic congestions than if each vehicle tries to solve these problems individually. Since communications in a vehicular sensing network are wireless communications naturally, the security of such a system typically relies on secret keys. In this part of the work, we target on generating secret keys for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure(V2I) under the assumption that the vehicular sensing network is supported by DSRC and 802.11p standard. Due to the lack of key management infrastructure especially in the V2V case, it becomes hard to establish secret keys. Note, even if there is a V2I communication system available as a part of the network, it is still desirable that two separate sets of secret keys are used for V2I and V2V communications. For example, a driver may query the traffic center for a section of road along his/her route to the destination through V2I communications. Without a secret key shared between the server and the individual user, this query may be overheard and disclosed to other users. Under certain circumstances, it can lead to serious privacy leakage. Therefore, secret key shared between the server and an

[1]Inspired by physical geocaches that store information and items at specific locations. Finding them with GPS receivers has become a popular pastime (http://en.wikipedia.org/wiki/Geocaching).

individual user should not be known by others. On the other hand, a driver sometimes may want to query for the local traffic condition without disclosing his/her exact current location to the remote server. This can be done through V2V communications on a distributed traffic alert system in which the traffic information are circled around locally. There are some other information the users may not want the server to know. For example, the users on multiple vehicles from the same family or company may want to keep their communications privately (watching family video/discussing creative business ideas) along the road while driving to the destination. In all the above cases, the secret keys shared between vehicles need to be kept unknown from the server.

Related Work: The process through which two parties share a secret key is called key agreement or key management. Traditional approaches include public key (asymmetric) cryptography (such as Diffie-Hellman key establishment [7]) and trusted third parties (TTP) [8] (such as the well-known Kerberos [9] scheme and Otway-Rees protocol [10]). As we discussed in the previous paragraph, as a part of the vehicular sensing network, the V2V communication has no central authority that can be relied on. Furthermore, since the node mobility is unrestricted, the topology may be unpredictable and makes the central authority assumption infeasible. Cost-effective processors with limited computational abilities make public-key cryptography almost impractical for embedded intelligence and ubiquitous computing applications, even without power consumption considerations. Prior work on key agreement in sensor networks and ad hoc networks has largely focused on pre-distribution protocols (e.g., [11–13]). In such protocols a large pool of symmetric keys is chosen and a random subset of the pool is distributed to each node. Thus, two nodes can establish a session key if they share a common key. However, this is hard to achieve if the nodes are mobile.

Proposed Solution: We propose untraditional methods to create secret keys. The core of the proposed schemes can be summarized by two words: **Reciprocity** and **Diversity**. Reciprocity represents the channel reciprocity theorem. Diversity includes space diversity, frequency diversity and time diversity. Channel reciprocity theorem and spatial decorrelation - a concept derived from space diversity - together can be used to generate shared secret keys between a pair of vehicles in an urban scenario or

a multi-path fading environment. In our early work [14], a random channel hopping scheme was originally designed for general wireless networks. By extending the idea, we develop a new probabilistic based key agreement for V2I communications. This extension on time and space axis, further improves the robustness of the secret key.

1.3 Privacy

The Challenges: Vehicular sensing networks provide great opportunities for traffic engineering. Based on the location information such as GPS data alone, many traditionally expensive and/or complicated traffic engineering work can be done with low cost. One example is queue length estimation at traffic light intersections [15]. However, collecting location traces raises many privacy concerns as discussed in the literatures [16, 17]. By tracking a vehicle, it becomes possible to identify the visited locations, thereby, breaching the privacy of the driver or the passengers. Furthermore, the location tracking information about a user can be misused by an adversary.

Related Work: At first glance, the privacy issue can be addressed through established anonymization techniques or the techniques of changing pseudonyms since such particular applications do not depend on vehicle identities. Some proposed privacy preserving methods address this problem via naive anonymization techniques which simply remove vehicle identifiers [18]. However as shown in [19, 20], naively anonymized (i.e., simply omitting names, vehicle identifiers, etc.) location traces can often easily be re-identified. Other methods attempt to protect privacy by perturbing or reducing the accuracy for either spatial or temporal information [21–23]. In these cases, however, transportation modelers usually hesitate to use such datasets for traffic applications, especially for those requiring high accuracy spatial and temporal information (e.g., for arterial performance measurements). A vehicular mix-zone [24, 25] defines a particular area (usually around intersections) where locations cannot be revealed so that an adversary cannot trace the movements of vehicles across these mix-zones. However, our target application introduces strict requirements on these locations: data is only of interest around intersections.

Proposed Solution: In the work under privacy domain, we mainly focus on providing a system architecture and algorithm to protect user privacy by reducing the link-ability between a user's discontinuous traces. While the system is studied under the assumption of the common location information available today and the availability of a location proxy server, we also discover that with further detailed location traces (when different driving characteristics can be extracted from the traces), a new privacy leakage problem is raised. In addition to that, we study the privacy protection problem when the available resource is limited[2]. Thus, our major contributions in privacy include three parts: 1) Propose a privacy model for vehicular sensing networks under general system assumptions; 2) Discover privacy leakage due to detailed location information available; 3) Study privacy protection and present privacy models under the assumption of limited resource. As discussed in more detail below.

In the first part, we develop a zone-aware privacy algorithm to filter location traces, which takes traffic density and uncertainty into account. This allows the algorithm to release location traces only in the intersection zones where data is needed by the application, yet still offer a fixed degree of privacy independent of traffic density. It is, to our knowledge, the first algorithm that combines these two aspects. More specifically, the algorithm seeks to achieve unlink-ability between released traces from any two zones. The zones are referred as VTL zones, since their locations can be marked by two or more Virtual Trip Lines (VTL) [26]. To be able to adjust the filtering algorithm based on the traffic density, the algorithm needs to be aware of all vehicles' location traces. Although we refer to a proxy server with access to these traces for the sake of simplicity, there are multiple usage scenarios for such an algorithm. Even if no proxy server exists, such an algorithm is still profitable to anonymized data before they are stored or transmitted to another party.

Nowadays, typical vehicle GPS receivers can provide location samples at an updated

[2]This is in regarding to both the privacy protection system and the adversary. Since no location proxy server is available to assist the system, the resource is limited. On the other hand, an adversary cannot directly obtain a large amount of location data by compromising the data channel between the proxy server and the application server. Without a location proxy server, the privacy algorithm has to be done locally in a distributed manner, which also means what the adversary can capture will be further constrained location information.

rate of 1 Hz and with an error range less than three meters. Given such precise location information, it is possible for an adversary to discover more details. In the second part of the research under privacy domain, we use machine learning models to extract critical movement features from location dataset and build up attacking models based on these features. We study different mobility characteristics such as acceleration and their combinations to see if they could allow an adversary to separate trip segments belonging to a truck from others. This would make linking trivial in a hypothetical case with only one truck in a dataset of cars. In more balanced datasets, it would still provide additional information that makes linking easier and reduce the system entropy in terms of privacy preservation. We also check whether a dataset with similar vehicles can still provide clues for identifying outliers through driving behavior. By studying the NGSIM location traces, we illustrate that the existing privacy models measuring the degree of privacy in anonymized location traces will no longer hold. Using data captured from vehicles, the fine-grained location traces reveal speed distribution and acceleration patterns that can be used to distinguish traces from different vehicle types (e.g., trucks and cars). Therefore, it is possible to identify outlier driving patterns such as higher speed, which could be used to link anonymous segments of location traces and eventually recover complete trips.

In the third part, we focus on designing the privacy preserving models for vehicular sensing networks under the assumption of limited resource. Firstly, the relationship between the prediction error rate and the distance to uncertainty ratio is studied through Monte Carlo simulation. The results can be used to help estimate real trace prediction error rate when the density to uncertainty ratio is known and it can help making quick decision on whether a privacy model is worth to be applied to the network system. Secondly, three privacy models are presented for vehicular networks without involving location proxy servers. Comparisons among the models are made through simulation and NGSIM dataset. Our results show that the distance based model is preferred when both application and privacy performance are considered equally important. The three models can also be used as references to develop other privacy models when more detailed information is available.

1.4 Thesis Structures

The entire thesis is organized as follows: In chapter 2, we present the research work in the field of efficiency, which is called the GeoCache and Boomerang algorithms. Chapter 3 describes the secret key agreement algorithms for vehicular sensing networks. The privacy part of work crosses three chapters. Chapter 4 proposes the VTL zone-based path cloaking algorithm. Chapter 5 presents linking anonymous location traces through driving characteristics. In chapter 6, we propose three privacy models under the assumption that there is no location proxy server and the information available is limited. Last chapter, chapter 7 concludes this thesis.

Chapter 2

Efficiency: The GeoCache and Boomerang Algorithms

2.1 Introduction

A direct consequence of fast growing up of vehicular sensing networks is the production of a vast amount of data, in terms of both type and volume. Example data types include pictures, videos, sound files, and plain text-based sensor readings. These data can potentially bring great convenience to the society as they can serve as traces of our lives and logs of the physical world.

Emerging mobile sensing [2–4] applications such as automotive traffic congestion monitoring (e.g., [26]) or pothole detection [27] require aggregation of sensor data from multiple mobile nodes that have passed the same geographic location to achieve accurate sensing performance. Current designs require mobile nodes to transmit all relevant sensor information through infrastructure networks (cellular, open Wi-Fi, etc.) to a central processing unit which aggregates the data. The volume of data generated is quite large, and the volume can continue to increase as new applications emerge that require continuous monitoring. Thus, this centralized approach suffers from the network capacity/cost concerns, in addition to the availability and privacy issues.

Particularly along rural highways cellular data service and open Wi-Fi access may not be available and cellular networks may be too costly to transmit high-volume sensor information such as video data before it is aggregated. In addition, frequent transmission of sensor data from mobile nodes, tagged with location information may raise privacy concerns.

To improve the efficiency of vehicular sensing networks, we propose a GeoCache

concept and a Boomerang protocols for its implementation. Rather than storing information in a particular node, geocache "stores" information at a particular geographic location. GeoCache are implemented by letting the car that is passing a geographic location carry the information about that location. As the vehicle moves further away, it transfers the information to a following vehicle closer to the reference location. Depending on the delay constraint of the application and node density this transfer can occur at different frequencies.

We study protocols that can retain geocache around the anchor location. Specifically, we consider two challenges in doing so: (i) returning the geocache with high probability to the anchor location if a node carrying the geocache becomes temporarily disconnected; (ii) minimizing the communication overhead for retaining the geocache near an anchor location. Prior work does not provide efficient protocols to address these challenges.

The boomerang protocol addresses these challenges by providing a trajectory-based return protocol that increases the probability that the geocache can be successfully returned to the anchor location if it has been carried away due to temporary disconnection. While this geocache return protocol is inspired by delay-tolerant geographic routing, it is unique in recording a node's trajectory as the node is moving away from the anchor location and using this trajectory as guidance for selecting nodes to carry the geocache back. Also, instead of each relay node sending geocache over the wireless link as soon as it receives the geocache, we let the relay node keep the geocache until it drives off the original trajectory. Thus, it exploits a characteristic of vehicular networks, namely that vehicles move on well-defined and usually bidirectional pathways. The increased return probability then also allows reducing communication overhead by purposefully allowing a node to briefly carry the information away from the anchor location and returning it even in connected networks.

In summary, the salient contributions of the work are:

- Outlining the geocache concept, making sensed data available at the anchor location, to support mobile sensing applications over a distributed network of mobile nodes.

- Designing a boomerang protocol which can periodically return geocache data to an anchor location with reduced communication overhead.
- Developing algorithms that can accurately detect whether a node is diverging from a recorded trajectory considering the complexity of the real-world road topology.
- Evaluation through a proof-of-concept implementation and showing through simulations using a portion of the south New Jersey highway network that the collection and use of return trajectory information in the boomerang protocol increases the probability of timely return to the anchor location by an average of 70% compared to the shortest-distance geographic routing.

The rest of this chapter is organized as follows. Section 2.2 discusses the platform assumption and network architecture. A formal definition of geocache is given in section 2.3. Section 2.4 describes different geocache anchoring protocols and the way to construct trajectory in detail. Section 2.5 presents our proof-of-concept experiment and the techniques to detect divergence from recorded trajectories. Section 2.6 compares the performance of different geocache protocols. Related work are presented in section 2.7, and section 2.8 provides the conclusion for this chapter.

2.2 Assumptions and Requirements

In this section, we first discuss the state-of-the-art in-car sensing and communicating technology. Following this, we discuss the envisioned road monitoring and alert applications that can be built upon the technology. In the end, we present the network architecture that can best support these applications.

2.2.1 Platform Assumptions

We envision that future automotive vehicles include sensing, communication and computing resources to enable sophisticated distributed sensing applications. As far as communication is concerned, in addition to the cellular access option, the automotive industry is also defining the IEEE 802.11p standard for Wireless Access in the Vehicular

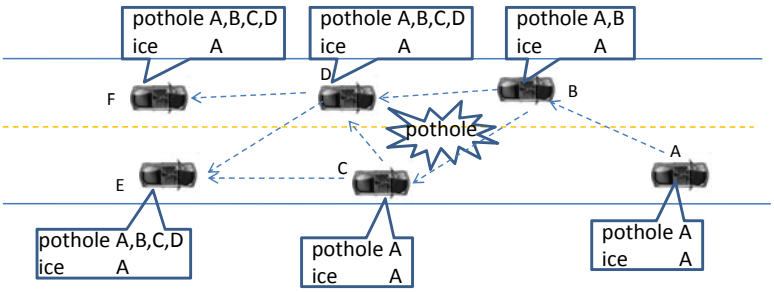


Figure 2.1: A location-centric peer-to-peer pothole detection scenario.

Table 2.1: Five interesting sample OBD-II reading

Mode(hex)	PID(hex)	Data(bytes)	Description	Min/Max	Units
01	05	1	Egine coolant temperature	-40/215	%
01	0A	1	Fuel pressure	0/765	kPa
01	0C	2	Engine RPM	0/16.383	rpm
01	0D	1	Vehicle speed	0/255	km/h
01	1F	2	Run time since engine start	0/65,535	sec

Environment (WAVE) to enable Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications in the 5.85-5.925 GHz band. For non-emergency applications, FCC regulations permit transmission power levels of up to 2W Equivalent Isotropically Radiated Power, which would provide a freespace communication range over 1 Kilometers and about 200 meters (at 10% PER) in a Rayleigh fading channel. The basic CSMA/CA protocol remains virtually unchanged from 802.11a/b/g except that association and authentication procedures have been replaced.

At the same time, most vehicles also already contain a rich set of sensors that provide information about engine and emission performance as well as the environment. In the United States, the On-Board-Diagnostics-II (OBD-II) specification has been mandatory since 1996; a similar EOBD specification is mandatory since 2001 in the European Union. These specifications provide a standardized interface for access to most engine performance and emission related in-car sensors through a 16-pin J1962 connector. A sample set of accessible sensor readings are shown in Table 2.1. The basic sensor readings shown in Table 2.1 reflect the internal settings of individual cars. Additionally, modern cars also include more advanced devices which can profile the surroundings. Examples of such sensors include front- and rear-view cameras, rain sensors, wheel traction sensors, accelerometers, gyroscopes and radar/lidar systems.

The existence of sensor devices with varying capabilities can help infer the presence of high-level events, which may be of relevance to a large number of drivers on the road. For instance, a car can detect a pothole on the highway by a combination of observed accelerometer registered shock, sudden braking, and camera image.

2.2.2 Location-Centric Peer-to-Peer Computing

Networking cars that are equipped with on-board computing, sensing and communication capabilities can enable a family of distributed sensing applications, which focus on monitoring the road/traffic health by detecting abnormal events on the road. Robust event detection often requires the aggregation and mining of readings from multiple vehicles that passed the event location. For example, inferring road hazards, such as objects on the road or slippery road conditions from sudden braking on one car can lead to false alarms as such sudden braking also frequently occurs due to driving mistakes. Therefore, achieving robust detection will require the detection of clusters of braking readings around the same time and location from multiple cars.

In this study, we seek to investigate the design issues involved in developing such a **RO**ad **M**onitoring and al**E**rt system (ROME). Fig. 2.1 illustrates a ROME scenario in which several cars collectively detect the presence of a pothole on the highway. In this simple example, cars will individually detect the pothole based on its local readings, but the base station will only confirm the presence of the pothole after gathering enough information from the passing cars. The gathering process can be as simple as counting the number of cars that detected vibrations, or as complicated as performing image recognition over several images. Information gathering can be implemented in the following ways:

- **Centralized:** This design is based on immediately communicating all sensor readings through a cellular link to a central server, which can store and analyze the data.

- **Query-Response:** The principle of this design is to store all sensor readings locally in each vehicle, until data is requested by an outside entity. Queries must

also use the cellular network, since the dispersion of cars storing similar sensor readings related to the same event increases with time and vehicular networks using short range communications with lower penetration rate remain frequently disconnected.

- **Location-centric P2P Processing:** This design involves each car that detects the event exchanging the information with its neighbors through short-ranged radio, and processing the information at the same time. Once the aggregated decision is reached, one of the cars will raise the alarm to the external server. Here, the computation is not tied to any node, but to the event location, and is carried out by the cars that pass by the location.

Among the three approaches, we advocate the location-centric P2P approach. First, compared to the other two alternatives, it provides better scalability. The centralized approach requires every car to contact the external server, which will soon saturate the cellular link given the number of events on a highway and cars that pass by. The query-response approach completely ignores the importance of location. After the cars that detected the event left, it is very difficult (or, costly) to gather their responds to the queries afterward. In addition to scalability, the location-centric P2P approach can help protect privacy because individual vehicle's sensor readings are not collected by a centralized entity. The example in Fig. 2.1 employs a distributed location-centric P2P architecture.

2.3 GeoCache: Concept and Abstractions

Before we present our location-centric peer-to-peer method of implementing ROME, we first present the concept of geocache. Environmental monitoring applications such as road monitoring can be expected to exhibit strong spatial locality in data access patterns, meaning that sensor data from a particular position is frequently accessed by other vehicles passing this position. Thus, geocache is a programming abstraction, which stores information in virtual geographic space rather than on a specific node (illustrated in Fig. 2.2). At a high level, the geocache subsystem bears similarity to

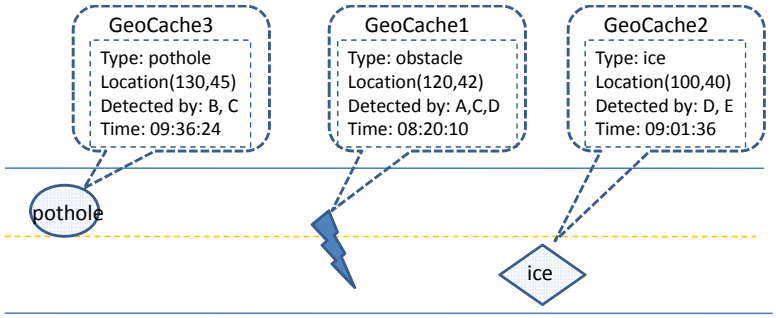


Figure 2.2: Three geocache examples.

a distributed geocache database, which records observations of different spots on the highway, and can be accessed by applications through a continuous query interface. This database is unique, however, in that the actual node that stores a geocache frequently changes as nodes passing by, to keep the information near the specified geocache location.

To efficiently operate in a highly mobile network with frequent disconnections, (i) GeoCache use opportunistic proactive dissemination methods rather than reactive query protocols; (ii) completeness of results is not guaranteed; and (iii) to reduce protocol overhead the system may transmit several geocache in a single packet, thus all geocache of the same type should share the same propagation parameters defined below. A geocache can be formally defined by the following attributes:

- GeoCache ID,
- Latitude, and longitude which describe the geographic anchor position of the geocache,
- Time which stands for the creation time of the geocache,
- Time-to-Live which denotes the duration until the geocache expires,
- Delay which denotes the geocache collection interval,
- Data which contains an application-defined data structure.

Time-to-live and delay constitute the geocache's propagation parameters. We note that a geocache should have a limited lifetime to ensure limited dissemination of the

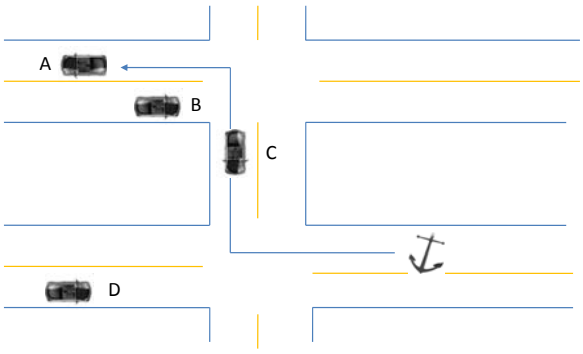


Figure 2.3: A single relay node is insufficient to bring back the data in some cases. In this example, after A hands off the data, we need B, C, D to return the data to its anchor location.

geocache. For instance, an event that was detected a long time ago should not be broadcasted among nodes. In order to ensure this temporal locality, a node invalidates a geocache if its time-to-live expired.

2.4 GeoCache Anchoring Protocols

The goal of the geocache anchoring protocol is to retain geocache data around the corresponding anchor location while minimizing communication overhead.

Intuitively, we envision the following anchoring process: the mobile node that currently carries the data (referred to as the carrier) moves away from the anchor location, either due to disconnected network or to reduce communication overhead. When possible, it will hand off the data to other nodes (referred to as the relay nodes), preferably those traveling in the opposite direction towards the anchor location. After receiving the data, a relay node will periodically examine whether another handoff is needed. This process repeats until data returns to the anchor location, and we call this protocol "boomerang" because the data returns to its origin like a boomerang.

To motivate how this boomerang approach can reduce communication overhead, let us consider a brief gedanken experiment. One could retain information at the anchor location simply by transferring the geocache to a following car whenever the boundary of the radio range approaches the anchor location. In an idealized model with constant radio range r, vehicle velocity v and high vehicle density, retaining the geocache for

a duration t would require $m = \frac{tv}{r}$ messages. The boomerang approach, however, reduces the number of transfers to $m = 2$ (one transfer to the opposite direction to return geocache and one transfer back to the anchor location) under ideal conditions. Thus, the boomerang approach has the potential to significantly reduce geocache communication overhead when the geocache is only needed periodically. Under more realistic assumptions, the number of messages may be larger because vehicles can choose different paths and may not return to the anchor location. The vehicle may also need to broadcast requests periodically to identify other carriers with geocache from the same anchor location that can be aggregated.

2.4.1 Protocol Description

The main challenge when implementing a boomerang protocol lies in the choice of the relay node at each handoff. The data may have traveled along a rather complicated route before the carrier looks for a relay, as illustrated in Fig. 2.3. A set of poorly selected relay nodes may incur a long delay in bringing back the data (the data may lose its value significantly after a long delay), or even lose the data. The task of choosing appropriate relay nodes is particularly daunting because at each handoff, neither the current carrier nor the nodes within the handoff range has knowledge beyond each node's current velocity vector. In this chapter, we propose a trajectory-based selection approach and compare it to a baseline shortest-distance-based selection scheme.

Shortest-Distance-Based Selection: Heuristics that fall in this category choose the node closest to the anchor location among those that are in the handoff range and moving towards the anchor location. They share the same rationale as many geo-routing algorithms such as the ones proposed in [28]. The heuristic we look at in this category is referred to as MaxProgress (Maximum Progress first). A simple example is given in Fig. 2.4.

Next let us look at the detailed handoff procedure. After traveling away from the anchor location for a certain amount of time, the carrier initiates a handoff by broadcasting the data along with the anchor location. A node within the handoff range becomes a candidate if its distance to the anchor location is decreasing. The candidate

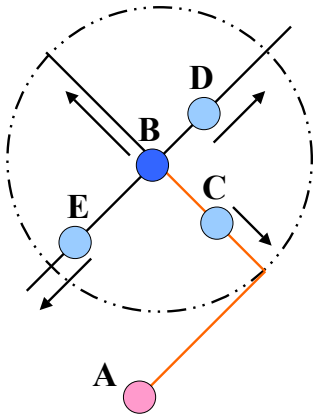


Figure 2.4: An example of handoff situation. In this case, B is the initial carrier. MaxProgress will choose node E as the relay node(because its distance with A is decreasing and it is currently the closest one to A among all the nodes), while RevTraj will choose C as the relay node.

calculates the ACK backoff time T as:

$$T = \frac{T_{max} * (d - d_0 + r)}{2r}, \tag{2.1}$$

where T_{max} is the maximum backoff time for all nodes; d is the distance between this receiver and the anchor location; d_0 is the distance between the carrier and the anchor location; r is the radio radius. Using this equation, we can distribute the ACK backoff times between 0 and T_{max}, and more importantly, the node with the shortest distance to the anchor location will have the smallest amount of backoff time. As a result, this node will be chosen as the next carrier.

The new relay node will keep moving until its distance to the anchor location starts increasing. At that time, it initiates another handoff procedure.

Trajectory-Based Selection: While the distance-based approach works well for geographic routing in ad-hoc networks, it may not be suitable for vehicular networks because it ignores the fact that vehicles only move along fixed roadways. Therefore, progress in Euclidean distance does not always yield a feasible path that returns to the anchor location. For instance, node E in Fig. 2.4 is on a path that never reaches A.

The above concerns lead us to the trajectory-based selection approaches. These approaches select new carriers from the nodes that are traveling in the opposite direction to the initial carrier's trajectory. The rationale is that the trajectory describes a general

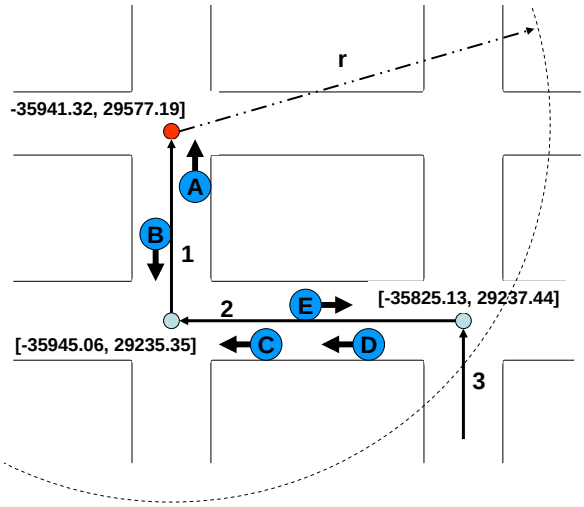


Figure 2.5: The trajectory-based handoff procedure.

feasible return path (with the exception of one-way path scenarios). The heuristic we consider in this study is thus called *RevTraj* (**Rev**erse **Traj**ectory). With this scheme, in Fig. 2.4, node C which is in the opposite direction of B's trajectory will be chosen as the next carrier.

The key component of RevTraj is a "trajectory history" which stores the trajectory: the path that the data has traveled so far. The trajectory history grows when a carrier is moving away from the anchor location, and shrinks when a relay node is moving towards the anchor location. Next, let us look at the detailed handoff process in a trajectory-based approach. In the discussion below, we assume trajectories are recorded as segments instead of continuous traces. As illustrated in Fig. 2.5, every turn leads to a new segment which can be represented by the coordinates of the two end points, e.g., sesgment 1 = [(-35945.06, 29235.35), (-35941.32, 29577.19)]. The trajectory can be stored through a stack structure. Below is the summary of the handoff procedure used in RevTraj:

1. *Handoff Initiation.* The current carrier broadcasts the data along with the trajectory history.

2. *Candidate Identification.* After receiving the trajectory history, every node in the handoff range pops the latest trajectory segments from the stack. We use

a parameter, *lookahead distance* LAD, to control how many segments will be examined. These lookahead segments can be numbered as $1, 2, ...LAD$, where segment 1 is the most recently traveled segment. In Fig. 2.5, LAD is 3. If the node finds itself on one of the lookahead segments (the details will be explained in Section 2.4.2), it becomes a candidate node and proceeds as below; otherwise it drops the data.

3. *Candidate Prioritization.* All the candidates are prioritized according to the three rules: (1) nodes traveling in the opposite direction of the trajectory get higher priority than those traveling in the same direction as the trajectory; (2) within each direction, nodes traveling on higher segment numbers get higher priority than those on lower segment numbers; and (3) for nodes traveling on the same segment, we give higher priority to those who are closer to the anchor location. The prioritization rules can be implemented if each candidate node calculates its ACK backoff time using the following equation:

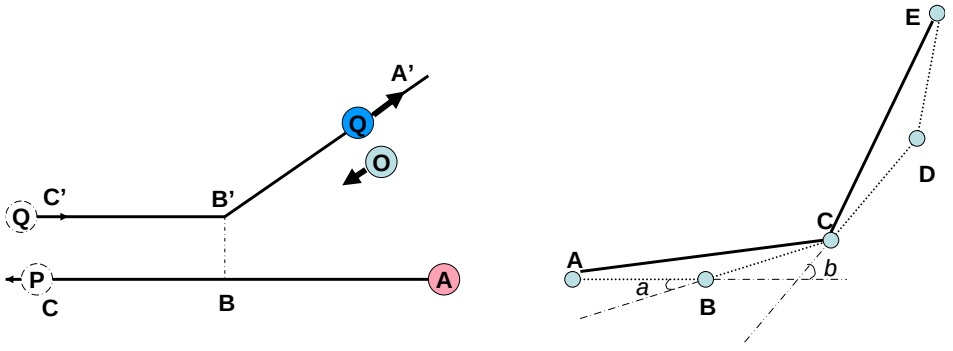
$$T = (\frac{LAD - s + (d - d_0 + r)/2r}{LAD} + \alpha)\frac{T_{max}}{2}, \tag{2.2}$$

where α is 0 if the candidate node is traveling in the opposite direction, 1 if in the same direction; s is the matched segment number; d is the distance between the candidate node and the anchor location; d_0 is the distance between the current carrier and the anchor location.

4. *Relay Selection.* The node with the smallest backoff time will send an ACK quickly than all the other candidates. To avoid the hidden and exposed terminal problem, we suggest the ACK be sent using a higher transmission power so that the rest of the candidate nodes can overhear and drop their ACKs.

5. *No Acknowledgement.* If the current carrier does not receive any ACK, it continues carrying the data and initiates another handoff later.

We further distinguish between prioritized trajectory-based anchoring as described above, and a non-prioritized one, where every candidate node can choose a random backoff time between 0 and T_{max}, and the winner becomes the next carrier.



(a) Segmented Path. **(b) Curve Segmentation.**

Figure 2.6: An illustration of segmented path and curve segmentation.

2.4.2 Constructing Trajectories from GPS Data

There are many challenges in the implementation of the trajectory-based boomerang protocol. For examples, how to efficiently record the path, and how a returning node detects its divergence from the path[1]. To illustrate some of them, consider the example in Fig. 2.6, where node P carries the geocache away from the anchor location A before handing off it to node Q. P records and compresses the GPS trace for the road stretch AC, and then sends it to Q. Since Q moving on the opposite lane, it keeps shrinking the path history. After traveling until B', it diverges from P's path and heads to A'. In this case, it needs to detect divergence and start extending the path history from B', so that it can hand off the data with the updated path $A \rightarrow B(B') \rightarrow A'$ to node O. The challenge in this implementation lies in developing path divergence detection algorithm that is robust to positioning inaccuracy and usable across a variety of road network scenarios.

Path Recording and Preprocessing: The boomerang system constructs a path from periodic (latitude, longitude) samples reported by a Global Navigation Satellite System (GNSS) receiver every second. First it uses the average value of consecutive samples to reduce redundancy and to smooth paths as illustrated in Fig. 2.7.

Next, it segments the path and retains only critical points by monitoring the angle

[1]Recall that the trajectory-based boomerang protocol uses a stored path to route data back to the anchor location. The returning node keeps the data while it travels on the path (in the opposite direction) and hands off the data when it diverges from the path.

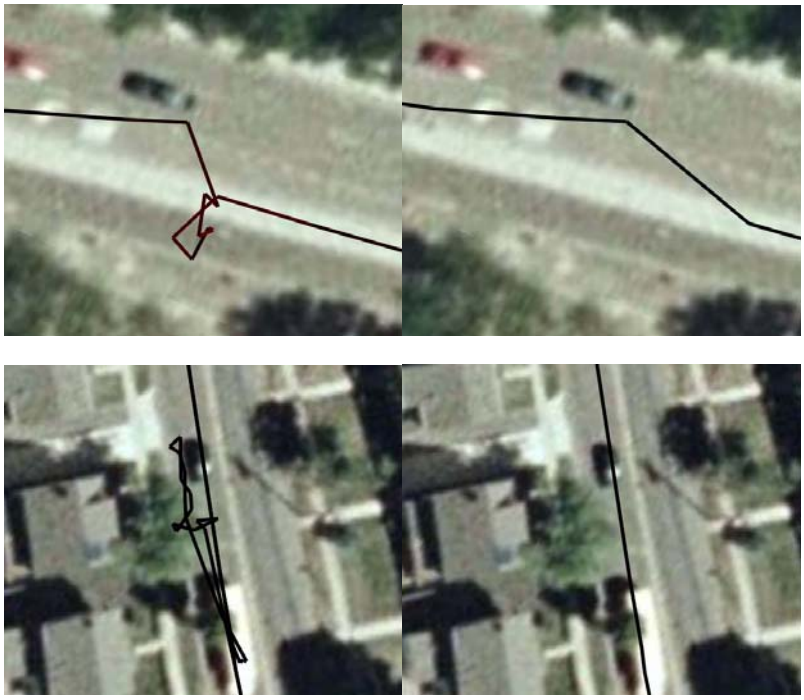


Figure 2.7: Trace before (the left two pictures) and after (the right two pictures) path smoothing pre-processing.

between the heading at the start of a segment and the current direction. If this angle exceeds some threshold, the algorithm adds a point to the stored trajectory and starts a new segment. Consider the set of consecutive GPS samples $A - E$ illustrated in Fig. 2.6(b). Assume a new segment starts at A, and the initial heading follows AB. When C is recorded, the algorithm checks angle a. Since angle a is under the threshold, C is still on the same segment. Next when D is recorded, angle b is checked. Since b is above the threshold, D is considered as on a different segment CD.

In practice, we use a maximum segment length to prevent the accumulated distance error when the change of angle is slow.

Path Manipulation and Divergence Detection: When returning the data, a node continuously shrinks the path history by replacing the path end point with its current position and removing points that have been passed. It must also check if it has diverged from the original path to determine whether a handoff is needed.

Divergence detection is based on the combination of distance and angle thresholds, since neither of them alone can provide robust detection algorithms. For examples, lane

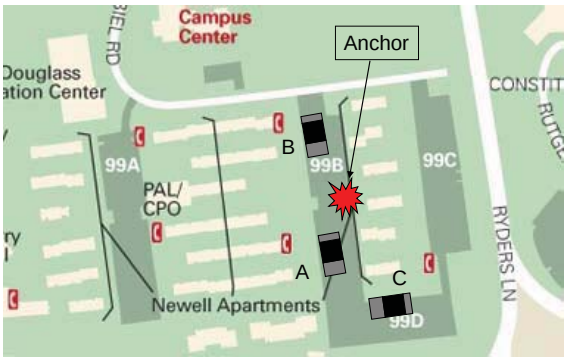


Figure 2.8: The experimental scenario.

changes or individual's driving behavior may lead to sudden direction change without diverging and the variety of road widths (e.g., 15–60 ft for city roads[2]) makes the selection of a single distance threshold difficult.

The algorithm monitors the following conditions when a new point is added: (1) the distance between its current location and the original path has exceeded the maximum road width d_{max}, or (2) the distance has exceeded the distance threshold d_0, and the angle change has exceeded the angle threshold h_0. Divergence is declared if n consecutive samples meet below conditions:

$$divergence = \begin{cases} 1 & d > d_{max} \text{ or} \\ & d > d_0 \text{ and } h > h_0 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Here, d_{max} is the maximum road width, which can be obtained from road design manuals. d_0 and h_0 are the thresholds for distance and heading difference.

2.5 Implementation and Lessons Learned

We implemented the trajectory-based Boomerang protocol on a Linux system that is equipped with Atheros AR5212 Mini PCI 802.11a/b/g wireless cards and conducted an experiment of three vehicles in a parking lot at Rutgers University, Cook Campus (see Fig. 2.8).

[2]http://www.greensboro-nc.gov/visitors/

2.5.1 Proof of Concept

The system includes three parts: data collection, data storage and data communication. Data are collected from in-car sensors and GPS device through PERL scripts. In-car sensor information is obtained via the On-Board Diagnostic's system (OBDII) through an ElmScan 5 USB device. GPS data are collected with a Pharos USB GPS device and a Panasonic network camera WV-NM100 provides further road information that can be stored in the geocache. The data storage module uses a local SQL database (MySQL) to maintain an event table and a task table. The event table is used to store all geocache. The task table maintains state information such as the scheduled time for handoff and the number of handoff attempts. Records across the two tables are connected through the location, time, data type and host ID information.

In the experiement, car A and B, both generated geocache when passing the anchor location. Then they kept driving with the data. Car C was driving in the opposite direction towards the anchor location. After some time, A started to hand off its data. Since C was moving towards the anchor, it caught A's data and responded with an ACK. Similarly, C also caught B's data. C then merged two geocache and carried them to the anchor location.

The proof-of-concept experiment revealed that most modules of the system, including trajectory recording, geocache handoff, and data aggregation worked as expected. However, it also revealed the challenge in the geocache return phase. Specifically, a return node needs to keep checking whether it is on the trajectory or not. If the return node is diverging from the original trajectory, it needs to schedule another handoff. Due to the inaccuracy of the GPS traces and the complexity of the roads topology, the divergence detection result can be inaccurate. In order to improve the algorithm, we had the following additional experiments.

2.5.2 Divergence Detection on Real-world Traces

Given the divergence detection algorithm (shown in Eq. 2.3), the key is to setup proper thresholds: d_0 and h_0. We use GPS samples taken from a driving experiment to find

Figure 2.9: The route traveled in the experiment are colored in red.

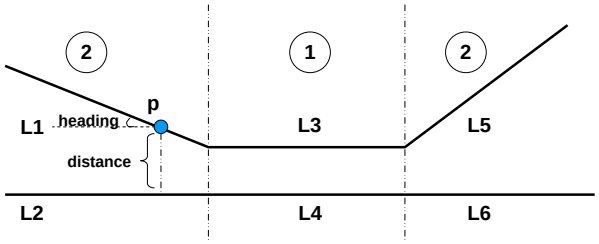


Figure 2.10: Dividing pairs of traces into two groups.

the best thresholds.

The sample set includes two hours of GPS data from New Brunswick, NJ. Two trips of traces were collected on both highways and local streets (see Fig. 2.9). Each of them covers 55 miles of road with an average speed of 35 mph. In the first trip, we stayed on the main loop and in the second trip we frequently turned into side streets.

The two sets of traces are then overlaid and manually divided into segments so that the paths either diverge or remain parallel in each segment as shown in Fig. 2.10. The samples in each segment are manually labeled as diverging or parallel and used as ground truth to compare with the results got from detection algorithms.

The location traces fall into two groups. Group one contains pairs of traces that are not divergent, and group two contains the remaining ones. As shown in Fig. 2.10, $(L3, L4)$ is in group one, while $(L1, L2)$ and $(L5, L6)$ are in group two. For each point on the segment for example p on $L1$, we record four values: d_p, h_p, x_1, x_2, where d_p is the distance between p and $L2$, h_p is the difference in angle, x_1 is the ground truth

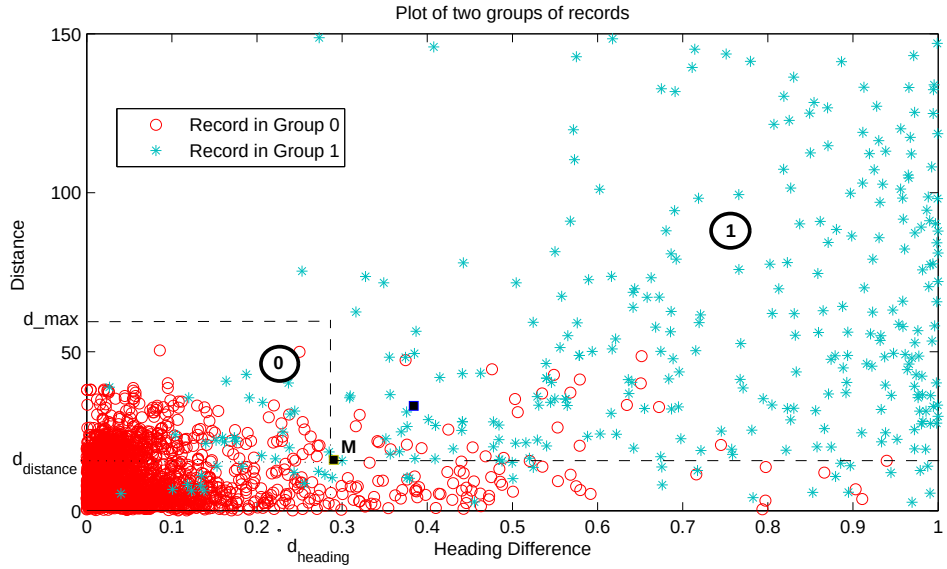


Figure 2.11: Using classification algorithm to build boundary between two groups of traces

of the divergence (1 for points in group two and 0 for points in group one), and x_2 is the result got from the detection algorithm. All the samples are ploted in Fig. 2.11 based on their d_p and h_p values. Different colors are used to indicate the ground truth. Classic classification algorithm then is used to find the best threshold values: d_M, h_M.

Based on the GPS traces collected from our driving experiment, when $d_0 = 16$ meters and $h_0 = 0.29$, the algorithm achieves a false positive rate of 0.013 and a false negative rate of 0.187 with an average delay of 2.26 samples, i.e. the divergence will be correctly detected at the 3rd sample.

2.6 Performance Evaluation

In this section, we study the performance of the geocache anchoring protocols through simulation. The return probabilities of the two protocols are compared.

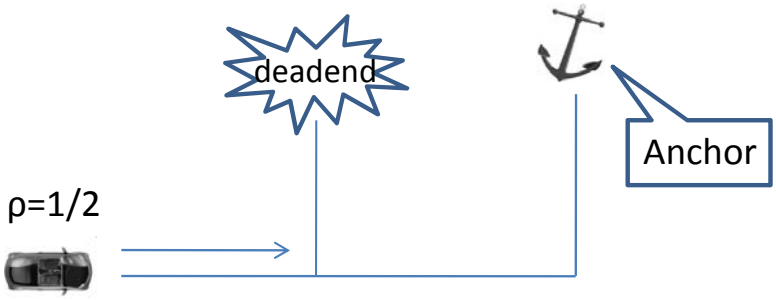


Figure 2.12: The impact on ρ value from a dead end.

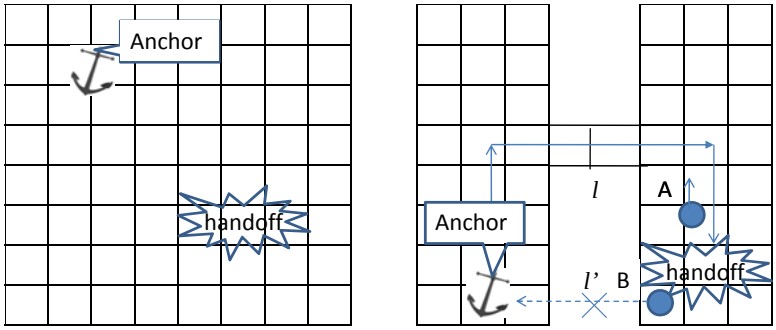


Figure 2.13: Two road topologies with different ρ values. (a). A fully-connected grid with $\rho = 1$ representing the Manhattan city road map.(b). A partially-connected twin-city grid with low ρ value, representing two cities being connected by a major highway.

2.6.1 Effect of The Road Connectivity

Intuitively, the MaxProgress method works better under fully or mostly connected road topologies. We capture the connectivity characteristic using parameter ρ:

$$\rho = \frac{N_{success}}{N_{total}}. \tag{2.4}$$

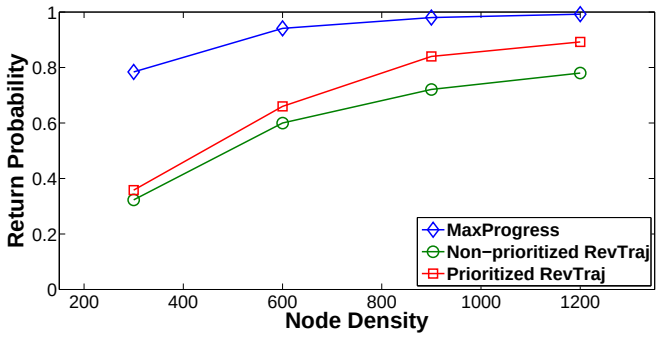
where $N_{success}$ is the number of successful paths in the partially-connected grid, and N_{total} is the total number of paths that head to (but not necessarily reach) the anchor location's direction. For a fully connected grid, we have $\rho = 1$ in Eq. 2.4. Our definition of ρ is different from the general concept of road connectivity. For both the numerator and the denominator, we only consider shortest-distance path, i.e. we do not backtrack once the road hits a dead end. According to this definition, road topology such as a dead end will result in a relatively low value of ρ as shown in Fig. 2.12.

A well-connected city road topology such as Manhattan's has a large ρ value, where all road segments have outlets at both ends. Many of others, however, do not, because

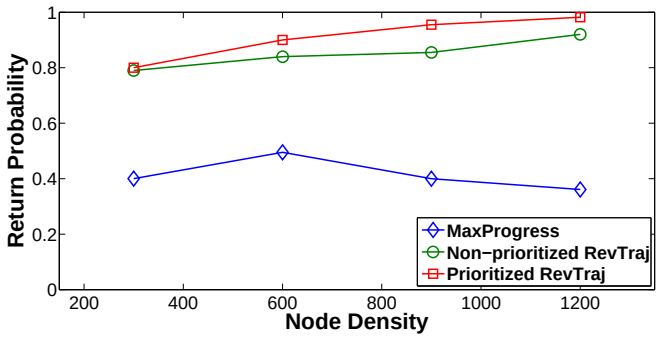
dead ends are common in either urban or rural road systems. According to [29], in the Digital Road Map Data Base (DRMap) for Japan, among all the 354,000 investigated roads, there are 22,000 dead ends in total. In partially-connected areas, we expect MaxProgress to exhibit sub-optimal performance. An example is given in Fig. 2.13(b). The geocache is handed off after traveling along the road l. When choosing the next carrier, MaxProgress always favors those that are physically closer to the anchor location, which is B in above case. But the nature of the road topology makes this choice a bad one, because the seemingly shortest path l' does not exist, whereas the longer alternative l is the only feasible path leading back to the anchor location. On the other hand, using RevTraj, node A who is moving towards the road l will be chosen as the next carrier. Following the trajectory, no matter how poorly connected the road topology is, we are always confident that the trajectory can lead to the anchor location. A trade-off is the probability of the RevTraj protocol finding a carrier is lower than that of the MaxProgress protocol especially when node density is low.

To verify above hypothesis, the performance of the two protocols are compared in different topologies depicted in Fig. 2.13 (a) and (b). Fig. 2.13 (a) represents a well-connected city road topology with $\rho = 1$, while Fig. 2.13 (b) shows a topology with a low ρ value, representing two cities being connected by a major highway. The simulation is done through NS-2 with 802.11 MAC and PHY layer protocols. The radio range is 250 meters. At each intersection, the probability for a car to turn left, right or go straight are all $\frac{1}{3}$. The length of each road segment is 300 meters. Vehicles are moving at the speed around 30 meters per second. For RevTraj, the number of look ahead segments is 3, meaning we look at the most recent 3 segments when comparing the trajectory.

Fig. 2.14 shows the return probability of the two protocols in which the node density is represented as the total number of vehicles existing in the network. As expected, when $\rho = 1$, MaxProgress outperforms RevTraj. On the other hand, when the value of ρ is small, the return probability of MaxProgress downgrades severely compared to the RevTraj schemes. The results are consistent with our previous analysis. Between the two RevTraj schemes, the prioritized and the non-prioritized schemes, the former performs better than the latter. Finally, we also can see that the performance of RevTraj



(a) GeoCache return probability in fully-connected grid.



(b) GeoCache return probability in twin-city grid.

Figure 2.14: Comparing RevTraj and MaxProgress in two road topologies. MaxProgress performs better in a $\rho = 1$ topology, but is significantly outperformed by RevTraj under a low ρ topology.

schemes is improved when the node density increases. This is because the probability of finding cars increases with the density.

2.6.2 Evaluating Anchoring Protocols

Next, we compare the performance using synthetic traffic traces generated by the Paramics Traffic model. The road topology is based on southern New Jersey highway system as shown in Fig. 2.15. The Paramics Traffic model captures the interactions of real world road traffic through a series of complex algorithms that describe car following, lane changing, gap acceptance, and spatial collision detection. The trace contains 984, 445 records from 5000 cars in 6am-7am off-peak traffic period. Free-space propagation is used as communication model. Every result shown in this section is an average over 5000 simulation rounds. In each round, an event occurs at a randomly picked location at a random time. An initial carrier will move for a period of $T_h = 750$ seconds

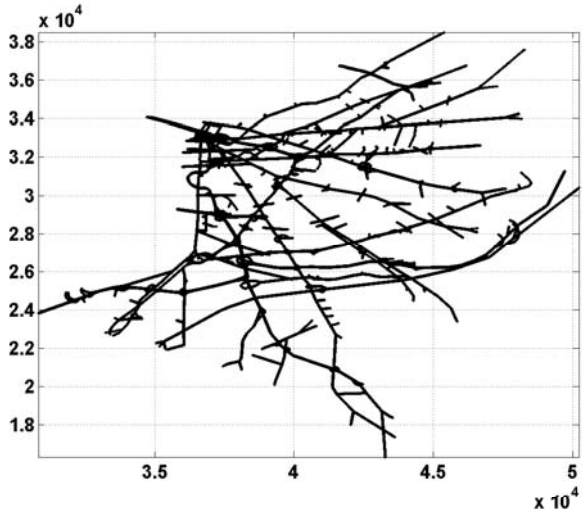


Figure 2.15: We evaluate the anchoring protocols using real traffic data from southern New Jersey. This is the roadmap of the traffic data.

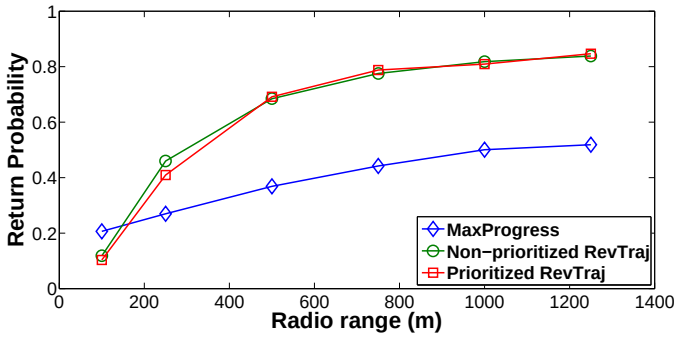
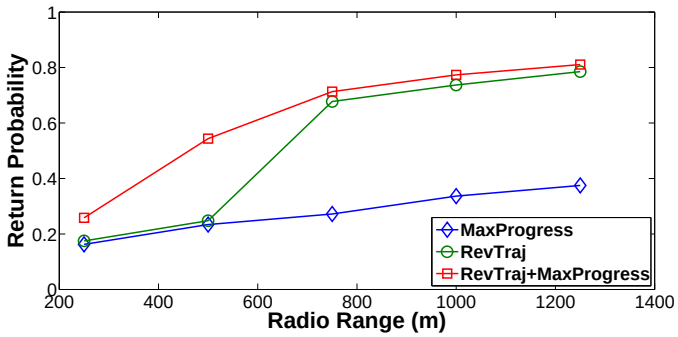
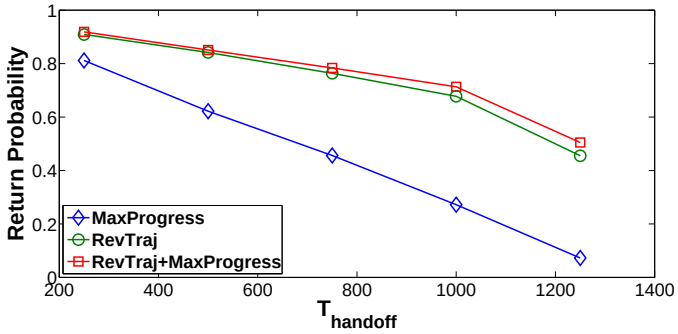


Figure 2.16: The comparison of three anchoring schemes when increasing the radio range radius. $T_h = 750$ seconds.

before handing off the geocache to others. A simulation round is end either when a successful return is made or time T_{end} elapses. After 5000 simulation rounds, the return probability is then calculated as the ratio of the number of successful returns over 5000. The results are shown in Fig. 2.16. Since the node density cannot change, we vary the radio range in the simulations to simulate different node densities. We find that except for extremely short radio ranges (100 meters), corresponding to extremely low node density, RevTraj significantly outperforms MaxProgress, with an improvement over 70%. This indicates that the real-world road topology has small ρ. Furthermore, we noted that all three schemes benefit from a large radio range or high density.



(a) Return probability when varying radio range values with T_h of 1000 seconds



(b) Return probability when varying T_h with radio range of 750 meters.

Figure 2.17: Performance evaluation between adaptive anchoring algorithm and other algorithms.

2.6.3 Adaptive Anchoring Scheme

Since both anchoring protocols have their own advantage over the other one under certain circumstances, it is natrual to develop a protocol which combines the two protocols and take the advantages from both. Just like RevTraj, the enhanced scheme keeps the trajectory to ensure a guaranteed return path to the anchor location. During a handoff process, if based on the RevTraj's defination the scheme cannot find proper carriers, it includes more candidates through the MaxProgress.

In Fig. 2.17 (a), $T_h = 1000$ seconds and the radio range is varying. Results show that across all radio ranges, the adaptive protocol has the highest probability of geocache return. RevTraj exhibits lower return probability when node density is low since it misses some chances that can lead to a successful return if MaxProgress was used. MaxProgress shows the worst performance. In Fig. 2.17 (b), the protocols are compared

with fixed radio range (750 meters) while varying the value of T_h. In all the cases, the adaptive scheme has the highest return probability. Finally, for all protocols, longer handoff time leads to lower return probability.

2.7 Related Work

This work spans the fields of mobile sensor networks and vehicular networks. Perhaps closest in spirit to the geocache programming abstraction are geographic hash tables [30], which provide a programming interface for data-centric storage in stationary sensor networks. Spatialviews [31] provides location-oriented programming language abstractions for mobile ad hoc networks, to ease application development and maintenance. This work does not address distribution of information at the protocol level, which is a key focus of our work.

a) Mobile sensor networks: Recent works in mobile sensor networks exploit mobility when it is not feasible to build a dense network of fixed sensors. Notably, Zebranet [2] places sensors on zebras to collect valuable zoology data. In under water sensor network [32], mobile nodes are robots that collect data from regions of interest. Several projects target specifically at vehicular sensing. CarTel [3], for example, is a comprehensive distributed mobile computing system used to collect, process and visualize data from sensors located on mobile units. It aims at exploring in-network computing on individual mobile units as we do, but it does not use inter-vehicle communication, which in our project, is a main focus to enable distributed aggregation of sensor readings from multiple cars. Another vehicular sensor network: MobEyes [4, 5], introduces MDHP (MobEyes Diffusion/Harvesting Processor), a protocol used to spread information within wireless sensor networks and build low-cost index of mobile storage. Although our projects bear similarities in that we also aim to develop low-cost yet efficient inter-vehicle communication protocol, MobEyes relies largely on an opportunistically broadcast approach, possibly with the emphasis of simplified protocol, while we aim at minimizing traffic overhead caused by more sophisticated schemes.

b) Inter-vehicle, geographic, and delay-tolerant communication: Many projects have

addressed scalable communication in mobile ad hoc networks (e.g., [33]), in sparse or disconnected mobile ad hoc networks (e.g., [1,34–36]), or through Infostations. In [37], the authors introduce Infostations to deliver data to mobile nodes. In [38,39], the authors aim at providing location-specific information to mobile devices, in which they developed schemes for detecting and transferring information of interest. All of these techniques adopt a server-client approach, but in our case, the information is provided by mobiles that have passed the location. The MaxProp [34] routing protocol is used to ensure effective routing of DTN (disruption-tolerant networks) messages via intermittently connected nodes. These protocols are based on different communication workloads, such as unicast between randomly chosen nodes, or multicast to random node sets. These techniques focus on delivering messages to certain nodes, while our protocols try to keep information around a certain location. In [1], designated mobile nodes (message ferries) store and carry messages. Our project differs in that virtually all nodes are "peers". In [40], the authors aim to guarantee message transmission in minimal time, at the expense of additional messaging overhead. Instead, our applications are more delay tolerant, and the main goal is to reduce communication overhead.

Geocast protocols [41–43] transmit messages to a predefined geographical region. They are suitable for location based services such as position-based advertising, publish and subscribe. Repeated geocasts or time stable geocasts [44] could also be used to maintain geocache in a certain area and bear similarities to our baseline scheme. It is different in concept though in that it requires the definition of a geographic region, which is not required in our case. Most geocast schemes concentrate on routing messages to the areas of interest, or distributing messages to all nodes [41,43], while geocache is established close to the anchor location and needs only be known to very few nodes. Further, time-stable geocasts continuously remain in the region of interest, while geocache can travel away from the anchor location. In [45], it mentions some trajectory concepts, but it fails to take into account the peculiarities of vehicular networks and still only forwards data to a node that is physically closer to the destination. Geopps [46] is maybe the most similar work to ours, however, it requires each mobile node to have full topology information which is not feasible in realistic scenario.

In [47], the authors examine the dissemination of availability reports about resources in mobile peer-to-peer networks. By opportunistically exchanging the reports, and decaying the relevance of the report as its age increases, the proposed algorithm is able to limit the global distribution of a report to a bounded spatial area and to the duration for which it is of interest. Although we are also interested in retaining spatio-temporal information to a local area, our focus is to maximize the probability to find certain information, instead of bounding its global distribution to a certain spatial area and duration.

c) Matching GPS observations: The problem of map matching based on GPS readings has been extensively studied. Existing work includes [48–50]. Even though we share some similarities with the map matching problem when using gps readings to identify road segments, we differ significantly with the general map matching problem in the use of road maps. Map matching solutions generally focus on matching a node's position to the nearest street presented in the map. This differs fundamentally from our work since we don't use street maps but only GPS readings of traversed paths. Therefore, the general map matching approach which involves searching and comparing nearby road segments could not be applied to our problem. Instead, we propose to use absolute distances and heading differences with the recorded road segment to determine divergence.

2.8 Conclusion

We have presented the trajectory-based boomerang protocol to periodically make available data at certain geographic locations in a highly mobile vehicular network. The boomerang protocol returns the geocache through nodes traveling toward the anchor location. To increase the probability of successful return, it records a node's trajectory while moving away from the anchor location then select nodes to return the geocache based on the trajectory (RevTraj). We compared this scheme with a shortest-distance scheme (MaxProgress), and demonstrated that the RevTraj scheme significantly outperforms its counterpart in realistic traffic simulation, with an improvement up to 70% in return probability.

Chapter 3

Security: Key Agreement Algorithms for VSN

3.1 Introduction

In this chapter, we consider two types of communication modes in vehicular sensing networks: Vehicle-to-Infrastructure (V2I) communication and Vehicle-to-Vehicle (V2V) communication. In V2I, individual vehicle speaks to the roadside infrastructure - unit (station) - to obtain or upload information to a remote traffic server or other application server. In V2V, private data communication is performed between pair of vehicles. Both communications are supported by Dedicated Short-Range Communications (DSRC) radio devices, which offer high data rate communication up to 1000 meters [51].

It is desired that two separate sets of secret keys can be used for V2I and V2V communications. For example, a driver may query the traffic center for a section of road along his/her route to the destination through V2I communication. Without a secret key shared between the server and the individual user, this query may be overheard and disclosed to the other users. Under some circumstances, this will result in serious privacy leakage. Therefore, secret key shared between the server and an individual user should not be known by others. On the other hand, sometimes a driver may want to query for the local traffic condition without disclosing his/her exactly current location to the remote server. This can be done through V2V communications on a distributed traffic alert system in which the traffic information are circled around locally. There are some other information the users may not want the server to know. For example, the users on multiple vehicles from the same family or company may want to keep communicating privately (watching family video/discussing creative business ideas) along the road while driving to the same destination. Police car perform tasks on the same road or a small area is another example. In all the above examples, the

secret key shared between a pair of vehicles needs to be kept unknown from the server.

The process through which two parties share a secret key is called key agreement or key management. Traditional approaches include public key (asymmetric) cryptography (such as Diffie-Hellman key establishment) and trusted third parties (TTP) [8]. However, neither approach fits the V2V communication nor V2I communication very well. To be more specific, TTP requires a trusted central server. However, for V2V communication, we do not assume there is a trusted third party or a central authority. The central authority assumption itself conflicts with our goal for a secure V2V communication. As discussed in the last paragraph, we do not want anyone except the two users to know the secret key. It is also unsecure for V2I communication because even though we can assume the infrastructure is connected to a trusted central server, the key distribution procedure itself is not secure, especially if the secret keys are distributed through wireless channels. Whether public key cryptography could be used in mobile ad hoc networks or not has been discussed for a long time. Due to its high computational cost, many prior research works [13, 52, 53] show that it is impractical for embedded intelligence and ubiquitous computing applications to use cost-effective processors with limited computational abilities. One issue with public key protocols is the number of certificates that need to be exchanged. With the proposed approach, certificates exchanging are avoided. Furthermore, public key cryptography is based on 'computational difficulty' which is not 'unconditional' secure. The security of such cryptography decreases while the adversary's hardware is improved or a new attacking algorithm is found. For examples, both RSA and ElGamal encryption have known attacks which have much low complexity than the brute force approach, and it has been known that the possibility of a man-in-the-middle attack is a potential security vulnerability in using asymmetric keys. Therefore, for both V2V and V2I communications, public key cryptography is not the best solution.

As discussed above, traditional approaches are not good choice. Therefore, in this work, we are looking for untraditional method to create secret keys for V2V and V2I communications.

By significantly extending our previous work [14, 54] and integrating the special

features of vehicular network, we develop a novel set of key agreement schemes for both V2V and V2I communications. The core of the proposed schemes can be summarized into two words: **Reciprocity** and **Diversity**. Reciprocity represents the channel reciprocity theorem. Diversity includes space diversity, frequency diversity and time diversity. Channel reciprocity theorem and spatial decorrelation - a concept derived from space diversity - together can be used to generate shared secret keys between a pair of vehicles for V2V communication in an urban scenario or a multi-path fading environment. To create a V2I secret key, the roadside units (RSUs) distribute seeds (pre-keys) through random channel hopping (different seeds are broadcast at different frequency channels in different time periods). When a vehicle enters the communication area of an RSU, it can tune its radio hardware onto a particular frequency channel and collect all the seeds from the channel. If two vehicles entering and leaving from the area of an RSU both at the same time but without always tuning the radio channel onto the same frequency, then with high probability, they will obtain different seeds in the end. This could also happen if two vehicles entering the communication area of an RSU at different time due to the space and time diversities. A vehicle can request other vehicles for additional seeds via the V2V secure communication. In the end, a secret key used for the V2I communication can be generated by XORing a set of seeds the vehicle collected. Due to the special property of the XOR operation, missing of one seed by the adversary, can results in a failure to obtain the final secret key.

The proposed V2V key agreement method is one kind of information-theoretic key establishment schemes or physical layer security methods. Compared to previous efforts, our scheme has higher key generation rate. This is important, since in vehicular networks the communication channels sometimes only exist in very short time period. For example, two vehicles moving from opposite directions at the speed of 30 meters per second will only have less than 17 seconds to communicate (802.11p has maximum 1000 meters communication range). To exchange a large volume of sensitivity information, it requires fast building up of a security communication channel. E.g., a police car can quickly identify a vehicle and its driver or other information by matching the plate number and the driver information on the fly. Further investigation and information

request could be done just during a couple of seconds when they bypass each other. Thus, a security channel must be set up as quickly as possible. Similar scenarios could also appear in traffic control and road monitor applications, in which crowd sourcing should be done in a transparent, fast and verifiable way (through security channel). To be specific, vehicles could exchange traffic or road condition information when they meet on the road. Such information can be used for individual use or can be uploaded to the traffic center for final analysis. However, certain level of authentication through a security channel is necessary to keep bogus data as little as possible.

To summarize, the main contributions of this work are:

1. We propose a secret key agreement scheme for V2V communication. This scheme is based on the channel reciprocity theorem and the spatial decorrelation property. It achieves strong information-theoretic security by extracting secret bits from the wireless channel between two legitimate users. It is different from previous efforts which also try to achieve information-theoretic security through channel characteristic. In this proposed scheme, the amount of increasing or decreasing of the Received Signal Strength (RSS) value is used to identify a secret bit instead of using the RSS value itself. This change helps to increase the key generation rate and prevent the attacks that have been found in prior methods.

2. We propose a secret key agreement scheme for V2I communication. This scheme exploits random channel hopping mechanism to create frequency diversity when distributing different seeds through RSUs. Unless two vehicles are listening to the same channel all the time, there is a high probability that these two individuals obtain at least one different seed. The final secret key is obtained through XOR operation on a set of received seeds, in which one seed difference is enough to cause a total different key. Due to the space and time diversities, the security can be further improved through seed exchanging between vehicles via the secure V2V communication.

3. We evaluate the proposed V2V key agreement and the V2I key agreement schemes through extensive simulation, experiment and numerical analysis. By comparing

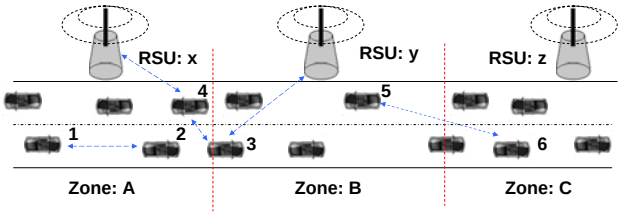


Figure 3.1: An illustration of vehicular communication networks.

to a baseline, a prior work [55], which also tries to achieve information-theoretic security through channel characteristic, it is shown that the proposed V2V secret key agreement scheme can generate strong secret key 100% faster than the baseline. We also show that the proposed V2I secret key agreement scheme can achieve strong security with extremely high probability even when the adversary has impractical capability comparing to a regular legitimate user.

The remainder of this chapter is organized as follows. Section 3.2 describes the system model. Section 3.3 introduces the main algorithms. To be specific, section 3.3.1 describes the V2V secret key agreement scheme and section 3.3.2 describes the V2I secret key agreement scheme. Section 3.4.1 and section 3.4.2 evaluate the two schemes respectively. Section 3.5 discusses several methods to further improve the V2V key agreement scheme. Section 3.6 reviews some related work. Section 3.7concludes our work.

3.2 System Model

Vehicular communication system contains two types of nodes: vehicle On-Board Unit (OBU) and RoadSide Unit (RSU). To support *Intelligent Transportation Systems (ITS)* applications, both OBU and RSU are Dedicated Short Range Communications (DSRC) devices. 802.11p is used as a special operation mode of IEEE 802.11 for vehicular networks called Wireless Access in Vehicular Environments (WAVE) on these devices. DSRC devices operate over seven 10 MHz channels in a 75 MHz region of the 5.9 GHz unlicensed band [51].

Fig.3.1 describes the whole system structure. All vehicles which can perform V2V

and V2I communications, has DSRC device installed which can communicate up to 1000 meters. As long as two vehicles are in each other's communication area, they can communicate, no matter if they are moving toward to the same direction, for example vehicle 1 and 2, or they are moving toward to the opposite direction, for example vehicle 5 and 6. Vehicles access remote traffic server or other application servers through RSUs. Each RSU can only cover a segment of the road. Therefore we divide the road into different zones. As shown in the figure, RSU-x covers zone A, RSU-y covers zone B, and RSU-z covers zone C. When vehicles are in the same zone, they all can listen to the same RSU. For example, vehicle 1, 2 and 4 all can communicate with RSU-x. However, even if two vehicles are very close to each other and can talk to each other such as vehicle 3 and 4, they might belong to different zones and cannot listen to the same RSU. Additionally, in this model, we allow an RSU jump between channels when broadcasting seeds.

We assume two types of adversaries. (1) The first kind type is interested in knowing the content of the private communication between two mobile users. The adversary could be any other vehicle as long as it is not the users themselves. The adversary could have much more powerful hardware than the users have. It could even be the traffic server which can access and control all the RSUs to monitor the communication between the two users. Finally, the adversary can even have the ability to combine information obtained from the server and multiple vehicles. However, we do assume that any device used by the adversary is not installed less than 1/2 wavelength away from any of the two vehicles' DSRC antennas and the adversary cannot sniffer inside the vehicle. This assumption is reasonable since otherwise, the user may notice the eavesdropper. (2) The second one is interested in eavesdropping the communication content between a vehicle and the server. Thus, the adversary should not be the target vehicle itself and also cannot be the server or the RSUs controlled by the server. The adversary can cooperate with multiple vehicles which are also moving in the same area near the target vehicle. We also assume that deploying multiple radio devices at every RSU along the entire vehicular networks is infeasible to the adversary. Finally, we assume none of the adversaries is interested in interrupting the key agreement process.

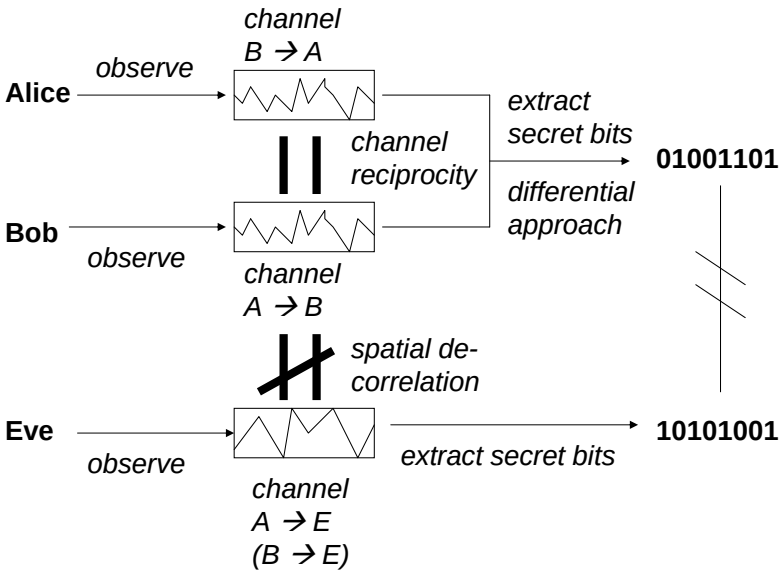


Figure 3.2: The proposed V2V key agreement flow chart.

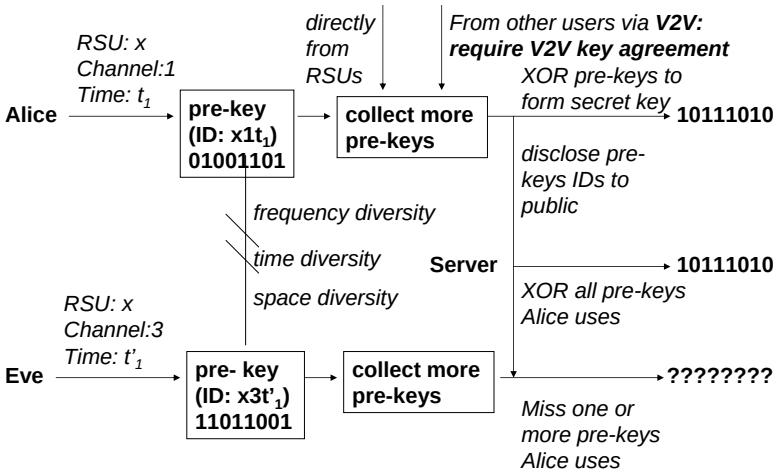


Figure 3.3: The proposed V2I key agreement flow chart.

3.3 Main Algorithm

In this section, we will describe the V2V and V2I secret key agreement schemes in detail. First, we give an overview on both schemes through Fig. 3.2 and Fig. 3.3 respectively.

As shown in Fig. 3.2, because of channel reciprocity, two vehicles - Alice and Bob - observe similar channel characteristic. Through the proposed differential approach, they can extract a sequence of bits from the channel at each side. The unique sequence of bits can be used to form a secret key for V2V communication. On the other side, Eve can observe either channel from A to E or from B to E. However, due to the spatial

decorrelation, both channel characteristics are different from channel A to B (and B to A). Therefore, the bit Eve can extract from the channel is different from the bits extracted at Alice or Bob side.

Fig. 3.3 describes the V2I key agreement scheme. Alice receives a seed from the RSU x at channel 1 and time t_1. Because of frequency, space and time diversity, Eve doesn't receive the same seed. For example, 1) she is not on the channel 1 and/or 2) she is not inside the communication range of the RSU x. Later, Eve receives seed from RSU x at channel 3 and time t_1'. However, these two seeds are totally independent. Alice continues collecting seeds along the road as well as exchanging seeds with other legitimate users when possible. In the end, Alice selects some of the seeds and uses XOR operation to form a secret key. The index of the selected seeds are disclosed to the server, thus, the server can execute the XOR operation on the same set of seeds to regenerate the key. On the other hand, if Eve misses one seed, she cannot form the same key.

Next, we will describe the V2V key agreement scheme and V2I key agreement scheme in more detail.

3.3.1 Vehicle-to-Vehicle: The Differential Approach

The V2V key agreement scheme is based on channel reciprocity theorem and spatial decorrelation property. A secret bit 1 or 0 can be extracted from the channel while the channel characteristic changes. Therefore, we propose a differential approach to capture such variations and generate secret bits.

Principle

Channel reciprocity describes the phenomenon that the communication nodes at the two ends of a channel will observe identical channel characteristic, such as channel impulse response or Received Signal Strength (RSS) value. Fig. 3.4 shows a period of RSS measurement under a multi-path (office) environment. Two users, Alice and Bob, alternatively transmit wireless signal to each other while moving in about 1 meter per second. Both Alice and Bob probe the channel and measure the RSS values at a rate

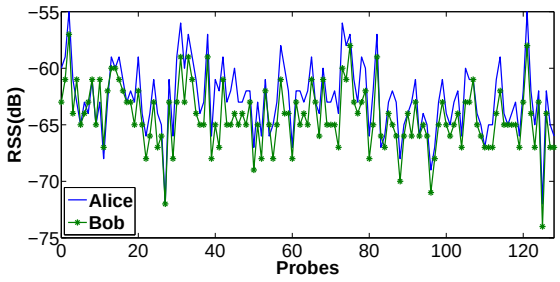


Figure 3.4: An illustration of the channel reciprocity.

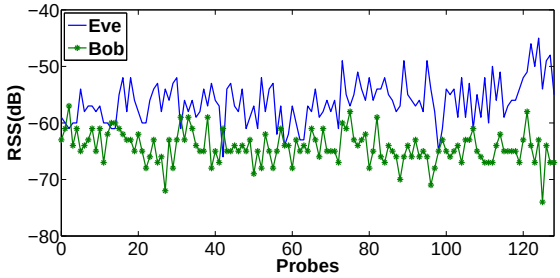


Figure 3.5: An illustration of the spatial decorrelation.

of 40 per second. As shown in Fig. 3.4, due to the channel reciprocity, the RSS values observed at Alice and Bob's sides for Alice-Bob and Bob-Alice channels respectively are highly correlated (=0.9120).

On the other hand, Eve, who is at a different location from Bob, observes different RSS values from Alice-Eve channel comparing to Bob's observation of the Alice-Bob channel, as shown in Fig. 3.5. The two curves shown in the figure are highly uncorrelated (=-0.1937) due to the spatial decorrelation property in multi-path (Rayleigh) fading channel. Theoretically, the spatial decorrelation property can be described by a Bessel function:

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} cos(xsin\theta) d\theta \tag{3.1}$$

in which x is the distance between Bob and Eve in the unit of wavelength λ, and θ is phase offset. As shown in Fig. 3.6, when the distance is larger than $\frac{\lambda}{2}$, the channels they individually share with Alice are uncorrelated.

The wireless channel between Alice and Bob can be described as complex and discrete function of time $h_t = H(t)e^{j\gamma_t}$. Alice sends Bob a signal $s(t_1) = A_{t_1}e^{j\phi_{t_1}}$ at time

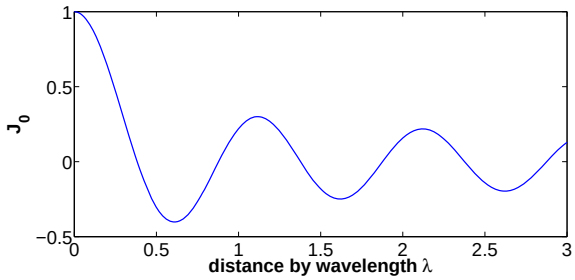


Figure 3.6: Using bessel function to describe spatial decorrelation.

t_1, then the received signal at Bob can be written as:

$$y_{t_1} = H_{t_1} A_{t_1} e^{j(\phi_{t_1} + \gamma_{t_1})} + n^b_{t_1} \tag{3.2}$$

where n^b_t is the noise terms which are independently and identically distributed complex Gaussian random variables. A signal $s(t_2) = A_{t_2} e^{j\phi_{t_2}}$ sent from Bob to Alice is received as:

$$y_{t_2} = H_{t_2} A_{t_2} e^{j(\phi_{t_2} + \gamma_{t_2})} + n^a_{t_2} \tag{3.3}$$

From t_1 to t_2, if the total changes in location on both Alice and Bob sides are much smaller than $\frac{\lambda}{2}$, then good estimated values of $\hat{h}_{t_1} \approx H_{t_1} e^{j\gamma_{t_1}}$ at Bob side and $\hat{h}_{t_2} \approx H_{t_2} e^{j\gamma_{t_2}}$ at Alice side are highly correlated according to equation 3.1. On the other side, assume Eve is not close to either Alice or Bob, she cannot obtain a proper estimation on either h_{t_1} or h_{t_2}. Furthermore, when Alice and Bob alternatively send each other probe signals, then the sequences of probed channel characteristics $[\hat{h}_{t_1}, \hat{h}_{t_3}, ..., \hat{h}_{t_{2n-1}}]$ are highly correlated to the sequence of probed channel characteristics $[\hat{h}_{t_2}, \hat{h}_{t_4}, ..., \hat{h}_{t_{2n}}]$.

Challenges

Several research groups have proposed key agreement schemes based on the channel reciprocity theorem and spatial decorrelation property. The core idea of most existing works to extract secret key from RSS values is Quantization, in which, one or two threshold values are either determined through a pre-probe phase such as in [55, 56] or post-phase process [57, 58]. The value of a secret bit is obtained by comparing the RSS values to the threshold values. By studying existing works, we feel the proposed scheme must at least fulfill the following challenges.

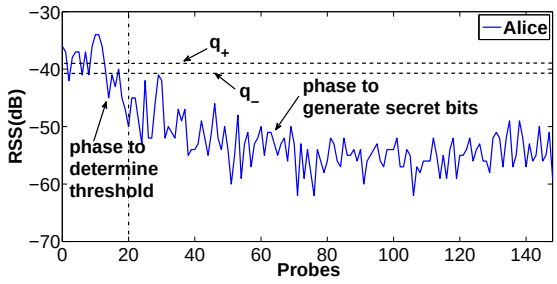


Figure 3.7: An example of the pre-probe method fails.

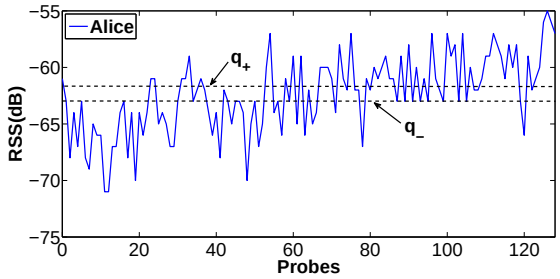


Figure 3.8: An example of the post-probe method fails.

Prevent Entropy Reduction. We introduce a metric: entropy, to evaluate the strength of a secret key:

$$H_i = -p_0 \log p_0 - p_1 \log p_1 \tag{3.4}$$

$$H_{average} = \frac{1}{N} \sum_{i=0}^{i=N} H_i \tag{3.5}$$

where N is the total length of the secret key, p_0 is the post test probability of a bit being 0 based on adversary's knowledge. The closer to 1 the value of $H_{average}$ is, the stronger the secret key is.

In the pre-probe method, the thresholds are determined in a pre-probe phase. This method relies on the assumption that the future probes are roughly and evenly distributed around the thresholds. However, this is not always true. As shown in Fig. 3.7, in which the thresholds q_+ and q_- are calculated according to [55]. If an adversary notices that the Euclidean distance between two vehicles is dramatically increasing during the bits extracting phase, he might easily predict the results. Therefore, the entropy of the resulted secret key is very low.

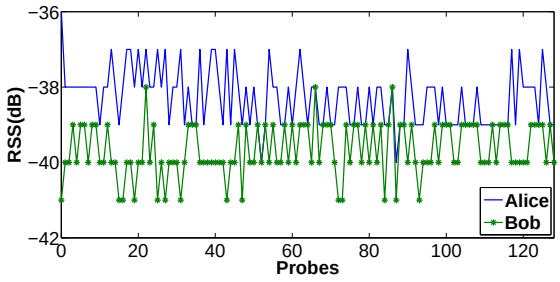


Figure 3.9: The impact of small fluctuations.

In the post-probe method, thresholds are determined after all RSS samples have been collected. As shown in [57], by inserting or removing intermediate objects between Alice and Bob, Eve can force the RSS curves following certain trends as shown in Fig. 3.7 and Fig. 3.8. Secret keys based on these two curves are very predictable from Eve's point of view. Therefore they have very low entropy.

Reduce the Impact of Small Fluctuation. It is known that small fluctuation may reduce the effect of channel reciprocity. Especially when the real channel variation is smaller than the small fluctuation caused by noise, interferes etc. As an example, Fig. 3.9 shows that when users are static, small fluctuations may dominate the RSS variation. Under such condition, no secret bit can be correctly extracted from the channel. The proposed algorithm should be able to reduce the impact of small fluctuation.

Increase the Secret Bit Generation Rate. In V2V, the communication is limited by the period of the encounter duration between two vehicles. When vehicles moving out of radio range, the communication stops. Sometimes the duration of the communication could be very short. In this work, we pursue a more efficient and fast key agreement scheme comparing to previous works.

Algorithm

Instead of using absolute thresholds, the proposed differential approach determines a secret bit based on the difference between two neighbor RSS values. To illustrate the basic concept, an example is shown in Fig. 3.10. In this method, whenever an increase between two RSS values is observed, a bit 1 is generated, and a bit 0 is generated for a decrease.

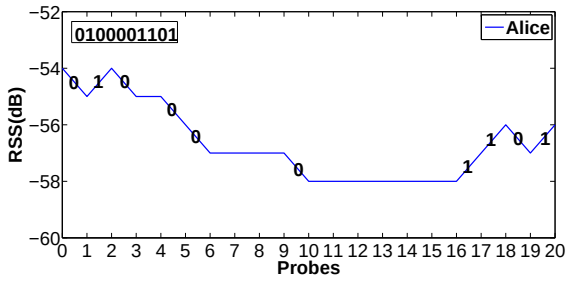


Figure 3.10: An illustration of the differential method.

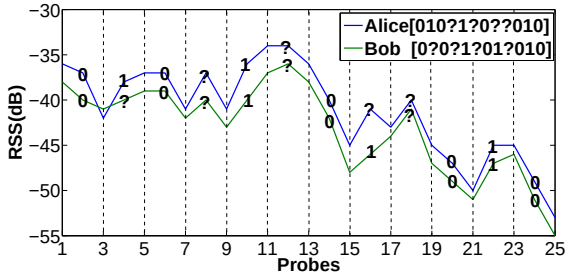


Figure 3.11: Differential approach: fixed interval.

The differential approach can be summarized in the following steps:

1. Sample collection: Both Alice and Bob collect a period T of RSS values using their maximum probe rate.

2. Segments division: Divide the sequence of probes into segments by every τ number of probes.

3. Small fluctuation removal: Using moving average method to reduce the influence of small fluctuation by width d.

$$Y = \frac{x_1 + x_2 + x_3 + ... + x_d}{d} \tag{3.6}$$

4. Bit extraction: Secret bit is generated by comparing a RSS sample of each segment (for example the first RSS value of the segment). Set bit to 1 if there is an increase by more than ϵ/d, and 0 if there is a decrease by more than ϵ/d. ϵ is an approximate estimate of the small fluctuation, it could be different for Alice and Bob[1]. Note, to reduce the computational load, we only need to calculate the

[1]Different device may have different accuracy on RSS value estimation.

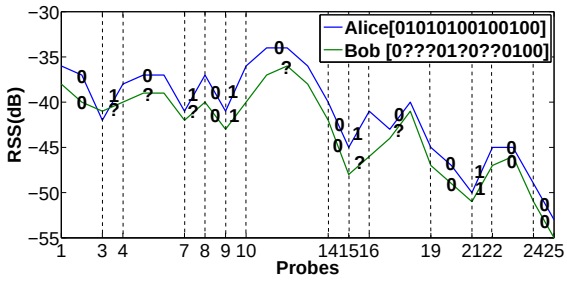


Figure 3.12: Differential approach: dynamic method.

moving average of one value in each segment.

5. Information exchange: Alice sends Bob only the positions of those probes which are used by her to generate secret bits. From those positions, Bob picks the ones he can also extract secret bits and replies back to Alice.

Fig. 3.11 gives a more concrete example for the key agreement scheme. For the sake of simplicity, in this example, we assume the moving average width $d = 1$, $\tau = 2$ and the value of ϵ for both Alice and Bob is equal to 3. Alice obtains a sequence of bits 010?1?0??010 by comparing the first RSS value of each segment. She is unsure of the bit values at positions '4,6,8,9' in the sequence. Then she sends Bob a message to disclose this information. On the other hand, Bob obtains bit sequence 0?0?1?01?010. In addition to what Alice is not sure of, Bob adds position '2' to the unsure bit list and informs Alice. After taking out the unsure bits, both Alice and Bob obtain the final bit sequence 0010010. To further improve efficiency, we introduce another parameter ϵ_2 related to the small fluctuation, $\epsilon_2 = a * \epsilon$, $0 < a < 1$. When only one of Alice and Bob is not sure about a bit at a specific position, she/he uses ϵ_2 instead of ϵ to identify a bit value. Through this way, more secret bits can be generated since ϵ_2 is smaller than ϵ. For the case in Fig. 3.11, assume $\epsilon_2 = 0.5 * \epsilon$, two more bits 1 will be generated from segment 2 and 8.

One of the advantages of using differential method is that it can prevent the attack described in [57]. Because, even the channel condition is improving or downgrading following an observable trend, this method will not generate bits all in 1 or 0 values.

To further reduce parameter dependence, we propose a dynamic differential approach in which the fixed interval τ is removed. In this approach, the first RSS sample is used as a reference. Every RSS sample starting from the second one will be compared with the reference until a difference larger than ϵ/d is observed. A bit is extracted depending on whether the difference is an increase or a decrease. Reference is updated at the position where a bit is extracted. The balance RSS samples will be compared with the updated reference until the next large difference appears. In the end, Alice sends Bob the positions of which she is able to extract secret bits. Upon receipt, Bob checks if he can also extract bits from these positions and then, sends the results back.

In Fig. 3.12, we assume $d = 1$ and $\epsilon = 3$. Based on the method described above, Alice extracts a bit sequence 01010100100100. Bob recognizes 0???01?0??0100 and recommends Alice to remove unsure bits at positions '2,3,4,7,9,10'. Therefore the resulted common bit sequence is 00100100. If using two ϵ thresholds, for example $\epsilon_2 = 0.5 * \epsilon$, the bit sequence Bob obtains becomes 0?0101001?0100. This updates the common bit sequence to 001010010100.

Note, the bit sequence both Alice and Bob obtain is not necessarily the final secret key. For example, they could reorder the bits in the sequence and/or remove some bits from the sequence.

3.3.2 Vehicle-to-Infrastructure: a Hierarchical Approach

As shown in Fig. 3.3, the proposed V2I key agreement scheme is a combination of different kind of diversities. Through this scheme, individual vehicle can create a secret key or update a session key between itself and the server.

Problem Statement

We have shown that traditional key agreement and public/private cryptography are not good candidates for secure V2I communication. The V2V secret key agreement scheme does not fit V2I either. That is because: 1) The V2V secret key agreement scheme requires a multi-path (Rayleigh) fading environment, however, in V2I, this may not exist. For example, when the RSU is installed on a much higher position comparing to

all the moving vehicles, channel characteristic is dominated by line-of-sight propagation. 2) RSUs are installed at fixed locations and may not be checked by people for certain time. Thus, an adversary may have an eavesdropper installed very close to the RSU device for a long period before anyone notices that. 3) Relying on a particular RSU to generate secret key between the vehicle and the server is not a very strong secure manner. If the adversary compromises the RSU, the secret key is no longer a secret.

Exploit Channel Diversity: The Frequency Hopping Method

Since it is not proper to reuse the solution of V2V in V2I, we propose to use a frequency hopping method and its extension based on time and space diversity properties. The principle of frequency hopping method is described below.

Assume an RSU (Alice) is the transmitter, one legitimate vehicle receiver (Bob) and one passive vehicle eavesdropper (Eve). Everyone can communicate on multiple, non-interfering channels, but receive on a single (or a few) channel at a given time. As in [59], we assume that the hardware Eve has is similar to Alice's and Bob's. Alice and Bob seek to establish a secret key without any prior shared information.

A) Basic Packet-Based Scheme. We first describe a basic packet-based protocol before proceeding to the final multi-agreement scheme. The idea underlying this scheme is that both parties of the key agreement process—Alice and Bob—randomly select a channel to send and listen to, respectively. If they are on the same channel, key information is successfully transferred and Bob sends an acknowledgment (ACK). Otherwise, a timeout occurs. Alice and Bob may select other channels and repeat the process. Alice must use a different key material, which we refer to as a seed, for every transmission attempt. If Alice receives an ACK, she knows that this seed will be used, otherwise she discards the seed.

While Bob could also select a new channel for each attempt, this would require Alice and Bob to maintain synchronized timers. Since Eve has no knowledge of which channel Bob is on, Bob remaining on the same channel will not affect the security of the scheme.

Fig. 3.13 illustrates this scheme. Alice sends Bob several 4-bit seeds through different

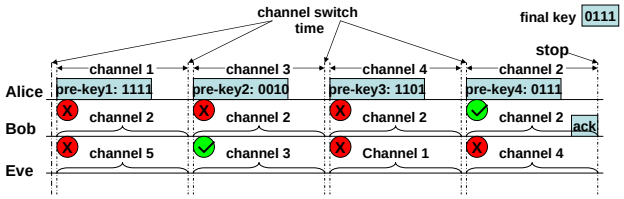


Figure 3.13: An illustration of packet based scheme.

channels. Bob successfully receives the last seed and sends an ACK. Even though Eve happens to overhear the second seed, it will not be of any use since the last seed is different.

Analysis. Given n channels, the probability that Alice and Bob are on the same channel is $p = \frac{1}{n}$. Assume that Eve can monitor one channel at a time as Bob, the probability she overhear Bob's secret key is $p_e = \frac{1}{n}$. To achieve a high level of security, the basic scheme requires a large number of channels. This is impractical because the number of available channels is often limited by the radio hardware, and the time required for a successful key exchanging increases with the number of channels. In fact, the probability that Alice and Bob successfully exchange a secret key in x attempts follows the **Geometric** distribution $P_X(x) = p(1-p)^{x-1}$. Thus the expected number of exchange attempts is $E[X] = \frac{1}{p} = n$, where n is the number of channels. This means that halving the probability of key overhearing p_e requires twice the number of channels and time required for key agreement.

B) Multi-Agreement Scheme. To address the limitation, we introduce a multi-agreement scheme. In this scheme, Bob and Alice will repeat the seed agreement multiple times. The process will end when Bob receives k seeds and the final secret key will be a XOR of all the seeds.

Analysis. By using k multiple seeds, the probability function for Alice and Bob to create a secret key in x packet exchange attempts, changes from the Geometric distribution to a **Pascal** distribution: $P_X(x) = \binom{x-1}{k-1} p^k (1-p)^{x-k}$, and the expected number of packet exchange attempts becomes $E[X] = \frac{k}{p} = kn$ where $p = \frac{1}{n}$. The probability for Eve to obtain secret key decreases to $p_e = (\frac{1}{n})^k$. Note that now doubling the value of k doubles the expected time to create a secret key, however, the value of

p_e is squared. This allows us to achieve a higher level of security in less time.

C) Move to V2I. In V2I communication, RSUs controlled by the server will be the legitimate senders (Alice) who broadcasts seeds through random channel hopping scheme. There is no immediately feedback: ACK. Instead, after a vehicle (Bob) has received enough seeds, it sends the server a message about the seeds it will use to form a secret key, as shown in Fig. 3.3. The message tells the server information about the time and from which RSUs the seeds are collected. Based on above knowledge, the server can also reproduce the key from its side.

Further Enhancement: The Space and Time Diversities

Space Diversity. As shown in the Fig. 3.1, when two vehicles are in different zones, even if they are very close, they cannot hear signal from the same RSU. Therefore, spatial difference increases the chance that an adversary missing a seed. When using XOR operation to form secret key, one missing is enough to fail the attack. Furthermore, a vehicle, especially the one from opposite direction, has a very high probability of having different seeds somewhere else. Thus, we could further improve the scheme by exchanging seeds between vehicles via the secure V2V channel created before. As an adversary, simply following the target and collecting all seeds along the road does not work anymore.

Time Diversity. A vehicle does not need use all received seeds to form a secret key, instead it leaves some seeds for future use. This kind of time diversity makes an adversary even harder to get all the seeds in forming a key. If the set of seeds across a long time span, it becomes even worse. Not to mention, a vehicle may also receive old seeds from other vehicles. To further prevent an adversary to collect same seeds from others, we limit that a vehicle can only give away a seed once, and the seed must be directly obtained from RSUs. On the other side, the server publish the ID of the seeds that have been used in generating secret key, so a vehicle could estimate if giving away a seed would threaten the strength of someone's secret key.

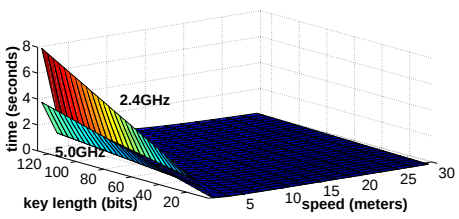


Figure 3.14: Time taken to generate a secret key while varying the key length and the nodes moving speed.

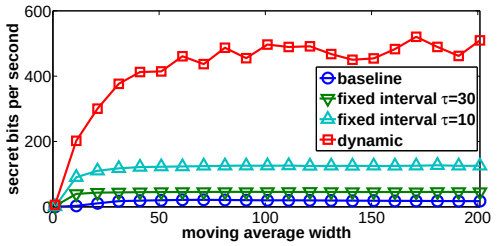


Figure 3.15: Comparison of the bit generation rate among schemes.

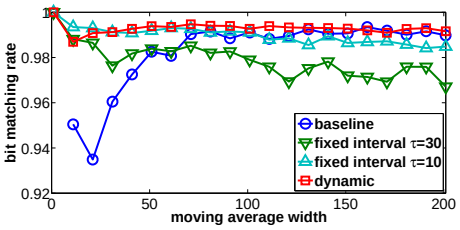


Figure 3.16: Comparison of the bit matching rate among schemes.

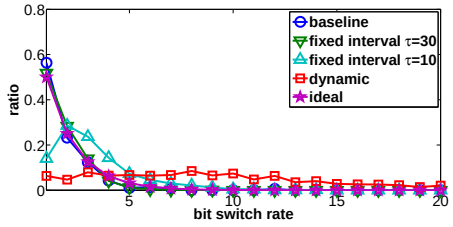


Figure 3.17: Comparison of the bit switch rate among schemes.

3.4 Experimental Results and Numerical Evaluation

In this section, we first evaluate the proposed V2V key agreement scheme based on real experimental data. Then, we show the simulation and numerical analysis results for the V2I key agreement scheme.

3.4.1 Vehicle-to-Vehicle Case

Fig.3.14 shows the time taken to generate different size of secret key theoretically. Through channel reciprocity and spatial decorrelation, generating 128-bit secret key takes just 8 seconds for 2.4GHz communication or 4 seconds for 5GHz communication at a moving speed of 1 meter per second of a mobile node. In a quick movement scenario, since the location changes quickly, the spatial decorrelation becomes fast. Therefore for a speed of 30 meters per second, it takes 0.13 seconds for 5GHz or 0.27 seconds for 2.4GHz to generate a 128-bit secret key.

The following results shown in this subsection are based on real experimental data. In the experiment, two mobile nodes, Alice and Bob, are moving in a multipath fading

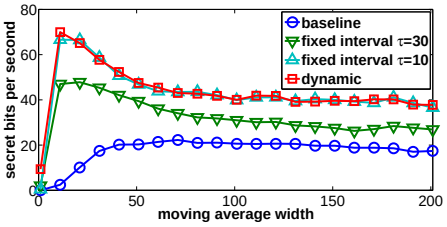


Figure 3.18: Comparison of high entropy (full) secret bit generation rate among schemes.

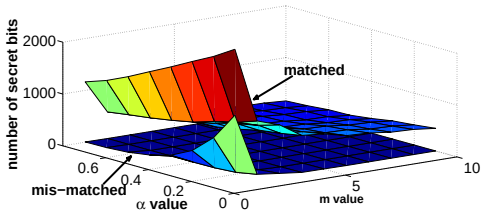


Figure 3.19: Comparison of the total bits generated from the sample data.

channel environment and collecting 50000 RSS value samples at the same time. We use Orbit mobile nodes [60] equipped with Atheros AR5212 Mini PCI wireless interfaces[2]. Our baseline is based on the protocol proposed in [55]. However, we have updated the scheme from pre-probe to post-probe. In [55], the authors use level-crossings and quantization to extract bits from correlated stochastic processes. To be more specific, two legitimate users use the channel statistics to determine scalars, q_+ and q_- serve as reference levels for quantizing. A secret key bit 1 or 0 is agreed if enough channel magnitude measurements are higher than q_+ or lower than q_- on both sides. Since post-probe method generally can achieve higher performance than pre-probe method due to its better threshold setting, we update the baseline by using post-probe method.

In Fig. 3.15, we compare the bit generation rates among four schemes: baseline [55], fixed interval differential (interval $\tau = 10$ and interval $\tau = 30$) and dynamic differential schemes. Fig. 3.16 shows the bit matching rate. Note, the results of baseline are already enhanced by subtracting the moving average and setting $\alpha = 0.125$ and $m = 4$. In all proposed schemes, $\epsilon = 6$ and $\epsilon_2 = 3$. Estimated value of $\lambda/2v$ is around 25, which means ideally, an uncorrelated secret bit can be generated every 25 probes. As shown in the figure, all differential approaches perform better than the baseline. In fixed interval method, the smaller τ, the higher generation rate. This is easy to be understood because small τ results in more RSS values to be compared and consequently more large scale variations may be caught. However, the negative part is it may generate correlated bits that have low entropy. This fact is shown in Fig. 3.17. In the figure, x indicates

[2]Although it use 802.11a instead of 802.11p for wireless communication and at a relative slow moving speed, the results is still instructive.

the length of a continuous 1 or 0 bit sequence, and y is the probability distribution of different length. Both fixed interval with $\tau = 10$ and the dynamic approach do not have the same distributions as the ideal one. Since a long set of 1 or 0 due to the correlation is easy to be predicted by an adversary, these two cases will generate bit sequence with low entropy. There are some methods to convert such a low entropy bit sequence into a high entropy bit sequence. For example, remove some redundant bits. In Fig. 3.18 we show the results after conversion. The dynamic approach and the fixed interval with $\tau = 10$ have similar and the highest bit rate.

Above figures also show the affect of moving average width. For all differential methods, the width should not be too small in order to remove the small fluctuations. Otherwise, it leads to a low bit generation rate. For the baseline, small width also doesn't work since it cannot remove large scale fading. On the other hand, moving average width should not be too large, because it will screen out some useful large variations. In the proposed approaches, the width could be estimated by $\lambda/2v$. However, for the baseline there is no proper method to estimate the width[3].

Next, we study each approach individually. The total number of secret bits can be generated from 50000 collected RSS samples with different parameter settings through baseline are shown in Fig. 3.20. We use moving average width 81 (an observed optimal setting from previous results) and α varies from 0.1 to 0.8 and m varies from 1 to 10. The baseline generates 2595 bits maximally with the mis-matching rate as high as 0.5094. To reducing the mis-matching rate, we have to limit the bit generation rate which results in a less than 1000 bits total output.

Next, we study the bit generation rate for the fixed interval approach. The moving average width d is set to 21, a value close to $\lambda/2v$. As shown in Fig. 3.20, the smaller the sample interval τ, the more secret bits can be generated. However, when τ is too small, for example $\tau = 1$, the bit generation rate dramatically decreases. This is because small τ does not make room to generate enough difference between two neighbors. In the figure, when τ is equal to 5, it generates more than 5777 secret bits with a bit

[3]This is because for the proposed methods, we need to remove the small fluctuations only, however in the baseline, it needs to remove a large scaling fading which is harder to estimate.

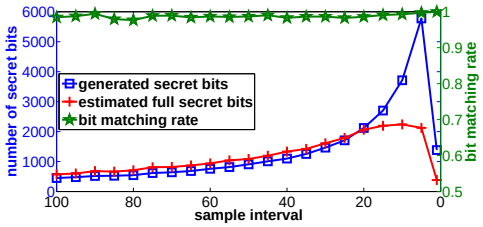


Figure 3.20: The impact of sample interval.

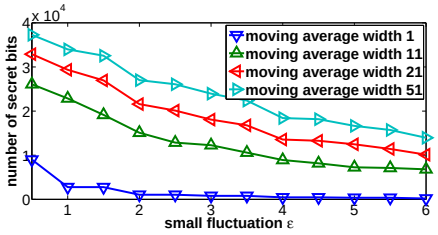


Figure 3.21: The impact of moving average width.

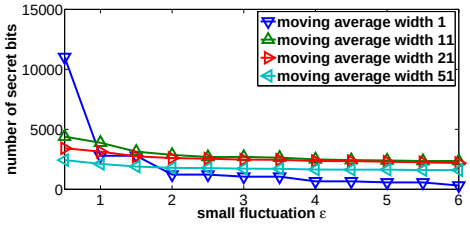


Figure 3.22: The impact of moving average width and parameter ϵ on high entropy (full) secret bits.

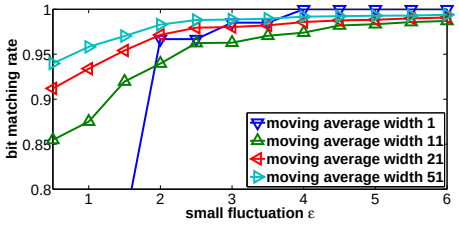


Figure 3.23: The impact of moving average width and parameter ϵ on bit matching rate.

matching rate of 0.99[4], which is much better than the best the baseline can achieve. After removing the bits with low entropy, it generates maximally 2239 bits when $\tau = 10$, which is twice as the baseline generates. When τ is set to 20, it generates more than 2000 bits. Another advantage of this approach over the baseline is that the parameter τ can be easily determined by making the sample interval close to $\lambda/2v$ in the unit of time.

At last, we discuss the dynamic differential approach. Fig. 3.21 shows that the larger the moving average width, the more bits are generated. However as we know, if the width is too large, some useful large scale variations will be removed improperly. Why Fig. 3.21 does not reflect this factor? This is because when ϵ/d is small, a large number of low entropy bits are generated. After removing those low entropy bits as shown in Fig. 3.22, large moving average width is no longer always preferred. As shown in Fig. 3.21, the smaller the value of ϵ, the more initial bits are generated. Fig. 3.22 shows that the trend becomes less significant after conversion. We also observe that

[4]Using information reconciliation, such as the ones mentioned in [61,62], even higher matching rate can be obtained by scarifying some bits generation rate.

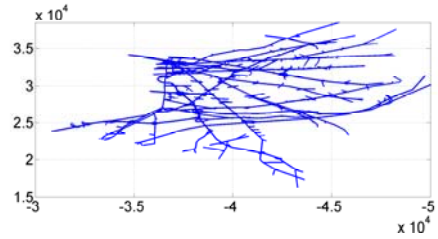


Figure 3.24: The map of southern New Jersey vehicular network.

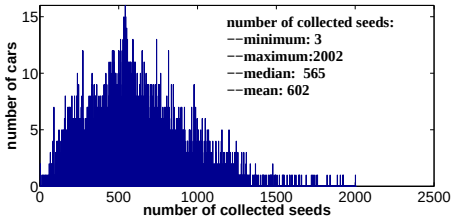


Figure 3.25: The histogram of the number of collected seeds per car.

the larger ϵ value, the higher the matching rate from Fig. 3.23. When ϵ reaches certain value, the matching rate stays stable. Finally, when moving average width is large, the affect of ϵ becomes small.

3.4.2 Vehicle-to-Infrastructure Case

In this subsection, we study the V2I key agreement scheme through simulation.

The simulation data are generated from Paramics Traffic Simulation model. As shown in Fig. 3.24, they are collected from the southern New Jersey highway network. Total data includes 984,445 records from 5000 cars in a 3395-second period. We assume every 1500 meters in vertical or horizontal distance, an RSU is deployed. Every 0.5 seconds, each RSU randomly picks one of three predefined channels to broadcast a seed. A vehicle receives the seed if it is tuning to the right channel and is in the right distance (less than 1000 meters Euclidean distance away). It is possible that a location is covered by more than one RSU. If a vehicle hear signal from multiple RSUs, it only receives the seed from the closest one.

In Fig. 3.25, we show the histogram of total received seeds for each vehicle. During the simulation, some vehicles collect more than 1000 seeds, and some collect less than 100 seeds. The number of seeds collected is basically proportional to the driving duration. As shown in the figure, the average number of seeds collected per car in this simulation is 602, and the median is 565. Considering most vehicles appear less than 3395 seconds in the simulation, in real world, one hour driving can bring a vehicle more than 500 seeds with high probability. It is easy to see that the probability of a certain number of seeds collected by a vehicle in one hour follows **Binomial** distribution. The

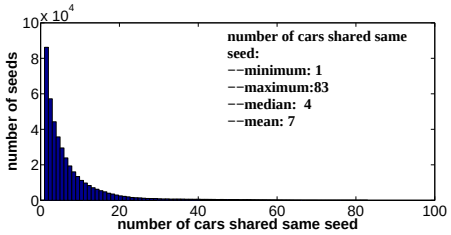


Figure 3.26: The histogram of the cars sharing the same seed.

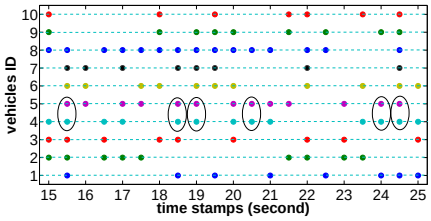


Figure 3.27: The impact of time, space and frequency diversity on seeds collecting.

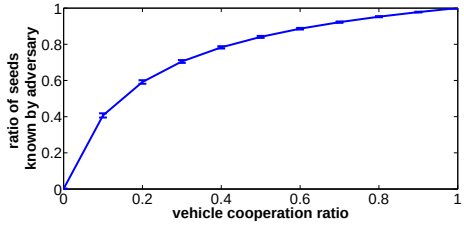


Figure 3.28: The impact of cooperation between adversary and other vehicles (comparing to total seeds collected in the networks).

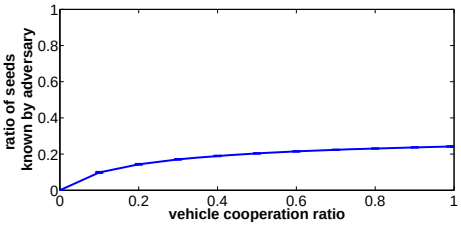


Figure 3.29: The impact of cooperation between adversary and other vehicles (comparing to total seeds generated by RSUs).

expectation number is $3600 * 2/3 = 2400$, and the probability that vehicle receives less than 500 seeds is ≈ 0.

In Fig. 3.26, the histogram chart describes the number of vehicles sharing the same seed. A seed is received by 7 vehicles on average. Most times a seed is only received by one vehicle, and the maximum number of vehicles sharing the same seed is 83. It is easy to see, the number of vehicles sharing the same seed is proportional to the total number of vehicles in the communication area of an RSU when it broadcasts the seed, and is inversely proportional to the number of available channels that can be chosen by the RSU. Recall that even if a seed is known by multiple vehicles include the adversary, as long as one seed is unknown to the adversary, she cannot form the right secret key.

Fig. 3.27 shows seeds collected by 10 vehicles from the 15th to 25th second. Most of them receive very different seeds. As a special case, vehicle 4 and 5 are driving very close to each other, and receive exactly the same seeds at the 15.5, 18.5, 19, 20.5, 24, 24.5th seconds. However, in all the other time, they still receive different seeds.

One of the strategies the adversary can use is to cooperate with multiple vehicles

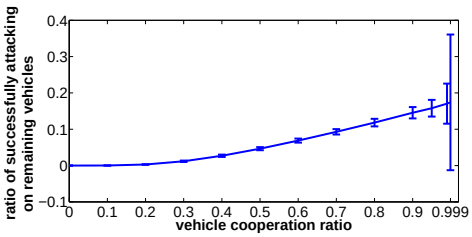


Figure 3.30: The impact of cooperation ratio on an adversary's successful attacking rate.

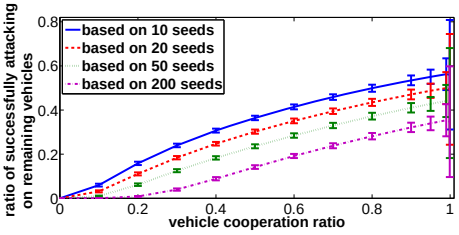


Figure 3.31: Successful attacking rate v.s. cooperation ratio v.s. number of seeds required.

and collect as many seeds as possible. In Fig. 3.28, we compare the seeds that are collected by an adversary through this strategy with the seeds collected by all the vehicles. Although it is helpful to cooperate with other vehicles, at 99.9% cooperation rate, an adversary still cannot guarantee to obtain all the seeds a vehicle received during the simulation. In the figure, we also show the standard deviation along the curve. In Fig. 3.29, the number of collected seeds is compared to the total number of seeds generated by the RSUs. Easy to see that trying to collect more seeds from other vehicles does not work well.

In Fig. 3.30, we study the successful attacking rate by the adversary when it has multiple partners. In this figure, the secret key is assumed generated based on all the seeds a vehicle collected during the simulation. The successful attacking rate is only 0.174 when 99.9% vehicles are cooperating with the adversary. This result is a disaster to the adversary. First, it is impossible to cooperate with so many vehicles in a large vehicular network. Second, the success rate is still very low. Thirdly, the standard deviation is too large to make the success rate reliable. Instead of using all received seeds, in Fig. 3.31, it assumes the secret key is generated based on fix number of seeds. When forming a secret key with more seeds, the success rate of an adversary to break the key becomes low. For example, 200 seeds correspond to 0.0003 for 10% vehicles, and 0.3472 for 99.9% vehicles.

A better strategy for an adversary is to closely follow the target with multiple sets of radio devices monitoring all the channels. Through this way, an adversary can collect all the seeds a vehicle receiving from RSUs. However, the target can also collect a

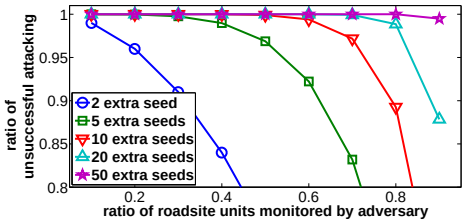


Figure 3.32: By collecting extra seeds from others, a vehicle can make trouble to an adversary even when its coverage ratio is high.

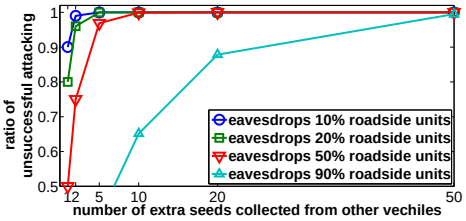


Figure 3.33: When a vehicle collects a large number of seeds from others, the attacking unsuccessful rate of an adversary quickly converge to 1.

seed from other vehicles through the secure V2V communication which is established through the differential approach. Furthermore, one seed will only be transferred once by a vehicle, it is very difficult to obtain the same seed by the adversary, unless the adversary also knows what the other vehicle has. Tracking all the vehicles a target might encounter along the road is impossible. Therefore, in the next two figures, we assume the adversary has deployed a large number of radio devices in the whole vehicular network and try to collect all seeds directly from RSUs.

As shown in Fig. 3.32, at 10% coverage, 5 extra seeds from other vehicles are already enough to make adversary fail all the time. Even when an adversary monitors 90% RSUs, 50 extra seeds can make adversary fail at probability of 99.5%. In Fig. 3.33, it shows that by including a large number of extra seeds collected from other vehicles, the attacking unsuccessful rate of an adversary quickly converge to 1.

Finally, we remind the reader that the proposed scheme also allows a vehicle to form a secret key with seeds collected at different time period. This will further reduce an adversary's ability to re-generate the secret key.

3.5 Discussions

3.5.1 Improvement on Differential Approach

Calculate the durations for data collection period T and τ

The data collection period T is related to the moving speed v of objects, assume the key length L, the wavelength of the wireless signal is λ. As we discussed in the earlier

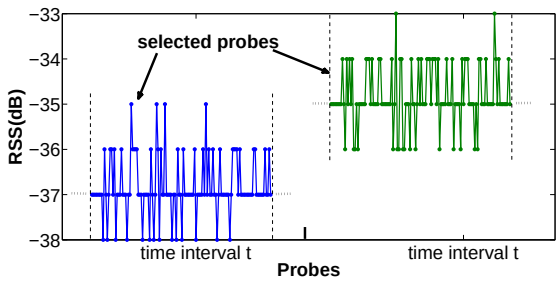


Figure 3.34: It is less likely RSS values in t_i and t_{i+1} period follow the same distribution

sections, in a Rayleigh channel, roughly every $\frac{\lambda}{2}$ location varies could generate an independent secret bit. Thus we could set $T \approx \frac{\frac{\lambda}{2} * L}{v} * \alpha$, where α is a predefined factor of safety parameter. For example, $\alpha = 1.2$. However, if there are not enough bits generated during T, Alice and Bob could always extend the time period until enough secret bits can be generated without reducing the key length L.

The value τ is needed for fixed interval approach only. It could be set to a number which results in the sample interval close to $\lambda/2v$ in the unit of time or smaller in case channel varies are bigger than expected.

Calculate the moving average width d

According to previous experiment, moving average width d can also be set according to $\lambda/2v$.

Estimate noise level ϵ

The small fluctuation parameter ϵ is an experience parameter. Since it is mostly related to the device itself and surrounding environment, most times it could be estimated in advance. Furthermore, when the probe rate is high, we could use large moving average width d, and the value of ϵ becomes not so critical. The effect of small fluctuation becomes very small after moving average process.

Non-parameter method when probe rate is high

The proposed differential approach still needs to estimate a parameter ϵ which is related to the range of small fluctuation. A non-parameter method could be implemented if

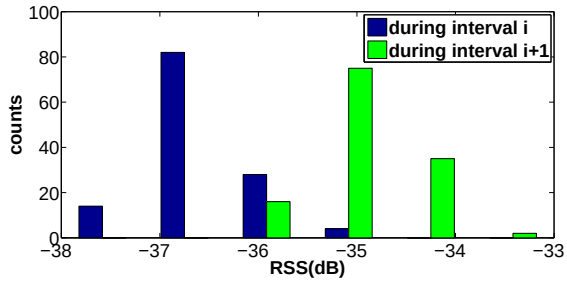


Figure 3.35: The histogram of the given example.

the maximum probe rate is much higher than Doppler shift rate.

We assume the fluctuation value (we should call it random error ε_i in the following text) is either uncorrelated random variables or independent normal random variables (ε_i are normally distributed)[5]. This assumption is reasonable when both Alice and Bob are static.

When the probe rate is really high, we assume we can collect n number of RSS values during an extremely short time period in which the mobile node can be assumed to be static. From each time interval $t_i = \frac{\lambda}{2v}$, we collect n RSS values at similar positions, for example in the middle of each t_i. In the case if there are enough channel varies between these two timestamps, the two RSS value sets might follow different distributions especially with different mean μ, as shown in Fig. 3.34.

The example shown in Fig. 3.34 tells us that these two sets of n are drawn from different distribution. However, in some cases, it might be hard to distinguish them. At that time, we use chi-square test.

let N_{ij} be the number of appearances of a particular RSS value j in the ith n probes. We define the hypothesis test:

H_0: $N'_{ij}s$ and $N'_{(i+1)j}s$ are drawn from same source

H_a: $N'_{ij}s$ and $N'_{(i+1)j}s$ are drawn from different sources

(case 1:) If the total number of probes is the same in two sets, then the chi-square

[5]Although RSS values we observed usually are quantized number which may destroy the normally distributed property, we ignore this factor since it doesn't affect the results much.

statistic is

$$\chi^2 = \sum_j \frac{(N_{ij} - N_{(i+1)j})^2}{N_{ij} + N_{(i+1)j}} \tag{3.7}$$

(case 2:) otherwise, we use

$$\chi^2 = \sum_j \frac{(\sqrt{N_{i+1}/N_i}N_{ij} - \sqrt{N_i/N_{i+1}}N_{(i+1)j})^2}{N_{ij} + N_{(i+1)j}} \tag{3.8}$$

where

$$N_i \equiv \sum_j N_{ij} \tag{3.9}$$

and the significance probability is calculated as:

$$p = f(\chi^2|v) = \frac{\chi^{(v-2)}e^{-x/2}}{2^{v/2}\Gamma(v/2)} \tag{3.10}$$

where v is the degree of freedom which is equal to the number of j in case 2 and number of j minus 1 in case 1. The Γ is the Gamma function.

For the example in Fig. 3.34 or its histogram format as shown in Fig. 3.35 which belongs to the case 1 (both sets have 128 samples), we got $\chi^2 = 200.0829$ and $p = 1.3433e^{-041}$. This indicates that the null hypothesis,H_0 is rather unlikely.

Finally, the good sample size also can be roughly estimated through iterations, as shown below (assume $1 - \alpha = 0.90$):

1. choose a margin of error E, use a small sample size to calculate mean μ and sample variance s^2.
2. compute preliminary sample size $n^* = (z_{\alpha/2} * s/E)^2$.
3. compute sample size $n = (t_{df,\alpha/2} * s/E)^2 = 1$, where $df = n^* - 1$.
4. if n and n^* have large difference, then repeat last step with $n^* = n$.

Let's start with an example, we choose the first 20 samples of the interval i from the Fig. 3.34. We got $\mu = -36.95$, $s^2 = 0.3658$, assume we pick margin of error $E = 0.1$, then after calculation, we got $n^* = 99$ and $n = 101$. Since n^* is similar to n, we claim a sample size of 100 is good enough.

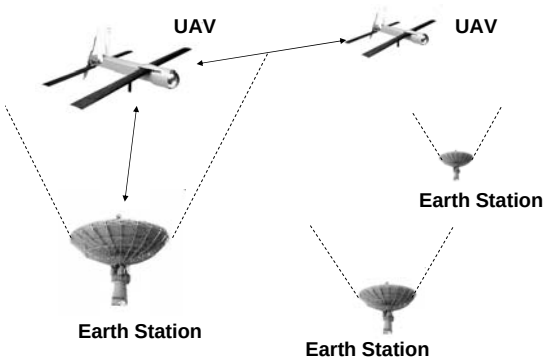


Figure 3.36: An UAV system.

Non-parameter method when probe rate is low

When the probe rate is low, to avoid using any parameter, we could use piecewise linear regression method to remove the parameter ϵ from the sequence of RSS samples:

$$y = a_0 + b_0 x + b_1(x - k_1)\delta_1 + ... + b_p(x - k_p)\delta_p \tag{3.11}$$

where $k_1 < k_2 ... < k_p$ are joint points and $\delta_p = 0$ if $x \leq k_p$ and $\delta_p = 1$ otherwise. However, this method requires more computational capability at each end user.

3.5.2 Application Extension

The proposed secret key agreement schemes can be extended to other applications such as the well known Unmanned Aerial Vehicle (UAV) System. A UAV is defined as a powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload [63]. On UAVs, the resources are even more limited than in vehicular networks. The proposed algorithm can help establish secret keys for secret communication between unmannered aircraft pairs, and between the unmannered aircraft and headquarter through earth station (as shown in Fig. 3.36). For communications between aircrafts, the spatial decorrelation property is different from V2V case. In UAV system, line-of-sight may exist between aircrafts. In such case, using Bessel function versus wavelength to describe the spatial decorrelation property (as in Equation (1)) may no longer be proper. Instead, longer

distance between the legitimate aircraft and adversary's aircraft is required to create enough decorrelation. However, this requirement is reasonable for UAV system, since aircrafts are moving much faster than vehicles and it is much difficult for an adversary to closely follow a target aircraft.

3.6 Related Work

Traditional key distribution protocols rely on infrastructure with online trusted third parties (TTP), such as the well-known Kerberos [9] scheme and Otway-Rees protocol [10]. Kerberos is based on the model presented by Needham and Schroeder [64] in which an online server known as key distribution center (KDC) shares a pre-initialized key with each user. Later on shared secret keys can be created among users by KDC sending each user a session key encrypted with users pre-initialized key. In Otway-Rees protocol, efficient mutual authentication is achieved through a third party too. However, in V2V communication, there is no central authority can be relied on as we discussed before. Furthermore, since the node mobility is unrestricted, the topology may be unpredictable making central authority assumption infeasible.

Diffie and Hellman discussed a public key distribution system and how it can be transformed into a one-way authentication system [7]. Their scheme, Diffie-Hellman key exchange, was based on the apparent difficulty of solving discrete logarithm problem [65]. An even earlier contribution is Merkle's "puzzle" [66]. In "puzzle" X sends N puzzles to Y. Each puzzle combines one ID and one key and it requires $O(N)$ effort to break. Y then randomly picks one puzzle to obtain the ID and the key. Next, Y informs X the ID and both X and Y will know the key. However, Z who also has all the N puzzles has to spend, on an average, $O(N^2)$ efforts to determine the key. Other well-known public key algorithms include RSA [67] based on factoring, Elliptic curves [68] based on Elliptic Curve Discrete Logarithm Problem, and Digital Signature Algorithms (DSA), etc. Cost-effective processors with limited computational abilities make public-key cryptography almost impractical for embedded intelligence and ubiquitous computing applications, even without power consumption considerations.

In [11], Rolf Blom presented a symmetric key generation system (SKGS), where each pair of users share one master key that is distributed at the start up time by a key generation authority. This master key is used to generate session keys later on. However, a network with n users implies that each user must have access to $n-1$ keys. If n is large, it becomes impossible to store all keys securely. The contribution of Rolf Blom is on finding a class of symmetric key based on an MDS code [69], for which the amount of secret data needed by each user is very small. On the other hand, a certain minimum number of K users have to cooperate to determine the keys used by other user pairs. Eschenauer and Gligor [12] proposed a key management scheme that relies on probabilistic key sharing among the nodes of a random graph and uses a simple shared-key discovery protocol for key distribution, revocation and node re-keying for large-scale distributed sensor networks. Chan [13] introduced Distributed Key Pre-distribution Scheme (DKPS) which is a fully distributed and self-organized key pre-distribution scheme without relying on any infrastructure support. The basic idea is based on Privacy Homomorphism (PH), introduced by Rivest et al. [70]. In DKPS, each node randomly picks keys from the universal set (which is publicly known), followed by a probability method to construct cover free family (CFF) in a distributed manner and security key discovery phase through modified Rivest's scheme (MRS), it eliminates the trusted third party from key pre-distribution scheme. Eschenauer and Gligor [12] proposed a key management scheme relies on probabilistic key sharing among the nodes of a random graph and uses a simple shared-key discovery protocol for key distribution, revocation and node re-keying for large-scale distributed sensor networks. Based on Eschenauer-Gligor's work, Chan et al. proposed a q-composite random key pre-distribution scheme [71] in which by increasing the value of common keys for each node pair, network resilience against node capture is improved. Using the giant component analysis [72], Hwang and Kim [73] claimed that they can further improve the security or the scalability of the previous pre-distribution schemes. Other key pre-distribution schemes are [74–80]. However, the strict requirement for pre-distribution might not be available all the time. For example, in vehicular networks, the cars (sharing no prior secret information) may just meet on the spot where there is likely to be no single

trustable proxy or TTP for key pre-distribution.

Wyner's wire tap channel [81] showed that perfectly secure communications is possible if Eve's channel is a degraded version of Bob's by hiding the message in the additional noise level seen by Eve. This work was extended by Csiszar and Korner to more general broadcast channel [82]. Later, in [83–85], Maurer and Wolf theoretically showed that even Eve's channel is better than Bob, secret key agreement is still possible in a satellite communication channel.

When two antennas A and B have no non-linear components radiate identical signals, the outputs of the antennas due to their excitation by the signal originating at the other antenna will also be identical. This behavior, known as the reciprocity theorem, arises from the reciprocity of the radiating and receiving patterns of antennas and applies when the medium between the antennas is linear and isotropic [86]. Based on the theorem, Hershey et al. [87] first presented the concept of using physical layer characteristics for key management. More recent work can be found in [55, 58, 88–90]. In [55], the authors use level-crossings and quantization to extract bits from correlated stochastic processes and shows that it is possible to achieve strong information-theoretic security by extracting secret bits from a multi-path fading channel at a rate of about 1 bit per second. In more detail, two legitimate users use the channel statistics to determine scalars, q_+ and q_- that serve as reference levels for quantizing. A secret key bit 1 or 0 is agreed if enough channel magnitude measurements are higher than q_+ or lower than q_- on both sides. However, the poor efficiency of this key agreement protocol may lose many interesting for real world implementation. In [88], the authors exploit the differences of arrival time among the direct wave and the delayed waves as the shared information between authorized users in a UWB Systems. Using Espar (Electronically Steerable Parasitic Array Radiator) antenna to measure the RSSI, [58] generates secret bits based on the median value of the RSSI profiles. In Zang et al.'s key dissemination protocol [89], channel state is used to XOR with secret key oriented from one of the users. Instead of key generation scheme, Xiao et al. [90], develop a wireless authentication protocol based on the channel characteristics. All above protocols require a relatively accurate measurement on end users. However, consider the difference

between each individual communication device, accurate and uniform threshold may not be available at different end users even with the channel reciprocity theorem.

Finally, some researchers started to exploit multi-channel characteristic of wireless devices to help improving security recently. Interested readers can refer [14, 59, 91].

3.7 Conclusion

In this chapter, we have presented a set of key agreement schemes to help establish secure communication channel in vehicular networks for both V2V and V2I modes. The proposed algorithm is based on two novel key agreement schemes: differential and channel hopping key agreement schemes and their extensions. It takes advantage of physical layer characteristic of a wireless channel and the natural characteristic of vehicular sensing networks. Specifically, besides the channel reciprocity property, it makes full use of all kinds of diversity properties existing in the channel and the vehicular networks. The unconditional security of the proposed algorithm is guaranteed by two factors: first is the well-known spatial decorrelation property and the second one is the complexity of the vehicular networks and individual randomness. Numerical evaluation through simulation and emulation has been studied.

Chapter 4

Privacy: VTL Zone-Based Path Cloaking Algorithm

Traffic engineering applications often benefit from collecting vehicle GPS traces in well-defined locations through vehicular sensing networks. As a motivating example, we will consider a traffic signal performance evaluation technique which requires GPS traces of vehicles traversing intersections. Since these applications do not require vehicle identity information, they are good candidates for data de-identification techniques. Prior techniques can either provide data from specific locations or guarantee a high degree of anonymity under low traffic conditions but do not achieve both. In this chapter, we propose a virtual trip line zone-based path cloaking algorithm that combines these features. Zones where data should be retained can be predefined over the intersections of interest and the path cloaking algorithm uses entropy-estimates to decide whether the data can be revealed. Result obtained from a traffic simulator show that the application success rate increased from 39 to 82% compared to a zone-unaware path cloaking algorithm, while achieving a similar degree of privacy.

4.1 Introduction

Location traces obtained through the Global Positioning System (GPS) are a promising source for extracting many types of transportation data. GPS traces from taxi fleets, delivery trucks, or mobile phones, for example, are already used to infer traffic congestion on highways [26]. This concept of estimating traffic system states from location traces generated by mobile sensors in individual vehicles, promises substantially lower costs, since it does not depend on infrastructure installed along roadways [15]. Within this concept, many other transportation applications are possible. Throughout this chapter, we will consider one motivating application: Mobile Sensor as Traffic Probes (MSTP),

which raises novel privacy challenges. This application focuses on state/performance estimation of signalized road intersections such as estimating real time delays, arrival volumes, and vehicle queue lengths. This has been a long-standing challenge in the transportation community especially when wide-area arterial networks are considered.

Recognizing that naively anonymized (i.e., simply omitting names, vehicle identifiers, etc.) location traces can often easily be re-identified, researchers have proposed several solutions [92, 93]. A mix-zone [24, 25] defines a particular area where locations cannot be revealed so that an adversary cannot trace the movements of vehicles across these mix-zones. This works well when the traffic density is high, but provides no guarantees when there is less traffic than anticipated. Furthermore, most mix-zone and related research [94, 95] have focused on finding the best location and size of zones for protecting privacy. However, our target application also introduces strict requirements on this location: data is only of interest around intersections. The uncertainty-aware path cloaking algorithm [17] also segments traces but was explicitly designed to achieve a defined level of privacy under all traffic conditions. It achieves this by filtering out points from the location traces whenever vehicles are trackable for a longer period of time, thus it provides no control of where data is omitted. The challenge in our motivating application lies in the requirement that location traces are needed only across intersections–exactly the area where mix-zones are often placed to hide information. While there are some safety critical applications that require data virtually everywhere, for most applications, data should only be provided in the limited areas where it is needed. This is also consistent with the basic principle of minimizing the information leakage.

In MSTP, GPS information as well as the speed of a vehicle is collected during a Virtual Trip Line (VTL) [26] zone which covers an intended intersection in order to analyze the intersection delay patterns, arrival volumes, and queue lengths. However, disclosing location information of a vehicle leads to potential threats to the location privacy of the user which has been shown in many research work [16, 17]. Furthermore, for applications similar to MSTP, if the trace data of multiple VTL zones can be linked together to identify an individual user, it can breach the privacy of the user as well.

We develop a zone-aware privacy algorithm to filter location traces, which still takes into account traffic density and uncertainty. This allows the algorithm to release location traces only in the intersection zones where data is needed by the application, yet still offer a fixed degree of privacy independent of traffic density. This is, to our knowledge, the first algorithm that combines these two aspects. More specifically, the algorithm seeks to achieve unlink-ability between released traces from any two different zones. We refer to the zones as VTL zones in the remainder of the chapter, since their locations can be marked by two or more Virtual Trip Lines (VTL) [26]. To be able to adjust the filtering to traffic density, the algorithm needs to be aware of all vehicles' location traces. While we will simply refer to a proxy server with access to these traces, we note that there are multiple different usage scenarios for such an algorithm. Even if no proxy server exists, the algorithm could be useful to anonymized data before it is stored or before it is transmitted to another party.

The rest of the chapter is organized as follows. In section 4.2, we provide more details on the motivating MSTP application. In Section 4.3, we describe our system and adversary model. We describe the main zone-aware path cloaking algorithm in section 4.4. In section 4.5, we evaluate the performance through simulation. Section 4.6 gives review on related work. Section 4.7 discusses a distributed solution. The whole chapter is concluded in section 4.8.

4.2 Traffic Signal Performance Evaluation

In this section, we briefly go through a traffic signal performance evaluation application called Mobile Sensors as Traffic Probes (MSTP) model. This application uses of VTL data from GPS cell phones for arterial performance measurements (PMs). VTLs are predefined, virtual locations on roadways. When crossing a VTL, the mobile phone equipped in a vehicle will report its speed. Ban et al. [15, 96] showed how to use intersection travel time to estimate real time arterial PMs including real time delay patterns and queue lengths. This can be shown in Fig. 4.1 for the real time delay pattern of an intersection, which is the travel time an imaginary vehicle will experience when arriving at the intersection at a given time. The figure shows a typical signalized

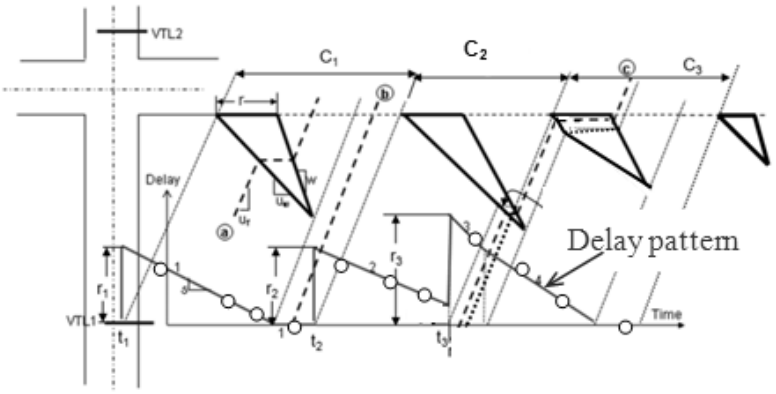


Figure 4.1: Intersection delay pattern estimation

intersection. Two VTLs are deployed upstream (VTL1) and downstream (VTL2) of the intersection. Under certain assumptions, we can use the bold solid triangles (or trapezoids) in the figure to represent how queue forms and dissipates based on the shockwave theory [97, 98]. The horizontal part of a triangle represents the duration of red time. As shown by the trajectories of vehicles (dashed lines), if a vehicle approaches the intersection in red time or if the queue length is not zero (e.g. trajectory a in the figure), the vehicle will join the end of the queue first and thus be delayed. The delay encountered by the vehicle is the horizontal part of trajectory a. Otherwise, if a vehicle arrives during green time and there is no queue (e.g. trajectory b), the vehicle will pass the intersection with no delay. By analyzing the geometry of the triangles, we can construct the theoretical delay curve as shown in the bottom of Fig. 4.1. The curve is piecewise linear and contains critical points, i.e. discontinuities and non-smoothness, that indicates traffic signal changes (such as the start of red) and traffic state changes (such as a queue is fully discharged) respectively. Mobile sensors can provide samples of intersection travel time, shown as circles along the delay curve in Fig. 4.1. These sample travel time can be used to identify the critical points and further to estimate the real time delay or queue length. In [15, 96], this is done via estimation algorithms to fit the sample travel time to piecewise linear curves.

Fig. 4.2 shows the results of applying the estimation algorithm to a field test in the Bay Area in California [96]. In the figure, the asterisks along the piecewise lines (i.e. the estimated delay curves) are the observed travel time and the plus signs at the bottom are

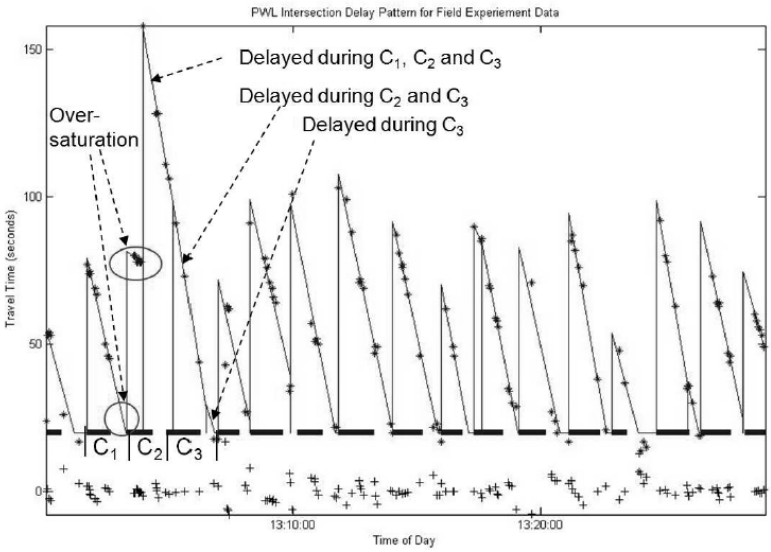


Figure 4.2: Field test results.

the errors. It is clear that the delay pattern can match well the observed sample travel time. Knowing the pattern will help identify traffic conditions, e.g. over-saturation (i.e. vehicles cannot be fully discharged within a cycle) as indicated in Fig. 4.2, or to estimate real time queue lengths [15].

When only a single intersection is considered, collecting travel time between the upstream and downstream VTLs, even the complete vehicle traces between the two VTLs, poses minimal privacy concerns because the distance between the two VTLs are usually very short (at most a few hundred feet). However if the above intersection PM estimation algorithm is applied to all intersections in a network, we need to collect travel time between the upstream and downstream VTLs of all intersections. This can result in sever privacy concerns as one may be able to re-identify a vehicle based on the arrival time at each VTL via the following steps. First, intersection delays (i.e., those between the upstream and downstream VTLs of an intersection) are the major component of arterial delay. The traverse time for a vehicle from the downstream VTL of an intersection to the upstream of the next intersection is usually the free flow travel time of the road (i.e., distance divided by the speed limit). Then if the arrival time of sample individual vehicles are known at the two VTLs, one may be able to analyze the arrival time and the traverse time to figure out which pair of arrival time (one at each

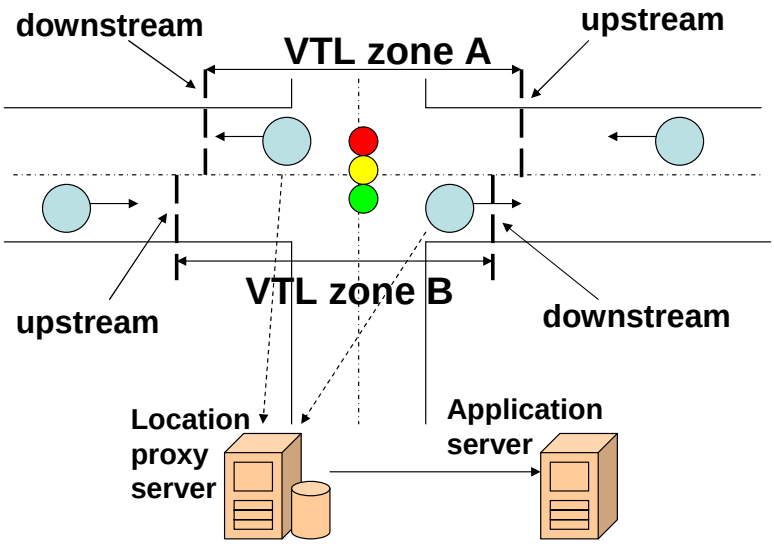


Figure 4.3: VTL zone-based privacy model system architecture.

of the two VTLs) belongs to the same vehicle. Since the intersection travel time are already collected for individual vehicles, one can repeat the above step for subsequent intersections and identify the same vehicle that passes all these intersections. As a result, the trace of a vehicle for an extended distance or a period of time may be reconstructed if intersection travel time for all intersections are collected at VTLs.

To overcome this issue and ensure privacy of individual vehicles, we need to design specific cloaking algorithms to determine whether VTL travel time of a vehicle can be released at a particular VTL location. This requires a zone-based cloaking algorithm.

4.3 Concept, Model and Problem

In this section we discuss the concept of VTL zone, our system and adversary model, and the privacy metric.

4.3.1 VTL Zone

A Virtual Trip Line (VTL) zone is an area of the road network between two virtual trip lines. VTL was originally introduced to allow an application to specify where vehicles should provide location updates. An update should be sent only when a vehicle is crossing a VTL. The VTL zone extends this concept to a continuous part of the road

network. Inside the zone, a vehicle provides its entire location trace. Fig. 4.3 illustrates this concept. A MSTP application, such as traffic monitoring at intersection, can benefit from receiving longer location traces before the traffic light and shorter traces after traffic light from vehicles. Because of this, two VTL zones, A and B, are shown on the same road to indicate different bounds. What are unshown in this example are the additional zones that could be deployed on the crossing road.

Conceptually, a VTL zone can be considered as the inverse of a mix-zone. Applications specify where they need data rather than specifying where data can be suppressed. We believe that this inverse encourages adherence to the privacy-by-design principle of minimal data collection, since developers have to expend effort to increase data collection rather than expending effort to suppress data collection.

4.3.2 System Model

Consider current infrastructure existing in the traffic monitoring system and the high on-road density, we propose a centralized system architecture as shown in Fig. 4.3. The centralized model is useful because: 1) historical data and long-term storage are necessary in many applications, for example, help government making road planning decisions; 2) based on existing infrastructure, centralized architecture is more efficient and easier to be implemented. For example, Google Maps for mobile [99]. Since long-term storage is more sensitive than in-memory processing, it's desirable to sanitize them before store them. Similar to [17], we assume there is a location proxy server between MSTP application server and the vehicles. The proposed VTL zone-aware path cloaking algorithm is running on the location proxy server. The application server is assumed to be untrusted, while the location proxy server is a trustworthy entity. Individual vehicle uploads its GPS trace and other data to the location proxy server with or without identification[1]. However, all identifiers must be removed before the data are passed to the application server. The secure of the transmission between vehicle and proxy can be guaranteed by using some cryptographic methods, such as the method

[1]Some applications may need authentication on the data source through ID.

described in [25], thus it is not our concern in this work. We assume the location proxy server is secure through techniques such as contractual obligations, privacy legislation or their combinations. As theoretical people always prefer a distributed solution, we also discuss the modification needed, the issues that might be introduced, and the alternative solutions if there is no location proxy server available in section 4.7.

4.3.3 Adversary Model

The objective of the adversary is to track the vehicles in the road network and breach the location information of their users. It is known that the longer an adversary can track a vehicle, the higher probability that the trace will contain a sensitive location and thus the privacy of the user will be disclosed. The adversary can be an internal part of the application server or some third party which is able to access all data from the location based service provider. Since the location information of a vehicle is collected exclusively from a VTL zone that only covers one intersection, individual VTL zone trace data of a vehicle is not a big concern here. However, if the adversary is aware that several sets of traces across multiple VTL zones are all from one vehicle, it may link the zones and obtain large trajectory information of the vehicle. Hence the chance to identify a user and break the privacy will be much higher. By using some statistical methods, this is possible even when anonymous technique is used [19, 20].

4.3.4 Problem Statement

We use Entropy $H = -\sum p_i \log p_i$ to describe the tracking uncertainty, in which p_i is the probability of a set of GPS traces from a particular VTL zone belonging to vehicle i. Lower values of H indicate more certainty or lower privacy. As we will see Later, H will be represented in a more specific format in the algorithm description section. Our target is to eliminate link-ability between any VTL zone pairs while keep as much trace data as available for the location based service provider. Therefore, the adversary is unable to identify an individual user through continuous VTL zones. On the other hand, the data accuracy is high enough for the application server with merely a small number of trace data unrevealed.

4.4 The VTL Zone-Aware Cloaking Algorithm

The goal of this cloaking algorithm is to eliminate link-ability between any VTL zone pairs *regardless of traffic density* while keeping as much trace data as possible to be used by the applications. This means, that the adversary should be unable to track an individual user through multiple VTL zones even if the traffic density becomes very low. It also means that when increased traffic density leads to a naturally higher level of privacy, the algorithm should filter less information.

To this end, we pair the VTL zone concept with a cloaking algorithm, which monitors tracking uncertainty and removes traces only until a defined privacy level is achieved. This proposed algorithm includes three major steps. As discussed in previous section, user privacy is in jeopardy if an adversary can reliably recognize that a trace from VTL zone A and a trace from VTL zone B both are from the same user. To link two traces, the adversary can estimate the likelihood that a user would have taken the path leading from A to B. We refer to this as the path likelihood. The adversary can also calculate the time interval between the traces from A and B and estimate the likelihood that the A-B trip would take this amount of time. We refer to the latter as the travel time likelihood.

Our cloaking algorithm therefore operates in three steps. The first two are to derive models for travel time likelihood and path likelihood. The third step is the actual cloaking step. Here, the algorithm evaluates the likelihoods for each pair of traces and compares them with other possible links of traces, which results in the tracking uncertainty. Only if the tracking uncertainty exceeds a threshold the pair of traces can be disclosed to the application.

4.4.1 Travel Time Likelihood

Given two VTL zones, the algorithm characterizes the travel time likelihood based on an empirically derived distribution. We assume that the adversary has no specific knowledge about the individual user under consideration, so the likelihood taking a certain amount of time can be derived from the empirical travel time distribution of

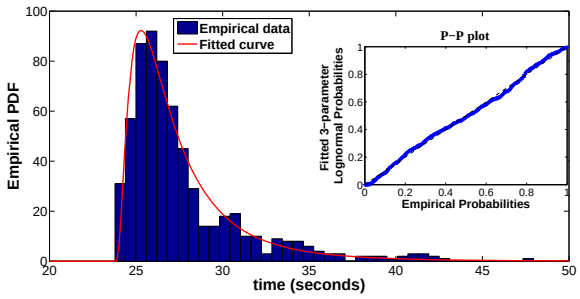


Figure 4.4: A 3-parameter log-normal distribution fits the empirical data well.

the population.

By analyzing simulation data that will be described in more details in section 4.5, we find that the travel time duration t between two zones follows a 3-parameter log-normal distribution, as shown in Fig. 4.4[2].

The probability density function of a 3-parameter log-normal distribution is:

$$f_T(t) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(t-\theta)} e^{(-\frac{(log(t-\theta)-\zeta)^2}{2\sigma^2})} & for\, t > \theta \\ 0 & for\, t \leq \theta \end{cases} \tag{4.1}$$

where θ is the threshold parameter, σ is the shape parameter and ζ is the scale parameter. The 3-parameter log-normal distribution can be fitted by Least Square Estimation (LSE) as shown in [101] in which the cumulative distribution function of 3-parameter log-normal

$$F_T(\hat{t}) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^{\hat{t}-\theta} \frac{e^{\frac{-(\ln(\delta)-\zeta)^2}{2\sigma^2}}}{\delta} d\delta \tag{4.2}$$

is used as input function for LSE:

$$\hat{b}_{opt} = \underset{b \in B}{\operatorname{argmin}}\, S(\hat{t}_1, ..., \hat{t}_n | b) \tag{4.3}$$

$$S = \sum_{i=1}^{n} w_i.(F_T(\hat{t}_i|b) - v_i)^2 \tag{4.4}$$

where $b = (\theta, \sigma, \zeta)$ and $\hat{t}_i$ are sorted historical travel time samples. $v_i = \frac{i-0.5}{n}$, and $w_i = \frac{1}{\sqrt{(v_i * (1-v_i))}}$ are weights which compensate for the variance of the fitted probabilities which is the highest near the median and lowest in the tails.

[2]Similar observations are also shown in [100]. Here, we emphasize using 3-parameter log-normal to describe the travel time distribution instead of general log-normal distribution.

This distribution is fitted for each pair of VTL zones and used to derive travel time likelihoods by the algorithm.

4.4.2 Path Likelihood

Since not all paths through the road network are equally alike, the algorithm also takes into account path likelihoods. The path likelihood $\rho_{a\to b}$ from a to b is defined empirically as

$$\rho_{a\to b} = \frac{\sum_{d\in D} k^d_{a\to b}}{\sum_{d\in D} k^d_a} \tag{4.5}$$

where d is the vehicle ID, D is the complete set of vehicle IDs, k^d_a is the number of times the vehicle d passes by a in the collected data set and $k^d_{a\to b}$ is the number of times vehicle d passes by both a and b in sequence.

Furthermore, the parameters of travel time distribution and the path likelihood of a VTL zone pair can be periodically updated at the location proxy server. For example, we calculate the path likelihood $\rho_{a\to b}$ from VTL zone a to b as an Exponential Moving Average (EMA) to give less weight to outdated data.

$$\rho_{a\to b} = \beta * \rho_{new} + (1-\beta) * \rho_{old} \tag{4.6}$$

where β is a predefined parameter. Since city traffic usually varies dramatically from peak hour to off hour, it makes sense to incorporate such updates for different time periods.

4.4.3 Trace Release

In order to decide if a set of trace samples from a vehicle can be released, the location proxy server needs to calculate the tracking uncertainty of this trace. The algorithm iterates over all VTL zones and for each VTL zone over all vehicles that have their latest released traces in that VTL zone. For a single trace, the tracking uncertainty is then (assume the current VTL zone is c):

$$H = -\sum_{v\in V\setminus c}\sum_{d\in D_v} p^d_{v\to c}\log p^d_{v\to c} \tag{4.7}$$

where V is the whole VTL zone set, D_v are the vehicles which have their latest disclosed data trace in VTL zone v. As shown in equation 4.8, $p^d_{v\to c}$ is the probability that the trace in c is generated by the same vehicle of trace d in zone v (after normalizing)

$$p^d_{v\to c} = \frac{\rho_{v\to c} * p_T(t^d_{v\to c})}{\sum_{v'\in V\setminus c}\sum_{d'\in D_{v'}} \rho_{v'\to c} * p_T(t^{d'}_{v'\to c})} \tag{4.8}$$

where $t^d_{v\to c}$ is the travel time assuming that vehicle d traveled from zone v to c (i.e., $t^d_{v\to c}$ is the time difference between the end of trace d in v and the beginning of trace under consideration in c). $p_T(t)$ is the discrete version of $f_T(t)$.

The trace in c can be released if the tracking uncertainty $H > \alpha$, where α is a specified confusion level that characterizes the degree of privacy. For applications which use coarse time unit, such as minute, multiple vehicles may enter the same zone within the same time period. Then below equation can be used to adjust the entropy value.

$$H_m = H - \log m \tag{4.9}$$

where m is the number of vehicles enter the zone at the same time.

When the mean travel time between two VTL zones is very large, the path likelihood tends to be very small compared to other possible source zones. Therefore, we believe it is safe to disregard those zone pairs. It will only have a minor effect on the degree of privacy while promising significant gains in computational efficiency.

4.5 Simulation Evaluations

In this section, we evaluate the proposed VTL zone-aware path cloaking algorithm through simulation. The tracking uncertainty α is set to be 0.95[3]. We compare the proposed VTL zone-aware path cloaking algorithm with a zone-unaware path cloaking algorithm from Hoh et al. [17].

First we compare both algorithms based on the privacy level can be achieved under same adversary model. Next, we compare the data quality obtained through different

[3]To see the impact of different uncertainty levels, please refer to our paper [102].

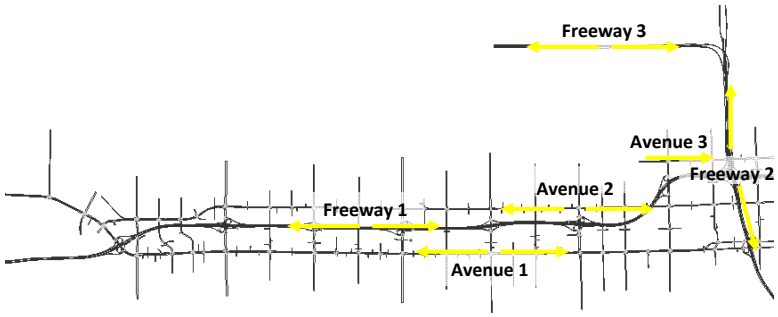


Figure 4.5: On this map total 102 VTL zones are deployed for the MSTP application.

privacy algorithms. And finally, by analyzing the simulation results, we also show computation overhead can be largely reduced by setting proper time threshold.

4.5.1 Traffic Simulator

The simulation is based on the traffic data generated from Liu and Jabari's Paramics Traffic Simulation model [103]. As shown in Fig. 4.5, timestamped location traces are collected from a sub-network of the SR41 corridor located in the city of Fresno, CA. The corridor comprises a stretch of the SR41 freeway and three parallel arterials, with a total of over 90 signalized intersection and 15 ramp metering controllers. It is approximately 16 miles in length and 4 miles in width. Overall, the network includes 20 arterials and 3 freeways. We marked VTL zones on the three avenues shown in Fig. 4.5 for all signalized intersections, which yields a total of 102 VTL zones. Data are collected for about 1 hour. We have implemented the algorithms in Java (except for the extensions described by equations 6 and 9). On one Intel(R) Xeon(R) 2.66GHz core, the simulation with one data set (full density of 1 hour data) takes about 13 minutes to run (without optimization).

4.5.2 Privacy Results

We assume a powerful adversary who can distinguish every individual record and knows the corresponding vehicle ID at one of the VTL zones. The adversary also has access to the exact empirical travel time and path likelihood data. The strategy of the adversary

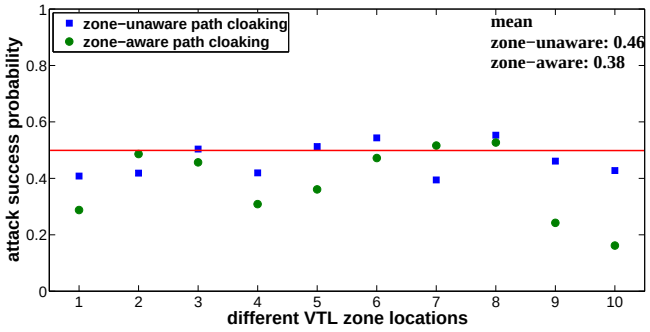


Figure 4.6: The adversary successfully identify the trace at almost the same probability under two privacy algorithms.

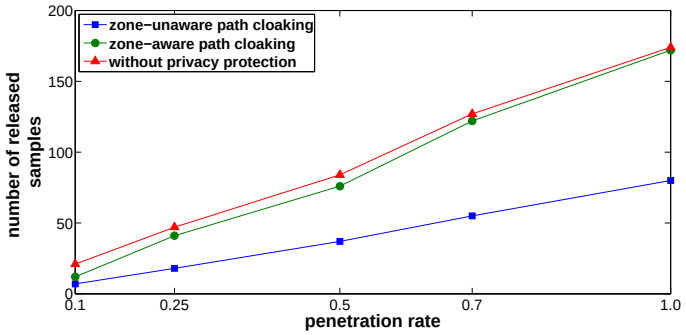


Figure 4.7: The proposed zone-aware algorithm releases more samples than zone-unaware algorithm does.

is to link the most likely traces.

The target of that adversary is to identify the records from the next directly connected VTL zone. As shown in Fig. 4.6, two schemes results in similar attack success probability over ten different VTL zones. In terms of mean value, the zone-aware algorithm is slightly better.[4]

4.5.3 Data Quality Results

Fig. 4.7 shows that the proposed algorithm releases more samples than the zone-unaware algorithm does. Note, that we only count location samples inside the VTL zones, which matter to the application. Overall, the amount of released location samples

[4]Since the uncertainty value of 0.95 is quite moderate, it is possible to see some points located above 0.5. However, by increasing the uncertainty threshold, both schemes show the decreasing in attack success rate.

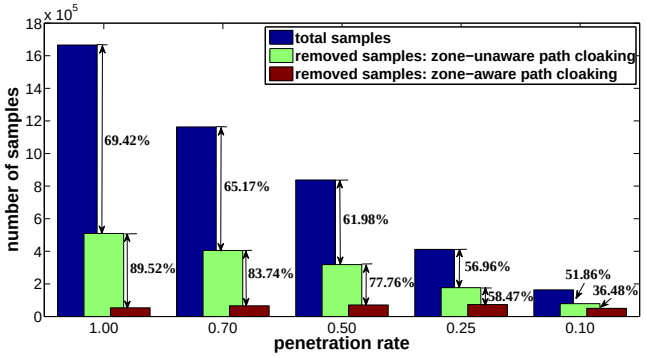


Figure 4.8: The impact of penetration rate. At every penetration rate, we also show two numbers. The upper one represents the ratio of remaining samples after zone-unaware cloaking algorithm comparing to the original number of samples, the lower percentage represents the ratio of remaining samples after zone-based comparing to the zone-unaware cloaking algorithm. Reader can also indirectly obtain the ratio of remaining samples after zone-based cloaking algorithm comparing to the original samples by multiplying these two numbers.

is much closer to the ideal case, labeled 'without privacy protection' in the figure, where no samples are suppressed. The penetration rate indicates the density of the traffic condition. For penetration rate less than 1, we generate the simulation data by randomly keeping vehicles based on the rate from the original (penetration rate=1.0) data.

Fig. 4.8 shows the impact of penetration rate on the whole data set which includes both location samples inside and outside the VTL zones. In all cases, the proposed algorithm keeps more samples than the zone-unaware one does. For example, when the penetration rate is 50%, Hoh's algorithm removes 38.02% samples, while our algorithm removes $(100-61.98)\% * (100-77.76)\% = 8.45\%$ samples. Even when the penetration rate is 10%, the proposed algorithm still outperforms the zone-unaware algorithm by releasing 36.48% more samples[5].

The increased amount of available data also leads to improved traffic monitoring application performance. As shown in Fig. 4.9, the success rate[6] is high and close to ideal with the results from the proposed scheme. Due to the smaller number of released

[5]The 10% penetration rate in the simulation is very low. Within one hour of observation, only 2388 vehicles appear in an area of $16 * 4 = 64\ mi^2$.

[6]Measured in terms of the ratio of the cycles where the model has enough sample data to compute the traffic condition over all the cycles.

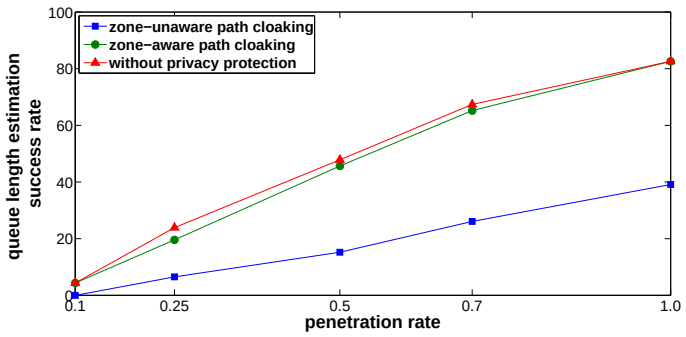


Figure 4.9: The proposed zone-aware algorithm which has more sample being released results in better performance on a traffic monitoring model (MSTP [15]).

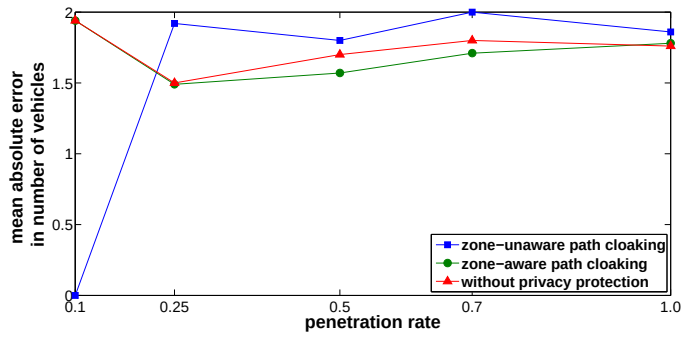


Figure 4.10: The output from the proposed zone-aware algorithm results in small mean absolute error than the output from the zone-unaware algorithm.

samples, the application does not perform well under the zone-unaware algorithm. At full penetration rate (1.0), the proposed zone-aware algorithm increases the success rate from 39% to 82% over the zone-unaware algorithm. We also observed a slight improvement in the mean absolute error with the zone-aware algorithm as shown in Fig. 4.10. Surprisingly, this occasionally result in slight improvements even over the ideal uncloaked data set, likely because the privacy algorithm also removed outliers that affected the application.

We could consider the data samples only useful in conducting the MSTP algorithms as the exact penetration rate (exclude the data samples filtered out by path cloaking algorithms and the data samples which are considered uncompleted for MSTP algorithms). Fig. 4.11 shows the exact penetration rate with and without privacy protection schemes. As expected, the exact penetration for all the cases should be lower than labeled penetration rate, and the zone-aware algorithm overperforms zone-unaware one.

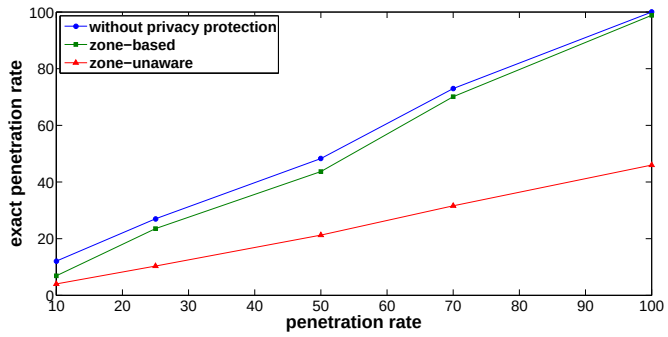


Figure 4.11: The exact penetration rate V.S. claimed penetration rate.

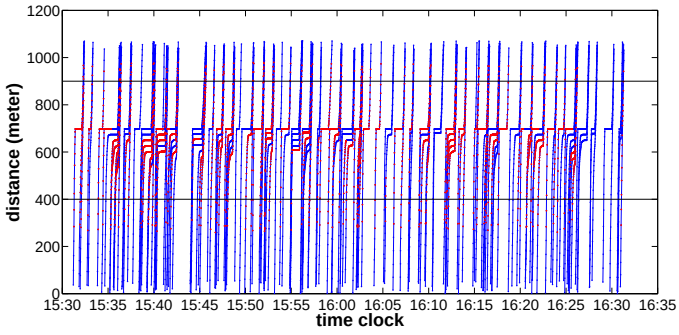


Figure 4.12: Trajectories before and after cloaking using uncertainty-aware path cloaking algorithm.

More detailed comparison can be found in table 4.1.

Fig. 4.12 shows examples of trajectories before and after cloaking using the zone-unaware algorithm. Blue solid lines and dots represent the entire trajectories, and red dots represent the privacy trajectories after cloaking. As we observe in the figure, about one third of samples are removed after cloaking. Some remaining vehicles only have discontinuous or incomplete trajectories, which sometimes may not be detected by both VTL1 and VTL2 (black horizontal solid lines). As a consequence, more than half of the travel time data are lost. Under this circumstance, the MSTP model does not perform well. In contrast, the proposed algorithm (Fig. 4.13) removes only 3% of the collected samples, and it achieves the same success rate as there is no sample get removed.

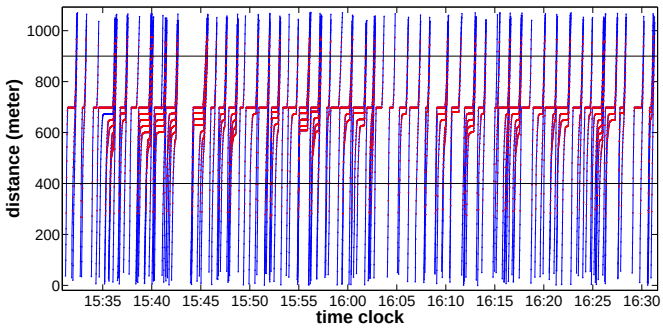


Figure 4.13: Trajectories before and after cloaking using VTL zone-based algorithm.

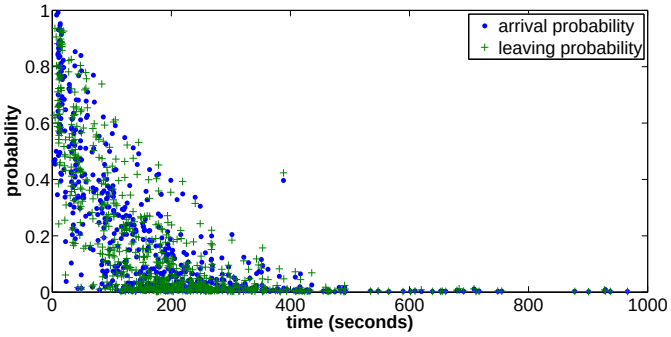


Figure 4.14: When the mean traveling time between two VTL zones is larger than 15 minutes (900 seconds), the maximum probability a vehicle leaving from the first VTL zone will arrive at the second one is less than 0.01 (arrival probability), on the other hand, the maximum probability a vehicle arriving at the second VTL zone is from the first one is less than 0.013 (leaving probability).

4.5.4 The Impact of Travel Time Threshold

In the simulation, there are total 102 VTL zones placed on the map. Theoretically, the VTL zone-based algorithm requires a large amount of workload since 102 zones forms 10302 possible different pairs. However, only 917 zone pairs can be found from the traffic dataset. More than 90% of VTL zone pairs don't exist and are out of consideration when running the proposed algorithm.

When the mean travel time between two VTL zones is larger than 15 minutes, both the arrival probability and the leaving probability are very low as shown in Fig. 4.14. Therefore, we believe it is safe to disregard the cases that have a travel time of more than 15 minutes between two zones. This doesn't affect much on the degree of privacy while improves the computation efficiency. Further study even shows that the ratio of

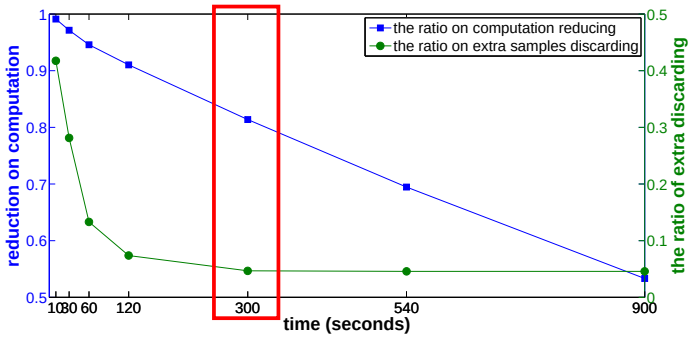


Figure 4.15: The computation is saved by more than 80% if we don't consider samples 5 minutes ago when we calculating entropy. Not too much data samples are discarded if we set the threshold to 5 minutes.

disregarded extra samples is less than 0.05 when ignoring all the cases that have more than 5 minutes travel time between two zones. However, the computational complexity is reduced by more than 80% as shown in Fig. 4.15.

4.6 Related Work

With the rapid development of mobile sensor networks recently [14], privacy preserving issue has attracted more attention than before.

A very common and practical technique people used in protecting privacy is anonymization [104]. However, recent studies [92, 93] have shown that removing identifiers from a dataset does not guarantee anonymity. An enhanced version of anonymization technique is called k-anonymity protection [19]. A release is protected by k-anonymity when the information for each individual contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release. Since the first time it was introduced by M. Gruteser and D. Grunwald [22] into location service fields, the k-anonymity concept has become the core of privacy preserving techniques such as temporal and spatial cloaking for location-based queries [23, 105].

The time-to-confusion criterion to evaluate privacy in a location dataset is introduced by Hoh et al. [17]. It is defined as the time that an adversary could correctly follow a trace. The privacy guarantees and data accuracy are both achieved at the same time through a zone-unaware path cloaking algorithm. The basic idea of this

path cloaking algorithm is to search for the maximum time-to-confusion by updating the entropy value of tracking uncertainty. As long as the entropy value is higher than a given confusion level, the data from the same trace will continue to be released without violating the maximum time-to-confusion metric. Our proposed algorithm is better than the zone-unaware path cloaking algorithm in terms of data accuracy as shown in section 4.5.

To enhance user privacy in location-based services, Beresford and Stajano introduce a method called mix-zone [24]. Mix-zone is defined as an area where no application can trace users' movements. Once a user enters a mix-zone, his/her identity is mixed with all other users. The basic idea is to let the user use different pseudonyms when entering and leaving a mix-zone. Since the attacker doesn't know the length of time frame a user passing through a mix-zone, to trace a user's movement, the attacker has to solve a mapping problem between different pseudonyms. In [25], Freudiger et al further extend the mix-zone concept to location privacy in vehicle networks. Related to mix-zone, [94] evaluates the effectiveness of changing pseudonyms among vehicles during mix-zone and [95] suggests vehicles to loosely synchronize updates pseudonyms (identifiers) when changing their velocities. Different from mix-zone, in our targeted applications, the location of a zone is predefined. Therefore, our task is to maximize the unlink-ability between two zones without the freedom to define the location and size of the zones. Although, pseudonyms are supposed to be a precondition of our proposed algorithm, the methods proposed in [94] and [95] could be a complimentary to it.

4.7 Discussion

Comparison with Mix-Zones. Recall that VTL zones are essentially the inverse of mix-zones; they describe where data should be available rather than where it is to be hidden. Apart from this distinction which we believe will encourage minimal data collection, the proposed approach emphasizes adaptation to varying traffic conditions. Of course, mix-zone size and location could also be continually readjusted to the traffic volume to ensure a constant degree of privacy. This will, however, affect applications such as the traffic queue estimation. Here, a mix-zone that becomes too large would

potentially cut off the end of the vehicle queue, leading to incorrect results.

Distributed Algorithms. The proposed solution assumes that the cloaking algorithm can be run in a centralized location as shown in Fig.4.3. It is also possible to remove the centralized location proxy server, and distribute the proposed algorithm onto the vehicles. For example, the application server could be used to compute the travel time distribution based on collected samples. It could also calculate the path likelihood. Every individual vehicle can locally estimate the overall probability and the entropy value to decide if it should release the sample to the server or not. Two issues need to be carefully considered. First is privacy, as many necessary information are needed from the application server for an individual mobile node to compute its entropy, the directly query process itself might introduce privacy leakage. The second issue is the computation and communication overhead introduced by the local computing. One way to address the privacy concern is by exploiting short-range communications between vehicles, as used for example in the geocache protocol [14]. Furthermore, by optimizing the time threshold as discussed in the last section, computation and communication overhead introduced by distributed computing can also be reduced.

4.8 Conclusion

We have presented a virtual trip line zone-based path cloaking algorithm. Our proposed algorithm can reduce link-ability between any VTL zone pair and minimize the number of trace samples that have to be removed to preserve location privacy. Simulation results show that the proposed algorithm significantly outperforms a zone-unaware cloaking algorithm in all kinds of traffic densities, and it increases the success rate of MSTP model, the targeted application, from 39% to 82% under full penetration rate.

Table 4.1: The performance of queue length estimation. Note, the success rate is not 100% even when no cloaking is used. This is because MSTP model doesn't guarantee success in all traffic scenarios.

Penetration rate (%)	Privacy	Exact Sample Size	Exact Penetration Rate (%)	Cycle with 2 or More Samples	Cycle in Total	Success Rate (%)	MAE (in vehicle number)
100	Before Cloaking	174	100.00	38	46	82.61	1.76
	Zone-based	172	98.85	38	46	82.61	1.78
	Zone-unaware	80	45.98	18	46	39.13	1.86
70	Before Cloaking	127	72.99	31	46	67.39	1.80
	Zone-based	122	70.11	30	46	65.22	1.71
	Zone-unaware	55	31.61	12	46	26.09	2.00
50	Before Cloaking	84	48.28	22	46	47.83	1.70
	Zone-based	76	43.68	21	46	45.65	1.57
	Zone-unaware	37	21.26	7	46	15.22	1.80
25	Before Cloaking	47	27.01	11	46	23.91	1.50
	Zone-based	41	23.56	9	46	19.57	1.49
	Zone-unaware	18	10.34	3	46	6.52	1.92
10	Before Cloaking	21	12.07	2	46	4.35	1.94
	Zone-based	12	6.90	2	46	4.35	1.94
	Zone-unaware	7	4.02	0	46	0.00	0.00

Chapter 5

Privacy: Linking Traces Through Driving Characteristics

5.1 Introduction

The broad adoption of location-based services has led to an increasing number of services and applications that monitor and record time-series location traces of peoples movements [3–5,106]. While some applications require knowledge of the user, a number of applications can enhance privacy by working with anonymous location records. Traffic engineering-related applications that monitor traffic states and other performance measures are one example of such applications.

As in other contexts, achieving strong anonymity in a dataset of location records requires more effort than just deleting identifiers such as user ids, login names, cell phone IDs, or IP addresses [92, 93, 104, 107]. For this reason, a frequently proposed technique to enhance the anonymity of location records is to delete portions of the location traces themselves. This includes trip starting/ending points, which might point to a sensitive location where a user does not want to be seen or a particularly identifying location such as a home, which would make it easier to re-identify a location trace [108–110].

Since it is difficult, however, to scrub all such locations from traces, a frequently proposed concept is to also delete portions of the data within trips, so that longer trips are divided into shorter unlikable segments. This limits privacy leakage of the information from one short segment, if a re-identification at one location were to occur. Variations of this concept are known under the names of mix-zones [24, 25]or path cloaking [22], and have been the subject of many follow-up studies (e.g., [17, 23, 94, 95, 105]). These approaches do have in common that they define and evaluate unlinkability primarily through the use of movement prediction models. Informally, two trip segments are unlinkable if it is difficult to predict the missing portion of the path with

sufficient precision to determine that these two segments are indeed recorded from the same user.

In this chapter, we test whether this assumption remains sufficient as the time-series location traces become more precise. For example, typical GPS receivers in vehicles can provide location samples at an updating rate of 1 Hz and with an error of less than 3 meters in most cases. Given such precise movement traces, are there other unique patterns embedded in the traces, which would allow linking of two location trace segments without using the prediction models?

One form of such unique patterns is outliers in vehicle moving characteristics. We studied a large set of vehicle traces and tried to identify such outliers. In particular, we considered outlier as a vehicle which exhibits higher speed, higher acceleration, or a larger number of lane changes compared to the other nearby vehicles, which are grouped inside the same anonymity group. The causes a vehicle may have very distinguishable moving characteristics comparing to others can be classified into two categories: intrinsic and extrinsic. Intrinsic factors are due to the physical nature of a vehicle, for example the size and weight of a delivery truck reduce the acceleration compared to passenger automobiles. Extrinsic factors are the result of the drivers' driving behavior or other external conditions. For example, many of us tend to drive fast when we are pressed for time. The intrinsic and extrinsic causes are not absolutely disconnected from each other, many times they are related and both contribute to distinguishable moving characteristics.

The rest of the chapter is organized as follows. In Section 5.2, we review some of the related work in location privacy and give more detailed introduction on mix-zone model. In section 5.3, we discuss our motivation with a set of data analysis results. Section 5.4, we study several possible learning models an adversary can use to attack existing mix-zone model both for vehicle classification and for general outlier detection. In section 5.5, we evaluate the proposed learning models and the performance downgrade on the mix-zone model. Section 5.6 concludes our work.

5.2 Background and Related Work

To improve user privacy protection in location-based services, Beresford and Stajano introduced a concept called mix-zone [24, 111]. Once a user enters a mix-zone, his/her identity is mixed with all other users. The theorem behind mix-zone is k-anonymity [19, 20]. The anonymity level in mix-zone is defined by the system entropy or confusion level. In general, higher entropy indicates better privacy.

In [25], Freudiger et al. further extend the mix-zone concept to location privacy in vehicular networks. They first illustrate a symmetric key establishment phase (CMIX protocol) to create cryptographic mix-zones at road intersections and then expand the study on the effect of mix-zones on privacy in vehicular networks. The authors also claim that vehicular mix-zones should be deployed at intersections where speed and direction of the vehicles change the most. Another difference between the original mix-zone concept [24, 111] and the vehicular mix-zone [25] is the way to compute the normalized transition probability between two pseudonyms. In vehicular mix-zone, a probability related to travel time distribution is also considered. Dahl et al. did a formal analysis of privacy for vehicular networks mix-zones in [112]. As some extension work to the vehicular mix-zones, Buttyan et al. [94] evaluate the effectiveness of changing pseudonyms among vehicles within a mix-zone and Mingyan et al. [95] suggest that vehicles should loosely synchronize updates to pseudonyms (identifiers) when changing their velocities. In [16], Dok et al. propose to combine a set of published solutions, namely: Mix-Zones, Silent Periods, and Group Signatures in order to improve the privacy of drivers. Carianha et al. in [113] improve location privacy of mix-zones via extensions to the CMIX protocol. However both improvements only focus on security key establishment phase, but not the privacy preserving mechanism in vehicular mix-zones. A recent work called Mobimix [114] studies the problem due to the assumption on uniform transition probability in mix-zone, and suggests to use non-rectangular mix-zone to improve privacy when transition probability is non-uniform distributed.

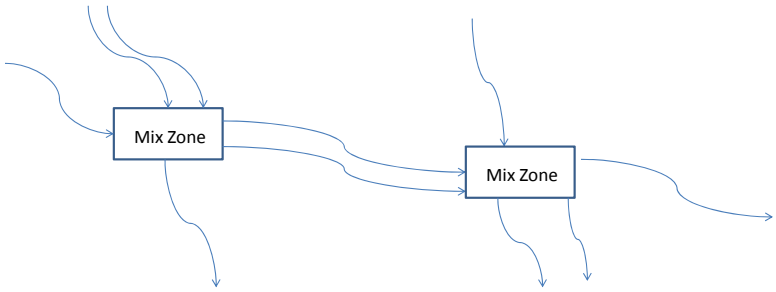


Figure 5.1: Bird's eye view on mix-zone.

5.2.1 The Underlying Theory of Mix-Zone

The underlying theorem in existing zone-based anonymization techniques is to split the trace of each vehicle into discontinuous segments, then mix all the segments within an anonymity set. To be more specific, a mix-zone is defined as an area where no application can trace users' movements. By using different pseudonyms before entering and after leaving a mix-zone, a user's identity is lost. To track a user's movement, an attacker has to solve a mapping problem between pre- and post-zone pseudonyms without knowing the user's path through the mix-zone or the length of time a user spending in the mix-zone. For example, in Fig.5.1 a continuous trace of a vehicle is cut into discontinuous segments by mix-zones. Inside mix-zone, a vehicle keeps silent and does not provide any location data.

It is expected that after sufficient mix with other members from the same anonymity set, a vehicle can make its future trace unlinkable from the old one. The ability to break the link between a pair of traces from the same vehicle through the mix-zone system is called unlink-ability. Generally, the system anonymity is also measured in terms of the unlink-ability through entropy. As in the classic vehicular mix-zone algorithm [25], it defines

$$H(l) = -\sum_{k=1}^{k=N} p_{k->l} log_2(p_{k->l}) \tag{5.1}$$

where $p_{k->l}$ denotes the normalized conditional probability of a vehicle coming with original pseudonym "k" and leaving with the new pseudonym "l", and

$$p_{k->l} = p_{n,e} q_{n,t}(t) \tag{5.2}$$

where $p_{n,e}$ is the probability of exiting at location "e" knowing that the vehicle entered at location "n" and $q_{n,e}(t)$ is the probability that the time needed to enter at "n" and exit at "e" is equal to "t" or in another words the delay characteristics.

In [25], Freudiger et. claim that the mix-zones are most beneficial to be placed in the intersections where the speed and direction of vehicles changes the most. Our proposed adversary models are completely independent from general location predication models assumed in the vehicular mix-zone. Thus the advantage of placing mix-zone around intersections no longer exists.[1]

In the next section, we will show that the existing mix-zone kind of algorithm is no longer able to provide high level anonymity in many cases, especially when an adversary has additional knowledge on vehicle's moving characteristics and other information.

5.3 Feasibility of Location Trace Outlier Detection

So far the analysis on privacy preserving models has been purely based on predicting and matching the reemergence of vehicles out of the mix-zone. That is, the adversary can link two trace segments if he/she can correctly predict where or when a vehicle will appear. The privacy results of such an analysis are based on the hidden assumption that no other way of linking vehicle traces exists. In this section, we will explore the feasibility of linking trace segments based on characteristics of their movement.

5.3.1 The Rise of an Outlier

Consider the example shown in Fig. 5.2 where one vehicle tends to drive significantly faster than other nearby vehicles.

Assume that all the vehicle traces outside the mix-zone are available to the adversary since vehicle a has very different movement characteristics (higher speed) both before the mix-zone and after, the adversary could assume that these two trace segments were

[1] Although it is possible to improve an attacking through datasets collected from intersection where the moving characteristics may appear more obviously, we do not discuss it in this chapter since our objective is to show the reader that an adversary can improve the link-ability on the traces of a target without using general location predication models.

Figure 5.2: A fast car appears as an outlier.

generated by the same vehicle and link them into a single trace. In practice, the success of such heuristics will depend on actual speed distributions as well as their tendency to maintain speed long enough to traverse a mix-zone. We refer to vehicle a an outlier because its movement pattern is very different from the others. The above example actually indicates that outliers could destroy the mix-zone mechanism for the easy link-ability between the traces.

In [115], Grubbs defined an outlier as: "one that appears to deviate markedly from other members of the sample in which it occurs." When considering all trace segments of a group of vehicles as samples, an outlier segment must appear to deviate markedly from the others. As discussed above, special movement characteristics could lead to an outlier. The cause of such special characteristics can be considered from two sides: intrinsic and extrinsic. The difference between a typical delivery truck and a typical passenger automobile is intrinsic (similarly between a motorcycle and an automobile). On the other hand, a particular driver's speed as mentioned in the above example is an extrinsic cause of being an outlier. Our work will focus on intrinsic factors first and then extrinsic factors.

According to the US Bureau of Transportation Statistics [116], there were 8,212,267 motorcycles, 10,770,054 trucks and 230,444,440 passenger automobiles in the US in 2010. Therefore, we will see only one truck for every 25 vehicles observed on average (Note that the ratio of trucks and motorcycles is only 4.4% and 3.3% of the total number of vehicles, respectively). From daily observation, trucks and motorcycles have different movement patterns from passenger automobiles. An adversary can exploit this knowledge to help him/her to identify a truck (or a motorcycle) target from a group of automobiles and to link the trace segments of the truck (or motorcycle) before and

after mix-zones. Considering the ratio of trucks and motorcycles to the total number of vehicles on the road, such a scenario will stand out and be easily observed.

5.3.2 The Observation of Real World Driving Characteristics

In this subsection, we use a set of results to illustrate the feasibility of outlier detection based on vehicle movement characteristics. Since extrinsic cases are based more on individual behavior, we will focus on intrinsic cases first. The following results are based on NGSIM [117] data which consists of detailed vehicle trajectories, wide-area detectors, and supporting data from researching driver behavior. The vehicle trajectory data was collected using digital video cameras to record the precise location of each vehicle on a 0.5- to 1.0-kilometer section of roadway every one-tenth of a second. The portion of data we use covers the southbound direction of highway US 101 (Hollywood Freeway) in Los Angeles, California. The video cameras were mounted on a 36-story building, 10 Universal City Plaza, which is located adjacent to the U.S. Highway 101 and Lankershim Boulevard interchange in the Universal City neighborhood. The vehicle trajectory data was transcribed from the video data using a customized software application developed for NGSIM. The total length of the road segment is about 2100 feet and there are five main lanes throughout the section. Lane numbers start from the left-most lane. The total time spans from 7:50am to 8:05am on June 15, 2005. There are 2086 passenger automobiles (autos), 53 trucks and 30 motorcycles (motos) traces in the dataset. Note, the distribution of vehicle types in this dataset is similar to the US Bureau of Transportation Statistics [116].

First, we show a histogram plot of mean acceleration for motos, trucks and autos in Fig. 5.3. As can be seen, motorcycles tend to have a larger mean acceleration compared to automobiles and trucks (e.g., most of the motorcycles show a mean acceleration higher than 4.2 ft/s^2). The acceleration of trucks tends to be smaller compared to other vehicle types. As shown in the figure, the minimum mean acceleration belongs to trucks. Under certain circumstances (e.g., the target is the only typical motorcycle among a group of automobiles), an adversary could easily identify the target solely based on the difference in the mean acceleration between these two types of vehicles.

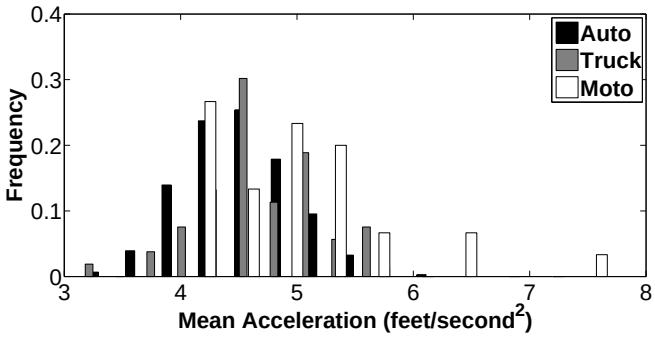


Figure 5.3: The histogram of mean acceleration for trucks, autos and motos.

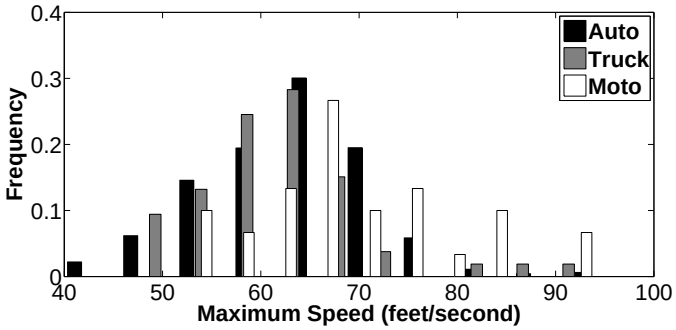


Figure 5.4: The histogram of maximum speed for trucks, autos and motos.

This ability to distinguish between vehicles increases the link-ability of vehicle traces before and after mix-zones.

Similar observations can also be seen in other vehicle movement characteristics. Fig. 5.4 shows the histogram plots of the maximum speed for trucks, automobiles and motorcycles. As with the mean acceleration, most motorcycles tend to have larger values of maximum speed while automobiles have both larger and smaller values than trucks. The former is quite consistent with our intuition while the latter is not. To explain why automobiles have both larger and smaller maximum speeds, we note that the drivers of automobiles cover a large range of the population, some of them tend to drive fast while many others tend to drive slowly (e.g., seniors). On the other hand, the people who drive trucks are mostly professional drivers that are either young or in their middle ages. Thus, it is also reasonable to see that the use of automobiles covers a larger range in terms of the maximum speed. However, from the figure, we can conclude again that under certain circumstances, it is easy for an adversary to identify a target

vehicle if he/she can exploit the maximum speed information (e.g., when the target is a typical motorcycle and the other vehicles are all common automobiles or trucks).

The above two examples do not show a clear way to distinguish a truck from other types of vehicles based on one particular movement characteristic. However, some trends can at least be observed. For example, trucks generally have a relatively small mean acceleration and maximum speed. Our proposed adversary model does not looking for a strong difference in one particular movement characteristic, instead, small differences in multiple characteristics can lead to a way to distinguish target vehicle from a group of vehicles.

In the next subsection, we will illustrate more possible movement characteristics that can be used for intrinsic cause outlier detection.

5.3.3 Exploiting Other Features

There are a few movement characteristics that may be useful for intrinsic cause outlier detection. In this subsection, we illustrate two other characteristics that are useful for identifying trucks from other vehicles.

Lane Changing

As more and more detailed and accurate location information becomes available, some movement characteristics (patterns) that were not available before, are easier to be obtained. For example, when the GPS or other location devices become accurate enough, recognizing the lane a vehicle is moving on becomes feasible. In the NGSIM [117] dataset, all vehicle traces are recorded with lane information. As another example, High Accuracy Nationwide Differential GPS (HA-NDGPS) system which is currently under development can provide 10-15 centimeter accuracy.

In Fig. 5.5, we show the histogram of dwell time ratios on left most lanes for trucks, automobiles and motorcycles. The question of which lanes to stay may appear as a driver's preference or an extrinsic factor at the first glance. However, if the special movement pattern appears in most trucks, we may presume that this is the result from intrinsic or mixed of the intrinsic and extrinsic factors. In fact, it is reasonable to assume

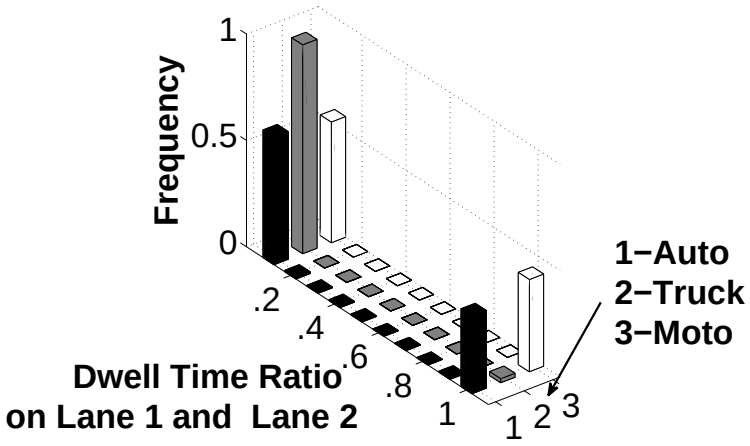


Figure 5.5: The histogram of dwell time ratios on left most lanes for trucks, autos and motos.

truck drivers tend to drive on the outer (right most) lanes instead of inner (left most) lanes because trucks can not move as flexible as small automobiles generally. Inner lanes are usually reserved for fast moving vehicles, slow in lane changing and speed changing discourage truck drivers to stay in the inner lanes. The result in Fig. 5.6 complement what is shown in Fig. 5.5. Trucks do not stay on the inner lanes most times, instead they tend to appear in the outer lanes. Above plots indicate the possibility for an adversary to detect a truck outlier out of automobiles and other type of vehicles.

Headway Distance

Some features or characteristics may not be available or only available from some vehicles. However, once available, they may increase an adversary's ability to detect certain types of vehicles or a special kind of target. One example is the headway information. Headway is defined as the head-to-head distance between a vehicle and its immediate frontward vehicle (front-headway) or the vehicle immediate afterward (back-headway). The maximum headway distance does not help the adversary much since the values can be very large when there is no vehicle nearby. Intuitively, headway distance (especially front-headway) can be useful for distinguishing trucks and small automobiles, because trucks tend to leave more space in front of them. However, the effectiveness of using

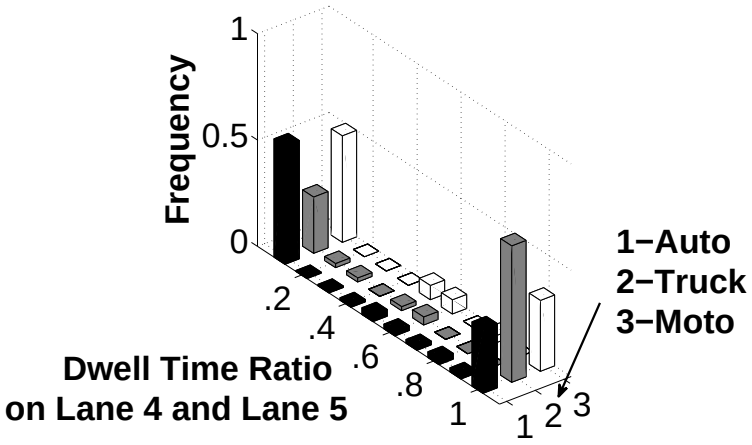


Figure 5.6: The histogram of dwell time ratios on right most lanes for trucks, autos and motos.

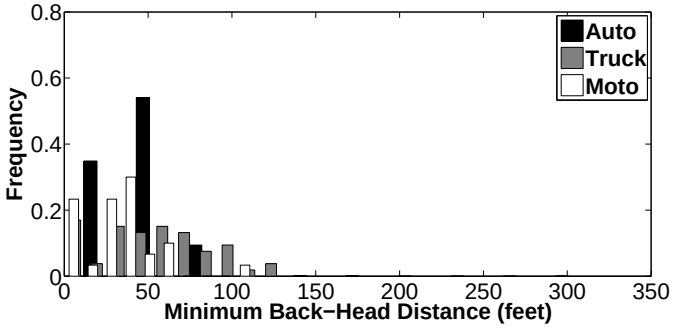


Figure 5.7: The histogram of minimum back-headway distance for trucks, autos and motos.

headway distance is strongly related to the penetration rate. If an adversary has obtained all vehicles' location traces, then it can compute the accurate headway distance for each one, otherwise, the value may not be correct. Consider a scenario, where vehicle a is immediately followed by vehicle b. The location trace of a is unknown by the adversary, then the front-headway distance of vehicle b can't be computed. Therefore, the use of headway distance is limited.

As shown in Fig. 5.7 and Fig. 5.8, trucks have the largest front-headway distance and back-headway distance compared to automobiles, while motorcycles have the smallest values. Although, this is not true for all the cases, the characteristic over large amount

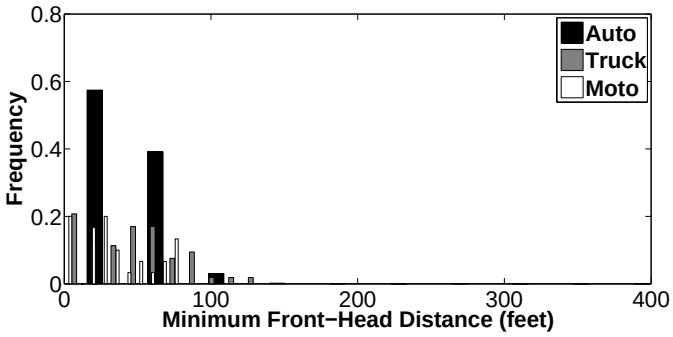


Figure 5.8: The histogram of minimum front-headway distance for trucks, autos and motos.

of samples will be useful when introducing such knowledge into a machine learning classification model. As we will show in the next section, one characteristic alone may not be able to differentiate all trucks or motorcycles from automobiles, however it can be very helpful when combined with other movement characteristics.

5.4 Trace Outlier Detection Algorithms

The previous section shows the feasibility of location trace outlier detection. In this section, we provide detailed methods for such detection.

5.4.1 Intrinsic Outlier Detection

Recall that intrinsic outlier movements are caused by less common types of vehicle. In particular, we consider the case where an adversary knows a priori that the vehicle of interest is a less common vehicle (e.g., a truck or motorcycle) when trying to track the vehicle. This task would become significantly easier, if the adversary can first filter out the traces from all other vehicle types because this would result in a smaller anonymity set and less probability of linking error.

To realize this type of vehicle filtering, we use machine learning classification approaches. An adversary uses historical data to train the classification model, and then uses the model to classify and filter vehicles crossing a mix-zone. Compared to the extrinsic techniques we will discuss later, a key advantage of this method is that the training dataset is not limited to traces from the target vehicle but can also use traces

from other vehicles of the same type. This results in a larger sample pool and can build a better model for outlier detection. However, the adversary must know the intrinsic nature of the target vehicle (e.g., vehicle type) and the characteristic must belong to a less common vehicle. If the confidence level is low, an adversary still has to resort to more general outlier detection strategies, to be discussed later.

In particular, we studied three classic machine learning models, and compared their performance. To be more specific, we studied Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Naive Bayes Models. LDA generates linear boundaries, QDA and NB can both generate quadratic boundaries. For more details about these learning models, the interested reader could consult [118].

The input of each machine learning model is the road segment traces obtained from NGSIM data [117]. Available information includes vehicle ID, global time, local X, local Y, vehicle length, vehicle width, vehicle velocity, vehicle acceleration, lane identification, preceding vehicle, following vehicle, space headway and vehicle class, etc. Local X and Y is the distance from the front center of the vehicle to the left most edge of the road segment and to the entry edge of the segment respectively. The space headway information only covers the front-headway distance, however, with the preceding vehicle and following vehicle information, back-headway distance can be computed and used in the adversary model. For more detailed information regarding the NGSIM data, please refer to [117].

Based on above dataset, we further extract the following movement characteristics:

1. speed (5 features): maximum speed, minimum speed, average speed, median speed, standard deviation of speed.
2. acceleration (8 features): maximum acceleration, maximum deceleration, average acceleration, average deceleration, median acceleration, median deceleration, standard deviation of acceleration, standard deviation of deceleration.
3. proportion (4 features): acceleration greater than 5 ft/s^2, deceleration greater than 5 ft/s^2, speed greater than 60 ft/s, speed less than 20 ft/s.
4. lane position (6 features): frequency on visiting lane 1, 2, 3, 4, 5, total number of lane switching.

5. headway (4 features): minimum distance to the vehicle in front, average distance to the vehicle in front, minimum distance to the vehicle after, average distance to the vehicle after.

Before illustrating our proposed adversary models in the next subsection, we go through several initial observations from studying.

- Some movement characteristics may help an adversary more than others to build the outlier detection models. For example, in the 1927 data samples collected from US101 highway between 7:50am and 8:05am (including both automobiles and trucks), the residual error of a QDA model on maximum speed is only 0.34. While with minimum speed, the error rate is 0.56 which is much larger.
- A movement characteristic itself may not be good enough to differentiate vehicles. However, a combination of such characteristics can be useful. For example, using LDA to distinguish three types of vehicles (trucks, automobiles and motorcycles) from the same dataset with average acceleration, the (residual) error rate is 0.41. If based on the combination of multiple dimensions: maximum speed, average acceleration, proportion of deceleration, the ratio of frequency of visiting lane 4 and 5, and the number of lane changes, the error rate reduces to 0.11.
- Increasing the total number of movement characteristics to be used may not always lead to better performance.
- To find a set of boundaries for classifying trucks, automobiles and motorcycles through one learning model is not easy. Instead, we choose to train a set of models each of which can distinguish a pair of vehicle types.

5.4.2 Feature Selection or Dimension Reduction

To build proper learning models, it is necessary to filter some characteristics (or features) which cause more noise than they contribute to the outlier detection. Therefore, we studied two methods to improve the performance.

Manually Feature Selection

The first strategy is to select a number of features which performs well individually. Multiple selected features are combined together to form learning models in high dimensions. For example, assume we extract a total of n features from the vehicle trace dataset, then for each feature, a learning model is built. All features are then ordered based on the performance in training dataset. Next, the top m features are selected to form the best combination. This method has several advantages. First, the resulted model is easier to be understood. Each feature corresponds to a movement characteristic which has clear definition. Second, the performance can be estimated. Since all the features have real physical meaning, it is relative easy to estimate the performance based on user's daily observation. One disadvantage is that such a method is slow. For every pair of features, the adversary needs to generate a learning model, and sort the results based on performance. Another disadvantage is that the result may not be optimal. Some features may have correlation, thus putting them together may not contribute more information to a learning model.

As part of our evaluation results, the basic features we manually selected are maximum speed, average acceleration, proportion of deceleration greater than 5 ft/s^2, frequency on visiting lane 4 and 5.

Principle Component Analysis Based Dimension Reduction

The second strategy is to use a dimension reduction method. In our study, we use Principle Component Analysis (PCA) to project all the features onto m best dimensions and then train the learning model based on these m dimensions. The advantage of this method is that it is fully automatic and can achieve better performance than the previous method. That is because if two features are strongly correlated, it will only appear as one dimension after projection in PCA. One disadvantage of such a method is the selected dimensions in the end may not have clear physical meaning, thus the results cannot be always interpreted easily.

Figure 5.9: An outlier detection scenario.

5.4.3 General Outlier Detection: Extrinsic

In the extrinsic case, it is not clear that we can build a learning model based on historical data from different drivers and different trips. For example, the special movement characteristics of a driver during a trip (e.g., due to time pressure) may not appear in his/her previous trip two days ago. And there is no training dataset available from general location trace data to indicate if a driver is under time pressure or not. Comparing to intrinsic, extrinsic is more unstable and more case dependent and/or time dependent. Therefore, the machine learning model for outlier detection must be built on the fly and in real time. Assume that before a mix-zone, the trajectory of a target vehicle a is known as shown in Fig. 5.9, the challenge for an adversary is now to identify the same target vehicle after the mix-zone. Since there is no historical data that can be used, the size of the training dataset relies on the length of the road segment before the mix-zone. While we still can use machine learning models, we need assume the target vehicle itself is a class. There are two strategies available for the adversary:

One-to-One

In this method, the trace from the target vehicle forms a class, and all the other traces from the remaining vehicles of the anonymity set form the second class. The learning model detects the target based on binary classification result of the dataset after the mix-zone. This method is easy to implement, however due to the small number of data samples available in the first class compared to the second class, the classification model may not be able to generate a good boundary for a binary classification model.

One-to-Many

In this method, not only the trace from the target vehicle forms a class, but also all other vehicles form classes independently. The adversary builds multiple binary classification learning models for the target with all the other vehicles. To detect the outlier, the adversary runs all learning models for each trace segment collected after mix-zone to determine if a trace looks more like the target or one of the others.

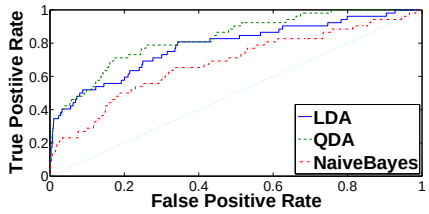
5.5 Evaluation

In this section, we study the proposed outlier detection techniques with the NGSIM [117] dataset. The True Positive Rate (TPR) and the False Positive Rate (FPR) are used as measures to characterize the ability of an adversary to track a vehicle after a mix-zone. The same dataset which has been studied in section 5.3.2 is used. Here we give a brief summary of the dataset again. All the location traces are collected from the US101 highway. The dataset covers the traffic data between 7:50am and 8:05am. It includes a total of 1875 automobiles, 52 trucks and 25 motorcycles, all of which have trajectories longer than 2000 feet. To date, we have focused more on intrinsic factors, since there is a relatively large training dataset available. A small number of vehicle traces are removed. These are those which 1) were less than 2000 feet long; 2) never had any vehicle driving in front; 3) never had any vehicle following behind. The first condition is to reduce the possible impact from factors other than the vehicle movement characteristics, the second and third condition are to make sure we can use the same dataset when comparing the performance of different outlier detection algorithms (with and without headway information).

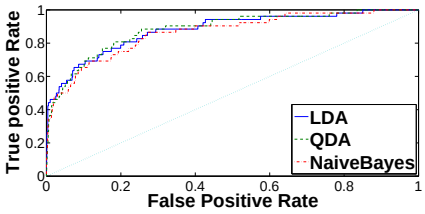
The results presented here focus on showing the Receiver Operating Characteristic (ROC) curves of different learning models. An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. It is not only a useful performance graphing method in general, but also very useful for domains with skewed class distribution and unequal classification error cost [119]. The data source input into the outlier detection algorithm mostly follows skewed class distributions. Thus, it is

beneficial to use the ROC curves for comparing the performance of different learning models. On the other hand, through the ROC curves, the trade-off between quantity of attacks and quality of attacks (in terms of confidence level) can be easily observed.

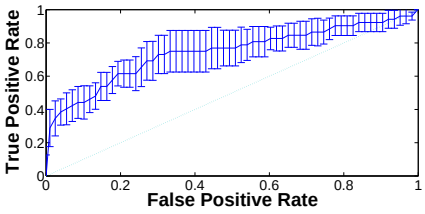
5.5.1 Outlier Detection: Intrinsic



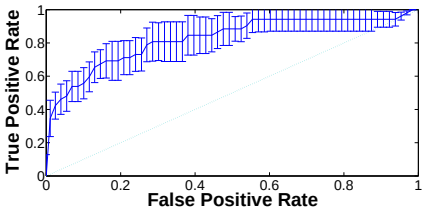
(a) performance comparison on the training dataset



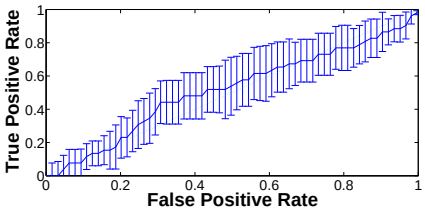
(b) performance comparison on the training dataset



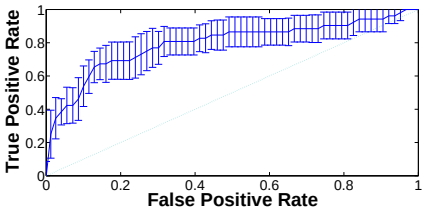
(c) 10-fold cross validation results for QDA learning model



(d) 10-fold cross validation results for QDA learning model



(e) 10-fold cross validation results for Naive Bayes learning model



(f) 10-fold cross validation results for Naive Bayes learning model

Figure 5.10: Detecting trucks from automobiles with manually feature selection method. Left side: without headway information; right side: with headway information.

First, we show the results based on manually feature selection. The performance of the three different machine learning models LDA, QDA and Naive Bayes for truck detection are shown in Fig. 5.10. The main objective is to show that by properly selecting a learning model an adversary can identify a truck or a motorcycle from common automobiles. This means that additional information leaking from the traces

could be used to compromise mix-zone protection. The features we manually selected are maximum speed, average acceleration, proportion of deceleration greater than 5 ft/s^2 , frequency of visiting lane 4 and 5. As shown in the ROC graphs, in general, to improve the confidence of tracking a target, an adversary has to sacrifice the quantity of the overall attacks. As the confidence level increases, the number of vehicles that can be tracked decreases. Nonetheless, even an adversary who can claim a very high confidence in tracking a target just occasionally may be unacceptable.

Fig. 5.10(a), Fig. 5.10(c) and Fig. 5.10(e) show results (for the LDA, QDA and Naive Bayes models) from the training dataset and results of 10-fold cross validation on the QDA as well as Naive Bayes model, respectively. As can be noticed from Fig. 5.10(a), all the three learning models generate better ROC curves than randomly guessing (the diagonal line in the figure) in training. This indicates that with machine learning classification models an adversary is able to identify trucks from automobiles. To be more specific, the ROC curve of the QDA model crossing the point at FPR=0.2 and TPR=0.7 indicates that an adversary can identify a truck with 70% success rate if it is indeed a truck. 20% of the time, the adversary will mis-identify an automobile as a truck. The 10-fold cross validation as shown in Fig. 5.10(c) and Fig. 5.10(e) evaluates the learning model by rotating training and testing data (in 9:1 ratio) 10 times. Although the Naive Bayes model does not perform very well, as can be inferred from Fig. 5.10(c), using QDA, an adversary can achieve higher success rate (about 96%) when he/she only focuses on tracking 40% of the trucks which show the most evidence of being a truck. This is a considerable improvement of the tracking ability of an adversary. For instance, in a group of ten vehicles containing one truck and nine automobiles passing through a mix-zone, an adversary only has about a 30% failure rate to mis-classify one of the nine automobiles as a truck while the real truck is able to be identified in 40% of cases. This dramatically reduces the protection of a mix-zone for such outlier vehicles, since the mix-zone model predicts a 90% failure rate. The QDA learning model can also help an adversary when there are multiple trucks in the anonymity group. For example, if there are two trucks in ten vehicles, an adversary only has a 28% failure rate to mis-classify one out of the eight automobiles as a truck

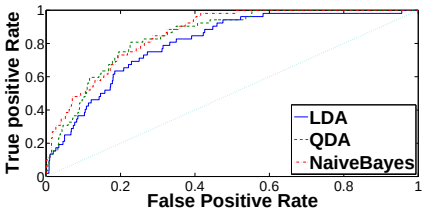
while it has a 40% success rate to identify the real trucks. With an overall success rate of more than 30%, he/she can correctly identify the trace of the target truck.

Fig. 5.10(b), Fig. 5.10(d) and Fig. 5.10(f) show the results when the headway information (which has been discussed in section 5.3.3) is included. Both training and testing results show that with headway information, an adversary will have a better chance to detect a truck under all three models. For example, as shown in Fig. 5.10(d), an adversary can achieve roughly 1% FPR under 35% TPR. This means that in a group of ten vehicles, one of which is a truck, passing through a mix-zone, the adversary only has a 8.6% chance to mis-classify one automobile as a truck while he/she is able to identify the real truck at a rate of 35%. This compares to a 90% failure rate with random guessing and shows that the protection of the mix-zone has deteriorated.

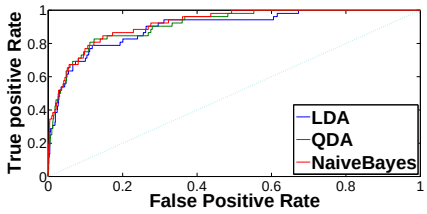
In Fig. 5.11, we assume that an adversary will perform a principal component analysis on the dataset. This operation projects all the characteristics (features) onto the 10 most important dimensions. The boundary is then generated only based on the 10 dimensions. Compared to Fig. 5.10, the performance is slightly better in both the training dataset and the testing dataset. This indicates that an adversary can achieve better results if he/she has a better learning model or data mining method. The better an adversary can do, the worse the current mix-zone model will be. In additional to all of the above observations, we also note that among all these three models, QDA performs the best and the headway information improves the performance of every learning model. We interpret this as evidence that further improvement is possible on outlier detection techniques if the adversary has better learning models or data mining algorithms.

In our study, we also evaluated the outlier detection techniques on motorcycle identification. Similar results were observed. For completion, we attach them: Fig. 5.12 and Fig. 5.13 here without further detailed discussions.

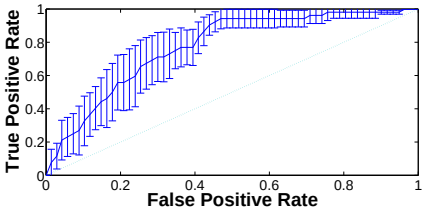
Since entropy is one of the most important factors in evaluating a mix-zone model, in Fig. 5.14, we compare the original system entropy of a mix-zone with the one after the QDA learning model is used. As can be seen from the figure, while varying the size of the anonymity set, the QDA learning model always reduces the system entropy by



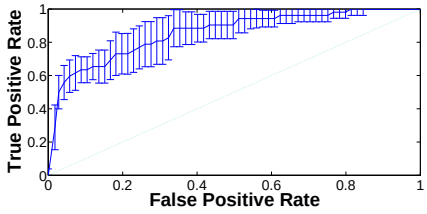
(a) performance comparison on the training dataset



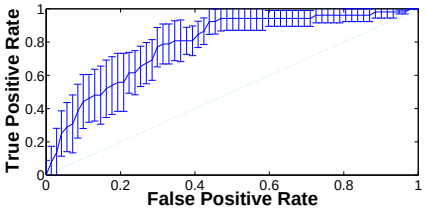
(b) performance comparison on the training dataset



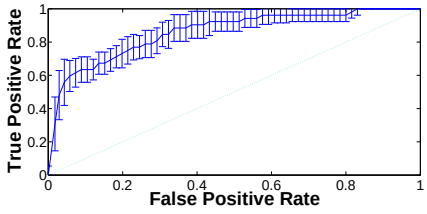
(c) 10-fold cross validation results for QDA learning model



(d) 10-fold cross validation results for QDA learning model



(e) 10-fold cross validation results for Naive Bayes learning model

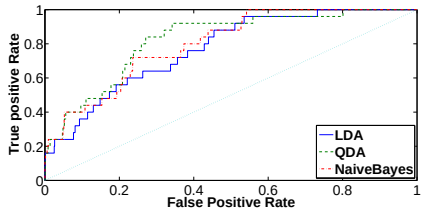


(f) 10-fold cross validation results for Naive Bayes learning model

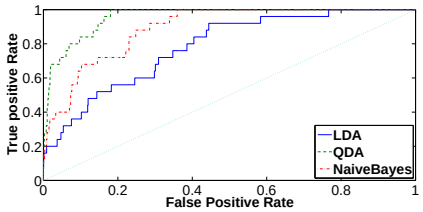
Figure 5.11: Detecting trucks from automobiles with PCA projection method on the first 10 dimensions. Left side: without headway information; right side: with headway information.

at least 1. This reduction helps the adversary significantly, considering that entropy is a logarithmic value.

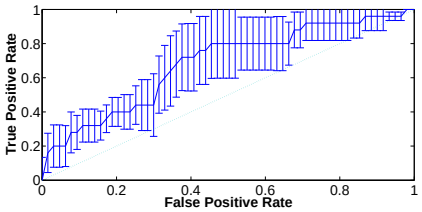
Finally, in another set of experiments, we create 55 groups of ten vehicles. Each group has one truck and the remaining are all automobiles which are moving closest to the truck. With the QDA model, an adversary can successfully identify trucks in the 22 of the 55 groups.



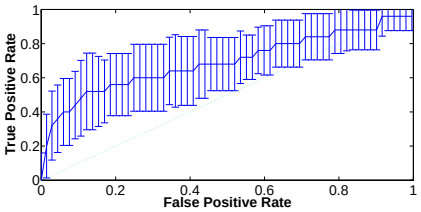
(a) performance comparison on the training dataset



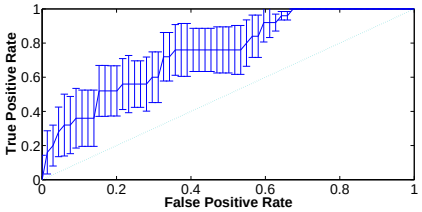
(b) performance comparison on the training dataset



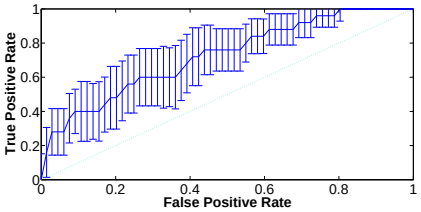
(c) 10-fold cross validation results for QDA learning model



(d) 10-fold cross validation results for QDA learning model



(e) 10-fold cross validation results for Naive Bayes learning model

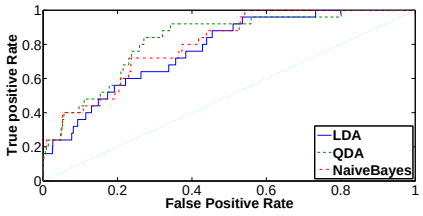


(f) 10-fold cross validation results for Naive Bayes learning model

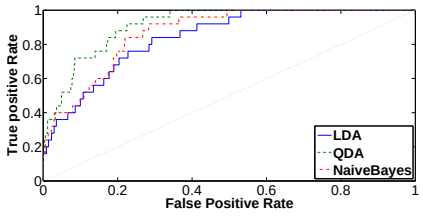
Figure 5.12: Detecting motorcycles from automobiles with manually feature selection method. Left side: without headway information; right side: with headway information.

5.5.2 More General Outlier Detection: Extrinsic

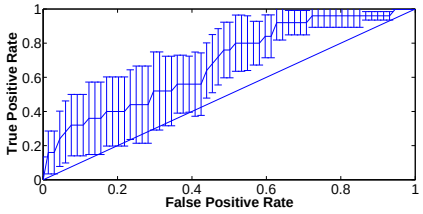
Last, in this subsection, we show preliminary results on more general outlier detection: extrinsic and/or mixed cases. In this experiment, all 2000 feet long trace segments are further divided into three portions. The vehicle trace dataset from the first and third portions are assumed to be available by the adversary. All trace dataset from the second portion of the road are removed in order to simulate a mix-zone model. Assume there is no intrinsic knowledge available such as vehicle type. The dataset from the first segment are used as training data. The adversary tries to link all the pairs of the traces belonging to the same vehicle. The learning model used is QDA. Different from previous work, in this part we assume the target can be any vehicle.



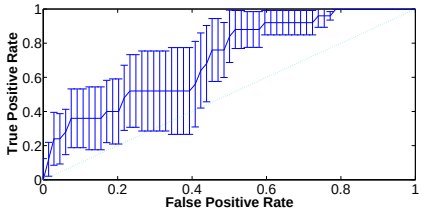
(a) performance comparison on the training dataset



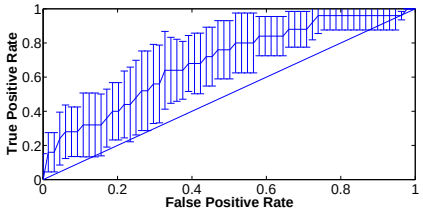
(b) performance comparison on the training dataset



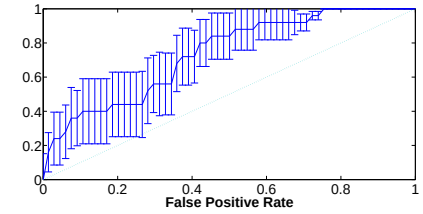
(c) 10-fold cross validation results for QDA learning model



(d) 10-fold cross validation results for QDA learning model



(e) 10-fold cross validation results for Naive Bayes learning model



(f) 10-fold cross validation results for Naive Bayes learning model

Figure 5.13: Detecting motorcycles from automobiles with PCA projection method on the first 10 dimensions. Left side: without headway information; right side: with headway information.

As shown in Fig. 5.15, the tracking rate, which is defined as the success rate at which an adversary can identify a particular target from a group of vehicles largely increases with the learning model compared to random guessing. For instance, in a ten vehicle anonymity set, the learning model has a 28% tracking rate, which is a good improvement for an adversary who originally had only a 10% tracking rate. As mentioned before, we define the anonymity set size as the number of nearby vehicles whose trace data are available to the adversary.

This figure has shown the tracking rate for any randomly picked vehicle from the dataset. This indicates that there are a large number of cases in which the target

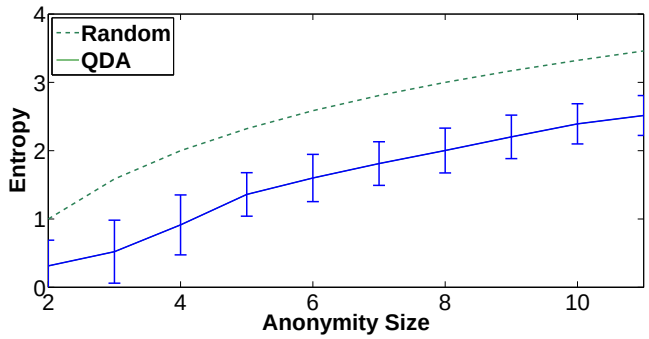


Figure 5.14: Entropy reduction due to outlier detection.

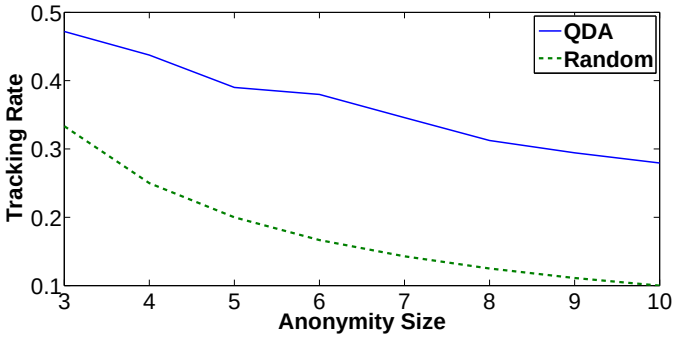


Figure 5.15: With the proposed generic outlier detection method, the rate of successfully tracking is increased.

may not have special movement characteristics compared to other vehicles. In practice, however, the adversary can ignore those cases and concentrate on outlier vehicles that can be tracked with high confidence.

Fig.5.16 shows the probability to find the special feature combination of any target vehicles in the data samples. For example, the chance to find a set of features for a vehicle to make it distinguishable from all other vehicles (a size of 10) is 30%, while the chance to find a set of features for a vehicle to make it distinguishable from at least 5 vehicles is 60%. This result is also consistent with previous figures and indicates that there is a big chance that an adversary can identify a special vehicles through machine learning models.

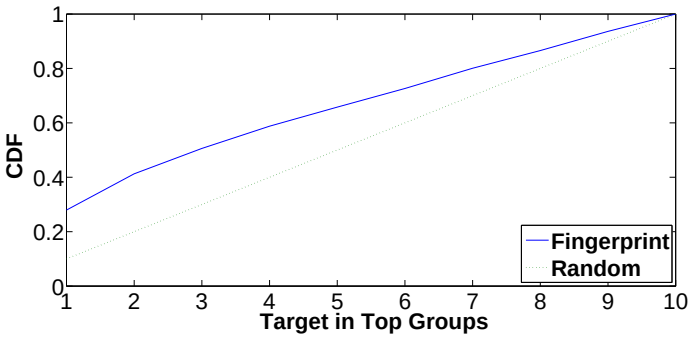


Figure 5.16: CDF indicate the target is in top groups.

5.6 Conclusion

In this chapter, we have studied whether existing models to measure the degree of privacy in anonymized location traces hold as location traces continue to become more precise. Using data captured from vehicles we have shown that fine-grained location traces reveal speed distribution and acceleration patterns that can be used to distinguish traces from different vehicle types (e.g., trucks and cars). Our analysis on NGSIM location trace shows that an adversary can identify 40% trucks from cars with success rate of 96%. We have also shown that it is possible to identify outlier driving patterns such as higher speed, which could be used to link anonymous segments of location traces and eventually recover complete trips. Our preliminary results show that the general outlier detection technique can improve an adversary's ability to identify a trace segment of any user from an average tracking rate of 10% to 28%. While this rate is still relatively small, and would be smaller still if a vehicle trip passes over multiple mix-zones, these findings show that movement characteristics reveal information. An immediate countermeasure is to revert back to coarser location traces but a full solution to this issue remains an open problem. We believe that further research is warranted to refine the definition of unlink-ability for very fine-grained location traces.

Chapter 6

Privacy: Design Models with Limited Resource

6.1 Introduction

In chapter 4, a privacy preserving model is developed for vehicular sensing networks: VTL zone-based path cloaking algorithm. This model can protect individual users' privacy by dynamically filtering location traces based on the current entropy of the system through a middleware location proxy server.

Then, a question was raised in chapter 5: What if an adversary can extract more information from the plain location information (vehicle trajectories)? We studied the possibility to link anonymous location traces through driving characteristics. The results indicate that when driving characteristics can be extracted from the location information, either the zone based privacy models need to be improved or a higher threshold should be used to filter more location data in order to achieve the desired privacy level.

In this chapter, we study a set of general privacy preserving models that can be directly distributed to a user's individual vehicle system. The main advantages of these models are that they avoid the requirement of a location proxy server. Secondly, it becomes possible for system designers to gauge the effects of these models even before performing the privacy preservation algorithm on the vehicular sensing network. Another advantage is the simplicity of this model in its mathematical representation. Furthermore, these models can be utilized in more general (not necessarily zone based) location applications services. At last, these models can be considered as references or lower bounds for any privacy model which requires more detailed location information from the individual user. One basic feature of these models is to move the data-filtering

process from the location proxy server to the individual vehicles. Therefore, the dependency on a trusted local proxy server is reduced. However, since the process is executed locally and in a distributed way, the information available for privacy processing is limited. Unlike our VTL zone-based privacy model in which the location proxy server can evaluate the system entropy in a particular area based on the collected traces from a zone, the proposed models do not have such resource. This limitation will significantly reduce the privacy protecting ability of a network system. Thus, to help individual vehicle make better data filtering decisions, it is assumed that 1) limited high level information can be collected by a third party (e.g. application server); 2) overall network information is available to all the members. These assumptions are still much weaker than the location proxy server assumption. In the proposed models, before being confirmed to be releasable, no detailed trace needs to be stored anywhere except in the vehicle itself.

The rest of this chapter is organized as follows. System assumption is described in section 6.2, followed by section 6.3 which reviews some of the concepts used in this chapter. In section 6.4, basic system performance is studied in terms of the data releasing rate and the prediction error rate. The proposed models are shown in section 6.5. Section 6.6 presents performance comparison among the proposed models through Monte Carlo Simulation, while in section 6.7, the proposed models are compared again using NGSIM real location traces. This chapter is concluded in section 6.8.

6.2 System Assumption

6.2.1 The Application Server

Many location based services require users to upload location information periodically. Clearly, not everyone is willing to share the information. A user should have the right to refuse uploading some location traces if he/she feels his/her privacy is threatened. However, a large number of traces missing from users will affect the performance of the location based service. Thus, from maintaining the quality of the location based service point of view, it is better for the application server to collect as many location

information as possible from every user.

Similar to the VTL zone-based path cloaking model and many other research works in the privacy domain, we focus on protecting users' privacy through trace filtering method. However, different from the VTL zone-based model, the proposed models are more independent from the central location proxy server. Individual vehicle plays the main role in performing data filtering algorithm. Basic network information such as traffic density may still be available from a third party or the application server.

6.2.2 The Adversary Model

The adversary is assumed to have access to all the location information from the application server. It is trying to track a target vehicle by linking the related anonymous traces. The strategy of the adversary is based on comparing the distances from everyone's current locations to target's expected location. To be more specific, assume the adversary knows that the trace of vehicle A before time t_0. At time t_1, there are n location traces available for the adversary. Among those, one belongs to vehicle A. Based on the information obtained from the trace before time t_0, the adversary tries to estimate vehicle A's future location at time t_1. By comparing all the n location traces with the estimated location of vehicle A at time t_1, the adversary finds the trace which has the shortest distance to the estimated location and links it to vehicle A. Since decision criteria are based on distance between the estimated location and the location of the traces, it is called distance based adversary model.

6.3 Basic Concepts

The models presented in this chapter focus on data filtering which is one of the methods can be used to improve a system's privacy preservation ability. In VTL zone-based path cloaking model, traces are all collected by a location proxy server, and then some are filtered (discarded) based on the system entropy and others are submitted to the application server. In the proposed models, the location proxy server is assumed not available or there is no guarantee that the filtered location traces will not be obtained

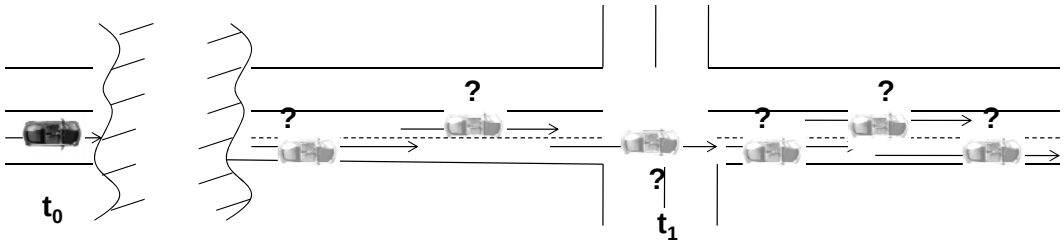


Figure 6.1: The future location of a target vehicle is uncertain.

by an adversary.

Two factors which impact the performance of a privacy system are discussed in this chapter: the density of the vehicular network and the uncertainty of a vehicle's future location.

Network density is usually defined as the total number of vehicles divided by the size of the entire vehicular network. In this study, we only consider relative simple road topology such as highway, and use the average distance between neighboring vehicles to indicate the density. Density has important meaning to privacy protection because when vehicles are moving closely to each other, it causes more troubles for an adversary to identify the target. Prediction error increases when vehicles are moving together as a group; on the contrary, it decreases when vehicles are far away from each other.

What is the uncertainty of a vehicle's future location? Fig.1 illustrates this concept in greater detail.

Assume a vehicle v has disclosed its location information at time t_0. Given a new location trace at t_1, how can an adversary determine if the new trace belongs to vehicle v? A vehicle leaving from the previous location at t_0 can only appear in a certain area of the road network after $t_1 - t_0$ time period. To put it more precisely, its future location is inside a certain area with high probability and all other locations with very low probability. In the one dimensional case (assume, for simplicity, there are no forks in the road), it is reasonable to assume that this future location follows a Gaussian distribution. When an adversary is trying to link the new trace with a target vehicle, the higher uncertainty of this vehicle's location at time t_1, the higher probability the adversary will make prediction error. On the other hand, by all means if it is known the uncertainty of a vehicle in terms of its future location is low, we need to be more

cautious while releasing location traces. The value of the uncertainty depends upon many factors, such as the distance between the location of target vehicle at t_0 and the location at t_1, the time difference between t_1 and t_0, the speed limit of the road, etc.

In this work, the prediction error rate is the criteria to evaluate a system's ability in privacy preservation and the data releasing rate is used to indicate the impact a privacy model has to the performance of the location based service. The prediction error rate is defined as the probability that an adversary cannot identify true trace of a target vehicle.

6.4 System Analysis

In this section, system analysis on a simplified vehicular network is presented. We assume the value of uncertainty follows a Gaussian distribution and the average distance between neighboring vehicles is fixed. However, even for such a preliminary case, a thorough theoretical analysis is complicated. Instead, we choose a Monte Carlo simulation method to study the system.

In the following subsections, we try to answer questions listed below:

1. In a simple scenario, will simulation result converge?
2. Does the number of vehicles in the whole networks affect the results?
3. How are the average distance and the standard deviation influencing the prediction error rate?

6.4.1 Simulation Setup

First, n number of points (corresponding to vehicles and their expected locations) are uniformly deployed along a straight line (corresponding to a road). Then another n number of points (corresponding to the true locations of the vehicles) are independently put on the network around each one's expected location following one Gaussian distribution. Based on this setup, we study the impact on the prediction error rate from

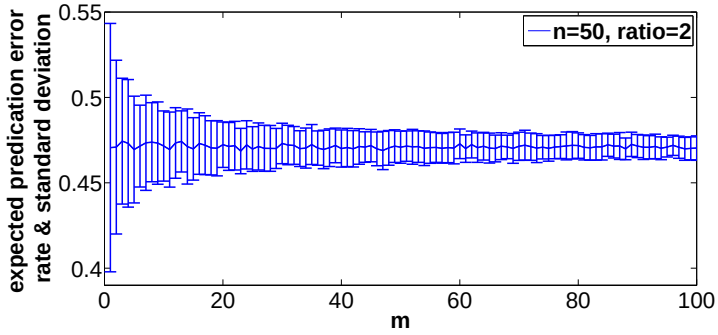


Figure 6.2: The impact from m on prediction error rate with n=50 and ratio=2.

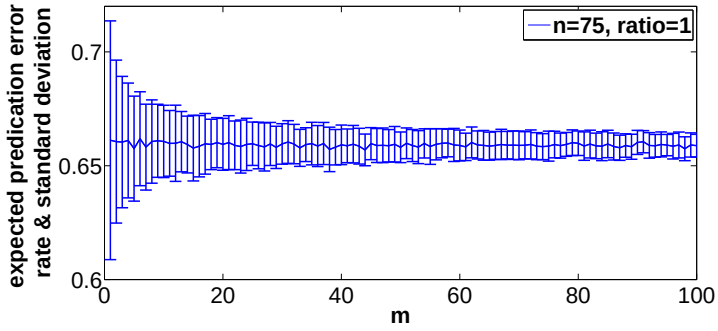


Figure 6.3: The impact from m on prediction error rate with n=75 and ratio=1.

various parameters, such as the total number of vehicles n, the ratio of the average distance to the standard deviation.

6.4.2 Does error rate converge when m increases?

The first step is to check if prediction error rate obtained through Monte Carlo simulation is stable and convergent. This is important because if the error rate is stable, it indicates 1) the Monte Carlo simulation is an appropriate method to be used to study this system, and 2) the ratio of the average distance to the standard deviation is a factor that directly impacts the prediction error rate.

In Fig. 6.2 and Fig. 6.3, the expected prediction error rate is compared with the number of simulation rounds in order to verify the output is stable. m is the number of simulation rounds, n is the total number of vehicles in the network; the ratio is the average distance between neighboring vehicles versus the standard deviation of the Gaussian distribution. As shown clearly, the prediction error rate is stable and

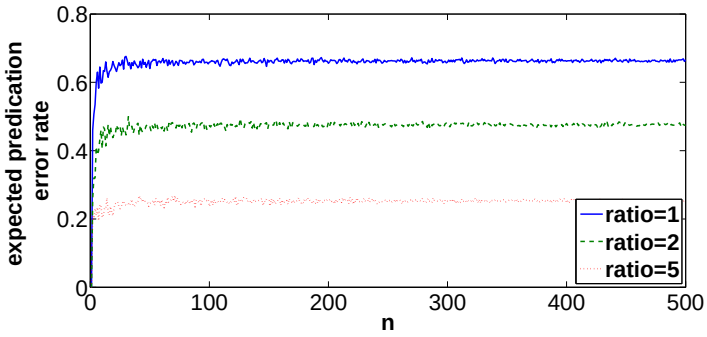


Figure 6.4: The impact of n on prediction error rate.

convergent, its variances decreases when m increases. This indicates that ratio between the average distance and the standard deviation has stabilizing influence on the prediction error rate, and the output from Monte Carlo simulation tells the real error rate. Another observation is that the expected prediction error rate can be estimated before repeating the simulation hundreds of times. The value obtained when $m = 100$ is a fair tradeoff between a correct estimation and the number of simulation rounds.

6.4.3 Does n affect the prediction error rate?

Intuitively we think the prediction error rate increases as the number of vehicles on the road increases. However, this is only partially true. The factor that truly impacts the prediction error rate is the density of the road network. Thus, when the number of vehicles increases largely if the scale of road map also increases, the change in the prediction error rate is still minor, as shown in Fig. 6.4. In the simulation, to keep the density unchanged we adjust the number of points along the straight line, at the same time varies the total length of the line. As illustrated in the figure, the expected prediction error rate is considerably stable, especially when n is larger than 100 for all three cases (ratio=1, ratio =2 and ratio=3). Note, each point on the curves is generated based on 500 rounds of simulation.

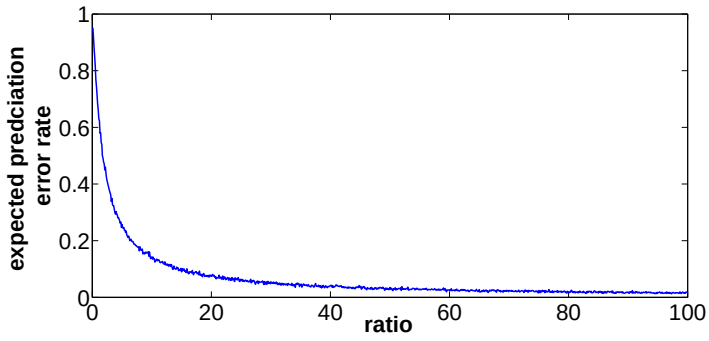


Figure 6.5: The prediction error rate decreases almost logarithmically when the density to uncertainty ratio increases.

6.4.4 The impact of density to uncertainty ratio on prediction error rate.

In this subsection, the influence on the prediction error rate from the ratio of the average distance (density) to the standard deviation (uncertainty) is studied. The curve in Fig. 6.5 is generated based on 100 rounds of simulation in which there are total 100 vehicles uniformly distributed in the network.

As can been seen, the expected prediction error rate decreases logarithmically when the ratio increases. The turning point is around 10, before which the error rate decrease rapidly. In the opposite, rate decreases slowly after the point. The error rate eventually gets close to 0 as the ratio continuously grows. Fig. 6.5 also provides an easy way to determine whether a privacy protection model is required for a vehicular network. For example, while a system's target prediction error rate is 0.8, once the density to uncertainty ratio reaches 0.5, it is not necessary to use any privacy model.

6.5 The Proposed Models

In this section, we propose three different privacy models based on the study results from section 6.4.

6.5.1 Randomly Dropping Model

As learned from last section, when the ratio between the average distance and the standard deviation is fixed, the network system has a fixed expected prediction error rate. This error rate indicates the original privacy level of a network system without using any privacy preserving model. In order to achieve higher privacy protection level, one of the methods is to adjust the value of the ratio since each ratio corresponds to a particular prediction error rate. Ostensibly, Fig. 6.5 shows the prediction error rate is inversely proportional to the ratio. In other words, instead of filtering traces, we should add traces to increase the prediction error rate. This is not what a filtering algorithm can achieve. However, above observation is only partially true since the overall prediction error rate is also impacted by those traces dropped from the system. This brings up the following discussion.

Assume there are total 100 traces, correspondingly 100 vehicle IDs available to an adversary. The current system prediction rate is 0.5, which means this adversary can link 50 traces to their corresponded vehicle IDs. Next, assume 20 traces (rate of 0.2) are dropped. If the prediction error rate does not change, then among the remaining 80 $(1 - 0.2 = 0.8)$ traces, on average, $80 * 0.5 = 40$ traces would be successfully linked to their vehicles IDs by the adversary. The overall prediction error rate in terms of the originally 100 traces and their corresponded 100 vehicles is $\frac{100-40}{100} = 0.6$ which can also be directly computed through $0.8 * 0.5 + 0.2 = 0.6$. We consider this overall prediction error rate is the true prediction rate, thus based on above example, we have

$$(1 - x) * r_o + x = r_t \tag{6.1}$$

where x is the rate of traces dropped (in above example, $x = 0.2$) , r_o is the system's original prediction error rate ($r_o = 0.5$) and r_t is the resulted true prediction error rate (0.6). If we only know $r_o = 0.5$ and $r_t = 0.6$ as desired prediction error rate, then we can compute the value of x as the rate of trace to be dropped from above equation too.

Based on above analysis, we propose the first privacy model: randomly dropping certain ratio of the records according to equation 6.1. The intuition behind this method is very straightforward. According to the equation, when x is increasing and $r_o < 1$ is

fixed, the value of r_t is monotonically increasing with x. Therefore by dropping traces, the value of x increases, so does the value of r_t. There is a unique solution existing that will satisfy this equation.

To execute this model, additional information are needed: the network density, the uncertainty of a vehicle's future location and the original system prediction error rate. The value of network density can be estimated without putting privacy in hazard. For example, let each vehicle send beacon signal periodically without location information. The uncertainty of a vehicle's future location is estimated based on historical location traces. The only difficulty is to obtain the prediction error rate. There are three possible methods: 1) On highway or other uncomplicated road networks, the simulation results as shown in Fig. 6.5 can be used to estimate the prediction error rate; 2) Before setting up the privacy preserving model, allow an observation period in which all traces are collected by a third party to generate the prediction error rate curve; 3) Allow application server or a third party to infrequently collect discontinuous traces only for the purpose of generating the prediction error rate curve. Note the second and the third method are still different from relying on location proxy server.

6.5.2 (Error) Rate Control Model

Above randomly dropping model assume that before getting the true prediction error rate of the total number of traces, the prediction error rate of the released traces is the same as the original prediction error rate. However, this might not be true because the total number of released traces (and their corresponded vehicle IDs) is less than the original total number of traces (and vehicle IDs). Thus, the average distance among the released vehicles are larger as well as the value of ratio. According to Fig. 6.5, we then will have smaller prediction error rate. Therefore, continuing use the original prediction value o_r is not fully correct.

In order to achieve the target prediction error rate, the privacy model should follow below equation instead:

$$(1 - x) * y + x = r_t \tag{6.2}$$

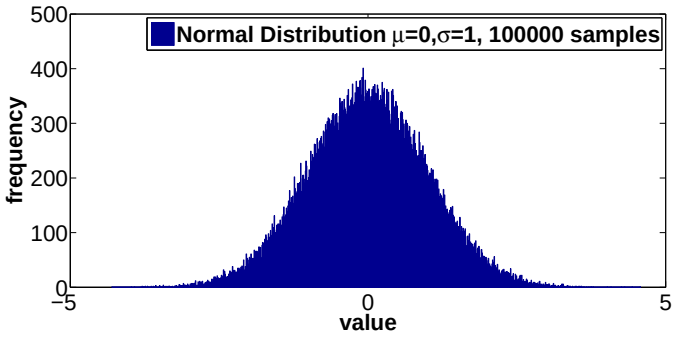


Figure 6.6: Original Distribution.

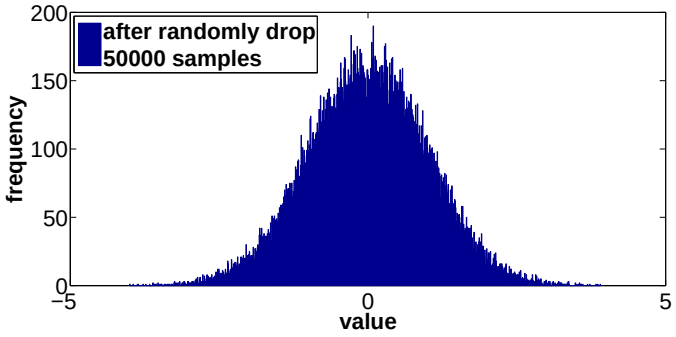


Figure 6.7: Distribution is not changed after randomly dropping

Where x is again the rate of traces to be dropped, Instead of using r_o, y is the actual prediction error rate when the ratio of network density to the uncertainty changes (as discussed above which can be referred from the prediction error rate curve), and r_t is the desired prediction error rate. However, the complicate of this equation is when you replace r_o with y, then the value of x obtained through this equation changes too, and because of this, the value of y needs to be adjusted again according to the above discussion and Fig. 6.5. There are two types of methods can be used in such a case. Firstly, we can use binary brute force search to find the best pair of x and y. Secondly, we can use optimization methods such as gradient descent or Newton's method to find the optimization value pair. In our experiment, for illustration purpose, we just use binary brute force search.

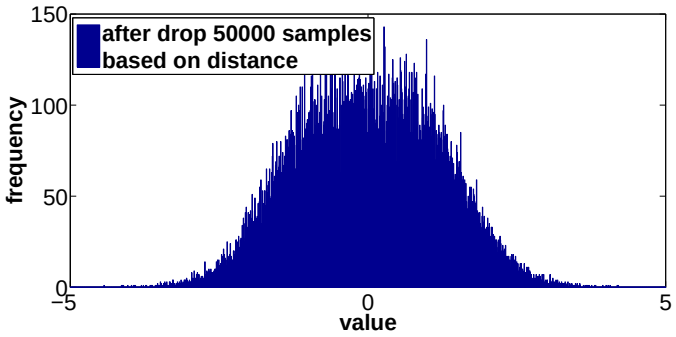


Figure 6.8: Distribution is changed after dropping with different probabilities based on the distance

6.5.3 Distance Based Model

At last, we present the third privacy model which is considered to be an upgraded version of the randomly dropping one. Recall in Fig. 6.5, when the value of the density to uncertainty ratio decreases, the prediction error rate of a system increases. While it is impossible to increase the network density through filtering method, there might be a chance to increase the uncertainty through filtering method. Instead of randomly dropping traces, given higher probability to drop those vehicles which are closer to their expected location, the underline standard deviation of the distribution get changed accordingly. In particular, it is moving in the direction of increasing. The following figures explain this point clearly. In Fig. 6.6, 100000 samples from normal distribution are randomly generated and the resulted histogram is shown. Fig. 6.7 is drawn after half of the samples are removed. As can be seen, the overall distribution is similar to the previous one shown in Fig. 6.6. However, while removing half of the samples based on a normal distribution of $\mu = 0$ and $\sigma = 0.75$, in Fig. 6.8 the underline distribution is changed and the resulted "standard deviation" (uncertainty) increases.

Thus, the third proposed model is called distance based model. It is expected to provide better privacy protection level compared to the randomly dropping one under the same data releasing rate.

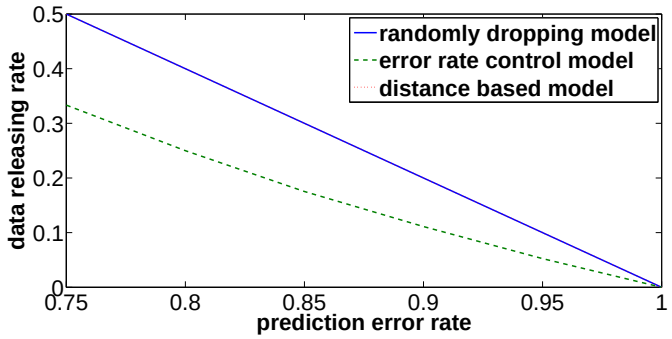


Figure 6.9: The origin system prediction rate is 0.5. This figure shows the data releasing rate of the three proposed models in order to improve the prediction error rate to 0.75 and up.

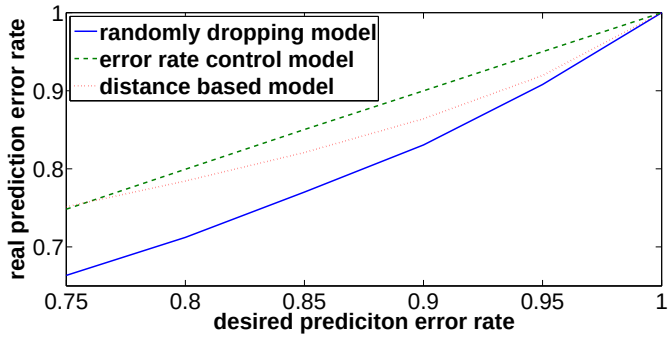


Figure 6.10: The original system prediction rate is 0.5. This figure shows the real prediction rate of the three privacy models after data filtering.

6.6 Simulation Dataset Based Performance Comparison

In this section, we evaluate the performance of the proposed models using the same simulation data from section 6.4. The basic simulation process is done through the following steps: 1) Adjust the value of ratio to achieve a desired original system prediction error rate; 2) Define a desired target prediction error rate; 3) Adjust the value of x. In both the randomly dropping and the distance based dropping models, the value of x is directly computed through equation 6.1, while in error rate control model, the value of x is obtained through equation 6.2; 4) Randomly drop traces. In both the randomly dropping and the error control models, we directly drop traces according to the value of x while in the distance based method, we drop the traces based on the distance between a vehicle expected location and its true location; 5) the resulted prediction error rate is obtained by adding up cases that there is any other vehicle appearing between the

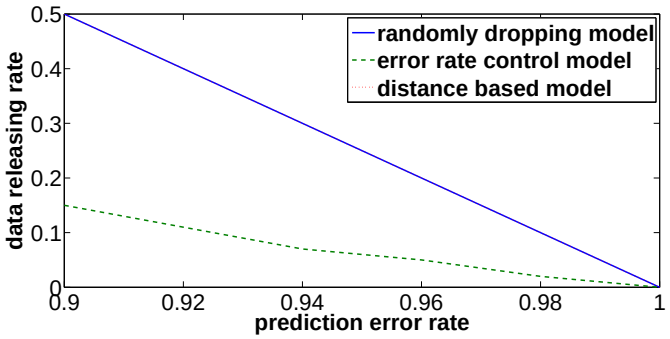


Figure 6.11: The origin system prediction rate is 0.8. This figure shows the data releasing rate of the three proposed models in order to improve the prediction error rate to 0.9 and up.

expected location of a vehicle and its true location.

In the first case, we assume the system originally has a ratio of 2.2 and a prediction error rate of 0.5. All three privacy models target on reducing the number of traces that can be identified by half or less, which corresponds to a target prediction error rate of 0.75 and up. Fig. 6.9 shows the data releasing rate varying while changing the targeted prediction error rate. As can been seen, the randomly dropping model and the distance based model have the same data releasing rate over different values of ratio. The rate control scheme release much less data than the other two. However, it almost achieves the target privacy protection level perfectly as shown in Fig. 6.10. Thus the rate control model sacrifices more application performance in order to have privacy well protected. The advantage of distance based model over the randomly dropping one is obvious: without dropping more location traces, the distance based model can achieve higher privacy protection level.

The difference of the three models becomes even more pronounced in Fig. 6.11 and Fig. 6.12, in which the models are trying to improve the prediction error rate from 0.8 to 0.9 and up with original density to uncertainty ratio at 0.54. When the original system's density to uncertainty ratio is low, distance based dropping model shows more significant advantage over the randomly dropping model. To some extent, this also can be explained from Fig. 6.5. When the original ratio is low, any small difference in the data releasing rate among the models can cause a significant value change in prediction

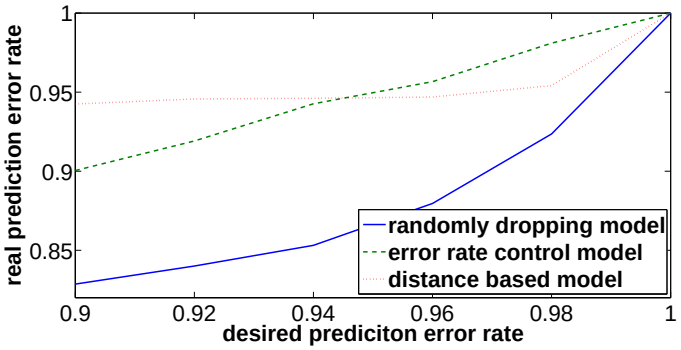


Figure 6.12: The original system prediction rate is 0.8. This figure shows the real prediction rate of the three privacy models after data filtering.

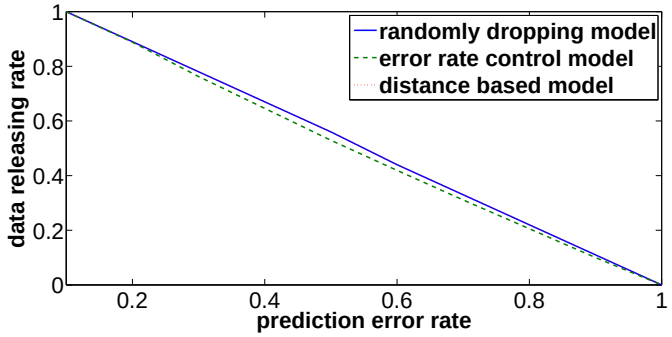


Figure 6.13: The origin system prediction rate is 0.1. This figure shows the data releasing rate of the three proposed models in order to improve the prediction error rate to 0.2 and up.

error rate. In the rate control model, the change of error rate leads to further adjustment on data releasing rate.

In the end, we study a case in which the initial network density to uncertainty ratio is 13.5 and the prediction error rate is 0.1. Fig. 6.13 and Fig. 6.14 show the corresponded data releasing rate and the consequential prediction error rate respectively. As can been seen from the figures, the difference among the three proposed models is very small. The distance based model performs closely to the rate control one in terms of privacy, at the same time, it also achieves the similar performance in application domain as the randomly dropping model.

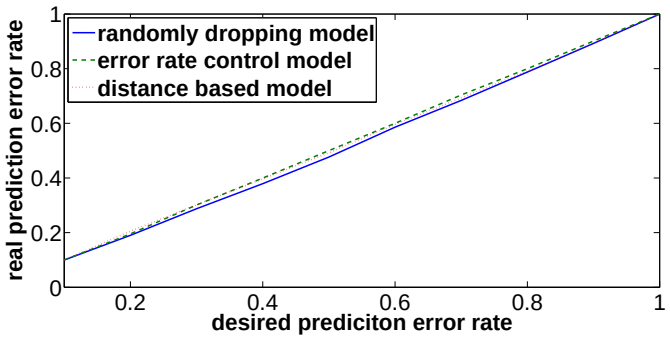


Figure 6.14: The original system prediction rate is 0.1. This figure shows the real prediction rate of the three privacy models after data filtering.

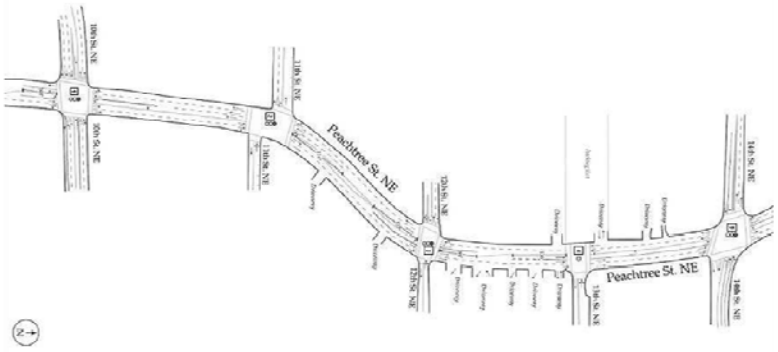


Figure 6.15: The road map of NGSIM traces.

6.7 Real Trace Based Performance Comparison

In this section, the three proposed models are compared using real world vehicle traces.

The evaluation was done using NGSIM [117] dataset collected at Peachtree St in Atlanta, Georgia. Vehicle traces were collected using recognition techniques via video images, which provides an continuous tracking (with a 10-Hz sampling frequency) and full penetration of the real traffic flow. The geometry of the network is shown in Fig. 6.15. Data collected between $4:00pm$ and $4:15pm$ are used in this experiment.

Similar to what we have done on the simulation data, firstly we fix the network system's original prediction error rate by adjusting the value of ratio (Note, we only need adjust the value of standard deviation to obtain desired value of ratio). Secondly, we drop the traces according to the results obtained from equation 6.1 and equation 6.2 based on predefined target prediction error rate. Finally, add up the cases that an

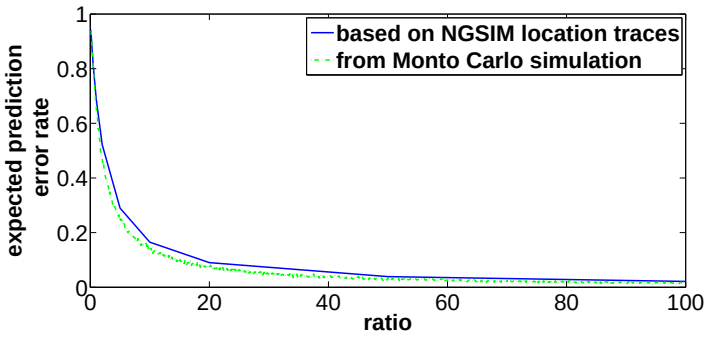


Figure 6.16: Prediction error rate V.S. density to uncertainty ratio.

adversary will make mistake on linking the true location and expected location pair to get the system prediction error rate.

In Fig. 6.16, the prediction error rate of the original NGSIM traces is plotted along with the one obtained from Monte Carlo simulation before any privacy protection model applied. Since it is hard to get detailed resolution as from the simulation, this curve is derived from 10 points only, which is enough to do the plot. The NGSIM data shows the same trend as the curve drawn from the Monte Carlo simulation. A small shift of the curve to the right indicates that the traces from NGSIM dataset are more difficult to be predicted. This is because in the simulation, we assume the average distance is fixed and the uncertainty of a vehicle's future location follows Gaussian distributions. However, the reality might differ, which makes more difficult for an adversary to make correct prediction.

Since the difference is so small that we expect to replace the prediction error rate curve of a real world scenario with the simulation results, especially when the network topology is simple, such as highway.

Fig. 6.17, 6.18 and 6.19 are the results obtained from NGSIM data by expanding the three cases discussed in section 6.6. Since the prediction error rate curves are very similar as shown in Fig. 6.16, the rate curve obtained from simulation is used as the reference to generate Fig. 6.17 to 6.19. Recall the information of the curve is used in the rate control model. As can been seen, the overall performance of these three models are all a little bit better than the simulation results. This is also consistent with the results shown in Fig. 6.16. Due to the smaller number of cases and dataset can

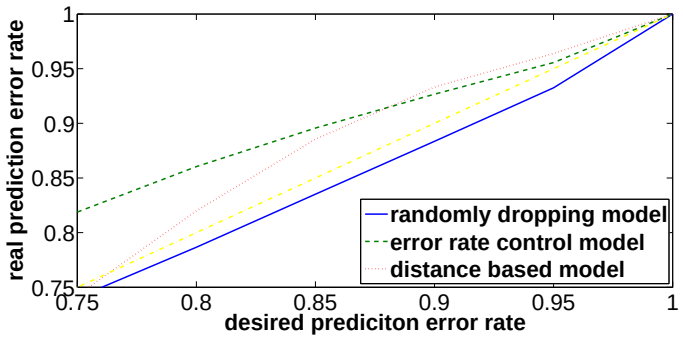


Figure 6.17: The original system error rate is 0.5. The privacy models try to achieve prediction error rate of more than 0.75.

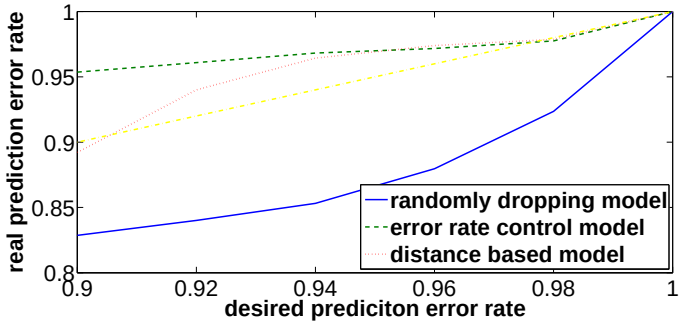


Figure 6.18: The original system error rate is 0.8. The privacy models try to achieve prediction error rate of more than 0.9.

be studied from the NGSIM data, it generates more fluctuated results than the Monte Carlo simulation does in general.

6.8 Conclusion

In this chapter, we extend our work in vehicular networks privacy domain under the assumption that there is no trusted location proxy server and the available information is limited.

The influence of the density to uncertainty ratio on the prediction error rate is evaluated first. The Monte Carlo simulation results provide a proper reference to check the error rate of a network system based on density to uncertainty ratio and help to decide if any privacy model needs to be applied. Then we propose three privacy models: randomly dropping, rate control and distance based models. It appears that if to achieve certain privacy requirement is the first priority, then the rate control model

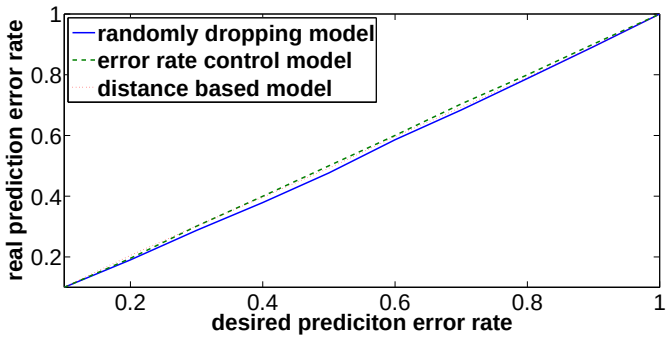


Figure 6.19: The original system error rate is 0.1. The privacy models try to achieve better prediction error rate.

is the best choice among the three. If both the application and the privacy performance need to be considered, then the best choice is the distance based model. We should not use randomly dropping models in either case.

What needs to be emphasized is that in the real world scenario, it is usually much more difficult for an adversary to successfully attack a target vehicle. In this work, we didn't consider the case that adversary models need to continuously track multiple traces, in which the difficulty actually increases exponentially. Therefore, if the ability and the expectation of the imaginary adversary are known to the system, a better designed privacy model can be chosen and consequentially release more location traces. The results in this chapter can be used as a lower bound for application performance (in terms of data releasing rate) of other privacy models, for example a much refined model VTL zone-based path cloaking.

Chapter 7

Conclusion

This thesis presented our research efforts on vehicular sensing networks in the field of efficiency, security and privacy.

On efficiency, in order to reduce the communication overhead in long distance communication such as vehicular to cellular based station we developed a geocache concept and boomerang anchoring protocol to bind sensing data around the event-detected location. The challenge of an anchoring protocol is to maintain the data around the event-detected location efficiently. The trajectory-based boomerang protocol helps returning aggregated data to the event-detected location through the trajectory of a geocache carrier. It has high return probability than a distance-based approach.

On security, we proposed secret key agreement schemes for V2V and V2I communication modes in vehicular sensing networks under the assumption that both modes need separate and independent set of keys. The V2V scheme is based on the channel reciprocity theorem and spatial decorrelation property. It does not need any third trust party for authorization, which better fits the mobile ad hoc scenario. The V2I scheme is a probabilistic based key agreement scheme. Because of the space diversity in vehicular sensing networks, the channel diversity of the radio device's receiving and transmitting frequency and the time diversity of the vehicular moving trajectory, this V2I secret key agreement scheme can achieve high level of security even when adversary is relatively much more powerful than a regular user. Both V2V and V2I schemes have simple design and can be easily implemented in real world.

On privacy, in additional to a privacy protection model proposed in vehicular sensing networks under general assumptions, we also discussed the possible privacy information leakage if detailed location traces are available. Furthermore, we studied the cases when

privacy system model need to be estimated or to be improved under the assumption of very limited resource. Therefore, our work in privacy domain for vehicular sensing networks covers three chapters.

First, a VTL zone-ware path cloaking algorithm was proposed to protect user privacy. It filters the vehicle traces based on entropy value. If the entropy of a trace is lower than the assigned threshold, the trace will be removed from the releasing list. The calculation of the entropy value is a combination of path likelihood and travel time likelihood. Because of recognizing the special characteristics of vehicular networks, the proposed algorithm has better performance than general path cloaking algorithms without zone-awareness. The proposed algorithm is also different from the mix-zone concept. It targets on the applications that location traces are needed only across the intersections-exactly the area where mix-zones are usually placed to hide information.

Second, we studied the impact of driving characteristics on privacy information leakage. We showed that different classes of vehicles can be identified through driving characteristics extracted from location information. We also demonstrated that it is possible to distinguish outlier through special driving patterns. This helps an adversary to link anonymous segments of location traces and eventually recover complete trips.

Third, we studied the impact on prediction error rate by the ratio of the network density to the uncertainty of a vehicle's future location. The result drawn from Monte Carlo simulation indicates that estimating the privacy protection capability of a network system can be done if the density to uncertainty ratio is known. Decision on whether a privacy model needs to be applied can be quickly concluded when such information is available. Three privacy models namely: randomly dropping, error rate control and distance based models were presented. They are designed under the assumption that there is no location proxy server and the available information is limited. The error rate control model is preferred when privacy requirement is the first priority, while the distance based model is preferred when privacy and performance are equally important. Finally, the proposed privacy models can be used as references for future privacy model designing when more information and resource are available.

References

[1] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *In Proceeding of 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004, pp. 187–198.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *ACM SIGOPS Operating Systems Review*, vol. 8, pp. 96–107, 2002.

[3] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proc. of 4th international conference on Embedded networked sensor systems*, 2006, pp. 125–138.

[4] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *Wireless Communications, IEEE*, vol. 13, no. 5, pp. 52 –57, 2006.

[5] M. G. P. B. U. Lee, E. Magistretti and A. Corradi, "Opportunistic dissemination and harvesting of urban monitoring information in vehicular sensor networks," in *UCLA, Tech. Rep*, 2007.

[6] B. M. Oki, M. Pfluegl, A. Siegel, and D. Skeen, "The information bus@an architecture for extensible distributed systems," in *Proceedings of the 14th ACM Symposium on Operating Systems Principles*, 1993, pp. 58–68.

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976. [Online]. Available: citeseer.ist.psu.edu/diffie76new.html

[8] P. A, S. R, W. V, C. D, and T. J, "Spins: Security protocols for sensor networks," *Wireless Nets*, vol. 8, no. 5, pp. 521–534, 2002.

[9] J. Steiner and J. I. Schiller, "An authentication service for open network systems," in *Usenix Conference Proceedings*, 1988, pp. 191–202.

[10] D. Otway and O. Rees, "Efficient and timely mutual authentication," *SIGOPS Oper. Syst. Rev.*, vol. 21, no. 1, pp. 8–10, 1987.

[11] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 335–338.

[12] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *ACM Conference on Computer and communication Security(CCS)*, 2004.

[13] A. C. f. Chan, "Distributed symmetric key management for mobile ad hoc networks," *IEEE INFOCOM*, vol. 4, pp. 2414–2424, 2004.

[14] B. Zan and M. Gruteser, "Random channel hopping schemes for key agreement in wireless networks," in *PIMRC '09: Proceedings of the 20th Personal, Indoor and Mobile Radio Communications Symposium 2009*, Tokyo, Japan, 2009.

[15] X. Ban, P. Hao, and Z. Sun, "Real time queue length estimation for signalized intersections using sample travel times," *Transportation Research Part C, in press*, 2011.

[16] F. Dotzer, "Privacy issues in vehicular ad hoc networks," in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. ACM Press, 2005.

[17] B. Hoh and M. Gruteser, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *Proceedings of ACM CCS*, 2007.

[18] S. Rass, S. Fuchs, M. Schaffer, and K. Kyamakya, "How to protect privacy in floating car data systems," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, ser. VANET '08. New York, NY, USA: ACM, 2008, pp. 17–22. [Online]. Available: http://doi.acm.org/10.1145/1410043.1410047

[19] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, 2002.

[20] V. Ciriani, S. D. C. di Vimercati, S. Foresti, and P. Samarati, "*k*-anonymity," *Secure Data Management in Decentralized Systems*, pp. 323–353, 2007.

[21] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "Random-data perturbation techniques and privacy-preserving data mining," *Knowl. Inf. Syst.*, vol. 7, no. 4, pp. 387–414, May 2005. [Online]. Available: http://dx.doi.org/10.1007/s10115-004-0173-6

[22] M. Gruteser, D. Grunwalddepartment, and C. Science, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM MobiSys*, 2003, pp. 31–42.

[23] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, 2005, pp. 620 –629.

[24] A. R. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW04*, 2004, pp. 127–131.

[25] J. Freudiger, M. Raya, M. Felegyhazi, P. Papadimitratos, and J.-P. Hubaux, "Mix-Zones for Location Privacy in Vehicular Networks," in *Proceedings of ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007.

[26] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08, 2008, pp. 15–28.

[27] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *The Sixth Annual International conference on Mobile Systems, Applications and Services (MobiSys 2008)*, Breckenridge, U.S.A., June 2008.

[28] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, no. 6, pp. 30 –39, nov/dec 2001.

[29] D. Raychaudhuri, M. Ott, and I. Secker, "Orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*, feb. 2005, pp. 308 – 309.

[30] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "Ght: A geographic hash table for datacentric storage," in *Proc. of 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 78–87.

[31] Y. Ni, U. Kremer, A. Stere, and L. Iftode, "Programming ad-hoc networks of mobile and resource-constrained devices," in *Proc. of the 2005 ACM SIGPLAN conference on Programming language design and implementation*, 2005, pp. 249–260.

[32] I. Vasilescu, K. Kotay, and D. Rus, "Data collection, storage, and retrieval with an underwater sensor network," in *Proc. of 3rd International conference on Embedded networked sensor systems*, 2004, pp. 154–165.

[33] J. Li, J. Jannotti, D. Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of 6th annual international conference on Mobile computing and networking*, 2000, pp. 120–130.

[34] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "Routing for vehicle-based disruption-tolerant networks," in *Proc. of 25th IEEE International Conference on Computer Communications*, 2006, pp. 1–11.

[35] Q. Li, D. Rus, M. Dunbabin, and P. Corke, "Sending messages to mobile users in disconnected ad-hoc wireless networks," in *Proc. of 6th annual international conference on Mobile computing and networking*, 2000, pp. 44–55.

[36] L. Briesemeister and G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *Proc. of 1st ACM international symposium on Mobile ad hoc networking and computing*, 2000, pp. 45–50.

[37] R. H. Frenkiel, B. R. Badrinath, J. Borres, and R. D. Yates, "The infostations challenge: balancing cost and ubiquity indelivering wireless data," *Personal Communications, IEEE*, vol. 7, pp. 66–71, 2000.

[38] Y. cai and T. Xu, "Design, analysis, and implementation of a large-scale real-time location-based information sharing system," in *Proc. of 6th International conference on Mobile Systems, Applications, and Services*, 2008.

[39] H. Lu, N. Lane, S. Eisenman, and A. Campbell, "Bubble-sensing: A new paradigm for binding a sensing task to the physical world using mobile phones," in *Proc. of International Workshop on Mobile Device and Urban Sensing*, 2008.

[40] L. Chisalita and N. Shahmehri, "A peer-to-peer approach to vehicular communication for the support of traffic safety applications," in *Proc. of 5th IEEE International Conference on Intelligent Transportation Systems*, 2002, p. 336C341.

[41] Y. B. Ko and N. H. Vaidya, "Flooding-based geocasting protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 7, pp. 471–480, 2002.

[42] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proc. of 4th ACM international symposium on Mobile ad hoc networking and computing*, 2003, pp. 233–244.

[43] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "Carnet: A scalable ad hoc wireless network system," in *Proc. of 9th ACM SIGOPS European Workshop*, 2000, pp. 61–65.

[44] C. Maihofer, T. Leinmller, and E. Schoch, "Abiding geocast: Time-stable geocast for ad hoc networks," in *Proc. of 2nd ACM international workshop on Vehicular ad hoc networks*, 2005, pp. 20–29.

[45] J. LeBrun, C.-N. Chuah, D.Ghosal, and M.Zhang, "Knowledge-based opportunistic forwarding invehicular wireless ad hoc networks," in *Proc. of the 61st IEEE conference on Vehicular Technology*, 2005.

[46] I. Leontiadis and C. Mascolo, "Geopps: Geographical opportunistic routing for vehicular networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, june 2007, pp. 1 –6.

[47] A. Sistla, O. Wolfson, and B. Xu, "Opportunistic data dissemination in mobile peer-to-peer networks," in *Advances in Spatial and Temporal Databases*, ser. Lecture Notes in Computer Science, C. Bauzer Medeiros, M. Egenhofer, and E. Bertino, Eds. Springer Berlin / Heidelberg, 2005, vol. 3633, pp. 923–923.

[48] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1-6, pp. 91 – 108, 2000.

[49] J. S. Greenfeld, "Matching gps observations to locations on a digital map," *81th Annual Meeting of the Transportation Research Board*, vol. 1, no. 3, pp. 164–173, 2002.

[50] M. Quddus, W. Ochieng, L. Zhao, and R. Noland, "A general map matching algorithm for transport telematics applications," *GPS Solutions*, vol. 7, pp. 157–167, 2003.

[51] "Dsrc standards: What's new?" Retrieved 2008-02-17. [Online]. Available: http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm

[52] M. Al-Shurman and S.-M. Yoo, "Key pre-distribution using mds codes in mobile ad hoc networks," in *Information Technology: New Generations, 2006. ITNG 2006. Third International Conference on*, april 2006, pp. 566 –567.

[53] C. Castelluccia, N. Saxena, and J. H. Yi, "Self-configurable key pre-distribution in mobile ad hoc networks," in *in: IFIP Networking Conference*, 2005, pp. 1083–1095.

[54] B. Zan, M. Gruteser, and F. Hu, "Improving robustness of key extraction from wireless channels with differential techniques," in *ICNC '12: Proceedings of the International Conference on Computing, Networking and Communications 2012*, Maui, Hawaii, USA, 2012.

[55] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: extracting a secret key from an unauthenticated wireless channel," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2008, pp. 128–139.

[56] B. Azimi-Sadjadi, A. Kiayias, A. Mercado, and B. Yener, "Robust key generation from signal envelopes in wireless networks," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 401–410.

[57] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2009, pp. 321–332.

[58] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *Antennas and Propagation, IEEE Transactions on*, vol. 53, no. 11, pp. 3776–3784, Nov. 2005.

[59] M. J. Miller and N. H. Vaidya, "Leveraging channel diversity for key establishment in wireless sensor networks," April 2006, pp. 1–12.

[60] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 3, march 2005, pp. 1664 – 1669 Vol. 3.

[61] "Lecture notes in computer science."

[62] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data." Springer-Verlag, 2004, pp. 523–540.

[63] "unmanned aerial vehicle," in *The Free Dictionary*, 2011. [Online]. Available: http://www.thefreedictionary.com/Unmanned+Aerial+Vehicle

[64] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[65] (2009) Discrete logarithm. [Online]. Available: http://en.wikipedia.org/wiki/Discrete_logarithm_problem

[66] R. C. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, 1978.

[67] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[68] N. Koblitz, "An elliptic curve implementation of the finite field digital signature algorithm," in *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology.* London, UK: Springer-Verlag, 1998, pp. 327–337.

[69] F. J. MacWilliams and N. Sloane, *Theory of Error-correcting Codes.* North-Holland, 1977.

[70] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms." Academic Press, 1978, pp. 169–177.

[71] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy.* Washington, DC, USA: IEEE Computer Society, 2003, p. 197.

[72] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[73] J. Hwang and Y. Kim, "Revisiting random key pre-distribution schemes for wireless sensor networks," in *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks.* New York, NY, USA: ACM, 2004, pp. 43–52.

[74] C. Blundo, L. A. F. Mattos, and D. R. Stinson, "Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution," in *CRYPTO*, 1996, pp. 387–400.

[75] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *CRYPTO*, 1992, pp. 471–486.

[76] T. Matsumoto and H. Imai, "On the key predistribution system: A practical solution to the key distribution problem," in *CRYPTO '87: A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology.* London, UK: Springer-Verlag, 1988, pp. 185–193.

[77] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[78] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security.* New York, NY, USA: ACM, 2003, pp. 52–61.

[79] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach," Nov. 2003, pp. 326–335.

[80] R. Di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *SASN '03: Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks.* New York, NY, USA: ACM, 2003, pp. 62–71.

[81] A. D. Wyner, "The wire-tap channel," vol. 54, no. 8, pp. 1355–1387, 1975.

[82] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 339–348, May 1978.

[83] U. M. Maurer, "Secret key agreement by public discussion from common information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, 1993.

[84] U. M. Maurer and S. Wolf, "Unconditionally secure key agreement and the intrinsic conditional information," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 499–514, 1999.

[85] ——, "Towards characterizing when information-theoretic secret key agreement is possible," in *ASIACRYPT*, 1996, pp. 196–209.

[86] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," in *Ultra-Wideband, 2007. ICUWB 2007. IEEE International Conference on*, Sept. 2007, pp. 270–275.

[87] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *Communications, IEEE Transactions on*, vol. 43, no. 1, pp. 3–6, Jan 1995.

[88] A. Kitaura, T. Sumi, K. Tachibana, H. Iwai, and H. Sasaoka, "A scheme of private key agreement based on delay profiles in uwb systems," March 2006, pp. 1–6.

[89] Z. Li, W. Xu, R. Miller, and W. Trappe, "Securing wireless systems via lower layer enforcements," in *WiSe '06: Proceedings of the 5th ACM workshop on Wireless security.* New York, NY, USA: ACM, 2006, pp. 33–42.

[90] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the ether: Using the physical layer for wireless authentication," in *Communications, 2007. ICC '07. IEEE International Conference on*, June 2007, pp. 4646–4651.

[91] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 64–78.

[92] J. Krumm, "Inference attacks on location tracks," in *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), volume 4480 of LNCS*. Springer-Verlag, 2007, pp. 127–143.

[93] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems," *Pervasive Computing, IEEE*, vol. 5, no. 4, pp. 38 –46, 2006.

[94] L. Buttyan, T. Holczer, and I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in vanets," in *Proceedings of Workshop on Security and Privacy in Ad hoc and Sensor Networks*, 2007.

[95] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, "Swing & swap: user-centric approaches towards maximizing location privacy," in *Proceedings of the 5th ACM WPES' 06*.

[96] X. Ban, R. Herring, P. Hao, and A. Bayen, "Delay pattern estimation for signalized intersections using sampled travel times," in *Transportation Research Record*, 2009, pp. 109–119.

[97] P. Richards, "Shock waves on the highway," in *Operations Research 4*, 1956, pp. 42–51.

[98] M. Lighthill and G. Whitham, "On kinematic waves i flood movement in long rivers. ii a theory of traffic flow on long crowded roads," in *In Proceedings of Royal Society (London) A229*, 1955, pp. 281–345.

[99] D. Barth, "the bright side of sitting in traffic: Crowdsourcing road congestion data." [Online]. Available: http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html

[100] M. A. P. S. Susilawati, Taylor and Sekhar, "Travel time reliability measurement for selected corridors in the adelaide metropolitan area," *Journal of Eastern Asia Society for Transportation Studies*, 2010.

[101] "Fitting a univariate distribution using cumulative probabilities." [Online]. Available: http://www.mathworks.com/products/statistics/demos.html?file=/products/demos/shipping/stats/

[102] Z. Sun, B. Zan, J. Ban, M. Gruteser, and P. Hao, "Evaluation of privacy preserving algorithms using traffic knowledge based adversary models," *to be presented in 14th International IEEE Conference on Intelligent Transportation Systems (ITSC 2011)*, 2011.

[103] H. Liu and S. Jabari, "Evaluation of corridor traffic management and planning strategies using microsimulation: a case study," *Transportation Research Record*, pp. 26–35, 2008.

[104] D. Chaum, C. O. T. Acm, R. Rivest, and D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, pp. 84–88, 1981.

[105] M. F. Mokbel, C. yin Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *In VLDB*, 2006, pp. 763–774.

[106] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: Traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, p. 2004, 2004.

[107] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, pp. 843–854, 1979.

[108] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Proceedings of the 7th International Conference on Pervasive Computing*, ser. Pervasive '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 390–397. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-01516-8_26

[109] D. Barth, "the bright side of sitting in traffic: Crowdsourcing road congestion data." [Online]. Available: http://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html

[110] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Achieving guaranteed anonymity in gps traces via uncertainty-aware path cloaking." *IEEE Trans. Mob. Comput.*, vol. 9, no. 8, pp. 1089–1107, 2010. [Online]. Available: http://dblp.uni-trier.de/db/journals/tmc/tmc9.html#HohGXA10

[111] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46 – 55, 2003.

[112] M. Dahl, S. Delaune, and G. Steel, "Formal analysis of privacy for vehicular mix-zones," in *Proceedings of the 15th European conference on Research in computer security*, ser. ESORICS'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 55–70. [Online]. Available: http://portal.acm.org/citation.cfm?id=1888881.1888886

[113] A. M. Carianha, L. P. Barreto, and G. Lima, "Improving location privacy in mix-zones for vanets," in *Proceedings of the 30th IEEE International Performance Computing and Communications Conference*, ser. PCCC '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/PCCC.2011.6108111

[114] B. Palanisamy and L. Liu, "Mobimix: Protecting location privacy with mix-zones over road networks," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ser. ICDE '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 494–505. [Online]. Available: http://dx.doi.org/10.1109/ICDE.2011.5767898

[115] F. E. Grubbs, *Procedures for Detecting Outlying Observations in Samples*. Defense Technical Information Center, 1974. [Online]. Available: http://books.google.com/books?id=jDS-NwAACAAJ

[116] (2012, accessed Sept. 14,) National transportation statistics 2012. online.

[117] (2012, accessed Sept. 14,) Next generation simulation (ngsim). online.

[118] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/

[119] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," Tech Report HPL-2003-4, HP Laboratories, Tech. Rep., 2004. [Online]. Available: http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf