ACHIEVING GUARANTEED ANONYMITY IN TIME-SERIES LOCATION DATA

BY BAIK HOH

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

Prof. Marco Gruteser

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2008

ABSTRACT OF THE DISSERTATION

Achieving Guaranteed Anonymity in Time-Series Location Data

by Baik Hoh

Dissertation Director: Prof. Marco Gruteser

Collaborative sensing networks anonymously aggregate location-tagged sensing information from a large number of users to monitor environments. However, sharing anonymous location-tagged sensing information from users raises serious privacy concern. Rendering the location traces anonymous before sharing them with application service providers or third parties often allows an adversary to follow anonymous location updates because a time-series of anonymous location data exhibit a spatio-temporal correlation between successive updates. Prior privacy techniques for location data such as spatial cloaking techniques based on k-anonymity and best-effort algorithms do not meet both data quality and privacy requirements at the same time. This raises the problem of guaranteed anonymity in a dataset of location traces while maintaining high data accuracy and integrity.

To overcome these challenges, we develop a novel privacy metric, called *Time-To-Confusion* to characterize the privacy implication of anonymous location traces and propose two different privacy-preserving techniques that achieve both the guaranteed location privacy of all users and high data quality. The *Time-To-Confusion* effectively captures how long an adversary can follow an anonymous user at a specified level of

confidence, given system parameters such as location accuracy, sampling frequency, and user density. Two different privacy mechanisms are designed with and without a trustworthy location privacy server in a time series of location updates. In the first solution, we propose an uncertainty-aware path cloaking algorithm in a trustworthy privacy server that determines the release of user location updates based on tracking uncertainty and maximum allowable tracking time. Our second solution does not require users to trust the centralized privacy server. Instead, we propose the novel concept of virtual trip lines where vehicles update their location and sensing information. This concept enables temporal cloaking in a distributed architecture where no single entity accesses all of identity, location, and timestamp information, yet incurring only a slight degradation of service quality. We evaluate two proposed algorithms with a case study of automotive traffic monitoring applications. We show that our proposed solutions effectively suppress worst case tracking bounds and home identification rates, while achieving significant data accuracy improvements.

Acknowledgements

I would like to express my deep gratitude and appreciation to my advisor, Professor Marco Gruteser. Throughout four years with him, his support, supervision, and inspiration encouraged me to finish my dissertation even though I was not exposed at all to privacy research before I met him.

My sincere thanks are extended to the members of my committeeDr. Hui Xiong, Dr. Roy Yates, and Dr. Yanyong Zhangfor their valuable suggestions for improving this research. Moreover, I would like to thank also to Dr. Wade Trappe and Dr. Dipankar Raychaudhuri who were with my committee members at my thesis proposal presentation.

Above all, infinite thanks go to Dr. Quinn Jacobson and Professor Murali Annavaram at USC. Working with them for nearly a year helps me find exciting aspects of industry career. I wish to extend my thanks to Professor Hui Xiong and Dr. Ansaf Alrabady at General Motors Research. The collaboration with them contributes to my thesis, especially Chapter 4 and 7. I am lucky to have invaluable massive dataset of anonymous GPS traces. Thanks to volunteers that consent to share their anonymous traces for research purposes.

I want to thank deeply to Dr. Jaewon Kang at Telcordia Research, my mentor and an alumni of WINLAB. Since I joined Rutgers in 2003, he has been a great mentor to me both in academic and personal life. I send my thanks to My WINLAB colleagues, Sangho Oh, Kishore Ramchandran, Mesut Ali Ergin, and Gayathri Chandrasekaran. Studying with them at WINLAB always gives me a pride.

Lastly, I thank deeply to my wife, Hana You, a well-trained and -disciplined biologist. I would never have completed my degree without her loving support. Every presentation practice with her improves my presentation skill, finally helping me

ready and confident with my Ph.D. thesis defense presentation. Her advices based on the lessons that she learned from NYU, Cornell, and Rockafellar Univ. inspired me througout my Ph.D. life with her. My deepest thanks go to my family PhDs, Dr. Joo Hoh, Professor Eunha Hoh at SDSU, and my mother, Younghee Lee who supported three PhDs during all her life. Their experiences and advices always encourage me to concentrate my PhD study. My elder syster, Inyoung Hoh and her husband, Joosang Park have been always my supporters and have always made me feel the love of family.

Dedication

To my wife, Hana You, my lifelong lover and companion

To my prarents, Dr. Joo Hoh and Younghee Lee, my great supporters

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

> While many people are focused on making computers do more, a few of us are focused on technology for ensuring that there are certain things computers will not do, such as invade your privacy. [87]
>
> – David Chaum

Due to the increasing prevalence of Global Positioning System (GPS) chips in consumer electronics and advances in wireless networking, *GPS traces* from a large number of individual users can be easily collected and shared. The availability of these traces has brought about a new class of mobile sensing networks called *collaborative sensing*, also called *participatory sensing* [8] or *community sensing* [71]. Collaborative sensing networks anonymously aggregate location-tagged sensing information from a large number of users to monitor environments. Examples include environmental monitoring applications (e.g., TIER [4], N-SMARTS [5], and Participatory Urbanism [9]), automotive traffic monitoring applications [57], and mobile worm/virus propagation monitoring [55].

These location traces, however, give rise to privacy concerns, because location traces can reveal visits to sensitive or private places (e.g., home, medical clinics) and associated information such as time of day or speed of travel.[1] Privacy of user location traces can be enhanced through standard data protection techniques such as policy-based disclosure, access control, or encryption. Unfortunately, these techniques are not feasible when a database is governed by an untrustworthy service provider, is publicly released to third parties, or is accidentally (or maliciously) made insecure by insiders.

[1] GPS technology can provide 10 to 15 meter accuracy that is enough accurate to pinpoint your house in dense populated residential area.

Because these applications do not depend on specific user identity information of location traces, at first glance, anonymization is of particular interest for applications that aggregate data from many users. Anonymization of location traces, however, poses special challenges because the high spatio-temporal correlation in the time-series nature of a GPS trace often allows re-identification of users. For example, it is often straightforward to identify the home or work locations based on a GPS trace, providing means to reidentify the user. Thus, removing identity information from a trace only provides weak anonymity. Achieving strong anonymity in a dataset of location traces is of concern as cheaper GPS chips are introduced and more location-based services are getting popular in commercial markets.

Existing solutions for strong anonymity have been influenced by database privacy and anonymous networking domains. In the area of database privacy, similar de-anonymization threats have been discovered such as in the Netflix dataset [76] and in public anonymous census data [88]. The need for strong anonymity in a publicly released database has motivated the development of k-anonymity algorithms [88, 80]. Stemming from k-anonymity concept, several spatial cloaking algorithms [51, 47, 75] have been known to provide strong anonymity. In addition, many best-effort approaches [24, 81, 74, 54, 26] based on the concept of David Chaum's MIX [30] create confusion to prevent an adversary from linking anonymous location updates.

However, existing techniques cannot meet both privacy requirements and accuracy requirements at the same time. Spatial cloaking algorithms provide strong anonymity, but they result in large spatial error that cannot be acceptable in automotive traffic monitoring applications. Meanwhile, many best-effort approaches effectively create confusion among anonymous traces in high user density areas, but they cannot guarantee location privacy in low density areas under the multi-target tracking threats [53]. This study aims for achieving the notion of guaranteed privacy regardless of user density, while meeting data integrity requirements. Data integrity can be achieved only if both data accuracy and data authenticity are guaranteed. We thus design privacy mechanisms that modify original location traces as little as possible and balance anonymity against authentication of originator of sensing information.

We observe that the exact privacy implications of anonymous GPS traces depend on how long an adversary tracks an anonymous user. The linkability of anonymous location updates is subject to many factors, especially location accuracy, sampling frequency, user density, and what knowledge (e.g., map information, personal information about the specific user to be tracked) is available to an adversary. Motivated by a well-known target tracking algorithm, Multi-Target Tracking (MTT) [79], we formalize the tracking attack to investigate the privacy threats of given traces. We introduce a formal privacy metric that captures the effect of the factors described above on the tracking time duration where tracking uncertainty remains lower than an uncertainty threshold. We call it *Time-To-Confusion*.

To achieve strong anonymity, we provide two different privacy preserving schemes: a centralized approach that requires a trustworthy privacy server and a distributed approach that does not requires a trustworthy privacy server. First, we develop a special location disclosure control algorithm that achieves a strong degree of anonymity, in other words, that limits *Time-To-Confusion* to less than the predefined tracking time threshold. The algorithm is deployed at a trustworthy privacy server between clients and application service providers. Specifically, we contribute toward

- introduction of a novel time-to-confusion metric to evaluate privacy in a set of location traces. This metric describes how long an individual vehicle can be tracked.
- development of an uncertainty-aware privacy algorithm that can guarantee a specified maximum time-to-confusion.
- demonstration through experiments on real-world GPS traces that this algorithm limits maximum time-to-confusion while providing more accurate location data than a random sampling baseline algorithm.

The centralized approach, however, still requires users to trust a centralized privacy server. To relax this requirement, we propose a novel traffic monitoring system design based on the concept of *virtual trip lines (VTLs)* and experimentally evaluate its accuracy. Virtual trip lines are geographical markers stored in the client, that trigger a

position and speed update whenever a probe vehicle passes. They enable the distribution of identity, location, and timestamp information among multiple entities so that no single entity owns all of them. Furthermore, through privacy-aware placement of these trip lines, clients need not rely on a trustworthy server. The key contributions of our second approach are:

- showing that sampling in space (through virtual trip lines) rather than in time leads to increased privacy by facilitating a distributed monitoring architecture where no single entity possesses an identity and accurate location information.
- designing a privacy-aware placement algorithm that creates the virtual trip line database.
- demonstrating that the virtual trip line concept can be implemented on a GPS-enabled cellular phone platform.
- evaluating accuracy and privacy through a 20 vehicle experiment on a highway segment.

1.1 Collaborative Sensing Applications

Collaborative sensing applications rely on the availability of periodic location updates provided by ever more cost-effective GPS chips. The applications that actively use GPS traces are not limited to intelligent transportation systems. Applications such as the following that make GPS traces available to external service providers can benefit from our study on guaranteed privacy in location traces:

- Traffic monitoring applications [41, 13, 98, 60]: Instead of camera or loop detectors on the roads, probe vehicles, which are equipped with GPS and sensors, are expected to be used in many traffic monitoring systems [57]. Our proposed privacy technique could enhance privacy protection, thereby increasing user participation rates in such schemes.

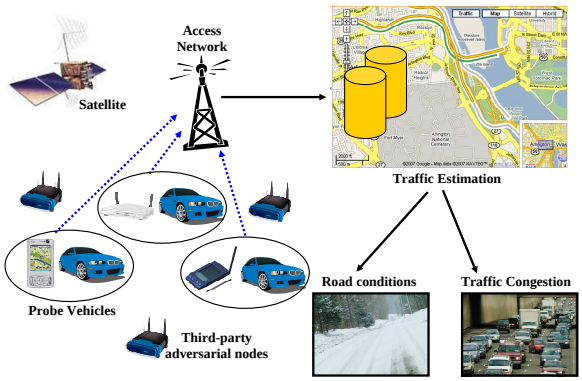


Figure 1.1: Traffic monitoring architecture comprises three entities: probe vehicles, communication service provider, and traffic monitoring service provider. Main tasks of traffic monitoring include building a real-time congestion map and sensing road conditions in real-time.

- Pollution monitoring applications: Sensors embedded into a cellphone can measure the level of air pollution. For example, the measurements from a large number of cellphone users enable to derive carbon dioxide concentrations over a wide area. [15, 19, 5, 9]

- "Pay as you drive" insurance [92]: This approach allows auto insurance carriers to customize insurance premiums to individual driving patterns. In return for potential discounts, drivers let the insurer install a GPS device that provides GPS traces to the insurer. The insurer can use the collected traces to develop a risk assessment model for mileage driven, roads taken, speed, time of day for trips, duration of rest periods, and other factors.

- Pedestrian/Vehicular flow monitoring: The flow information of pedestrians and vehicles is useful for developing human mobility [50]. Human mobility model is invaluable for diverse research areas such as urban planning, epidemic prevention [55], and emergency response. These applications might gain more popularity once indoor localization techniques are prevalent.

Recently, a Nokia-initiated cooperation, SensorPlanet [8] has introduced the concept of using GPS-enabled smartphones as sensors for all above purposes.

The traffic monitoring application that will serve as case study aims to provide estimates of current travel time for routes using real-time traffic flow information. Traffic

flow information is derived from probe-vehicle speed readings on road segments. [61] This approach promises reduction in infrastructure installation and maintenance cost (e.g., cameras or loop detectors), while extending sensing coverage to less traveled roadways [36, 44]. Dai et al. [36] observed that a penetration rate of 5% is required for reliable traffic status estimation.

The probe vehicles use on-board GPS receivers [98] (or GPS-enabled mobile devices) and cellular communications (or WiFi [60]) to periodically report records with the following parameters to traffic information systems: latitude, longitude, time, and speed. A central traffic monitoring system stores them in a database for real-time and historical traffic analysis. From this information the system can estimate current mean vehicle speed, which can be fed into navigation systems or can be used to build a real-time congestion map (e.g., a congestion index). Estimated traffic information can then be broadcasted to subscribers or made available through a web interface, where drivers can access it through their navigation systems or from home or office computers. Figure 1.1 illustrates this architecture.

1.2 Overview of Dissertation

The remainder of this study is organized as follows. After chapter 1 has presented several motivating applications that fall into our research domain, focusing on automotive traffic monitoring applications, we summarize and highlight the prominent related works in chapter 2. Chapter 3 describes security and privacy challenges that might exist for those applications. Chapter 4 elaborates on inference attacks, the focus of our study. Given threat models, we explain our novel privacy metric and design goals in chapter 5. In chapter 6, we propose a basic architecture that provides countermeasures to security and privacy challenges described in chapter 2, mainly developing the architecture for authenticated but anonymous data collection. On top of this proposed architecture, we introduce privacy-preserving mechanisms for preventing an inference attack at untrustworthy service providers or third parties in chapter 7 and 8. Chapter

7 summarizes our case study on developing an uncertainty-aware path cloaking algorithm that is deployed at a trustworthy privacy server and evaluating it on real GPS location traces that are collected from Detroit downtown and suburban areas. Chapter 8 focuses on highway traffic monitoring by identifying possible tracking threats on highway road segments, and it proposes a decentralized approach where no single entity has all knowledge on identity, location, and timestamp information. In chapter 9, we conclude our two different approaches by pointing out areas for further study.

Chapter 2
Background

> Privacy concerns due to the misuse of new technological inventions can be traced back at least to Louis Brandeis article "The right to privacy" addressing photography in 1890. Since then technological advances have posed repeated challenges. Twenty years ago, David Chaum introduced the view of privacy in a networked world.
>
> – An anonymous graduate student

2.1 Related Works

There have been several prototypes for traffic monitoring systems and real deployments in industry. MIT CarTel [60] proposed to use the unused bandwidth of open wireless hotspots to deliver the GPS-based location and speed measurements of probe vehicles to the central server for traffic data mining. Recently, Jakob et al. demonstrate that the similar system is effective in locating potholes in recent study [41]. Previous studies using cell phone based traffic monitoring [28], [84], [91], [97] investigate the use of triangulation-based positioning technology to locate phones, and because of the poor quality position estimate (100m accuracy), vehicle speeds could not be consistently determined. Recently, Yoon et al. [98] propose to use cellular network as a delivery method of GPS-based sensing information from probe vehicles. Since most traffic monitoring applications that have been proposed so far do not depend on the specific identification information about probe vehicles, the anonymization of sensing information has been a solution in practical deployments [13, 14, 12]. However, in recent years, several studies [72, 57, 53] analyzed the privacy risk of GPS traces and found that naive anonymization (i.e., omitting obvious identifiers from a dataset) does

not guarantee anonymity due to a spatio-temporal correlation between periodic location updates. This observation along with recent database privacy compromises (e.g., Netflix [76], AOL search logs [22], Cellular user tracking [64]) raises an urgent need for stronger protection mechanisms.

Many different research communities such as networking, pervasive computing, cryptography, and data mining have addressed the problem of location privacy. Early technological solutions for data privacy include data encryption for communication and storage, and operating system and database access control and auditing. Over the past few decades, as information technology has permeated our lives, several new research directions have influenced the development of techniques for GPS traces.

k-anonymity. The problem of guaranteeing anonymity in database has long been paid attention since k-anonymity [80, 88] was proposed, but a solution or a privacy metric has not been studied for time-series location dataset. The k-anonymity concept has been easily deployed for location-based services [51, 75, 47]. As shown in section 4.2, these solutions can provide sufficient accuracy for applications such as point-of-interest queries in high density scenarios, but they do not achieve the high accuracy requirements of traffic monitoring applications with low penetration rates. In addition, a series of cloaking boxes applied to periodic location updates still allow an adversary to follow a target [95]. Many studies have subsequently extended the k-anonymity concept to allow cloaking through the use of Hilbert curves [65], efficient cloaking of paths [95], and cloaking algorithms for l-diversity as well as k-anonymity [21, 90]. Bettini et al. [25] recently provide a formal framework to define attack scenarios, defense techniques, and assumptions on the amount of knowledge that is accessible by an adversary. Most recently, several works presented new kinds of attacks where achieving k-anonymity is insufficient for guaranteed anonymity specifically when external knowledge is available to an adversary such as query types [83] *and* service responses and user's behaviors upon a service request [77]. However, none of these works focuses on preserving guaranteed privacy and high data quality in time-series location data.

Best effort algorithm. Specifically, two major research areas, privacy-preserving

data mining [18, 17] and anonymous communication [30], have several candidates as stronger protection techniques in time-series location data. Random perturbation approaches are not applicable since they cannot provide sufficient data accuracy and noise with small variance may be sometimes filtered by advanced signal processing techniques [69]. Another group of candidates are best effort location data protection algorithms [24, 81, 74, 54, 26, 63, 59], which have in common that they create areas of confusion where the traces from several users converge. While these algorithms successfully achieve better accuracy and a defined level of privacy in such an area of confusion, they cannot provide overall privacy guarantees because these areas of confusion might not occur in lower-density areas. The work on measuring communication anonymity [82, 38] also inspired us to use entropy in defining time-to-confusion.

Anonymous Communication. Anonymity has also been extensively studied in the networking domain. Starting from Chaum's anonymous communication work [30], researchers have developed MIX networks such as Onion Routing [49] or Tor [39]. Privacy of *location information* has been extensively investigated at the network level. Network-level privacy techniques such as mixes and pseudonyms have been developed for cellular networks [43] and mobile IP [42]. The use of silent periods [52, 81, 74, 59], periods of no communication, was proposed for wireless networks to reduce exposure to tracking. Sharing a similar approach with *swing & swap* by Li et al. [74], Jiang et al. [63] combined three known concepts (silent period, pseudonym update, and control of transmission) to maximize the size of anonymity set. For sensor networks, two research groups, Kamat et al. [66] and Deng et al. [37] developed routing algorithms to protect the location of message senders or receivers (i.e., base station). These approaches are largely complementary to our work; they could be used in relaying (encrypted) GPS readings to the traffic monitoring service provider. The work on measuring communication anonymity [82, 38] also inspired us to use entropy in defining *time to confusion*.

Miscelleneous. Another proposed approach builds on privacy policy languages [34, 35] and their location-oriented extensions [85] to allow users (or their automated agents) to make more informed decisions about data sharing. Such policies may be enforced

through access control mechanisms for spatio-temporal data [46, 99]. Using these approaches, data can only be shared if the data provider trusts the data consumer. Recently, two research groups, Apu et al. [68] and Andreas et al. [71] proposed a privacy-preserving data collection architecture for collaborative sensing applications. However, both groups do not consider inference attacks that utilizes existing correlation between location-based updates.

2.2 Most Relevant Studies

Spatial cloaking [51] provides a countermeasure against these risks. It dynamically adjusts the resolution of position samples to maintain a constant degree of privacy in situations with different user densities. Given a set of traces from different users, the spatial cloaking algorithm achieves k-anonymity by determining a square that encloses the current positions of at least k users. Square corners are chosen from an external reference grid, so that they do not reveal any clues about current user positions. The position samples of the k users are then replaced with the square (or its center point).

The privacy risks for single positions are compounded for longer GPS traces, which contain more than one position sample. If a user can be identified at any one point, an adversary can infer which buildings (e.g., stores, clubs, medical clinics, entertainment venues) a person visited and accurately measure time spent at work or at home. If the frequency of location samples is high (at least one every few minutes), one may also infer speed limit violations while driving, for example, even if the GPS device does not report speed information. Further identification risks are higher, because a person could now be identified through knowledge about the frequency of his or her visits to each location in the trace [57].

A countermeasure against these particular trace risks is *path segmentation* [24, 54, 81], which divides several anonymous traces into shorter traces, or in the extreme, into a set of anonymous samples. Intuitively, this might reduce the risks to those associated with anonymous samples. However, an adversary may frequently be able to reconstruct the complete traces by "following the footsteps" (if one segment begins where

another one ends the trajectory of both points into the same direction, they likely belong together). This can be automated through location tracking algorithms that exploit the spatio-temporal correlation between subsequent samples, such as *multiple target tracking* [54, 79]. In essence, these algorithms predict a user's next position based on the previous trajectory and add the sample closest to the prediction to the trace. This approach fails, if many potential users are near the predicted position—thus, the segmentation approach is only effective in areas where user density is high and many users share common paths. Target tracking algorithms can also filter noise from the location samples, thus privacy techniques that add random noise to each sample may be ineffective unless the noise component is very large compared to the range of possible positions.

Better privacy protection for GPS traces can also be provided through special disclosure control algorithms such as *origin-destination cloaking* (ODC) [56]. ODC is designed for GIS applications that primarily involve users in motion, such as traffic monitoring applications in the automotive domain. ODC cloaking aims to suppress the parts of location traces that are close to locations that a user has visited, but allow release of location information when the user is moving. The intuition behind this approach is that visited locations provide likely avenues for identification and reveal potentially sensitive information. With ODC the exact visited building remains hidden, as only the general area is known. Thus, both restricted space identification and compiling a dossier of visited locations become more difficult.

2.3 Other Location Privacy Risks

Besides de-anonymizing an anonymous location trace dataset using sophisticated data mining techniques such as MTT, several other location privacy threats have been identified, and a category of applications presented in the above might pose those attacks.

Device Identification. Recently, several studies have shown that device-oriented characteristics such as MAC/IP address [52, 59], device clock skewness [70], and device manufacturer-dependent protocol design [45] allow an adversary to identify or

even track anonymous devices. Anonymous location-based services might pose this threat if they use WLAN hotspots for delivering users' sensor readings to external application service providers.

Transmitter Localization. Jiang et al. [63] investigated the privacy implications of using a localization technique as a tracking method, which uses more than three receivers to localize the transmitter. To address this threat, they propose a method of controlling transmit power, while a Cambridge group [93] suggests a method of forming the beam of a transmitted signal using an antenna array to prevent malicious receivers from receiving the signal. The localization attack might exist when cellular networks are used in delivering sensing information from users to external service providers.

Of the three categories of tracking threats, device identification attack is relatively easy to address because rendering devices anonymous by randomizing device IDs and adding noise to hardware-dependent characteristics can achieve unlinkability. However, the target tracking and the localization attacks utilize characteristics that are not dependent on the device itself, namely, spatio-temporal correlation and wireless radio signal propagation, respectively. Thus, rendering devices anonymous is not enough, and a special understanding on fingerprinting (based on spatio-temporal collelation or wireless radio signal propagation) is required to provide strong anonymity.

Chapter 3
Security and Privacy Challenges

In this chapter, we describe security and privacy challenges that might exist in automotive traffic monitoring applications. In addition, we highlight the inference attack on collected location traces and clearly define the problem to be addressed by explaining why key related works are not sufficient to solve these challenges.

3.1 Security and Privacy Requirements

The primary security and privacy challenges that traffic monitoring applications face are ensuring *integrity* of the data samples containing speed and position information and maintaining *privacy* for the drivers that supply the samples. The complete system might also require access control to restrict access to the traffic congestion information to paying customers. Such secondary requirements can be met with state-of-the-art technology—we concentrate on the integrity and privacy requirements.

The integrity of the computed congestion index relies on genuine speed and position data from the probe vehicles. Data integrity may be affected by malfunctioning probes or malicious parties that modify sensor readings. While malicious attacks on traffic monitoring may sound far-fetched, currently available gray-market devices to reduce travel time (e.g., radar detectors, infrared transmitters to change traffic lights) make such manipulations quite plausible. These devices might manipulate the congestion index to divert traffic away from a road to reduce travel time for a particular driver, or may divert traffic to a particular roadway to increase revenue at a particular store. It is also possible that other service providers might try to undermine the information quality of a competing traffic monitoring service.

Proactively addressing user privacy concerns in the architecture increases the potential for user adoption of the traffic monitoring service and reduces the risk of public data handling mishaps. Location information collected by probe vehicles raises privacy concerns because it is often precise enough to pinpoint the exact buildings that drivers visited, at least in suburban areas where buildings have dedicated parking lots. Reconstructing an individuals trace could provide a detailed movement profile that allows sensitive inferences. For example, recurring visits to a medical clinic can indicate illness or visits to activist organizations could hint at political opinions. While the traces of all participants deserve protection, the location traces belonging to political leaders, celebrities, or business leaders would likely undergo particular scrutiny. For example, frequent meetings between chief executives might indicate a pending merger or acquisition, highly desirable information for competitors and stock market speculators.

Data integrity and privacy can be compromised by different entities, as enumerated here:

1. Data Integrity

    - Compromised vehicles: Drivers or third parties could modify the hardware or software to report incorrect position or speed readings for a vehicle.[1]
    - Impostor devices: A device could spoof other authorized devices. This compromise is of particular concern in the form of the Sybil attack [40], where a device claims multiple different entities. Traffic monitoring accuracy will degrade if many vehicles simultaneously report incorrect information.
    - Network intermediaries: The transmission of vehicle data over wireless and wired communication links enables modification of reports by intermediate network entities.

2. Privacy

[1] Such modifications are well-known in the tachographs installed in European trucks. These devices record the vehicles driving times and speed to allow authorities to check adherence to mandatory driver rest periods.

- Eavesdroppers: Unauthorized third parties could monitor network transmissions for vehicle position readings and unique identifiers that allow tracking of vehicles. In particular, third parties could monitor wireless transmissions around particularly sensitive locations to record which vehicles recurrently visit the area. Network identifiers, such as the international mobile subscriber identifiers (IMSI) in the GSM cell phone system, help identify recurring visits.
- Spyware: Parties with access to the on-board vehicle system could install software that directly reports vehicle positions to other network servers.
- Insiders: Privacy breaches through insiders are particularly insidious at the traffic monitoring server, which receives and stores reports from large numbers of vehicles. While access control mechanisms provide some protection, there are typically a number of individuals with root access to the system (e.g., system administrators).[2]

3.1.1 Location Privacy at Telematics Service Provider

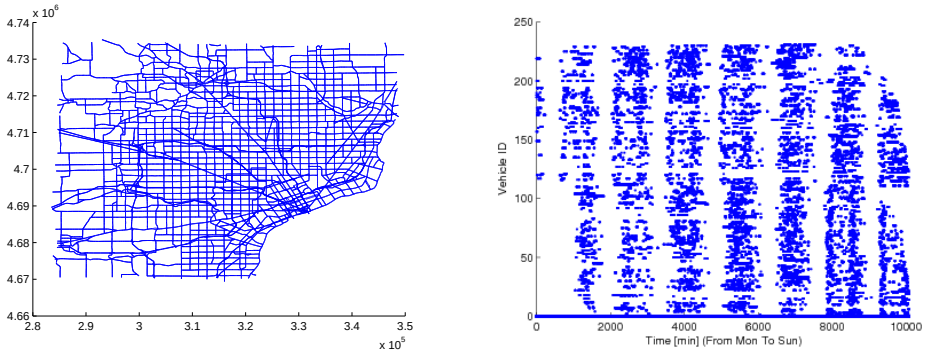
At the Telematics Service Provider, an intruder could gain access to a set of decrypted location samples from the TSP's database. While the data is likely well protected by the service provider, experience has shown that such data breaches still occur. Since anonymous location samples from all vehicles are mixed in this database, it appears that private information is difficult to extract. If multiple vehicles cross a similar path, it is difficult to discern which sample belongs to which vehicle.

However, in this study, we highlight two risk scenarios arising from data mining techniques, home identification and tracking, where privacy might be compromised even if the anonymous data collection architecture has been deployed. Home identification allows an intruder to identify the locations of the homes from probe vehicle

[2]On April 8, 2006, Information Week posted a chronology of data breaches reported since the ChoicePoint incident. Most of them were due to authorized employees at the targeted companies.

drivers. This serves as a first step towards identifying the driver. Tracking allows reconstructing paths from the anonymous traces and might be used to link the driver to sensitive places that were visited. These techniques are most useful in conjunction; a privacy compromise requires both identifying the driver and acquiring sensitive information about the individual. Indeed, in the next chapter, we describe more details of some potential risks raised by data mining techniques and define threat model based on them.

3.2 Data Quality Metrics and Requirements



(a) 70km x 70km road network with cell weights indicating the busiest areas

(b) Temporal distribution of GPS traces for 312 vehicles

Figure 3.1: Traffic monitoring system and spatio-temporal distribution of real-world dataset

There exists a tradeoff between data quality (or its utility) and the degree of privacy in data privacy algorithms, because each algorithm introduces unavoidable data modifications such as omission, perturbation, or generalization of a datum to increase privacy. To evaluate privacy algorithms meaningfully, we first discuss data quality requirements and metrics for the traffic monitoring application.

The application represents a road map as a graph comprising a set of road segments, where each road segment describes a stretch of road between two intersections. Generating the congestion map then proceeds in three steps: Mapping new GPS samples to road segments, computing mean road segment speed, and inferring a congestion index (e.g., by comparing current mean speed on a road segment of interest with its free flow

speed). Algorithm 1 illustrates a typical implementation that computes a congestion index for each roadway in more detail. [3]

Algorithm 1 Sample traffic monitoring algorithm

```
1: // Periodically recalculates the congestion index given new location samples and list of
   road segments
2: CalculateCongestionIndex(new location samples, road segments)
   // Map samples to road segments
3: for every new location sample (pos, vel) do
4:     Find the road segment s with minimal euclidian distance to pos
5:     cumVel[s] += vel
6:     numSamples[s] ++
7: end for
   // Calculate mean actual link velocity and congestion index for each link
8: for every road segments s do
9:     v[s]=cumVel[s] / numSamples[s]
10:    congIdx[s] = vfree[s]/v[s] - 1
11: end for
```

Algorithm 1 firstly matches a location sample to one of road segments in map database by projecting it onto every nearby road line segments, calculating the distance between the location and the projected point, and selecting the nearest road segment. [78] Given a set of location samples and its corresponding road segments, algorithm 1 collects anonymized speed information $(v_1, v_2, ..., v_N)$ from N anonymous probe vehicles which are spotted on each road segment during a pre-defined update interval $T_{interval}$, and averages the readings to compute the mean spot speed, $\overline{v} = \frac{\sum v_i}{N}$. Finally, mean spot speed directly allows obtaining mean link travel time by $T_{link} = \frac{L}{\overline{v}}$, where L is the length of road segment. The congestion index indicates the proportion of travel time on the link that is delay time (i.e., excess travel time above the free-flow travel time). Thus, function g calculates the congestion index (C_{idx}), which is defined as $\frac{T_{link}-T_{free}}{T_{free}}$ where T_{free} is the free-flow travel time. A congestion index near zero will indicate very low levels of congestion, while an index greater than 2 will generally correspond to congested conditions.

[3] The Federal Highway Administration provides multiple definitions for a congestion index. Here we adopt the 'Travel Time Index' definition as an example. This definition assumes that drivers will observe the speed limit so that the speed limit could be used for calculating the free-flow travel time.

Parameter	Requirement
Spatial Accuracy	100m
Sample Interval	1min
Delay	few minutes

Table 3.1: Traffic monitoring system data requirements

Mapping GPS samples onto road segments requires high *spatial accuracy*. Consider that two different parallel road segments (with traffic flow in same direction) may be only about 10m apart, as on the New Jersey Turnpike, for example. Cayford and Johnson [29] showed, however, that using tracking algorithms the correct road can be determined in 98.4% of all surface streets and 98.9% of all freeways if the location system provides a spatial accuracy of 100m and updates in 1s intervals. When increasing the update interval from 1s to 45s, the correctly determined roads drop from 99.5% to 98% (at 50m spatial accuracy). Therefore, to maintain high road mapping accuracy at the 1min sample interval for our data traces, we can assume that a minimum spatial accuracy of 100 m is needed.

Another important data quality requirement is *road coverage*, which primarily depends on the penetration rate, the percentage of vehicles carrying the traffic monitoring equipment. To achieve high coverage these systems aim at a minimum penetration rate of 3 (for freeways) to 5% (for surface streets) [36], but during the initial deployment phase penetration rates may be much lower. Thus privacy algorithms must offer protection even with in low deployment densities. Road coverage can also be reduced through privacy algorithms. Thus, we measure a *relative weighted coverage metric* for the privacy algorithms, which is based on the following heuristics. First, road coverage decreases as more samples are withheld. Second, probe-vehicle based traffic monitoring aims to extend traffic monitoring beyond a few key routes, but information from busier roadways is certainly more important than from low-traffic routes. Third, coverage is fundamentally limited by the number of probe vehicles on roads, thus we only consider coverage relative to the original dataset.

To measure the effect of removed samples on road coverage, relative weighted

coverage first assigns each location-sample a weight, depending on how busy the area around this sample is. Then, it divides the sum of weighted location samples from modified (or partially removed) traces by the sum of weighted location samples from the original traces. To estimate these weights for our dataset we divide the area into 1km by 1km grid cells and count the number of location samples n_i emanating from each cell i over one day in the original traces. The resulting weights for each cell are overlaid on the road map in Figure 3.1(a). The weights are normalized with the sum of weights over all samples, so that the relative weighted road coverage for the original dataset is equal to 1. More precisely, the weight for all samples in cell i equals $w_i = \frac{n_i}{\sum_j n_j^2}$. With these weights, relative weighted road coverage for a set of location samples L is then defined as $\sum_{l \in L} w_{c(l)}$, where the function c returns the cell index in which the specified location sample lies.

In summary, we can measure data quality for a traffic monitoring application through the relative weighted road coverage, where we consider a road segment covered if a data sample with sub-100m accuracy is available. Table 3.1 summarize key system parameters and requirements that we will assume in the following sections.

3.3 Research Directions

Applications that have access to private GPS traces from large numbers of users are relatively new. Thus this area provides many topics for further research. In summary, we observe that:

- *Risk analysis and privacy metrics.* Little practical experience with such applications exists. Privacy risks are typically identified by studying analogies to risks in other information systems. Improved privacy frameworks and metrics are needed to guide analysis of privacy risks in applications. These frameworks should include quantitative guidance on parameters such as user density, sampling frequency, and trace duration.

- *Ensuring guaranteed privacy and high data quality.* Both achieving guaranteed privacy and high data integrity is especially challenging because providing anonymity and perturbing user location for user privacy conflict with authentication of the update messages from users and data quality. This problem becomes more challenging if we assume that a service provider is not trustworthy.

- *Usable privacy preferences.* Because increased privacy protection usually reduces the quality of service provided by the application, a complete privacy solution should allow users to choose or specify different disclosure options. This requires research on user interfaces to understand how users can best express these preferences. It also requires research in privacy algorithms that must remain secure even if some users disclose more detailed information than others.

- *Maintaining privacy when using multiple techniques.* When different anonymization techniques are simultaneously used, for example, to satisfy different application requirements, an adversary with access to the different produced datasets may be able to infer private information. Further work is needed in understanding these risks and offering appropriate solutions.

- *Analysis and Penetration testing.* The described privacy algorithms are relatively new and should be subjected to more rigorous security analysis. As with other security techniques, only continued analysis and penetration testing over time will provide a good understanding of the exact level of protection they offer.

Chapter 4
Evaluation of Existing Privacy Algorithms

Throughout this section, we analyze the privacy risk in anonymous location traces and then obtain an empirical privacy risk model based on major factors such as user density and sampling interval. Moreover, we evaluate the existing privacy schemes and identify their weaknesses.

4.1 Privacy Leakage Through Anonymous Location Traces

Monitoring a vehicle's movements can reveal driver's sensitive information particularly in the United States where a person's life pattern heavily relies on automobile. First, knowing either an origin or a destination of the trip can reveal information about a driver's health, lifestyle, departure/arrival, or political associations. Second, distances between buildings are large so that the location samples often precisely point to the visited building (e.g., homes and work places).

Even after anonymization, some of this information may be recovered, as simply removing identifiers from a dataset does not always provide strong anonymity guarantees, which was the motivation for our study.

4.1.1 Insider Attacks

In traffic monitoring applications, revealing location traces to an adversary might pose a privacy threat. A complete traffic monitoring system clearly exhibits multiple possible points of attacks. Partial location traces could be collected through methods including malware on the on-board vehicle computing system, eavesdropping on wireless

communications, or compromises at the traffic monitoring server. In particular, to prevent compromises at infrastructure components, many techniques are proposed such as access control, cryptographic encryption/decryption, and policies. However, these techniques cannot completely prevent an accidental or intentional disclosure by legal employees [10] in the victim companies or through remote break-ins. A study [7] scrutinizes recent data breaches in the United States since 2005, some of which resulted in significant financial loss to customers. It reports that 217,551,182 records involved in data breaches contain sensitive personal information. It means that nearly everyone living in the United States has one sensitive record breached.

4.1.2 Inference Attacks

Once adversary gains access to a dataset of location samples, she can re-identify anonymous traces through data analysis. Indeed, we highlight some potential risks, home identification and target tracking raised by advanced data mining techniques. We demonstrate that driver's privacy can be compromised even if the anonymous data collection architecture is deployed.

Home Identification. Clustering [62] can be an effective tool for home identification. In particular, clustering promises to group a set of location samples that likely belong to the same destination, and the centroid of this cluster of endpoints provides a good estimate of the destination, where anonymized location samples with low-to-zero speed might be candidates for endpoints. Its computation automatically smoothes out noisy GPS location samples around destinations and allows automatic identification of repeatedly visited places. Such noisy GPS samples are due to the response delay to the first lock of GPS signals, GPS measurement inaccuracies, and possibly usage of different parking spots. For example, figure 4.1(a) shows a sample scenario where GPS samples cluster precisely on a single home's driveway. In contrast we found that trip origins are usually harder to identify, likely because the first GPS samples are drifted away from the exact destinations due to the receiver's GPS acquisition delay after power on. Figure 4.1(b) illustrates a case where trip endpoint variance remains

(a) Well clustered destinations

(b) Noisy originations due to GPS signal acquisition delay

Figure 4.1: Place identification example. Determining which building a driver visited is possible in left scenario because trip endpoints (shown by the markers) cluster denser than nearby homes.

too high for building identification.

Since home identification provides higher risk than any other destination [1], we develop a clustering algorithm for general place identification [20] into home identification technique by adding a set of heuristic rules to filter out irrelevant location samples. For instance, we can differentiate parking GPS location samples from moving GPS location samples by looking at GPS speed information. Also, the time information can be used to tell home location from other kinds of destinations. If the marked time is from 4 PM in the afternoon to midnight and there is no subsequent moving GPS location samples detected before the morning of the second day, the destination is more likely a home instead of a working place.

Target Tracking. Target tracking techniques can be used to reconstruct paths from anonymous samples or segments [79, 53]. This technique is particularly useful once a home location has been identified. Knowing a home position itself poses limited privacy risks, if no potentially sensitive information can be linked to this home. Privacy risks are beyond just knowing a home position itself, if potentially sensitive information or places can be linked to this home. Target tracking techniques can allow an adversary to follow the traces reported by a vehicle to other locations, thereby linking information

[1] We believe that home identification provides the highest risk, since their usually exists a one-to-one mapping between a typical suburban home and a household, and home owners and occupants are public knowledge though telephone white pages or real estate records.

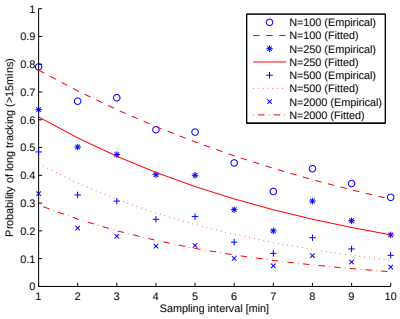


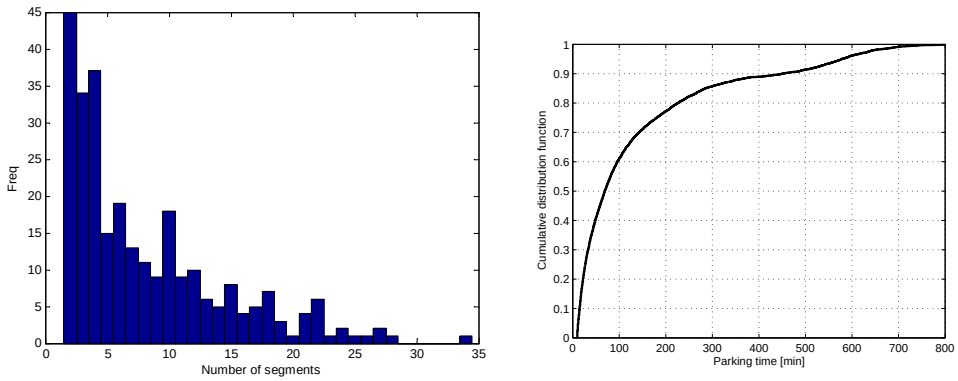
Figure 4.2: Dependency of Tracking Duration on Sampling Period and Probe Density

about other places to the driver identity.

However, if multiple vehicles cross a similar path, it is difficult to discern which sample belongs to which vehicle. Furthermore, target tracking techniques do not work well in urban area due to the poor availability of GPS signals in urban environment: buildings, bridges, or tunnels often block GPS signals. In other words, target tracking techniques are more effective in suburban area (sparse region of GPS traces).

4.1.3 Case Study: Anonymous GPS Traces on Suburban Areas

Through the analysis of real week-long GPS traces from 312 probe vehicles, we first attempt to build an empirical privacy model that estimates tracking duration from key parameters. This empirical model can give system designers an estimate of tracking risk for the collected location data from users without access to the detailed dataset. If it is combined with human mobility model, it can further relates tracking risk to likelihood of having visited an identifying location. As a preliminary example, we consider how the following two key parameters affect tracking duration: the density of users and the sampling interval with which location samples are updated. Figure 4.2 illustrates how long an adversary tracks anonymous users in different user densities and sampling intervals. This data is empirically derived using the tracking model described in section 8.1 over real GPS traces covering a suburban area (see figure 3.1(a)) for 24 hours. As evident, the tracking time appears to follow an exponential function as either the sampling interval or the probe density increases.



(a) Number of Trips in Each Vehicle's Trace. (b) Parking Time Cumulative Distribution.

Figure 4.3: Statistics on Vehicular Movement Patterns.

Parking Linking. Each probe vehicle's trace consists of multiple trips (2 to 13 trips per day) as shown in figure 4.3(a). Target tracking model can link two successive trips beyond linking between successive location samples. Specifically, the spatio-temporal correlation existing between the last sample of the previous trip and the first sample of the next trip enables this type of linking. We empirically obtain the distribution of time deviation between two successive trips as shown in figure 4.3(b). With this CDF of time deviation and an empirically fitted PDF of distance deviation (exhibiting quite similar pattern to figure 4.6(b)), we first apply target tracking model against paths of 315 different users with a duration of 2.5 days to measure the total tracking duration. We call this *path tracking* that characterizes the adversary's chances of correctly following the complete 2.5 day path of an individual user over all pseudonym changes.

Figure 4.4 shows the path tracking performance over 2.5 days. We illustrate the 70 longest tracked traces out of 315 users. We plot each traced trace in terms of total time, tracking time, and travel time. Total time denotes the whole time duration between the origin of the first trip to the destination of the last trip of a single probe vehicle, tracking time describes how long an adversary follows the vehicle including parking time, and finally travel time only measures the driving time. We observe that many tracking outliers are present even though each location sample is anonymized. Furthermore, those tracking outliers are found in low density areas.

From the above preliminary experiment, we found that calibrating the sampling

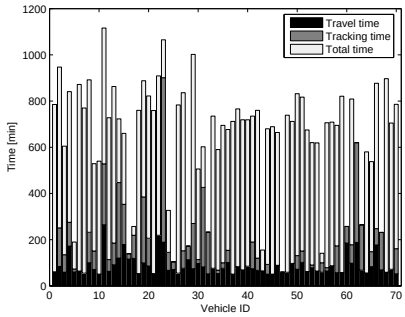


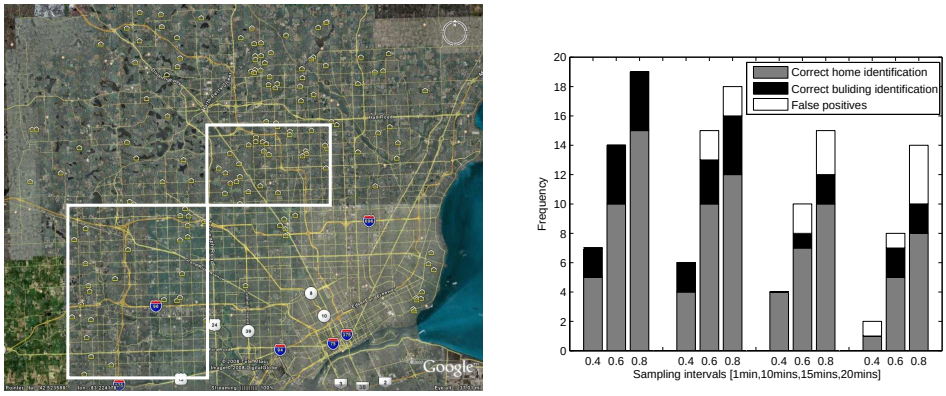
Figure 4.4: Path Tracking Performance.

interval for a given user density can allow tracking outliers, and some of them stretch over multiple trips. In addition, it is a time-consuming job to derive an empirical model for different areas, since the empirical privacy model depends on not only several major parameters (that we considered) but also road network, signalized intersections, and car-following dynamics.

Protection against Home Identification. To examine the effectiveness of reducing sampling frequency, we measure home identification rate, meaning how many homes out of the total (65 home locations shown in figure 4.5(a)) are correctly detected, and the false positives rate, meaning how many are incorrect among the estimated home locations. Such false positives can be caused by many vehicles waiting at traffic lights or stop signs in residential areas or shifting of the cluster centroid to a neighbor's house due to some inaccurate location reports. Sometimes a small degree of shift can cause a false positive in highly dense residential areas. We thus use an entropy to capture this uncertainty in adversary's decisions. For an attack algorithm, refer to the section 8.1. To calculate an uncertainty, we measure distance between a cluster and each of five nearest homes from it. For each distance, we assign a likelihood by computing a probability,

$$\hat{p_i} = e^{-\frac{d_i}{\mu}}$$

, normalize all likelihoods for five corresponding candidates, and calculate an entropy. In addition to the standard 1 min sample interval, we consider the following reduced frequencies: 10 min, 15 min, and 20min intervals.



(a) Plausible Home Locations. (b) Home Identification Rate.

Figure 4.5: Plausible home locations in two target regions (in white rectangles) according to manual inspection. The study considered a total of 65 homes in chosen area.(Left) Four different sampling intervals are depicted by four circles and the specific parameter is next to each mark. Original location traces have one location report per minute, which corresponds to 1 minute interval.(Right)

Figure 4.5(b) demonstrates that reducing sampling frequency on anonymous location traces does not necessarily address the privacy problem. Figure 4.5(b) shows the home identification uncertainty for each centroid returned from the clustering procedure described earlier, on data sets with sampling intervals of 1, 10, 15, and 20 min. The different bars show the number of correct home identifications for different uncertainty thresholds. Presumably, an adversary will only select locations with high certainty to reduce false positives. Note that even with a sampling interval of 20 minutes, the adversary can still correctly identify a home with high certainty. Reductions in sampling frequency can reduce the probability that samples are taken nearby a driver's home, but this probability is also a function of the length of the trace. If location traces are never discarded, sufficient samples around a user's home will eventually be available.

When taking 0.6 as a threshold, an adversary correctly locates 10, 10, 7, and 5 homes under 1 minute, 10 minutes, 15 minutes, and 20 minutes, respectively. The number of correct centroids by an adversary increases up to 15, 12, 10, and 8 homes with a 0.8 threshold. While this data suppression technique can reduce the home identification risk and thereby increase privacy, it is noted that even with a sampling interval

of 10 minutes, the adversary can still correctly identify a home with high certainty.

4.2 Existing Privacy Algorithms

Several techniques have been proposed to increase location privacy. However, we are aware of only one class of techniques, spatial cloaking algorithms for k-anonymity, that can guarantee a defined degree of anonymity for all users. Among known algorithms, we test the feasibility of subsampling techniques and spatial cloaking techniques based on k-anonymity as privacy algorithms for automotive traffic monitoring applications. Both of them cannot achieve high accuracy and strong privacy at the same time.

4.2.1 Best Effort Algorithms for Probabilistic Privacy

Given that in dense environments paths from many drivers cross, drivers intuitively enjoy a degree of anonymity, similar to that of a person walking through an inner-city crowd. Thus, Tang et al. [89] lay out a set of privacy guidelines and suggest that the sampling frequency, with which probes send position updates, should be limited to larger intervals. The authors mention that a sample interval of 10min appears suitable to maintain privacy, although the choice appears somewhat arbitrary (for reference, a typical consumer GPS chipset implementation offers a maximum sampling frequency of 1 Hz). We refer to data collection with reduced sampling frequency as subsampling.

Other best effort algorithms suppress information only in certain high-density areas rather than uniformly over the traces as the subsampling approach. The motivation for these algorithms that path suppression in high density areas increases the chance for confusing or mixing several different traces. This approach was first proposed by Beresford and Stajano [24]. The path confusion [54] algorithm also concentrates on such high-density areas although it perturbs location samples rather than suppressing them. These techniques increase the chance of confusion in high-density areas, but they also cannot guarantee strong privacy in low-density areas where paths only infrequently meet. Thus, in-terms of worst-case privacy guarantees their advantage over subsampling remains unclear.

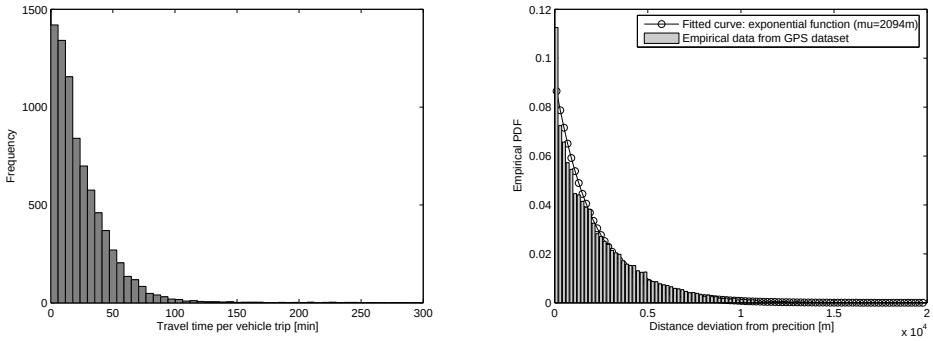
	Random sampling (50% removal)		Anonymization (no removal)	
	15min	20min	15min	20min
D=500, Un=0.45	45/47	28/29	3300/3480	1117/1172
D=2000, Un=0.7	18/30	10/17	1302/1394	908/958

Table 4.1: Empirical confidence in subsampling. D denotes user density and Un denotes uncertainty threshold.

We choose the subsampling algorithm as a best effort baseline algorithm. Table 4.1 shows an adversary's tracking performance over an anonymous set of samples with 1 min (no removal) and 2 min (50% removal) sampling intervals. For a probe vehicle density of 500 vehicles per a 70km^2 region, the tracking algorithm returns 3480 segments of 15 min duration and 1172 segments of 20 min duration. Both reducing the sampling interval and increasing probe vehicle density reduces tracking performance. For example, with 2000 vehicles on a same area and 2 min sampling interval, 17 segments of 20 min duration can be identified. Precision of the tracking algorithm is about 95% in all cases, meaning that only 5% of the returned segments do not match an actual vehicles path, except in the 2000 vehicle 2 min case, where relatively few segments can be tracked (in this case precision drops to 60 percent). These example results were obtained with a tracking model that we will describe in detail in the following section.

To understand the implications of these tracking durations (15min and 20 min), let us consider figure 4.6(a), which depicts the histogram of per-trip travel time in the GPS dataset. The data shows a large number of very short trips, for example 30% of trips are shorter than 10 min, 50% of trips shorter than 18min. This empirical result also coincides with the empirical statistics from real GPS traces in Krumm's work [73] (Krumm observes 14.4 min per trip as a median). This means that by following a trace for only 10min, an adversary may be able to track a vehicle from its home to a sensitive destination.

These results illustrate that protecting *all* drivers of probe vehicles through subsampling remains difficult. One minute sampling intervals are already large for a traffic



(a) Empirical distribution of travel times per vehicle trip.

(b) Empirical probability distribution function of distance deviation from prediction to correct sample.

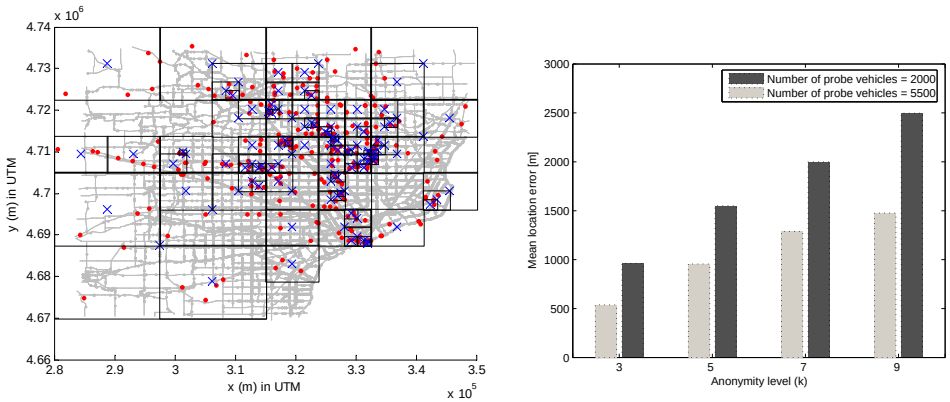
Figure 4.6: Fitting distance errors in tracking using an exponential function

monitoring application but protecting all drivers even in low density areas would require a further significant increase in the sampling interval. Moreover, it is difficult to choose this sampling interval since traffic densities can change substantially over time and space.

4.2.2 Spatial Cloaking for Guaranteed Privacy

k-anonymity [88, 80] formalizes the notion of strong anonymity and complementary algorithms exist to anonymize database tables. They key idea underlying these algorithms is to generalize a data record until it is indistinguishable from the records of at least $k - 1$ other individuals. Specifically, for location information, spatial cloaking algorithms have been proposed [51, 47] that reduce the spatial accuracy of each location sample until it meets the k-anonymity constraint. To achieve this, the algorithms require knowledge of nearby vehicles positions, thus they are usually implemented on a trusted server with access to all vehicles current position.

k-anonymous datasets produced with known algorithms cannot meet traffic monitoring accuracy requirements. Figure 4.7(b) shows the spatial accuracy results obtained after applying a spatial cloaking algorithm to guarantee k-anonymity of each sample. We use the same dataset in section 7.2.1 so that we could directly compare k-anonymity with our proposed solution in terms of spatial accuracy. The results were obtained with



(a) Snapshot of Spatial Cloaking Applied to GPS Traces.

(b) Data Accuracy.

Figure 4.7: Data accuracy of samples processed with spatial cloaking algorithm fails to meet the accuracy requirement in our scenario.(Right)

the CliqueCloak algorithm [47], which to our knowledge achieves the best accuracy. *The results show that even for very low privacy settings, $k = 3$, location error remains close to 1000m for an emulated deployment of 2000 vehicles, far over the accuracy requirement of the traffic monitoring application.* While these results can be expected to improve with increased penetration rates as the deployment case of 5500 vehicles shows 500m for $k = 3$ (indeed, [51] shows that median accuracies of 125 meters and below can be obtained when *all* vehicles act as probes), other privacy approaches are necessary to enable probe systems operating with lower penetration rates.

In summary, we observe that:

- **Observation 1.** Spatial cloaking algorithms that can achieve a guaranteed privacy level for all drivers fail to provide sufficient spatial accuracy for the range of user densities studied in our deployment. For $k = 3$ spatial accuracy remains over 1000m, for probe deployments of 2000 vehicles, one order of magnitude over the applications accuracy requirement. Thus, they are not suitable for probe vehicle systems that operate with low probe densities, or are incremental deployed over a longer time period.

- **Observation 2.** Ad hoc privacy techniques such as subsampling improve privacy but fail to provide a defined level of privacy for all users. 7% of users could

be tracked longer than 15min with only a 5% false positive rate. A 15 min tracking duration is sufficient to follow about 45% of all vehicle trips from origin to destination.

- **Observation 3.** While the evaluated home identification intrusion technique suffered from many false positives, this mechanism is at least effective as an automated pre-filtering step, followed by manual inspection. To provide a high degree of privacy protection, traffic monitoring systems should employ more sophisticated data suppression mechanisms.

These observations raise the question of alternate definitions and measures for anonymity in location traces as well as the need of enhanced privacy schemes.

Chapter 5
Privacy Metrics and Threat Models

> A recent study [7] scrutinizes data breaches in the United States since 2005, some of which resulted in significant financial loss to customers. It reports that 217,551,182 records involved in data breaches contain sensitive personal information.
>
> – *A Chronology of Data Breaches*, PrivacyRights.org

We present a novel privacy metric for location traces, **time-to-confusion** and provide more formal description on adversary models for target tracking and home identification in this section. In addition, we also provide a specific threat model, particularly focusing on highways, where more regular traffic flows increase the tracking risks.

5.1 Target Tracking

The degree of privacy risk is strongly subject to a tracking duration, how long an adversary can follow a vehicle. For a complete privacy breach, the tracked trace should have a privacy sensitive event (e.g., a sensitive destination) and the driver generating this trace should be identified. The chance of both events taking place increases with longer traces.

Since consecutive location samples from a vehicle exhibit temporal and spatial correlation, driving paths of individual vehicles can be reconstructed from a mix of anonymous samples belonging to several vehicles. This process can be formalized and automated through target tracking algorithms [53]. These algorithms generally predict the target position using the last known speed and heading information and then decide which next sample to link to the same vehicle through Maximum Likelihood Detection [94]. If multiple candidate samples exist, the algorithm chooses the one with

the highest a posteriori probability based on a probability model of distance and time deviations from the prediction (in our evaluation, we assume a strong adversary with a good model of these deviations). If several of these samples appear similarly likely, no decision with high certainty is possible and tracking stops.

Particularly when an adversary uses map information in the target tracking, a driving direction can be incorporated in computing a likelihood. Suppose a vehicle is running on a straightly stretched highway. In this case, anonymous location samples in a opposite way, even if near a predicted position, can be dropped out of the candidate set. However, using this heading information is not so effective in a relatively large sample interval such as 1 minute in our dataset.

Privacy Metrics. Under the above adversary model, we measure the degree of privacy as the *Mean Time To Confusion (MTTC)*, the time that an adversary could correctly follow a trace. Note that this includes time while a user remains stationary unless otherwise specified. More specifically, the time to confusion is the tracking time between two points where the adversary reached confusion (i.e., could not determine the next sample with sufficient certainty).

To formally describe the novel privacy metric, we define a linkability between two location samples first, then we define a time-to-confusion from it.

Definition 1. *Assume a set of anonymous location samples that are collected during the observation time instants,* $t_1, \ldots, t_n$*,*

$$M = \{\langle m_{1,t_1}, \ldots, m_{k_1,t_1}\rangle, \langle m_{1,t_2}, \ldots, m_{k_2,t_2}\rangle, \ldots, \langle m_{1,t_n}, \ldots, m_{k_n,t_n}\rangle\}$$

, where $k_1, \ldots, k_n$ *denotes the number of different samples received during each quantized time interval. An arbitrary sample* m_{i,t_j} *in the set* M *is said to be linkable to the sample* m_{q,t_r} *if the following conditions hold: (1)*$t_1 < t_r < t_j$*, (2)* $m_{q,t_r} = N(m_{i,t_j})$*, where* $N(\cdot)$ *is a function that returns the sample closest to a predicted position, and (3)* $U(m_{i,t_j}, m_{q,t_r}) \leq U_{th}$*, where* U_{th} *is an uncertainty threshold and* $U(\cdot)$ *computes the entropy for this tracking step as described below.*

Each sample has three properties: (a) its predecessor, (b) the aggregated number of locations, and (c) total elapsed time, which are denoted by $m_{i,t_j}.M$, $m_{i,t_j}.L$, and

$m_{i,t_j}.T$, respectively. The predecessor of the sample, $m_{i,t_j}.M$ is chosen as m_{q,t_r} that has the lowest $U(m_{i,t_j}, m_{q,t_r})$. We update (b) and (c) of each sample as shown below.

$$m_{i,t_j}.L = m_{i,t_j}.M.L + 1$$
$$m_{i,t_j}.T = m_{i,t_j}.M.T + \Delta t$$

, where Δt denotes the time difference between two linkable samples. If a sample is not *linakble*, its M is null and its L and T are zeros, meaning that the sample could be a starting point of another tracking.

Definition 2. *If the sample m_{i,t_j} is not linkable to any other, we trace back its predecessors, say, $m_{i,t_j}.M, m_{i,t_j}.M.M, \ldots, m_{i,t_j}.M.\ldots.M$ until we meet NULL. Let P be the set of all collected samples, then we call it a traceable path. We define the Time-to-Confusion of the traceable path as $m_{i,t_j}.T$.*

As privacy measures, we obtain the maximum and median *Time-to-Confusion* of all *traceable paths* P_i for a given set M.

Tracking Uncertainty. Inspired by the use of entropy in anonymous communication systems [82, 38], we use information theoretic metrics to measure uncertainty or confusion in tracking denoted by $U(\cdot)$.

For any point on the trace, *Tracking Uncertainty* is defined as $H = -\sum p_i \log p_i$, where p_i denotes the probability that location sample i belongs to the vehicle currently tracked. Lower values of H indicate more certainty or lower privacy. Given no other information than the set of location samples, intuitively the probability for a sample reported at time t is high, if the sample lies close to the predicted position of the vehicle at time t and if no other samples at the same time are close to the vehicle. As one step further, we can also express tracking confidence C on adversary's trial by calculating $(1 - H)$.

Empirically, we found that distances of the correct sample to the predicted position appear monotonically decreasing in figure 4.6(b). Therefore, we compute the probability p_i for a given location sample by first evaluating the exponential function

$$\hat{p_i} = e^{-\frac{d_i}{\mu}}$$

for every candidate sample and then normalizing all $\hat{p_i}$ to obtain p_i. The parameter μ can be interpreted as a distance difference that can be considered very significant. We obtain the value of μ from empirical pdf of distance deviation in figure 4.6(b) which we fit with exponential function using unconstrained nonlinear minimization (μ is 2094 meters).

The following algorithm is not dependent on the use of an exponential function for estimating the probability that a location sample belongs to the same trace. It does assume, however, that a publicly-known 'best' tracking model exists and that the adversary does not have any better tracking capabilities. In the thesis, we have empirically derived this probability model by fitting an exponential function.

Overall, the mean time to confusion can then be defined as the mean tracking time during which uncertainty stays below a confusion threshold. If the uncertainty threshold is chosen high, tracking times increase but so also does the number of false positives (following incorrect traces). Since the adversary cannot easily distinguish correct tracks and false positives, we assume that high uncertainty thresholds will be used.

5.2 Clustering-based Home Identification Algorithm

For the home identification algorithm, we use a k-means clustering algorithm [1] on anonymous location samples to identify frequently visited places after we drop high speed samples and day-time samples. Endpoints near a visited building likely have low-to-zero velocity, and vehicles are often parked at homes overnight. We then refine the resulting clusters using several heuristics.

First, we adapt a conventional k-means clustering algorithm to estimate the number of visited places automatically. Since the number of places visited in the traces is a priori unknown, this algorithm repeatedly merges clusters until any newly merged cluster would have an element farther from the centroid than a specified distance threshold.

[1] As noted in earlier work on place reconstruction from location traces [20, 67], clustering can be an effective tool to identify relevant (i.e., frequently visited) places.

1.	Drop location samples with too high speed ($> 1m/s$) from all vehicles (i.e., remaining samples contain the candidate trip endpoints).
2.	Select a target region of interest to improve computational efficiency, and drop samples outside this region.
3.	Apply k-means pair-wise clustering algorithm to samples in target region and store the returned cluster centroids.
4.	Filter the candidate home locations out of all centroids using two heuristics (A:arrival time and B:zoning information).

Table 5.1: Adaptive k-means Clustering for Home Identification.

Second, home locations are typically in residential zones. We thus drop clusters located on roads.

In a practical use of this algorithm to GPS traces over a large area, it is recommended to select a target region of interest to improve computational efficiency, and drop samples outside this region.

In table 5.1, step 3 repeats to calculate the centroids of clusters until it finally groups all location samples into the optimum number of clusters. K-means pair-wise clustering in Step 3 does not have a prior knowledge on the optimum number of clusters at the initial run. Thus it uses all locations obtained after Step 2 as initial clusters and keeps merging them in close proximity into smaller number of clusters at each run. Merging process stops if every centroid has all its elements within a certain limit distance on the average. The limit distance should be re-selected according to different home densities. If home density is too dense, it should be kept small enough to differentiate locations of other vehicles living close each other. In our simulations, we use a value of 100m for this threshold which we derive from the actual home density in the region.

After the algorithm reaches the optimum number of clusters, it moves to the filtering step based on *heuristic A*, where all centroids outside residential areas are eliminated. In our experiment, we manually eliminated centroids located outside residential areas by plotting and checking them on the satellite imagery of Google Earth. However, this process could be automated by obtaining GIS database such as city zoning information.

5.3 Variants of Target Tracking Algorithms

We also consider different possible tracking algorithms other than what we described in previous subsection. However, some of them are not applied to our situation, and others show only an incremental gain compared to the computation complexity that they introduced.

Linear Kalman Model. We observe that linear Kalman model does not enhance tracking capability. Linear Kalman model is an effective tool to estimate the state (e.g., position, speed, and acceleration) of system (e.g., vehicles) given a time-series of noisy observations. Accurately estimated state enables to predict the next position of the moving target as close as possible to its correct position. Of course, this accuracy increases as time interval between two adjacent samples decreases. However, our GPS dataset contains enough accurate location and position readings.

Multi-target tracking [79]. The Multi-target tracking has two distinctions from the Single-target tracking that we use in our study. It looks not only a moving target but also neighbors surrounding it when enumerating all possible hypotheses. This approach helps prune unlikely hypotheses in advance. Furthermore, it chooses the most likely path of moving target by computing the likelihood of multiple samples (including the chosen samples in previous decision cycles).

Pruning through map information. If we use road network information, we can achieve better pruning over a set of hypotheses. For example, even though two observed samples are in near distance, it might look obvious that they do not belong to a same user if they are sampled in two different parallel roads stretched in same directions. We expect the use of map information helps tracking. In our study, we emphasize on automated attack for massive number of targets without sophisticated knowledge such as map information or a prior knowledge on subjects to be tracked.

Accommodating Pseudo-Identifier Estimation Identifying pseudo- or quasi-identifiers from an anonymous message header enhances tracking capability. Theoretically, it is shown that any additional information can reduce anonymity, or so called uncertainty. [33] However, it holds as long as additional information is truly reliable (or

correct). The tracking algorithm described so far does not provide a formal way of using faulty but helpful additional information in tracking anonymous users such as the estimated pseudo-identifier that we explain in this section. The Multi-target tracking comprises of three steps: state prediction, hypotheses generation and selection, and state update, among which we insert the notion of pseudo-identifier estimation likelihood. In previous section, we describe a method of calculating a likelihood of two messages being originated from the same user (or device) by estimating their pseudo-identifiers based on their message headers. We generate and compute hypotheses with pseudo-identifier similarity as well as message's location proximity.

Chapter 6
Architecture

> People are willing to give up liberties for vague promises of security because they think they have no choice. What they're not being told is that they can have both.
>
> – Bruce Schneier, Nature 413, 773 (25 October 2001)

6.1 Design Criteria and Approaches

We aim to achieve both high quality traffic information and strong privacy protection in this traffic monitoring system. There exists an inherent tradeoff between these requirements because privacy-enhancing technologies such as spatial cloaking [51] reduce accuracy of traffic monitoring. One may expect, however, that a privacy-preserving design motivates more users to participate in such a system, which would improve the quality of traffic information. Our main objectives are the following:

Privacy. We aim to achieve privacy protection by design so that no single entity, not even an insider at the service provider, can identify or track a user.

Data Integrity. The system should not allow adversaries to insert bogus data, which would reduce the data quality of traffic information. This is especially challenging because it conflicts with the desire for anonymity.

Smartphone Client. The client software must cope with the resource constraints of current smartphone platforms.

We do not consider energy consumption because we assume that participants are using their phones in a charging dashboard mount to view navigation and traffic information as shown in figure 8.11.

We address the problem of building a privacy-preserving and secure architecture for automotive traffic monitoring applications that provides countermeasures against security and privacy challenges described in section 3.1. To do so, the focus of the thesis is three-fold. First, we design an architecture for collaborative sensing applications, more specifically an automotive traffic monitoring that provides users guaranteed privacy against inference attacks while achieving high service quality. As a solution, we design a centralized scheme that relies on the existence of a trusted location proxy. This approach suppresses the chance of several successive location samples being linked since a partially reconstructed trajectory could act as a quasi-identifier. The use of a trusted location proxy reduces the risk of trusting entities that are not honest and it is easier to understand/develop for system designers. Second, as an alternative design option for an architecture, we provide a distributed architecture design while meeting the same requirements as in the centralized architecture. In it, we discuss how reliance on a trustworthy privacy server may be relaxed. Third, against the known security and privacy risks, we provide countermeasures that are commonly among the two different architectures.

While conducting research on the above three themes, we take the following assumptions and methodologies.

- We evaluate our centralized architecture through the trace-driven simulation where real GPS traces of 312 probe vehicles are collected and their privacy is measured against our proposed algorithm.

- Designing a centralized architecture, we assume that a trustworthy privacy server is available to execute centralized algorithm. Also, we highlight an automated attack scenario that compromise a massive number of users. Moreover, we assume that adversary has no prior information about the subjects being tracked.

- We verify the feasibility of our distributed architecture through building a prototype, where we use GPS-enabled smartphones and servers that are connected to the Internet.

We first present the common architecture that are equipped with cryptographic primitives against security and privacy challenges and then explain two different architectures that prevent an adversary from tracking or re-identifying anonymous probe vehicles in next chapters.

6.1.1 Real-world GPS Trace Collection

For trace-driven simulations, we have offline collected a dataset containing GPS traces from 312 volunteer drivers driving in a large US city and its suburban area for a week. The collected traces, which are similar to a dataset of real deployments [13, 14, 12] covered the 70km by 70km region as depicted in figure 3.1(a). To protect drivers' privacy, no specific information about the vehicles or drivers is known to the authors. Each GPS sample is consisted of vehicle ID, timestamp, longitude, latitude, velocity, and heading information. Each sample is recorded every minute only while a vehicle is being in the state of ignition, so that the collected traces contain temporal gaps. These temporal gaps are due to one of three situations: when the vehicle is parked with its ignition switched off, when the GPS reception is lost (e.g., due to obstruction from high-rise buildings), or when the receiver is still in the process of acquiring the satellite fix. Because the traces do not contain information about ignition and GPS receiver status, we assume that a gap longer than 10 min indicates that the vehicle was parked. Figure 3.1(b) illustrates the distribution of gaps in the traces of around 312 vehicles. Each dot represents a received data sample. We refer to the parts of a trace between two gaps longer than 10 min as a *trip*.

6.2 Common Architecture

To resolve the tension between data integrity and privacy, the architecture assigns the authentication and filtering functions and the actual data analysis to separate entities. One entity knows the identity of the vehicle but does not have access to position and speed information, while the other entity knows position and speed but not identity.

The architecture also relies on encryption to prevent eavesdropping, tamper-proof hardware to reduce the risk of node-compromise and spyware, and data sanitization to further strengthen data integrity.

Figure 6.1 illustrates the entities and cryptographic schemes involved in transmitting a data sample from a vehicle. We distinguish the communication server (CS) and traffic server (TSP). The Communication server, which is provided by communication service provider, maintains network connections and authenticates users but does not access the location and speed data. The TSP receives anonymous data from the CS, decrypts and sanitizes the data and computes the real-time congestion maps. In a real implementation the communication server could be provided by a cellular phone service provider and the traffic server by a telematics service provider. The two parties would likely enter a contractual relationship as business partners but are assumed not to collude. They should not exchange any privacy sensitive information beyond those specified in this architecture. To further increase user confidence, the information exchange between these parties could be audited by an independent agency.

One key pair enables encryption between the TSP and vehicles. Every vehicle knows the public key K_{TSP} of the TSP and uses it to encrypt a location sample. We refer to this encrypted message as 'data segment (DS)'. Since the DS can only be decrypted with TSP's private key, K'_{TSP}, this layer of encryption protects location privacy against eavesdroppers.

The CS shares a separate symmetric key K_{veh} with each vehicle and knows the network identifiers (e.g., IMSI in GSM networks). Using this key, the CS can authenticate incoming data samples and ensure that they are transmitted from authorized probe vehicles. If valid, the CS then removes all network identifiers and the MAC from the vehicle and attaches its own message authentication code using a third key K_{CS} established between the TSP and the CS.

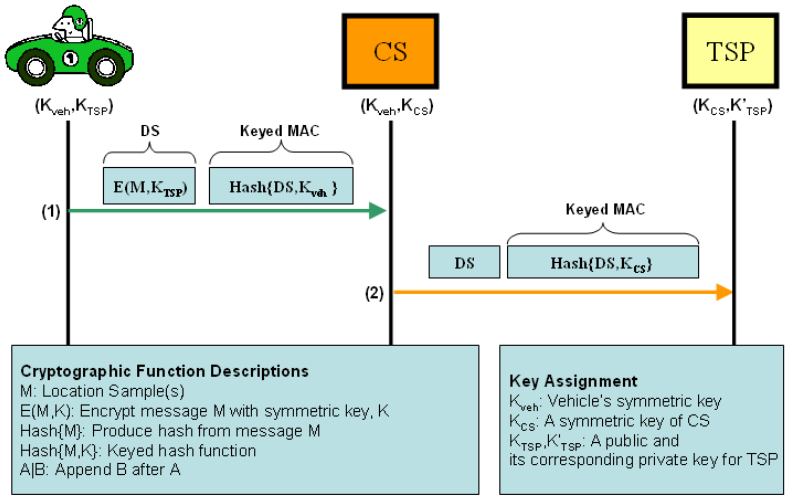


Figure 6.1: Traffic monitoring architecture to ensure data integrity and anonymous data collection.

6.2.1 Key Management: Distribution and Storage

The proposed architecture requires K_{veh} to be stored in vehicles. If an intruder can easily extract secret vehicle keys from multiple cars, the intruder could insert large numbers of incorrect data samples into the traffic monitoring system. Thus the key should be stored in tamper-proof hardware. The TSP's public key K_{TSP}, on the other hand, need not be stored in tamper-proof hardware as long as its integrity and authenticity can be verified.

Vehicles' keys can be initially embedded by the manufacturer and updated during regular government vehicle inspections, or during regular maintenance. This allows replacing keys if they have been compromised. If more frequent key updates are necessary, the architecture can be extended to allow over-the-air provisioning of new keys.

6.2.2 A Sanitizer for Traffic Monitoring Systems

The cryptographic authentication mechanisms can address Sybil attacks (provided that keys are hard to generate) and message modifications by network intermediaries but cannot prevent incorrect reports from compromised vehicles. Thus, the TSP should sanitize received data.

There are several existing techniques on sanity check, such as outlier detection, consistency checking, and rule-based classification [16, 23]. These techniques can be leveraged to build a sanitizer component in traffic monitoring systems. For example, the sanitizer may test the integrity of the subject data by comparing the speed information which anonymous vehicle claims on specific road segment at specific time with (1) statistics reported by other vehicles in the same situation, (2) statistics collected at one month ago in the same situation, or (3) adjacent location sample data reported by the same vehicle. As an example, if a malicious vehicle sends a fake message reporting low speed (severe traffic jam) but the sanitizer finds that the majority of probe vehicles on the same road segment at a similar time reports high speed, this can be easily detected as a fake message.

The system can be extended to actively blacklist vehicles that submit apparently incorrect data. Since identities are only maintained at the communication server, the TSP has to return the message with the incorrect data to the CS. The CS in turn looks up the originator of this message (this requires buffering messages for a certain time window) and drops all further messages from this vehicle until its integrity can be established through other means.

6.2.3 Discussion

The architecture presented above can provide privacy guarantees against basic eavesdropping and insider attacks through encryption and the separation of identity and position information. In this section, we consider more sophisticated intrusions at the communication and telematics service provider.

Integrity of Communication Server. The proposed architecture assumes that the CS is trustworthy with respect to data integrity. The architecture provides no cryptographic protection against the CS spoofing, replaying, or dropping messages. In order to relax this trustworthiness assumption, replayed messages could be easily filtered by the sanitizer at the TSP, since no two messages should contain the exact same GPS timestamp and positions. A basic degree of protection against spoofed messages could

also be added through an additional symmetric key, K_{INT}, shared by all vehicles and the TSP. This key can be used to generate a MAC for each location update message that can be verified by the TSP without being able to identify the vehicle. It is expected, however, that this key would need to be updated regularly, since a key shared by a large number of vehicles is difficult to keep secret. Identifying dropped messages proves most difficult.

For a more comprehensive solution, the TSP should continuously monitor the quality of the traffic data by cross-checking with other data sources and monitoring consumer complaints. This monitoring should enable the TSP to identify if a continuous bias in the data is inserted by the CS. It may also make the use of the additional authentication key (K_{CS}) unnecessary.

In this architecture we have deliberately emphasized protection of privacy, since privacy leaks are often more difficult to identify than integrity problems. Since the communication service provider and the telematics service provider will enter a contractual relationship of mutual benefit, it can be expected that both parties have an interest in maintaining data integrity and monitoring the possibility of insider attacks. Individual drivers, however, posses less resources to verify that their private data has not been compromised.

Localization Attack on Network Operators. While the communication server could likely use wireless network localization methods to obtain the position of the mobile node, these methods can be expected to be significantly less accurate.

For example, cell phone localization techniques in the United States were designed to Federal Communications Commission specifications. The E911 Phase II mandate states that the system should locate 67% of calls within 100 meters and 95% of calls within 300 meters. Thus, commonly used technologies such as Uplink Time Difference of Arrival can be expected to provide an order of magnitude less accuracy than in-vehicle GPS, which typically achieves better than 10 m accuracy. More precise may be assisted GPS (A-GPS) technology, which relies on GPS chips in cell phone handsets. A-GPS could be easily disabled, however, for in-vehicle deployment.

Chapter 7
Centralized Approach: Uncertainty-Aware Path Cloaking

In previous chapters, we introduce two novel attacks that re-identify anonymous location traces: target tracking and home identification, and through the analysis of privacy risks in a set of GPS traces from 312 vehicles, we observe that known privacy algorithms cannot achieve accuracy requirements or fail to provide privacy guarantees for drivers in low-density areas.

To overcome these challenges, we propose an uncertainty-aware path cloaking algorithm, a disclosure control algorithm that selectively reveals GPS samples to limit the maximum time-to-confusion for all vehicles. We demonstrate that this algorithm effectively guarantees tracking outliers, while achieving significant data accuracy improvements compared to known algorithms. In particular, we enhance the algorithm to suppresses a risk of identifying drivers' homes by forcing anonymous location samples to be revealed around home locations, where mostly low uncertainty is observed.

7.1 Path Privacy-Preserving Mechanism

Throughout this section, we develop a disclosure control algorithm that provides privacy guarantee even for users driving in low density areas. Given a maximum allowable time-to-confusion and a tracking uncertainty threshold, the algorithm can control the release of a stream of received position samples to maintain the tracking time bounds. It is challenging to design the algorithm to maximize the number of released samples while preserving privacy, and this becomes more challenging particularly when each sample has a different value in traffic estimation.

Since the algorithm must be aware of the positions of other vehicles, we develop a centralized solution that relies on a trustworthy privacy server. Trusted location proxy prevents a complete knowledge of user's location and its identity from being exposed to external service providers. Further, it suppresses the chance of several successive location samples being linked since a partially reconstructed trajectory could act as a quasi-identifier. The use of trusted location proxy is widely accepted because it reduces the risk of trusting entities (i.e., you would have only one single point of attack rather than worrying about all dishonest external service providers) and it is easier to develop.

We first consider the stepwise tracking model without the possibility of path reacquisition. We observe that a specified maximum time to confusion (for a given uncertainty level) can be guaranteed if the algorithm only reveals location samples when (i) time since the last point of confusion is less than the maximum specified time to confusion or (ii) at the current time tracking uncertainty is above the threshold.

Algorithm 2 shows how this idea can be implemented. Note that it describes processing of data from a single time interval, it would be repeated for each subsequent time slot with the state in the vehicle objects maintained. It takes as input the set of GPS samples reported at time t (v.currentGPSSample updated for each vehicle), the maximum time to confusion (confusionTimeout), and the associated uncertainty threshold (confusionLevel). Its output is a set of GPS samples that can be published while maintaining the specified privacy guarantees.

The algorithm proceeds as follows. It first identifies the vehicles that can be safely revealed because less time than confusionTimeout has passed since the last point of confusion (line 12f.) Second, it identifies a set of vehicles that can be revealed because current tracking uncertainty is higher than specified in confusionLevel (line 15-30). Finally, it updates the time of the last confusion point and the last visible GPS sample for each vehicle (line 32ff., the latter is needed for path prediction in the uncertainty calculation). This step can only be performed when the set of revealed GPS samples had been decided, since confusion should only be calculated over the revealed samples.

Algorithm 2 Uncertainty-aware privacy algorithm

```
1: // Determines which location samples can be release while maintaining privacy guarantee.
2: releaseSet = releaseCandidates = {}
3: for all vehicles v do
4:    if start of trip then
5:       v.lastConfusionTime = t
6:    else
7:       v.predictedPos = v.lastVisible.position +
8:       (t-v.lastVisible.time)*v.LastVisible.speed
9:    end if
10:
11:   // release all vehicles below time to confusion threshold
12:   if t - v.lastConfusionTime < confusionTimeout then
13:      add v to releaseSet
14:   else
15:      // consider release of others dependent on uncertainty
16:      v.dependencies = k vehicles closest to the predictedPos
17:      if uncertainty(v.predictedPos, v.dependencies) > confusionLevel then
18:         add v to releaseCandidates
19:      end if
20:   end if
21: end for
22:
23: // prune releaseCandidates
24: for all  v ∈ releaseCandidates do
25:   if ∃ w ∈ v.dependencies. w ∋ releaseCandidates ∪ releaseSet then
26:      delete v from releaseCandidates
27:   end if
28: end for
29: repeat pruning until no more candidates to remove
30: releaseSet = releaseSet ∪ releaseCandidates
31:
32: // release GPS samples and update time of confusion
33: for all v ∈ releaseSet do
34:   publish v.currentGPSSample
35:   v.lastVisible = v.currentGPSSample
36:   neighbors = k closest vehicles to v.predictedPos in releaseSet
37:   if uncertainty(v.predictedPos, neighbors) >= confusionLevel then
38:      v.lastConfusionTime=t
39:   end if
40: end for
```

The second step relies on several approximations. To reduce computational complexity it calculates tracking uncertainty only with the k closest samples to the prediction point, rather than with all samples reported at time t. This is a conservative approximation, since uncertainty would increase if additional samples are taken into

account (see proof in appendix A). In section 7.3, we use two most relevant candidates in tracking uncertainty computation ($k = 2$). Further, it builds a set of releaseCandidates since uncertainty should only be calculated with released samples, but the set of released samples is not determined yet. The algorithm subsequently prunes the candidate set until only vehicles remain who meet the uncertainty threshold. The key property to achieve after the pruning step is that $\forall$ v $\in$ releaseCandidates, uncertainty(v.predictedPos, k closest neighbors in releaseSet $\cup$ releaseCandidates) $\geq$ confusionLevel. The algorithm uses the approximation of calculating the k closest neighbors before the pruning phase, and ensuring during pruning that only vehicles remain if all k neighbors are in the set. While this approximation could be improved in order to release more samples, the current version is sufficient to maintain the privacy guarantee.

7.1.1 Algorithm Extensions for the Reacquisition Tracking Model

The algorithm described so far does not provide adequate privacy guarantees under the reacquisition tracking model because it only ensures a single point of confusion after the maximum time to confusion has expired. Recall that under the reacquisition model an adversary skips samples with high confusion under certain conditions and thus may be able to reacquire the correct trace even after a point of confusion.

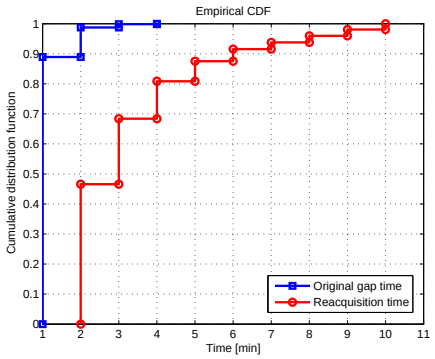


Figure 7.1: Cumulative distribution function of reacquisitions

We observe that such reacquisitions are only possible over short time-scales, since movements after more than several minutes become too unpredictable. To verify this assumption, figure 7.1 shows the longest reacquisition and distribution of reacquisition

length in minutes, empirically obtained from our dataset. As expected, no reacquisitions occur over gaps longer than 10 minutes. Thus, the following extensions can prevent reacquisitions within a time window w. For the experiments reported in the following section we set $w = 10$.

- **After the *confusionTimeout* expires:** In addition to maintaining confusion from the last released position, it is calculated from every prior released location sample (of the same vehicle) within the last w minutes. Samples can only be released if all these confusion values are above the confusion threshold.

- **Before the *confusionTimeout* expires:** Every released sample must maintain confusion to any samples which are released during the last w minutes *and* before the *confusionTimeout* was last reset.

7.1.2 Algorithm Extensions for the Home Identification Attack

The proposed algorithm, by virtue of its design, automatically identify the low density areas and removes location samples in those areas. This property of the algorithm helps preventing home identification since it removes location samples around homes.

To protect your private originations or destinations such as homes or hospitals, intuitively, you need to remove your footprints until you get into crowds, in other words, high confusion. It is the principle behind both two modifications that the algorithm uses its tracking uncertainty computation to detect the confusion. To make the proposed algorithm to do so, we need to deal with some challenges and add some modifications to the algorithm described so far:

- First, we modified the algorithm not to apply *windowing* until users go into high confusion. Once the algorithm detects high confusion, it starts to apply *windowing*.

- Second, we modified the algorithm to disable *windowing* during last T_{guard} minutes, where we use 5 minutes for its value in our experiments.

7.2 Experimental Evaluation

In this section, we present the experimental evaluation of the proposed privacy preserving techniques. Specifically, we demonstrate the effectiveness of our proposed techniques for privacy protection in the analysis of GPS traces. The analysis of the evaluation includes privacy preservation against home identification and target tracking attacks. Also, we evaluate how our proposed privacy preserving techniques can maintain the quality-of-service for the traffic monitoring application.

7.2.1 Experimental Setup

Experimental Data Sets. Throughout the experiments, we used (offline collected) real GPS traces from 312 probe vehicles in our trace-driven simulations. Conducting the target tracking and home identification experiments on real GPS traces, we capture real vehicle movements, density, GPS inaccuracies, and road network artifacts. In the experiments, we first applied privacy preserving techniques (i.e., the proposed one and the baseline) on the GPS traces and then measured the performance of privacy protection using target tracking and home identification techniques on these privacy-preserved GPS traces.

Since target tracking typically is only effective for a short time period, we only use 24-hour GPS traces out of a set of week-long GPS traces. This approach helps create a high density scenario (500 and 2000 probe vehicles on a 70km^2 region) with a limited number of probe vehicles. We overlay GPS traces of different volunteer drivers at the same time frame (24 hours) of different dates. This overlay method has a limitation in that it generates similar routes by aggregating GPS traces from the same set of drivers. However, we still believe that it provides insights into higher density scenarios. We will revisit this limitation in section 8.6.

Evaluation Metrics. In our experiments, we applied the following metrics to evaluate our privacy preserving algorithms for GPS traces.

Home Identification Rate. This metric measures the percentage of likely correct

home position identifications. Since no ground truth is available, we have manually inspected the *unmodified* traces and chosen selected 65 traces, where the vehicle visited one residential building significantly more frequently than others. We marked the position of this building as a likely real home position and measure which percentage of these positions is also selected by the automated home identification algorithm based on the *privacy-enhanced* traces. We also measure *false positives*, positions selected by the algorithm that do not match the manually chosen ones, to provide an indication of the precision of the algorithm.

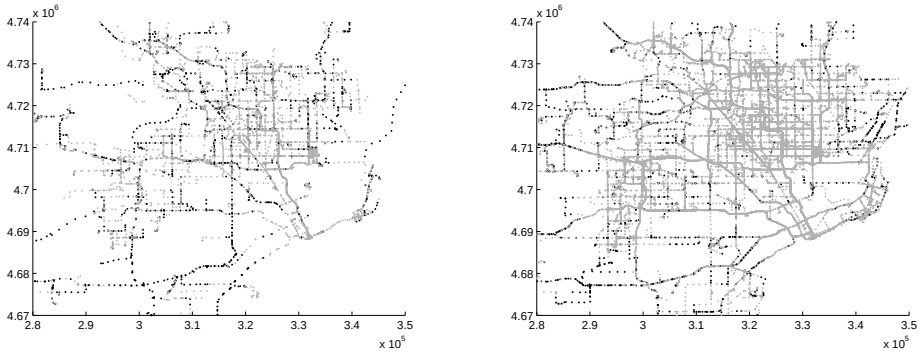
Tracking Time. Minimizing tracking time reduces the risk that an adversary can correlate an identity with sensitive locations. We use *time to confusion (TTC)*, which we defined in section 8.1 as a privacy metric, to measure the tracking duration. To better demonstrate the bounded privacy protection of our proposed algorithm, we report two statistics: the maximum value of TTC and the median value of TTC.

(Relative) Weighted Road Coverage. Through this metric, we measure the data quality that the privacy-preserved traces provide for the traffic monitoring applications. Also, this metric captures the value of each sample based on whether sampled on busier roads or not. Since privacy protection techniques, in general, incur a tradeoff between privacy protection and quality-of-service, our proposed solution aims to provide reasonable privacy protection while delivering the same road coverage for satisfying the need of the traffic monitoring applications. In this thesis, we use *relative* road coverage as we defined in section 8.2. In addition to this metric, we also provide the percentage of released location sample compared to the original traces which we consider 100%. Note that both metrics are normalized by values of the original GPS traces.

Ground truth. Since the real home addresses are unavailable (driver identities were omitted in the dataset for privacy reasons), we manually inspected the unmodified week-long traces overlayed on satellite images to identify plausible home locations as ground truth. To reduce the time taken for manual inspection, we choose a subset of the region covered by the 312 traces in the dataset (each trace corresponds to one driver). The subset contains the two residential regions (together a 25km-by-25km area) marked with rectangles in Figure 4.5(a) and contains 65 plausible homes found

through manual inspection. Since no ground truth for drivers' homes is available, all of the 65 reference locations from manual inspection contained a single home that stood out as a likely home location. The drivers visited this home much more frequently at night than others. Therefore, we believe the manual inspection provide a reasonable approximation of real home locatoins.

7.2.2 Snapshots of Privacy-preserving GPS Traces



(a) Snapshot of privacy-preserving GPS traces generated by uncertainty-aware path cloaking at off-peak time (over 1.5 hours) in a high density scenario

(b) Snapshot of privacy-preserving GPS traces generated by uncertainty-aware path cloaking algorithm at peak time (over 1.5 hour) in a high density scenario

Figure 7.2: Uncertainty-aware privacy algorithm removes more samples in low-density areas, in which vehicles could be easily tracked. Gray dots indicate released location samples, black ones denote removed samples.

Before evaluating the performance of our proposed technique, let us compare the privacy-preserved GPS traces generated by the proposed path cloaking algorithm with the original GPS traces to highlight major changes in modified traces. Figures 7.2(a) and 7.2(b) show both in a high user density scenario for off-peak (over 1.5 hours at 10am) and peak time (over 1.5 hour at 5pm), respectively. Gray dots indicate released location samples while black dots illustrate samples removed by path cloaking. We observe two characteristics from these traces. First, uncertainty-aware path cloaking removes fewer location samples at peak time and second, it retains more location samples within the presumably busier downtown area. This illustrates how the algorithm, by virtue of its design, retains information on busier roads where traffic information is

most valuable.

7.3 Results

The following target tracking experiment illustrates how the path cloaking algorithm prevents an adversary from reconstructing an individual's path using the cleansed GPS traces *and* locating an individual's home. Specifically, we compare our uncertainty-aware privacy algorithm and its *with-reacquisition* version with random subsampling in terms of maximum and median TTC for configurations that produce the same number of released location samples (as a metric of data quality). Also, we compare a set of same algorithms each other in terms of home identification rate and the number of released location samples. We evaluate the effectiveness of our proposed privacy preserving algorithms by answering the following questions:

- Do uncertainty-aware privacy algorithms effectively limit tracking time (i.e., guarantee time-to-confusion)? Are these limits maintained even in low-user density scenarios?
- How does the average tracking time allowed by path cloaking compare to the subsampling baseline, at the same data quality level.
- How are the results affected by the choice of data quality metric (percentage of released location samples vs relative weighted road coverage)?

7.3.1 Protection Against Target Tracking

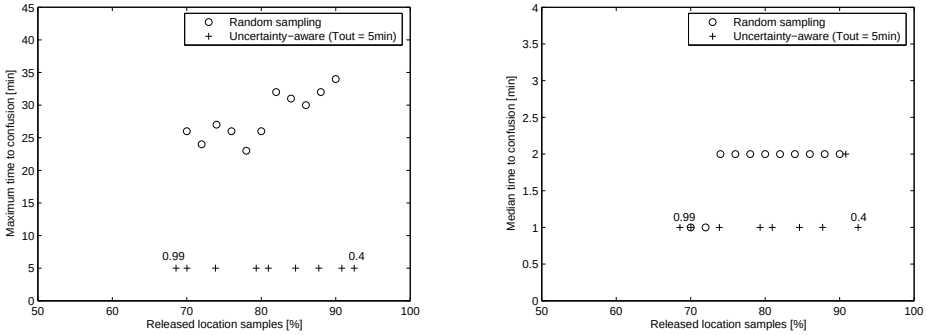
Throughout the results presented in the following subsections, one graph depicts many experiment trials, where one trial comprises the following steps. We first apply a privacy algorithm to the low-density (500 vehicle) or high-density (2000 vehicle) dataset generated from the 312 original vehicle traces. We then remove vehicle identifiers and execute the target tracking algorithm (see Sec. 8.1) to measure tracking time for the first 312 vehicles. For each vehicle, we compute the tracking time starting from each

sample of the trace and report the maximum. One data point shown in the graph then corresponds to the median or maximum over the 312 vehicle tracking times computed for one trial. For each graph, these trials are then repeated with different uncertainty thresholds for the path cloaking algorithms and different probabilities of removal in the subsampling algorithm.

Bounded Tracking Time without Reacquisition. First, we ascertain whether the uncertainty-aware privacy algorithm guarantees bounded tracking under the no reacquisition tracking assumption . Figures 7.3(a) and 7.3(b) show the maximum and median tracking time plotted against the relative amount of released location samples, respectively, for a high density scenario with 2000 vehicles in the 70km-by-70km area. Figure 7.3(a) shows results for the uncertainty-aware privacy algorithm (marked with +) for varying uncertainty levels with timeout fixed at 5 minutes and for the random subsampling algorithm for varying probabilities of removal. Since the configuration parameters from these algorithms are not directly comparable, the graph shows the percentage of released location samples on the x-axis, allowing comparison of TTC at the same data quality level. Also note that graph compares the algorithms in terms of maximum tracking time, to illustrate differences in tracking time variance and outliers. During tracking we set the adversary's uncertainty threshold to 0.4. This means that the adversary will give up tracking if at any point the uncertainty level rises above this threshold, because the correct trace cannot be determined. A 0.4 uncertainty level corresponds to a minimum probability of 0.92 for the most probable next location sample.

As evident from the data, the uncertainty-aware privacy algorithm effectively limits time to confusion to 5 min, except for very low privacy settings (i.e., low uncertainty threshold less than 0.4), while the random sampling algorithm allows some vehicles to be tracked up to about 35min. Our proposed algorithm can release up to 92.5% of original location samples while achieving the bounded tracking property.

In figure 7.3(b), we see that naturally occurring crossings and merges in the paths of nearby vehicles lowers median TTC to 1 or 2 minutes (with reacquisition it would be higher, though). However, with random subsampling (20% removal), about 15% of



(a) The Maximum Value of TTC using Uncertainty-aware privacy algorithm without Reacquisition

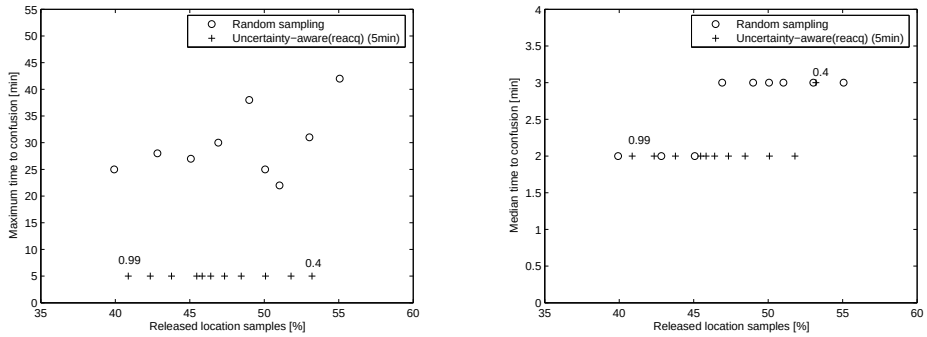
(b) The Median Value of TTC using Uncertainty-aware privacy algorithm without Reacquisition

Figure 7.3: Maximum / Median tracking duration for different privacy algorithms in high density scenarios (2000 vehicles / 1600 sqm). The Uncertainty-aware privacy algorithm outperforms random sampling for a given number of released location samples.

vehicles (34 out of 233) can still be tracked longer than 10 minutes. The uncertainty-aware path cloaking can guarantee the specified maximum tracking time of 5min even for these vehicles with higher data quality, removing only 17.5% of samples.

Dependence on Reacquisition and Density. We now repeat the same experiment under the reacquisition tracking model, where an adversary may skip ahead over a point of confusion. Figure 7.4(a) (note scaled x-axis) shows that the uncertainty-aware privacy algorithm with reacquisition extensions can also effectively limit tracking time under this model, while subsampling allows a worst case tracking time of 42 min. Figure 7.4(b) also shows that the median tracking time is increased by one minute due to the change in tracking model. The maximum allowable amount of released location samples is decreased compared to that of figure 7.3.

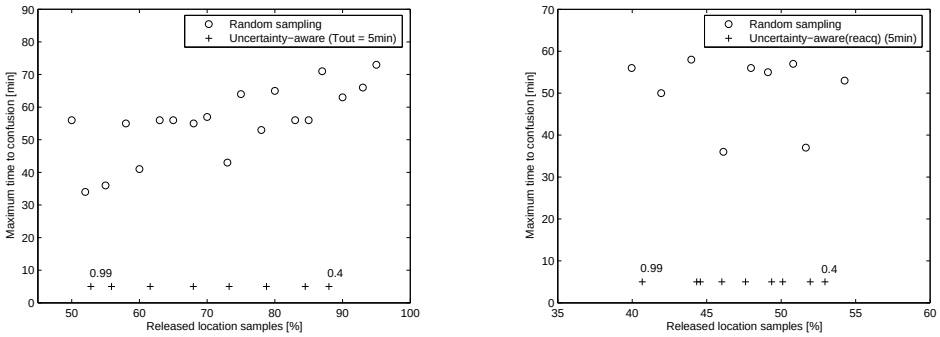
Let us now investigate whether the privacy guarantee is also maintained in a very low user density scenario with only 500 probe vehicles. Figure 7.5 shows that this is indeed the case both with and without the reacquisition model. While subsampling allows a longer maximum TTC due to the low user density, our proposed scheme still preserves the maximum TTC guarantee of 5 minutes by removing 1.8% to 14.8% more samples (for uncertainty thresholds between 0.4 and 0.99). The same result can be observed in figure 7.5(b) with reacquisition, except that the difference in samples removed



(a) The Maximum Value of TTC using the (with reacquisition) Uncertainty-aware privacy algorithm

(b) The Median Value of TTC using the (with reacquisition) Uncertainty-aware privacy algorithm

Figure 7.4: Maximum / Median tracking duration for different privacy algorithms in high density scenarios (2000 vehicles / 1600 sqm) under the reacquisition tracking model.



(a) The Maximum Value of TTC using the Uncertainty-aware privacy algorithm without Reacquisition

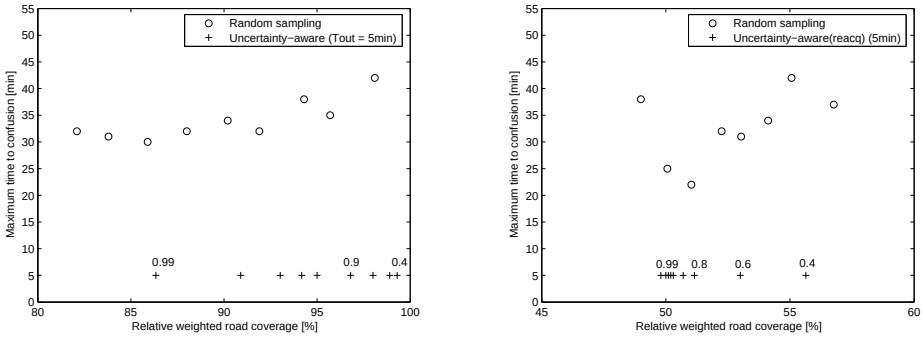
(b) The Maximum Value of TTC using the (with reacquisition) Uncertainty-aware privacy algorithm

Figure 7.5: The Uncertainty-aware privacy algorithm and its (with reacquisition) version outperform a random subsampling at a given range of sample removal also in the low density scenarios (500 vehicles / 1600 sqm).

is not as pronounced. Compared to the high density scenario, our proposed algorithm requires removing more samples to achieve the bounded tracking property in the lower user density scenario.

Quality of Service Analysis. So far, we have measured quality of service in terms of the percentage of samples removed by the algorithm. Since samples in higher density areas are more important for the traffic monitoring application, the benefits of our proposed privacy algorithm are even more significant if we consider *relative weighted*

road coverage. More details are shown in figure 7.6. Figure 7.7(b) further shows that the uncertainty-aware privacy algorithm achieves a relative weighted road coverage similar to that of original location traces even though the actual number of released location samples is lower than that of original location traces as shown in figure 7.7(a). Figure 7.2 explains this results, in that the algorithm retains most samples in high-density areas and removes most from lower densities. However, the uncertainty-aware privacy algorithm with reacquisition extensions provides a slight improvement of relative QoS for weighted road coverage. More detailed statistics on this improvement are provided in table 7.1.



(a) Comparison of Maximum TTC against Weighted Road Coverage in High Density Scenario (Uncertainty-aware privacy algorithm)

(b) Comparison of Maximum TTC against Weighted Road Coverage in High Density Scenario ((with reacquisition) Uncertainty-aware privacy algorithm)

Figure 7.6: Time-to-confusion advantages of uncertainty-aware path cloaking become even more pronounced when comparing algorithms with the traffic-monitoring-specific (Relative) Weighted Road Coverage data quality metric.

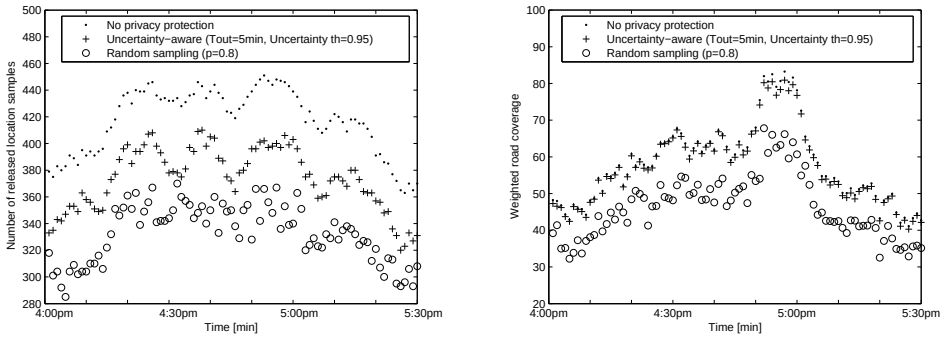
7.3.2 Protection Against Home Identification

In this section, we present the experimental evaluation of additional enhancements to our uncertainty-aware path cloaking algorithm. Specifically, we demonstrate the effectiveness of our proposed algorithm for home identification protection against map-aware adversary.

The following experiment illustrates how an uncertainty-aware path cloaking algorithm suppresses a risk of home identification. Specifically, we compare our proposed

	QoS metrics	
	Released location samples	Weighted road coverage
Original traces	100%	100%
Uncertainty-aware privacy (5min,0.95)	81%	95.0%
Random sampling (0.8)	80%	79.3%
(with reacq) Uncertainty-aware (5min,0.4)	53.2%	55.6%
Random sampling (0.53)	53%	52.9%

Table 7.1: Quality of service enhancement in each of Uncertainty-aware privacy algorithm, (with reacquisition) Uncertainty-aware privacy algorithm, and random sampling compared to the QoS level which original traces can achieve.



(a) Number of Released Location Samples in Peak Time

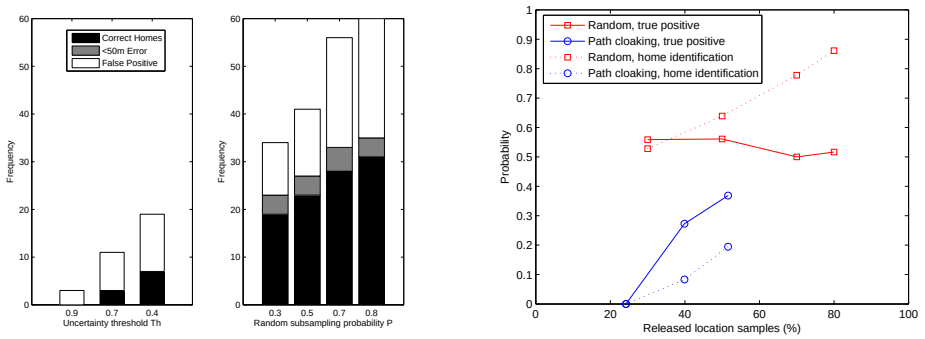
(b) Weighted Road Coverage in Peak Time

Figure 7.7: The Uncertainty-aware privacy algorithm removes more samples in low density area, leading to enhanced QoS in the high density regions, where traffic monitoring information is most valuable.

algorithm with random subsampling that we used in target tracking analysis in a previous section in terms of home identification risks and data quality.

Out of 65 manually inspected home locations in our dataset, we only select 37 homes as marked by ”house” symbol in figure 4.5(a). Because we need to apply an uncertainty-aware path cloaking algorithm over week-long traces [1], we could not overlay GPS traces of different dates to create high density scenario. We only consider a relatively low density scenario where we perturbed 312 week-long traces with our proposed technique and random subsampling, and applied home identification algortthm to them.

[1]Repeated visits are very important factors in home identification experiments.

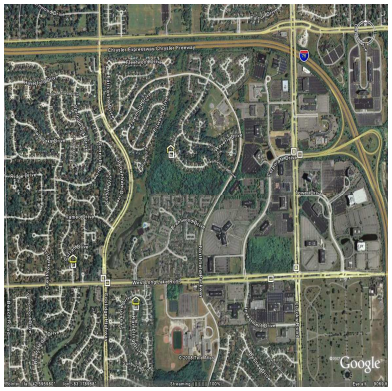


(a) Home Identification Details. (b) Random Sampling vs. Path Cloaking.

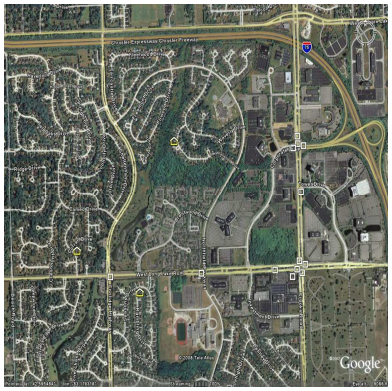
Figure 7.8: Both random sampling and path cloaking suppress home identification with lower samples revealed. However, in terms of true positive, an adversary can achieve the constant rate even against lower pecentage of revealed samples in random sampling.

For random subsampling, we varied a probability of anonymous location sample selection from 0.3 to 0.8. To have similar range of total number of release location samples, we varied uncertainty threshold from 0.9 to 0.4. For a set of returned clusters by each method, we count the followings among a set of clusters: (1) the number of clusters that exactly point to correct homes; (2) the number of clusters that are located within 50m from corrrect homes; and (3) the number of clusters that point to other buildings or homes that are not in a set of manually identified homes (or so called false positive). Each bar in figure 7.8(a) represents a tuple of three numbers for each method. Note that removing 70% of location traces still allows 19 out of 37 homes identified correctly and 4 homes narrowed down within 50m.

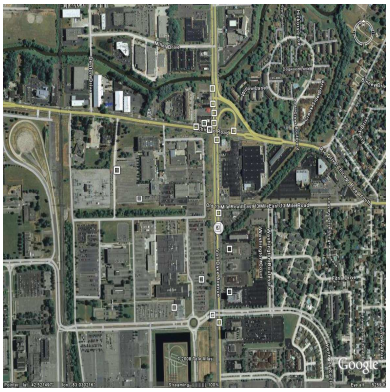
To enable a more meaningful evaluation of our proposed technique and random subsampling, we plot a quality versus privacy graph in figure 7.8(b). Recall that we only use 312 traces, so that more location samples must be witheld than what we observed in target tracking analysis with identical values of uncertainty threshold. For our proposed technique, we plot two different metrics, *true positive* meaning how many homes are correct among the estimated home locations, and *home identification rate*, meaning how many homes out of 37 manually identified homes are correctly detected. The former metric is more meaningful to an adversary who does not know a priori users' home locations. Meanwhile, the latter metric is useful for system designers



(a) Clusters after Random Subsampling



(b) Clusters after Uncertainty-Aware



(c) Clusters around Work Places after Uncertainty-Aware Patch Cloaking

Figure 7.9: The uncertainty-aware path cloaking pushes clusters towards roads from residential areas. However, note that it still leaves clusters near destinations such as work places where multiple users visit at the same time. 'House' symbol and rectangle symbol depict manually identified home and estimated home location, respectively.

that should determine proper parameters such as uncertainty threshold in our proposed technique or a selection probability in random subsampling to limit the absolute maximum number of correctly identified homes. Note that random subsampling returns a constant level of true positive even though we decrease a selection probability. Compared to random subsampling techniques, our proposed techniques better preserve user privacy against home identification attack.

Protection Against Place Identification. The uncertainty-aware path cloaking automatically detects how sensitive originations and destinations of trips are. It removes location samples around private places such as homes. By removing them, it pushes

out centroids of an adversary's clustering towards roads. Meanwhile, for destinations where many different users visit at the same time such as shopping malls or work places as shown in figure 7.9(c), it allows location samples even near to specific shops or several entrances. This leads to centroids remained near destinations, but it does not compromise user privacy much because the destinations such as shopping malls cannot be hardly mapped to a single identity. Two snapshots in figure 7.9 in the below qualitatively illustrate our observations.

7.4 Discussion

In this section, we discuss the limitation of the dataset and the experiment, the possible extensions of our proposed algorithm, and some future directions.

Map-based Tracking. An adversary can enhance tracking model in several ways if the knowledge on map and road network is available. Tracking performance would be improved if an adversary uses road network distance instead of Euclidean distance in computing likelihoods for candidate samples. Also, the algorithm could adjust the predicted location based on actual roadway positions [32]. For example, the adversary could assign a lower probability to a sample if no direct road connection exists, even though the sample is near the predicted position. To deal with this enhanced tracking, our proposed algorithm could also take these road maps into account when computing tracking uncertainty. The complete analysis remains an open problem for future work.

Prior Knowledge on Subjects. In this work, we assume that an adversary does not have any a priori knowledge about the subject being tracked when we develop inference attack models. Even if home identification and tracking in general remain difficult, an analyst could infer sensitive information by focusing on a select individual. For example, given the home and work position of an individual, it is possible to determine when the person left home and passed an accident site because the tracking analysis a priori knows the destination of the trip. The detailed analysis of this case also remains for future work. A similar approach can be found in recent study by Narayanan and Shmatikov [76], where they demonstrated that IMDb database helps de-anonymize the

public Netflix dataset.

Relaxing trust in location server. The algorithm described so far relies on a trustworthy location server, since the algorithm needs the full GPS traces of all vehicles. A fully distributed algorithm poses a research challenge by itself, since clients would need to monitor the positions of neighboring cars, which again raises privacy and trust issues. It also appears possible, though, to relax the trust assumptions in the location server through a hybrid approach, with additional in-vehicle disclosure control based on coarser information about neighbors. Since data quality would only be marginally affected by missing updates in low-density areas, one could devise schemes to inform vehicle of the approximate probe density in their area. Then vehicles could reduce location updates to the server in the most sensitive low-density areas. To prevent spoofing of such density information, further research could investigate data cross-validation schemes or secure multi-party computation schemes to compute density. Along with this direction, recent several studies [48, 31, 86] address the problem of developing more distributed and more client-oriented solutions without the involvement of a service provider or a trust proxy.

Dataset limitations. We need to point out that the tracking results can be affected by the choice of probe vehicles. In our dataset, most drivers shared the same workplace. Thus, the workplace acted as a place of confusion, where the tracking algorithms failed. A random sample of the population would probably improve tracking performance. This would cause both our proposed algorithms and the random sampling method to remove more samples to meet the maximum TTC. The performance gap between them might also change from what we have observed in our study. In addition, our method of overlaying multiple datasets to create one high-density scenario may not be entirely faithful in representing true traffic conditions. Due to this overlay, some of the vehicles may also be driven by the same driver on similar routes, creating a further bias towards reduced tracking performance. Nonetheless, we observed that naive anonymization is problematic and our proposed algorithm saves a lot more samples than the baseline. We still believe our current results provide a valuable first step towards understanding tracking performance in probe vehicle scenarios.

Privacy risk model and client-based solutions. The privacy risk model that relates several major system parameters to a degree of privacy risk might be a challenging task. Major system parameters such as user density and sampling interval influence the tracking performance. Also, several dynamics behind vehicle mobility such as the car-following model, signalized intersections, and road network types (e.g., rural, urban, and city) gives uncertainty in target tracking. The privacy risk model can give system designers an idea of privacy risk in collected dataset for a given set of system parameters and environment.

Guaranteed protection against home identification. Our proposed scheme dramatically suppresses the home identification rate than the baseline, but it is not an optimal solution that guarantees no single identified home and saves as many samples as possible. Designing guaranteed protection algorithm remains one of future works.

Path cloaking and k-anonymity. We could have several variations of uncertainty-aware path cloaking algorithms by tweaking the tracking uncertainty computation in several different ways. Using heading information as well as distance gap in a likelihood computation must be one of variations. Another example might be to introduce a concept of *k-anonymity* in a tracking uncertainty computation. Specifically, we compute a tracking uncertainty using all k candidate samples within a fixed bounding region around a predicted position. We only release location samples if k is greater than k_{th}, a threshold mendating k_{th}-1 users in a fixed boundary, as well as the computed uncertainty is greater than a predefined minimum tracking uncertainty. This approach uses a combination of k values and tracking uncertainty as a condition upon releasing a sample.

7.5 Conclusions

In this chapter, we have proposed a novel time-to-confusion metric to characterize the degree of privacy in an anonymous set of location traces. We presented two different privacy risks in anonymous location traces: target tracking and home identification. We then developed an uncertainty-aware privacy algorithm, which can guarantee a defined

maximum time-to-confusion for all vehicles, even those driving in low density areas. We showed through experiments with real-world GPS traces that the algorithm can effectively guarantee a maximum time-to-confusion, while a random sampling baseline algorithm allows tracking outliers for vehicles in low density regions at the same data accuracy level. Furthermore, we observed that our proposed algorithm automatically removes anonymous location samples in low uncertainty near origins and destinations, thereby reducing a risk of home identification.

Acknowledgment

The author would like to thank anonymous volunteering drivers who provide us their GPS traces for the purposes of this study.

Chapter 8

Distributed Approach: k-Anonymous Location Updates via VTL-based Temporal Cloaking

The architecture described so far in the previous chapter requires a trustworthy proxy server. Since this approach reduces the risk of trusting dishonest entities from the number of external service providers to the proxy server, it is still exposed to the inference attack by an insider at the proxy server.

Motivated by a millionaire problem [96], we split secret information (a complete knowledge of probe vehicle's identity and fine-grained location information for a given time) over multiple parties so that no single entity has a complete knowledge of it. This approach removes the reliance on the proxy server. Specifically, we propose a system based on virtual trip lines and an associated cloaking technique. Virtual trip lines are geographic markers that indicate where vehicles should provide location updates. These markers can be placed to avoid particularly privacy-sensitive locations. They also allow aggregating and cloaking several location updates based on trip line identifiers, without knowing the actual geographic locations of these trip lines. Thus they facilitate the design of a distributed architecture, where no single entity has a complete knowledge of probe identities or fine-grained location information.

Contributions. Several studies have demonstrated the feasibility of traffic flow estimation through analysis, simulations, and experiments [44, 36, 98]. Several open questions remain, however, before such a system is likely to be realized. First, it is unclear how such a system can quickly be bootstrapped since the service is only useful with sufficient participants. While telematics platforms or navigations system hardware is capable of performing these functions, these platforms are not openly programmable and thus hard to retrofit for this purpose. Second, it is not known how the quality

of the obtained traffic information compares with those collected through conventional methods (e.g., loop detectors). Third, the system requires that cars reveal their positions to a traffic monitoring organization, raising privacy concerns. Our earlier work [58] has proposed privacy enhancing technologies that can alleviate concerns. These solutions, however, still require users to trust centralized privacy servers.

To address these challenges, we propose a novel traffic monitoring system design based on the concept of *virtual trip lines (VTLs)* and experimentally evaluate its accuracy. Virtual trip lines are geographical markers stored in the client, which trigger a position and speed update whenever a probe vehicle passes. Through privacy-aware placement of these trip lines, clients need not rely on a trustworthy server. The system is designed for GPS-enabled cell phones to enable rapid software deployment to a large and increasing number of programmable smart phones. The key contributions of this work are:

- Arguing that sampling in space (through virtual trip lines) rather than in time leads to increased privacy because it allows omitting location samples from more sensitive areas.
- Describing a privacy-aware placement approach that creates the virtual trip line database.
- Demonstrating that the virtual trip line concept can be implemented on a GPS-enabled cellular phone platform.
- Evaluating accuracy and privacy through a 20 vehicle experiment on a highway segment.

Impacts. We believe the work presented in this chapter is relevant for two practical reasons. First, many different businesses are currently competing to provide traffic monitoring services and it is likely that some cellular handset based solutions will emerge that do not involve the cellular network operator at all. From this perspective it is interesting to ask how privacy preserving traffic monitoring could be implemented

by separate entities that currently do not yet have access to identity and location information, without requiring users to trust and reveal their information to these additional organizations. Second, one can rarely have complete trust in one entity which would imply trust in every single employee with access to the records (recall that insiders are responsible for a majority of privacy leaks). One can usually only have a high level of trust, if that organization appears serious in putting protection mechanisms in place to address these risks. From this perspective it is interesting to ask how a single company could improve privacy to protect its reputation and its customers from such insider (and outsider attacks).

We argue that privacy in traffic monitoring can be improved through VTLs because (a) they allow careful placement so that one can avoid transmitting location data in more privacy sensitive areas (which is more difficult to implement with temporal sampling) and (b) the usage of a VTL pseudonym allows us to perform temporal cloaking while no single entity has access to identity, location, and time information.

We expect that in actual implementations of this architecture different mappings will emerge. One extreme case may be three separate companies/organizations implementing the system with no involvement of the network operator (the only limitation is that one of the identities needs to be able to approximately, at a very coarse level of 10s of miles or more, verify client location claims. This verification could be provided by a network operator but other forms of verification are also plausible). Another extreme case would be a cellular network operator creating three entities within the company to improve privacy of their traffic monitoring system. Hybrid solutions between the two, as shown in the chapter, are also possible. Clearly, the first would be more preferable from a privacy perspective, but in the end both lead to a significant improvement in privacy over a naive implementation.

8.1 Privacy Risks and Threat Model

Traffic monitoring through GPS-equipped vehicles raises significant privacy concerns, however, because the external traffic monitoring entity acquires fine-grained movement

traces of the probe vehicle drivers. These location traces might reveal sensitive places that drivers have visited, from which, for example, medical conditions, political affiliations, romantic relationship, speeding, or potential involvement in traffic accidents could be inferred.

Threat Model and Assumptions. This work assumes that adversaries can compromise any single infrastructure component to extract information and can eavesdrop on network communications. We assume that different infrastructure parties do not collude and that a driver's own handset is trustworthy. We believe this model is useful in light of the many data breaches that occur due to dishonest insiders, hacked servers, stolen computers, or lost storage media (see [7] for an extensive list, including a dishonest insider case that released 4500 records from California's FasTrak automated road toll collection system). These cases usually involve the compromise of log files or databases in a single system component and motivate our approach of ensuring that no single infrastructure component can accumulate sensitive information.

We consider sensitive information any information from which the precise location of an individual at a given time can be inferred. Since traffic monitoring does not need to rely on individual node identities, only on the aggregated statistics from a large number of probe vehicles, an obvious privacy measure is to anonymize the location data by removing identifiers such as network addresses. This approach is insufficient, however, because drivers can often be re-identified by correlating anonymous location traces with identified data from other sources. For example, home locations can be identified from anonymous GPS traces [72, 57] which may be correlated with address databases to infer the likely driver. Similarly, records on work locations or automatic toll booth records could help identify drivers. Even if anonymous point location samples from several drivers are mixed, it can be possible to reconstruct individual traces because successive flow updates from the same vehicle inherently share a high spatio-temporal correlation. If overall vehicle density is low, location updates close in time and space likely originate from the same vehicle. This approach is formalized in target tracking models [79].

As an example of tracking anonymous updates, consider the following problem:

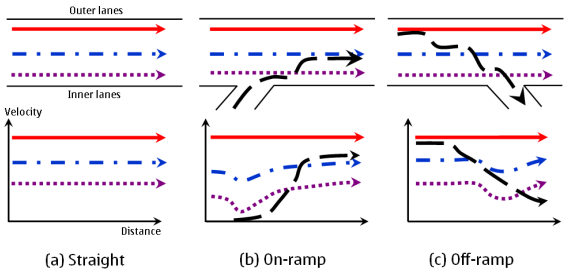


Figure 8.1: Driving Patterns and Speed Variations in Highway Traffic.

given a time series of anonymous location and speed samples mixed from multiple users, extract a subset of samples generated by the same vehicle. To this end, an adversary can predict the next location update based on the prior reported speed $\hat{x}_{t+\Delta t} = v_t \cdot \Delta t + x_t$ of the actual reported updates, where x_t and $x_{t+\Delta t}$ are locations at time t and $t + \Delta t$, respectively, and v_t is the reported speed at t. The adversary then associates the prior location update with the next update closest to the prediction, or more formally with the most likely update, where likelihood can be described through a conditional probability $P(x_{t+1}|x_t)$ that primarily depends on spatial and temporal proximity to the prediction. The probability can be modelled through a probability density function (pdf) of distance (or time) differences between the predicted update and an actual update (under the assumption that the distance difference is independent of the given location sample).

Knowing speed patterns further helps tracking anonymous location samples if it is combined with map information. For example, consider the traffic scenarios depicted in figure 8.1. On straight sections (a) vehicles on high-occupancy vehicle (HOV) or overtaking lanes often experience less variance in speed. Vehicles entering at an on-ramp (b) or exiting after an off-ramp (c) usually drive slower than main road traffic. These general observations can be formally introduced into the tracking model by assigning an a priori probability derived from the speed deviations. For example, to identify the next location sample after an on-ramp for a vehicle that generated x_t on the main route before the ramp, an adversary could assign a lower probability to location updates with low speed. These low speed samples are likely generated by vehicles that just entered after the ramp.

Privacy Metrics. As observed in [58], the degree of privacy risk depends on how long an adversary successfully tracks a vehicle. Longer tracking increases the likelihood that an adversary can identify a vehicle and observe it visiting sensitive places. We thus adopt the *time-to-confusion* [58] metric and its variant *distance-to-confusion*, which measures the time or distance over which tracking may be possible. Distance-to-confusion is defined as the travel distance until tracking uncertainty rises above a defined threshold. Tracking uncertainty is calculated separately for each location update in a trace as the entropy $H = -\sum p_i \log p_i$, where the p_i are the normalized probabilities derived from the likelihood values described later. These likelihood values are calculated for every location update generated within a temporal and spatial window after the location update under consideration.

These tracking risks and the observations regarding increased risks at certain locations further motivate the virtual trip line solution described next. Compared to a periodic update approach, where clients provide location and speed updates at regular time intervals, virtual trip lines can be placed in a way to avoid updates from sensitive areas.

8.2 Traffic Monitoring with Virtual Trip Lines

We introduce the concept of virtual trip lines for privacy-preserving monitoring and describe two architectures that embody it. The first architecture seeks to provide probabilistic privacy guarantees with virtual trip lines. The extended second architecture demonstrates how virtual trip lines can help computing k-anonymous location updates via temporal cloaking, without using a single trusted server.

8.2.1 Virtual Trip Line Concept

The proposed traffic monitoring system builds on the novel concept of virtual trip lines and the notion of separating the communication and traffic monitoring responsibilities (as introduced in [57]). A *virtual trip line* (VTL) is a line in geographic space that, when crossed, triggers a client's location update to the traffic monitoring server. More

specifically, it is defined by

$$[id, x1, y1, x2, y2, d]$$

where id, is the trip line ID, $x1$, $y1$, $x2$, and $y2$ are the (x, y) coordinates of two line endpoints, and d is a default direction vector (e.g., N-S or E-W). When a vehicle traverses the trip line its location update comprises time, trip line ID, speed, and the direction of crossing. The trip lines are pregenerated and stored in clients.

Virtual trip lines control disclosure of location updates by sampling in space rather than sampling in time, since clients generate updates at predefined geographic locations, compared to sending updates at periodic time intervals. The rationale for this approach is that in certain locations traffic information is more valuable and certain locations are more privacy-sensitive than others. Through careful placement of trip lines the system can thus better manage data quality and privacy than through a uniform sampling interval. In addition, the ability to store trip lines on the clients can reduce the dependency on trustworthy infrastructure for coordination.

8.2.2 Architecture for Probabilistic Privacy

To achieve the anonymization of flow updates from clients while authenticating the sender of flow updates, we split the actions of authentication and data processing onto two different entities, an ID proxy server and a traffic monitoring server. By separately encrypting the identification information and the sensing measurements (i.e., trip line ID, speed, and direction) with different keys, we prevent each entity from observing both the identification and the sensing measurements.

Figure 8.2 shows the resulting system architecture. It comprises four key entities: probe vehicles with the cell phone handsets, an ID proxy server, a traffic monitoring service provider, and a VTL generator. Each probe vehicle carries a GPS-enabled mobile handset that executes the client application. This application is responsible for the following functions: downloading and caching trip lines from the VTL server, detecting trip line traversal, and sending measurements to the service provider. To determine trip line traversals, probe vehicles check if the line between the current GPS

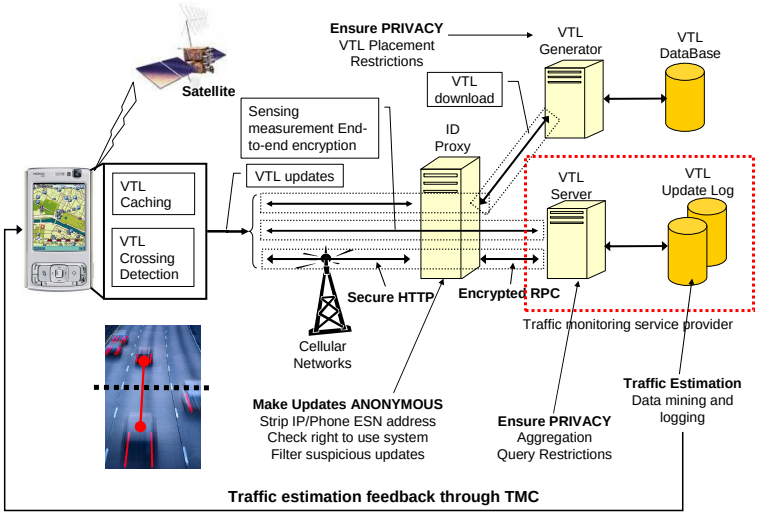


Figure 8.2: Virtual Trip Line: Privacy-Preserving Traffic monitoring System Architecture.

position and the previous GPS position intersects with any of the trip lines in its cache. Upon traversal, handsets create a VTL update comprising trip line ID, speed readings, timestamps, and the direction of traversal and encrypt it with the VTL server's public key. Handsets then transmit this update to the ID proxy server over an encrypted and authenticated communication link set up for each handset separately. Each handset and the ID proxy share an authentication key in advance.

The ID proxy's responsibility is to first authenticate each client to prevent unauthorized updates and then forward anonymized updates to the VTL server. Since the VTL update is encrypted with the VTL server's key, the ID proxy server cannot access the VTL update content. It has knowledge of which phone transmitted a VTL update, but no knowledge of the phones position. The ID proxy server strips off the identifying information and forwards the anonymous VTL update to the VTL server over another secure communication link.

The VTL server aggregates updates from a large number of probe vehicles and uses them for estimating the real-time traffic status. The VTL generator determines the position of trip lines, stores them in a database, and distributes trip lines to probe vehicles when any download request from probe vehicles is received. Similar to the ID proxy, each handset and the VTL generator should share an authentication key in advance.

The VTL generator first authenticates each download requester to prevent unauthorized requests and can encrypts trip lines with a key agreed upon between the requester and the VTL generator.[1] Both the download request message and the response message are integrity protected by a message authentication code.

Discussion. The above architecture improves location privacy of probe vehicle drivers through several mechanisms. First, the VTL server must follow specific restrictions on trip line placements that we will describe in section 8.3. This means that a handset will only generate updates in areas that are deemed less sensitive and not send any information in other areas. By splitting identity-related and location-related processing, a breach at any single entity would not reveal the precise position of an identified individual. A breach at the ID proxy would only reveal which phones are generating updates (or are moving) but not their precise positions. Similarly, a breach at the VTL server would provide precise position samples but not the individual's identities. Separating the VTL server from the VTL generator prevents active attacks that modify trip line placement to obtain more sensitive data. This is, however, only a probabilistic guarantee because tracking and eventual identification of outlier trips may still be possible. For example, tracking would be straightforward for a single probe vehicle driving along on empty roadway at night. The outlier problem in sparse traffic situations can be alleviated by changing trip lines based on traffic density heuristics. Trip lines could be locally deactivated by the client based on time of day or the clients speed. They could also be deactivated by the VTL generator based on traffic observations from other sources such as loop detectors. At the cost of increased complexity, the system can also offer k-anonymity guaratnees regardless of traffic density. We will describe this approach next.

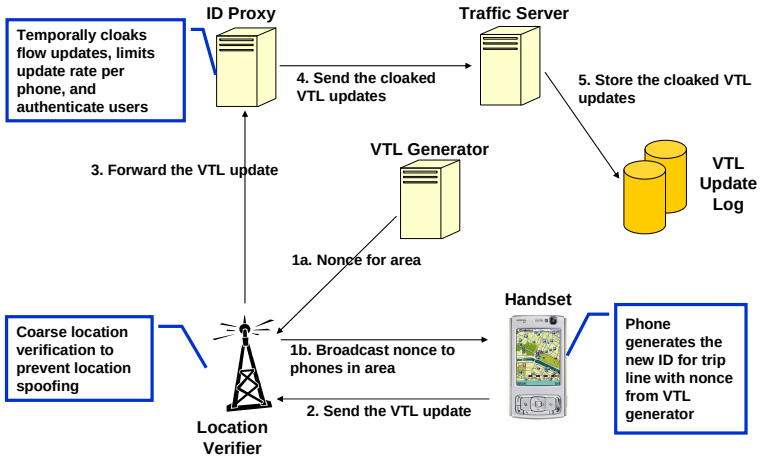


Figure 8.3: Distributed Architecture for VTL-based Temporal Cloaking.

8.2.3 Extensions for VTL-based Temporal Cloaking

We propose a distributed VTL-based temporal cloaking scheme that reduces timestamp accuracy to guarantee a degree of k-anonymity in the dataset accumulated at the VTL server. This provides a stronger privacy guarantee than probabilistic privacy, it prevents tracking or identification of individual phones based on anonymous position updates even if the density of phones is very low. The key challenge in applying temporal cloaking is concealing the location of probe vehicles from the cloaking entity. To calculate the time interval for nodes at the same location the cloaking entity typically needs access to the detailed records of each data subject [88, 51], which itself can raise privacy concerns.

Using virtual trip lines, however, it is possible to execute the cloaking function without access to precise location information. The cloaking entity can aggregate updates by trip line ID, without knowing the mapping of trip line IDs to locations. It renders each location update k-anonymous by replacing VTL timestamp with a time window during which at least k updates were generated from the same VTL (i.e., $k-1$

[1]While VTL positions are not highly sensitive, encryption reduces the possibility of timing analysis (see section 8.1).

other phones passed the VTL). In effect k VTL updates are aggregated into a new update $(vtlid, \frac{s_1 \ldots s_k}{k}, \max(t_1 \ldots t_k))$, where s_i denotes the speed reading of each VTL update i. Since now k-phones generate the same update, it becomes harder to track one individual phone. The cloaking function can be executed at the ID proxy, if handsets add a VTL ID to the update that can be accessed by the ID proxy.

Beyond the cloaking function at the ID Proxy, two further changes are needed in the architecture to prevent an adversary from obtaining the mapping of VTL IDs to actual VTL locations. The system uses two techniques to reduce privacy leakage in the event of phone database compromises. First, the road network is divided into tiles and phones can only obtain the trip line ID to location mapping for the area in which the phone is located. This assumes that the approximate position of a phone can be verified (for example, through the cellular network). Second, the VTL server periodically randomizes the VTL ID for each trip line and updates phone databases with the new VTL IDs for their respective location.

This leads to the extended distributed architecture depicted in Figure 8.3, where again no central entity has knowledge of all three types of information: location, timestamp, and identity information. As before, VTL updates from phones to the ID proxy are encrypted, so that network eavesdroppers do not learn position information. It first checks the authenticity of the message and limits the update rate per phone to prevent spoofing of updates. It then strips off the identification information and forwards the anonymous update to the VTL server. Knowing the mapping of VTL IDs to locations, the VTL server can calculate road segment travel times. This architecture differs, in that the ID proxy server cloaks anonymous updates with the same VTL ID before forwarding to the VTL server. It also requires a Location Verifier entity, which can coarsely verify phone location claims (e.g., in range of a cellular base station) and distribute the VTL ID updates to only the phones that are actually present within a specified tile. Table 8.1 summarizes the roles of each entity and how information is split across them.

The temporal cloaking approach can be vulnerable to spoofing attacks unless it is equipped with proper protection. For instance, malicious clients can send many updates

Entity	Role	Identity	Location	Time
Handset	Sensing	Yes	Accurate	Accurate
Location Verifier	Distributing VTL ID Updates	Yes	Coarse	Accurate
ID proxy	Anonymization and Cloaking	Yes	Not Available	Accurate
Traffic Server	Computing Traffic Congestion	No	Accurate	Cloaked

Table 8.1: Entity roles and splitting of sensitive information across entities

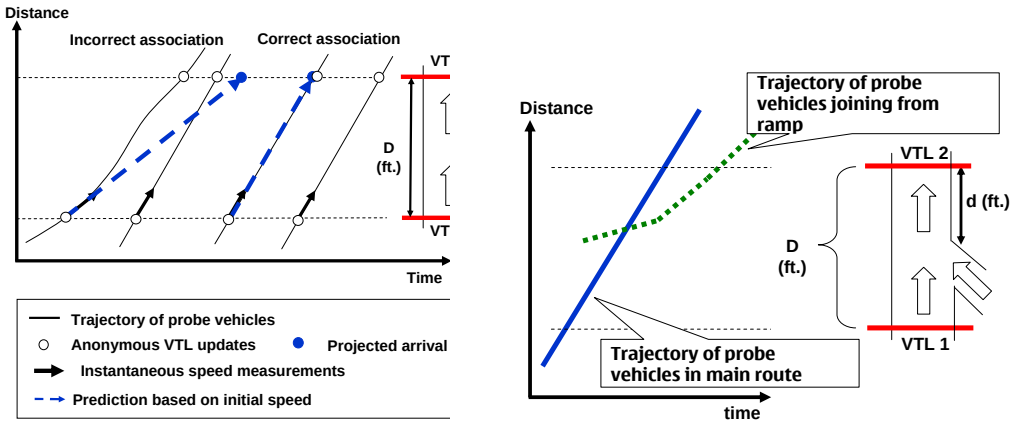
to shorten the cloaking time window. To prevent this denial of service attack, the ID proxy server limits the update rate per phone.

To reduce network bandwidth consumption of the periodic VTL updates, clients can independently update the VTL IDs based on a single nonce per geographic area (tile). The VTL server generates the nonces using a cryptographically secure pseudo-random number generator and distributes each nonce to the clients currently in the tile area. Both clients and server can then compute $VTLID_{new} = h(nonce, VTLID_{old})$, where h is a secure hash function such as SHA.

Discussion. Temporal cloaking fits well with the travel time estimation method used in the VTL system because the mean speed calculation does not depend on accurate timestamp information. To estimate the travel time, the VTL server calculates the mean speed for a trip line only based on the speed information in the flow updates. Typically, the travel time would be periodically recomputed. The use of temporal cloaking adaptively changes this update interval so that at least k phones have crossed the trip line. If k is chosen large, it reduces the update frequency. Even with temporal cloaking, however, the travel time algorithm would need speed reports from several vehicles to provide reliable estimates.

8.3 Trip Line Placement

This section describes placement algorithms that choose virtual trip line locations to maximize travel time accuracy and preserve privacy. A basic algorithm, the even placement approach, takes as input a partial road network graph. For each road segment, which refers to a stretch of road between two intersections or merges, the algorithm observes an exclusion zone at the beginning of the segment and then places equidistant



(a) Linking based on Prediction. (b) Linking around On-Ramp.

Figure 8.4: Linking attack scenarios on straight highway section and on-ramp section.

trip lines orthogonally to the road. Key to preserving privacy are the procedures for determining appropriate trip line spacing parameters and exclusion zone sizes, which we discuss in the following subsection. Privacy is also significantly improved by selecting only higher traffic roads for trip line placement, such as highways and aterials, which usually are less sensitive areas.

8.3.1 Placement Privacy Constraints

The algorithm considers two types of placement constraints, *exclusion areas* around sensitive locations and *minimum spacing* constraints to reduce tracking ability along straight roadways.

Determining Minimum Spacing. The minimum spacing constraint is particularly important on highways, where more regular traffic flows increase the tracking risks. Thus, we focus our derivations on straight highway scenarios. Minimum spacing for longer road segments is determined based on a tracking uncertainty threshold. Recall that to prevent linking compromises, an adversary should not be able to determine with high confidence that two anonymous VTL updates were generated by the same handset.

Tracking uncertainty defines the level of confusion that an adversary encounters when associating two successive anonymous flow updates to each other. We define tracking uncertainty as the entropy $H = -\sum p_i \log p_i$, where p_i denotes the probability

(from the adversary's perspective) that anonymous flow update i at the next trip line was generated by the same phone as a given anonymous flow update at a previous trip line. The probability p_i is calculated based on an empirically derived pdf model that takes into account the time difference between the predicted arrival time at the next trip line and the actual timestamp of flow update i. We fit an empirical pdf of time deviation with an exponential function, $\hat{p_i} = \frac{1}{\alpha}e^{-\frac{t_i}{\beta}}$, where we obtain the values of α and β by using unconstrained nonlinear minimization.

Consider the example scenario in Figure 8.4(a). In scenario (a) the adversary projects the arrival time at VTL 2 based on the phone's speed report at VTL 1. The projected arrival time is the endpoint of the dashed line (the solid lines indicate phones' actual paths). There are two actual flow updates at VTL 2 (indicated through points). The adversary would calculate the time difference between the projected arrival times, assign probabilities p_1 and p_2 using the pdf, and determine entropy. Compared to the second example in scenario (a) entropy is high indicating that an adversary cannot determine the correct VTL update with high confidence. In fact, the closest update would be incorrect in this scenario. Tracking uncertainty calculated through the entropy is maximized when there is more than one anonymous flow update with similar time-differences. In other words, lower values of H indicate more certainty or lower privacy. To reduce computational complexity, we only consider VTL updates within a time window w of the projected arrival time during the entropy calculation.

In general tracking uncertainty is dependent on the spacing between VTLs, the penetration rate, and speed variations of vehicles. If speed remains constant, as in the second example of figure 8.4(a), the projected arrival times match well and tracking uncertainty is low. Higher penetration rates lead to more VTL updates around the projected arrival time, which decreases certainty. As spacing increases, the likelihood that speeds and the order of vehicles remain unchanged decreases, leading to more uncertainty. Speed variations on highways are frequently caused by congestion—thus road segments with lower average speed tend to increase tracking uncertainty.

We empirically validate these observations through simulations using the PARAMICS vehicle traffic simulator [11]. Figure 8.5 depicts the minimum spacing required

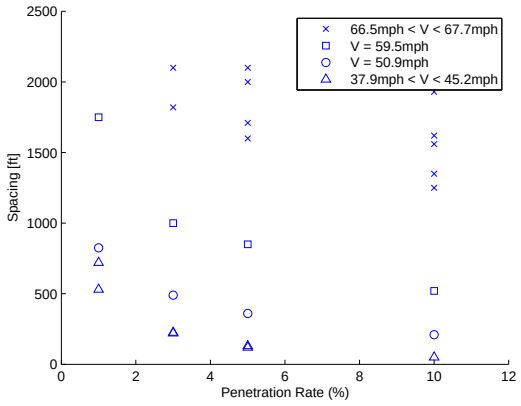


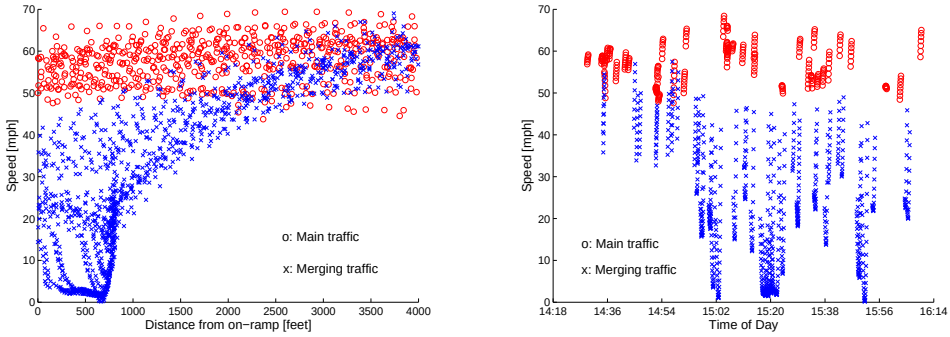
Figure 8.5: Minimum Spacing Constraints for Straight Highway Section.

to achieve a minimum mean tracking uncertainty of 0.2 for different penetration rates and different levels of congestion (or mean speed of traffic). We choose a reasonably low uncertainty threshold, which ensures to an adversary a longer tracking that could have privacy events such as two different places (e.g., origin and destination).[2] The uncertainty value of 0.2 corresponds to an obvious tracking case in which the most likely hypothesis has a likelihood of 0.97. The penetration rates used were 1%, 3%, 5% and 10%. To evaluate different levels of congestion, we used traces from seven 15 min time periods distributed over one day. We also used three different highway sections (between the junction of CA92 and the junction of Tennyson Rd., between the junction of Tennyson Rd. and the junction of Industrial Rd., and between the junction of Industrial Rd. and the junction of Alvarado-Niles Rd.) to reduce location-dependent effects. The simulations show that the needed minimum spacing decreases with slower average speed and higher penetration rate.

The clear dependency of the tracking uncertainty on the penetration rate and the average speed allows creating a model that provides the required minimum spacing for a given penetration rate and the average speed of the target road segment.

Determining Exclusion Areas. Additional tracking risks are present at ramps and many intersections because of large speed variations. Vehicles leaving the main direction of travel move slower and vehicles joining the main direction of travel from

[2]Two recent studies [72, 57] observe about 15 minutes as a median trip time.



(a) Speed Measurements over Distance.

(b) Speed Measurements over Time.

Figure 8.6: Exclusion Area Constraints for Highway On-ramp Section.

ramps accelerate from a very low speed while vehicles staying on the main direction maintain their speed. Figure 8.4(b) depicts the latter case. If trip lines are placed immediately before or after intersections, an adversary may be able to follow vehicles paths based on speed differences.

In a case study, we determine the required size of an exclusion area using the 20 vehicle dataset described in the next section (PARAMICS does not model ramp trajectories in sufficient detail). Figure 8.6(a) shows the difference in speed between the merging traffic (red circles) and the main traffic (blue crosses) with increasing distance from the on-ramp. As expected, we observe that near the on-ramp an adversary can distinguish main and merging traffic through a simple speed threshold (about 40 mph in our scenario). The graph also shows that speeds converge at a distance of about 3000 ft from the on-ramp, which can be used as an exclusion zone size.

The figure 8.6(b) depicts a weekday time interval between 1:30 PM to 4:00 PM, during which the mean speed of the test road segment decreases due to the increasing congestion, as reported by the PeMS highway measurement database [3]. As evident, the speed variation between main route traffic and merging traffic increases with congestion. This may require longer exclusion during congestion.

In addition to merges and intersections, where detailed information would be especially important for an adversary to track which path a vehicle takes, exclusion zones can also be places around other sensitive places. These may be places that could allow

sensitive inferences, such as a medical clinics, or locations where the driver of handsets may be identified (e.g., suburban home locations [72] or automatic toll booth plazas).

8.4 Implementation

We have completely implemented the probabilistic privacy architecture and use this implementation for the experiments in the following section. The implementation uses Nokia N95 smartphone handsets, which include a full Global Positioning System receiver that can be accessed by application software.

8.4.1 Map Tiles and Trip Lines

In our system, we recursively divide the geographic region of interest into four smaller rectangles (or quadrants), and the minimum quadrant size is 1m by 1m. We convert the GPS location of a user into a Mercator projection using the WSG84 world model. Mercator projects the world into a square planar surface. A *zoom* of 25 is assumed to be the maximum precision that location can be specified in. By default every GPS location is converted into 25 bit x and y values with *zoom* set to 25. By using the quadrant representation the mobile device can efficiently control the granularity by simply changing the *zoom* level. In this encoding, the world is treated as a square grid of four quadrants with *zoom* level 2, where x and y are the offsets from the top left corner of the world.

This reprentation makes it easy to specify the specific map tile. We define a map tile as a container that groups all trip lines within it. When a client wants to download all virtual trip lines within the San Francisco Bay Area, it sends the VTL server the triplet, $(zoom, x, y)$ for the corresponding region. In our implementation, we choose 12 as the default zoom level, which corresponds to an 8 km by 8 km square.

This representation also helps in reducing storage size and bandwidth consumption. Since the general area is identified by the quadrant, we only store the 13 least significant bits of the trip line end point coordinates instead of the full 25 bits used for typical UTM coordinates. This decreases storage consumption to 68bits (15 bit id, 1 bit

direction, $4 \cdot 13$ bits coordinates) per trip line. As an example of required storage and bandwidth consumption, consider the section of the San Francisco Bay Area shown in figure 8.7. The total road network in the white titles shown in the left figure contains about 20,000 road segments, according to the Digital Line Graph 1:24K scale maps of the San Francisco Bay Area Regional Database (BARD [1], managed by USGS). Assuming that the system on average places one trip line per segment this results in 166KB of storage.

8.4.2 Client Device and Software

We implemented the client software using J2ME (Java Platform, Micro Edition) on an Nokia N95 handset. This Symbian OS handset uses an ARM11-based Texas Instruments OMAP2420 processor running at 330MHz, and it contains 64MB RAM and 160MB internal memory. Its storage can be expanded up to 8GB with flash memory. We use the JSR 179 library (Location API for J2ME) [2] for communicating with the internal TI GPS5300 NaviLink 4.0 single-chip GPS/A-GPS module to set the sampling period and retrieve the position readings. This setup did not provide speed information. Instead, we calculate the mean speed using two successive location readings (in our implementation, every 3 seconds). The client software registers the task for checking the traversal of trip lines as an event handler for GPS module location updates, which is automatically invoked whenever a new position reading becomes available.

The communication between the handset and the ID proxy server, to send VTL updates or to request VTL downloads, is implemented via HTTPS GET/POST messages. The client software encrypts the message content but not the handset identification information using the public key of the VTL server so that only the VTL server with the corresponding private key can decrypt the message. To save network bandwidth and to reduce delay, we cache the downloaded trip lines for the nine map tiles closest to the current position in local memory. When a vehicle crosses a tile boundary, it initiates VTL download background threads for the missing tiles.

8.4.3 Servers and Databases

VTL Server. At the bottom of the hierarchy of our server implementation is a backend database server. The database server contains two databases. First is a VTL database which holds GPS coordinates of all trip lines. In future we plan to enhance our trip line database to hold meta data associated with that trip line. For instance, the meta data for a trip line can contain the posted speed limit at that trip line which can be used by the client application to decide if it is going over the speed limit in which case the client application can disable VTL updates. Write access to this database is restricted only to traffic administrators who can add, delete or update a VTL.

Figure 8.7: Road networks extracted from Bay Area DLG files (Left) and Trip Lines per road segment in Palo Alto CA (Right).

The second database is the VTL update database. This database stores the VTL updates sent by the mobile device whenever the mobile device chooses to send an update after crossing a VTL. The update database simply appends every VTL update along with a time stamp on when the update was received. To sanitize bogus VTL updates from the clients, the VTL update database also keeps both the encrypted and decrypted versions of the VTL update for further investigation in collaboration with the ID proxy server. When bogus VTL updates are detected in the VTL update database, their encrypted versions are compared to the encrypted version stored in the ID proxy server to blacklist the originator of bogus VTL updates.

We use Microsoft SQL to implement the databases, and we develop the VTL server

using J2EE (Java Platform, Enterprise Edition) and JDBC (Java Database Connectivity) to control the SQL databases that are connected to the VTL server. While we have used only a single DB server in this prototype, the two databases should ideally be implemented by different entities to prevent active trip line modification attacks by a compromised traffic monitoring entity.

ID Proxy Server. On top of the database server is the ID Proxy server. The identification proxy server is envisioned to be operated by an entity that is independent of the traffic service provider. We implement the ID proxy server as a servlet-based web server that takes in HTTPS GET/POST messages from clients and forwards messages to the VTL server. The HTTP message received by the proxy server from the client has two components. The first component contains the mobile device identification information, namely phone number of the message origin. This component of the message is required for all cell phone communications as operator needs to appropriately charge for data communication costs. The second component of the message contains information that is intended for the database server. The proxy server strips all the identification information from the message, namely the first component of the message, and passes on the second component of the message to the application server. We implemented the secure channel between ID proxy server and the VTL server using WSDL (Web Service Definition Language)-RPC (Remote Procedure Call) over J2EE Server.

8.5 Experimental Evaluation

We evaluate our system first in terms of travel time estimation accuracy and then analyze privacy-accuracy tradeoffs.

8.5.1 Traffic Flow Estimation Accuracy

GPS Speed Accuracy. A first experiment was run to estimate the position and speed accuracy of a single cell phone carried onboard a vehicle. The experiment route consisted of a single 7 mile loop on I-80 near Berkeley, CA, and VTLs were placed evenly

on the highway every 0.2 miles. Speed and position measurements were stored locally on the phone every 3 seconds, and speed measurements were sent over the wireless access provider's data network every time a VTL was crossed. The speed measurements were computed using two consecutive position measurements. In order to validate this calculation, the vehicle speed was also recorded directly from the speedometer on a laptop with a clock synchronized with the N95. In Figure 8.8, the speed measured directly from the vehicle speedometer is compared to the speeds measured by the VTLs and the speed stored in the phone log. Timestamp of each record denotes the elapsed time since midnight of the experiment day. On average, the vehicle odometer reported a speed 3 mph slower than the GPS. The position data was accurate enough to correctly place the vehicle on either the correct or neighboring lane of travel.

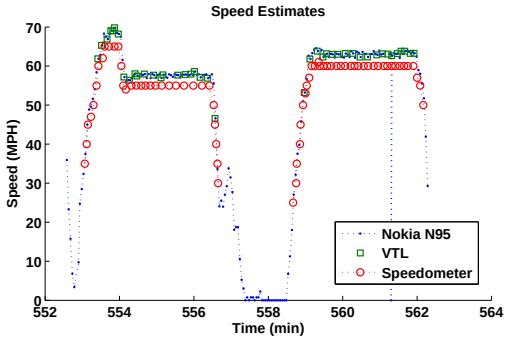


Figure 8.8: Comparison of the speed measurements recorded from the N95 (dots), the VTLs (boxes) and the vehicle speedometer (circles) as a function of time.

Figure 8.9: Satellite image of the first experiment site I-80 near Berkeley, CA. The red lines represent the locations of the VTLs, the blue squares show the speed recorded by the VTL, the green squares represent the position and speed stored in the phone log. The brown circles represent the readings from the vehicle speedometer.

To further validate the position accuracy of GPS enabled cell phones, the vehicle was driven on a frontage road along the highway, which poses a very important problem for cell phone based traffic monitoring. Frontage roads typically have slow moving traffic with speed limits of 25 or 35 mph and run alongside the freeway. Without high precision position accuracy, this traffic can be incorrectly identified as freeway traffic. In our study, the VTLs were only placed on the freeway, and they did not detect the vehicle as it traveled on the frontage road, despite the freeway an frontage road being separated in some locations by as little as 30 ft. Although the test was only conducted using a single phone, it presented promise that the technology can be used for advanced traffic monitoring applications.

Experimental Setup. A second experiment was conducted to demonstrate the feasibility of cell phone based travel time estimation in practice. Note that the accuracy experiments do not use placement constraints or temporal cloaking. We consider temporal cloaking separately in section 8.5.3. For two hours, twenty vehicles were driven back and forth on a 4 mile section of I-880 south of Oakland CA as shown in figure 8.10. The length (i.e., 4 miles) of test road segment was chosen to have 1% to 2% penetration rate given 20 participants and approximate round travel time. In order to observe a more natural mixing phenomena (in which vehicles pass each other) half of the drivers were instructed to drive a slightly shorter, 3 mile section of the highway (red circle) after the completion of the first lap. The location of this experiment was specifically selected because it featured both free flowing traffic at greater than 50 mph, and congested, stop and go traffic. An accident just north of the experiment site further added to the complexity of the northbound traffic flow. As observed by the drivers of the experiment, this accident created "shear" in the traffic flow, where vehicles in adjacent lanes of traffic were traveling at significantly different speeds.

We expect that actual users will place their phones into a dashboard car holder (depicted in the left of fig 8.11) to be able to view navigation and travel time information while driving. Since we did not have sufficient car holders available during the experiment, we placed the cell phones onto the dashboard as shown in the right of figure 8.11. For this experiment, 45 VTLs were first evenly placed to record the speed

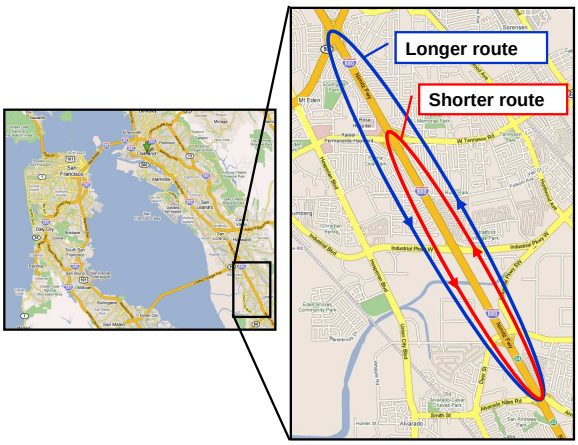


Figure 8.10: I880 Highway Segment for 20 Car Experiment.

Figure 8.11: Experimental Setup in a Car for 20 Car Experiment.

measurements from the 20 vehicles. Each phone also stored speed and position measurements to a local log every three seconds, following the same protocol as the first experiment. To estimate the travel time, the *instantaneous travel time* was computed, which assumes traffic conditions remain unchanged on every link[3] from the time the vehicle enters the link until it leaves the link. Therefore, the travel time of the section can be computed by simply summing those of the constituent links at the time a vehicle enters the route. The travel time of each link is computed with the length of a link and the mean speed that is obtained by averaging out speed readings from probe vehicles during *an aggregation interval*. The aggregation interval can vary from 10s of seconds to few minutes, depending on traffic condition. Its effect on travel time estimation accuracy will be examined in section 8.5.3.

We then run the DP algorithm to compute the optimal VTL locations. Using *the instantaneous travel time method*, we plot actual travel times versus predicted travel

[3]Each VTL is placed in the middle of its respective link and the conditions on the entire link are given by the VTL reading.

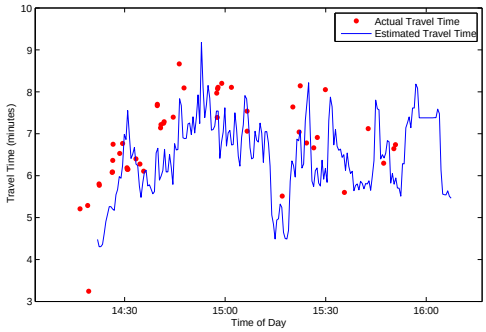


Figure 8.12: Actual travel times compared with an estimate given by the instantaneous method (30 second aggregation interval).

times in figure 8.12. We obtain ground truth for actual travel time by checking logging data of each probe vehicle. Since variates between actual travel times and predicted travel times are positive and negative, we calculate Root Mean Squared (RMS) error between two sets to see the expected magnitude of a travel time estimation error. For a given 30 second aggregation interval, we achieved a RMS error of about 80 seconds.

8.5.2 Privacy-Accuracy Tradeoffs

This section analyzes the travel time estimation accuracy and privacy preservation of our probabilistic approach, the VTL-based placement algorithm.

To analyze privacy, we measure the *distance-to-confusion* with two different sets of anonymous flow updates from both the evenly spaced VTLs (with exclusion area) and the evenly spaced VTLs (without exclusion area). We call the latter *spatial periodic sampling*. We use the repeated south bound trips of the 20 probe vehicles, which contain the effect of merging traffic from the shorter loop (see figure 8.13). The south bound direction also has lower traffic volume than the north bound direction, providing a more challenging environment to protect privcay. On the experiment day, we verified from the PeMS [3] highway measurement database that our test road segment (on south bound from Route 92 to Alvarado) experienced about 5000 vehs/h as a traffic volume and an average speed of 55 mph. Because we have 88 traces from 20 probe vehicles during our 100 min test period, the penetration rate is about 1% to 2%. Based on the

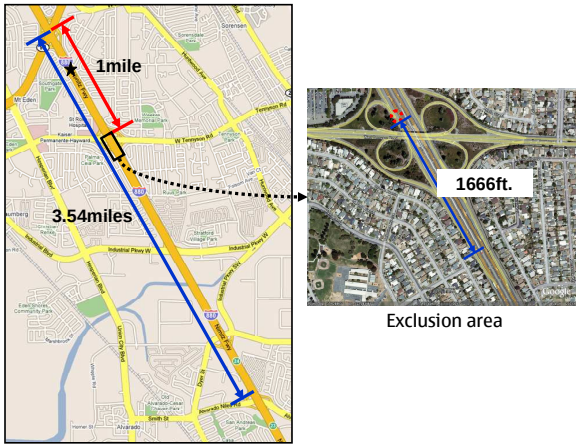
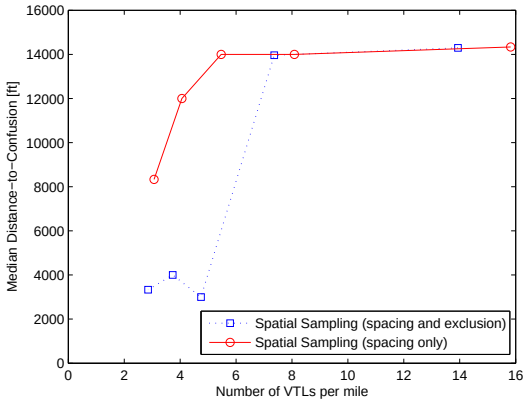


Figure 8.13: Exclusion Area on Test Road Segment. Tracking starts from the point marked by star.

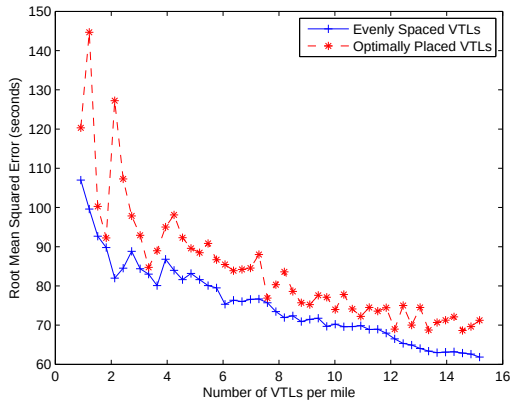
reported average speed and the penetration rate, we obtain the approximate value of the required minimum spacing (800 ft.) from the empirical result graph as shown in figure 8.5. At the on-ramp, we define a 1670 ft (500 meters) exclusion area. Given the fixed exclusion area, we generated different sets of equidistant trip lines with minimum spacing varying from 333 ft (100 meters) to 1670 ft (500 meters).

When we measure the *distance-to-confusion*, we use an uncertainty threshold of 0.2, meaning that tracking stops when it incorrectly links updates from different handsets, or when the uncertainty at any step rises above this threshold. We choose the probe vehicles of the main route as the test vectors. Among the set of anonymous updates that are reported at the same VTL, we measure the time deviation of each of them from the projected arrival time of the target probe vehicle, then we calculate the entropy using the empirically obtained probability distribution function of the time deviation between the projected arrival time and each timestamps of anonymous updates at the corresponding VTL. This empirical pdf was measured from the PARAMICS traffic dataset that have similar average speed and traffic volume. In linking anonymous flow updates that the spatial periodic sampling techniques generate, the adversary removes from the candidate set several anonymous flow updates that have speed measurements less than 40 mph within the 500 meter distance from the on-ramp, because an adversary has a knowledge on general trend around on-ramps as shown in figure 8.6(a). This

leads to better tracking performance by reducing the number of likely hypotheses.



(a) Spatial Sampling (exclusion and spacing vs. spacing only)



(b) Evenly spaced vs. Optimally placed

Figure 8.14: Comparison of privacy and travel time accuracy over different VTL spacings. Spatial sampling with exclusion zones better preserves location privacy.

The results are shown in figure 8.14(a) which plots the median *distance-to-confusion* against the total number of anonymous flow updates for each case. The dotted curve shows the VTL-based placement cases, 1666, 1333, 1000, 666, and 333 feet from the left to the right. The solid curve shows the spatial periodic sampling techniques for the same spacings. We observe that the dotted line drops at spacing of 1000 feet. As we expect from the figure 8.5, two successive anonymous updates that are sampled longer than 800 feet apart experience high tracking uncertainty. Another major reason for the drop in the curve is the existence of the exclusion area. The anonymous updates from the merging traffic can cause high uncertainty outside the exclusion area

since the speed measurements look similar to those from the main route traffic. Note that the periodic sampling in time behaves similarly as the spatial periodic sampling by increasing a time interval, but it cannot support the location awareness in sampling. Thus, the location-aware sampling via trip lines better preserves privacy than the periodic sampling in time. Also, the spatial sampling based on trip lines naturally perturbs position and timestamp information since the reported measurements are actually sampled when probe vehicles already pass the position of trip lines. This noisy measurement also can cause the reduction of distance to confusion in high penetration rate.

To study the traffic flow estimation accuracy tradeoff incurred by larger VTL spacings, figure 8.14(b) shows the root-mean-square error over the same range of VTL spacings. The travel time estimation generally improves with an increasing number of VTLs, both for evenly spaced and for optimally placed VTLs. In particular, one can see that the error from optimally placed VTLs from the DP algorithm is lower than the naive approach of evenly spaced VTLs. Because of the previously mentioned shearing present on the highway, the error of travel time estimation algorithms, and the variability of the drivers themselves, there will always be some error present in travel time which cannot be removed by increasing the number of VTLs alone. Note that this variability is present in figure 8.12, where we plot actual travel times versus predicted travel times using the instantaneous method earlier. The graph is not monotonically decreasing because travel time error is inherently variable and sometimes by adding a VTL we actually produce worse estimates if the additional VTL does not precisely capture the correct speed for the long link that it covers.

8.5.3 Guaranteed Privacy via VTL-based Temporal Cloaking

To compare the travel time accuracy of temporal cloaking with that of a baseline temporal periodic sampling techniques, we measure the RMS error between estimated travel times and the actual travel times that are collected from 20 probe vehicles over

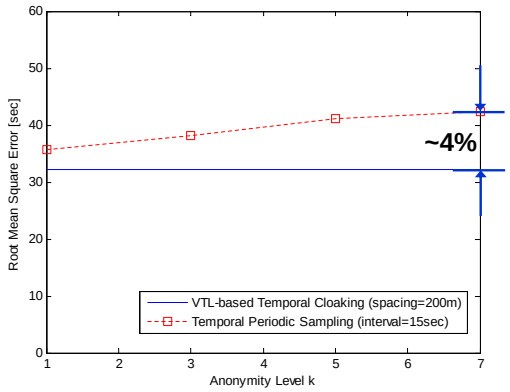


Figure 8.15: Travel Time Accuracy versus Anonymity k.

the shorter route (north bound) in figure 8.10. The mean of actual travel time over this shorter route is 265.13 seconds and its 95% confidence interval is $(254.9; 275.3)$.

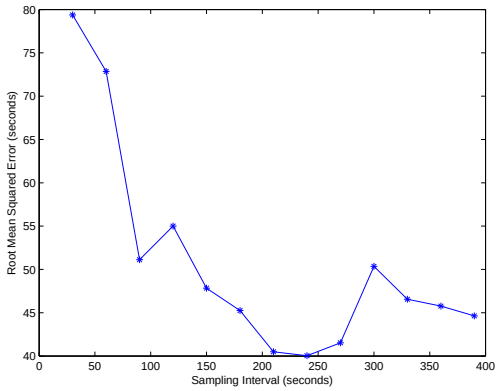


Figure 8.16: Travel time estimate errors by different aggregation intervals using 15 VTLs.

The travel time estimator divides the road into 200m intervals and separately calculates mean speed based on the periodic or VTL reports for each segment using an aggregation interval of a 200s window (parameters empirically chosen to provide good performance from figure 8.16). Our temporal cloaking method evenly place a trip line every 200 meters without an exclusion area. The sampling interval of 15 seconds for the periodic sampling technique is chosen so that both the VTL approach and periodic sampling produce the same number of updates. This allows us to compare both techniques based on similar input information and network overhead. Figure 8.15 shows that temporal cloaking reduces accuracy by only up to 4% for a k value of 7.

Larger k-anonymity parameters lead to longer aggregation intervals. Figure 8.16 illustrates the effect on the quality of the travel time estimates in terms of the length of the aggregation interval. During the aggregation interval, we sum anonymous flow updates for the corresponding trip lines to calculate the average speed of the link that is specified by the trip line. The general trend is that a longer sampling interval provides more accurate travel time estimates. However, the RMS error increases again after 250 seconds, indicating that the aggregation interval should be shorter than the changing period of traffic conditions.

8.6 Discussion

In addition to the privacy benefits, a key advantage of virtual trip lines over physical traffic sensors is the flexibility with which they can be deployed. For example, when roadwork is performed, VTLs can be deployed throughout the construction region, providing accurate travel time estimates in an area which often creates significant congestion. Because there is almost no additional cost to deploy the VTLs, and they do not interfere with the construction work or the highway traffic, they can be placed to adjust to the temporarily changed traffic patterns. One could even envision a VTL placement strategy which changes on a much shorter time period, with optimal placement strategies for the morning and evening rush hours, or holiday traffic patterns.

8.6.1 Security

This system significantly improves privacy protection over earlier proposals, by distributing the traffic monitoring functions among multiple entities, none of which have access to both location and identity records.

The system protects privacy against passive attacks under the assumption that only a single infrastructure component is compromised. One passive attack that remains an open problem for further study is timing analysis by network eavesdroppers or by the ID Proxy. Given knowledge of the exact trip line locations, which every handset could learn over time, and public travel time information on the road network an

adversary could estimate the time needed to travel between any two trip lines. The adversary could then attempt to match a sequence of observed VTL update message inter-arrival times to these trip line locations. One may expect that the natural variability of driving times provides some protection against this approach. Protection could be further strengthened against network eavesdroppers by inserting random message delays. Under the temporal cloaking scheme, however, the ID proxy also obtains trip line identifier information. If they are used for extended durations, an adversary may match them to actual VTL positions based on the sequence in which probe vehicles have pass them. This threat can be alleviated through frequent VTL ID updates. Quantifying these threats and choosing exact tile size and update frquency parameters to balance privacy and network overhead concerns remain open research problems.

The system also protects the privacy of most users against active attacks that compromise a single infrastructure component and a small fraction of handsets. It does not protect user privacy against injecting malware directly onto users' phones, which obtains GPS readings and transfers them to an external party. This challenge remains outside of the scope of this thesis, because this vulnerability is present on all networked and programmable GPS devices even without the use of a traffic monitoring system. Instead, the objective of the presented architecture is to limit the effect of such compromises on other phones. For the temporal cloaking approach, compromised phones result in two concerns. First, an adversary at the ID proxy can learn the current temporary trip line IDs. To limit the effectiveness of this attack, the architecture periodically changes trip lines and verifies the approximtate location of each phone so that a tile of trip line updates can only be sent to phones in the same location. Second, a handset could spoof trip line updates at a certain location to limit the effectivness of temporal cloaking. Our proposed architecture already eliminates updates from unauthorized phones and can easily limits the update rate per phone and verify that the approximate phone position matches the claimed update. This renders extended tracking of individual difficult because it would either require a large number of compromised phones spread around the area in which the individul moves, or set of compromised phones that move together with the indivdual. The system could also incorporate other sanity

checks and blacklist phones that deliver suspicious updates.

The same methods also offer protection against spoofing attacks that seek to reduce the accuracy of traffic monitoring data. The system does not offer full protection against any active attack on traffic monitoring accuracy, however. For example, a compromised ID proxy could drop messages to reduce accuracy. These challenges remain an open problem for further work.

8.6.2 Involvement of Cellular Networks Operators

While this work was based on cellular handsets, the question of how to improve location privacy within cellular networks themselves is out of scope of this work. The Phase II E911 requirements [6] mandate that cellular networks can locate subscriber phones within 150-300m 95% of the time, and AGPS solutions could achieve similar accuracy as the traffic monitoring system on open-sky roadways. In addition, the phones are identified through IMSI (International Mobile Subscriber Identity, in the GSM system) and operators typically know their owner's names and addresses. While precise phone location information is accessible, to our knowledge, it is not widely collected and stored by operators at this level of accuracy.

To slow the further proliferation of such data, this work has investigated how traffic monitoring services can be offered without access to sensitive location information. It was primarily motivated by third party organizations that currently do not yet have access to identity and location information and want to implement privacy-preserving traffic monitoring services. The solution, however, is general enough so that in actual implementations different levels of involvement of network operators are possible. One case may be four separate organizations implementing each one infrastructure component of the system with no involvement of the network operator.[4] Another extreme case would be a cellular network operator creating separate entities within the company to protect itself against dishonest insiders and accidental data breaches of their customers

[4]The only limitation is that for temporal cloaking one of the identities needs to be able to approximately (at the level of a tile size) verify client location claims. This verification could be provided by a network operator but other forms of verification are also plausible.

records. Clearly, the first would be more preferable from a privacy perspective, but in the end both lead to a significant improvement in privacy over a naive implementation.

8.7 Conclusions

This chapter described an automotive traffic monitoring system implemented on a GPS smartphone platform. The system uses the concept of virtual trip lines to govern when phones reveal a location update to the traffic monitoring infrastructure. It improves privacy, through a system design that separates identity- and location-related processing, so that no single entity has access to both location and identity information. Virtual trip lines can be easily omitted around particularly sensitive locations. Virtual trip lines also allow the application of temporal cloaking techniques to ensure k-anonymity properties of the stored dataset, without having access to the actual location records of phones. We demonstrate the feasibility of implementing this system on a smartphone platform and conduct a 20 vehicle experiment on a highway segment. Results show that even with this low number of probe vehicles, travel time estimates can be provided with less than 15% error, and the privacy techniques lead to less than 5% reduction in the accuracy of travel time estimates for k values less than 7.

Acknowledgment

The author would like to thank Ryan Herring, Dan Work, Juan-Carlos Herrera, and Alexandre Bayen at UC Berkeley for helping conduct the experiment in chapter 8 and produce the figure 8.8, figure 8.9, and figure 8.12 for the purposes of this study.

Chapter 9
Thesis Summary

He Loved Big Brother.

– The last sentence from *1984 by George Orwell*

In the thesis, we claim privacy risks in recently emerging applications, so called collaborative sensing that frequently monitor and log users' location information. Development of cheap localization techniques and statistical data mining techniques accelerates the proliferation of this kind of applications and the magnitude of privacy risks keeps increasing. One challenge is to maintain or balance privacy against security and quality requirements.

We identified two inference attacks that enables re-identification in anonymous location database. Based on them, we propose a novel privacy metric that estimates how long an adversary track an anonymous user, which in turn leads to re-identification with additional information such as a prior knowledge on the subject being tracked, map information, and so on. To prevent these risks, we develop two architecture solutions, one centralized architecture and one distributed architecture where no single entity can have a complete knowledge of user's identity and location for a given time. Lastly, we find our proposed architecture feasible through the use of GPS high-end phones.

The thesis can contribute to several research/industry communities. First, identified attack scenarios help develop privacy guidelines and privacy evaluation model. Second, our distributed/centralized architectures give system designers more flexible options for building collaborative sensing applications. Third, our novel privacy metric can be generally used for evaluating location privacy in mobile computing applications that require users' trajectories.

As future research directions, it would be valuable to develop privacy risk model that estimates the privacy risk on anonymous location database and gives a foundation for building privacy guidelines. The privacy risk model should relate user density, sampling interval, types of road network, and other major factors to the degree of privacy risk. The privacy risk model needs to accommodate possible threats in all communication layers (e.g., physical layer device identification, location tracking in networking or application layer, and device tracking based on any possible quasi-identifiers). Following the work, it would be also interesting to develop purely client-based solutions that depend on not infra-structure but privacy guidelines learned from privacy risk model. To provide 'privacy guarantee' in this client-based solution, one could devise peer-to-peer cooperation techniques. This approach faces an interesting challenge on trust management between peers.

Appendix A

Proof on the Conservative Approximation of Uncertainty Calculation

Theorem A. *Given n non-zero probabilities $p_0, p_1, \ldots p_n$, let $H(S_i)$ be the entropy calculated over the normalized probabilities of the $i \leq n$ most probable hypotheses. Then, $H(S_i) \leq H(S_n)$.*

Proof. Let us order the probabilities so that $p_1 \geq p_2 \geq p_3 \geq ... \geq p_n$. We then refer to the set which includes the normalized probabilities from the first to the ith one $\frac{p_1}{\sum_i p_i}, \ldots \frac{p_i}{\sum_i p_i}$ as S_i. The entropy of S_1 is 0, since the event is certain, and thus $S_1 \leq S_2$. More generally, we know from [33] that the following relation holds between $H(S_i)$ and $H(S_{i+1})$.

$$\alpha H(p_1, p_2, ..., p_i) + H(\alpha, 1-\alpha) = H(\alpha p_1, \alpha p_2, ..., \alpha p_i, 1-\alpha) \tag{A.1}$$

Since we ordered the probabilities (descending) and $(1-\alpha)$ is the $(i+1)$th probability in S_{i+1}, we also know that $(1-\alpha) \leq \frac{1}{i+1}$. Thus, $\frac{i}{i+1} \leq \alpha \leq 1$ holds, given that $\alpha \leq 1$ as a probability. In terms of $H(S_i)$ and $H(S_{i+1})$ equation A.1 can be rewritten as $\alpha H(S_i) + H(\alpha, 1-\alpha) = H(S_{i+1})$. Subtracting $H(S_i)$ from both sides yields equation A.2:

$$H(S_{i+1}) - H(S_i) = H(\alpha, 1-\alpha) - (1-\alpha)H(S_i) \tag{A.2}$$

We now show that this equation must be positive or zero to prove our theorem. If $\alpha = 1$ this obviously holds. Otherwise, the right side of the equation A.2 is minimized with the maximum value of $H(S_i)$, which is $\log i$ and is obtained with all equal probabilities. Thus, we now consider equation A.3.

$$H(\alpha, 1-\alpha) - (1-\alpha)H(S_i) \geq H(\alpha, 1-\alpha) - (1-\alpha)\log i \tag{A.3}$$

Since $f(\alpha) = H(\alpha, 1-\alpha) - (1-\alpha)\log i$ is a monotonically increasing function and $\alpha \geq \frac{i}{i+1}$, its minimum is obtained at $f(\frac{i}{i+1}) = (i+1)\{\log(i+1) - \log(i)\} \geq 0$. Therefore, $H(S_i) \leq H(S_{i+1})$ and by induction $H(S_i) \leq H(S_n)$ holds.

□

Appendix B
Multi Target Tracking

The tracking systems community knows the problem of linking location samples to probable users as the data association problem in multi-target tracking systems. Radar provides one typical application: the system must assign anonymous radar echos to a set of tracked targets. The key idea of such algorithms is to compare the positions of new location samples with the predicted positions of all known targets and choose an assignment that minimizes the error.

We chose Reid's multiple hypothesis tracking algorithm [79], which is based on Kalman filtering. This algorithm is one of the basic works in the field [27, p. 325]. Although, we do not currently use its capability to maintain multiple hypotheses, we have chosen it because we plan to experiment with this feature in future work.

Here, we will summarize our implementation of the algorithm. We refer the reader to the original work [79] for a more in depth discussion and the derivation of the equations. Additional information, also on the Kalman filter, can be found in [27]. The algorithm operates in three steps: First it predicts a new system state, then generates hypotheses for the assignment of new samples to targets and selects the most likely hypotheses, and finally it adjusts the system state with information from the new samples.

We simplified Reid's algorithm in a number of points. First, we do not consider random track initiation. Second, we assume all samples are taken at a fixed sample rate. Finally, as already mentioned, after every step only one hypothesis survives, which means that at each step likelihood is calculated under the assumption that the previous assignments were correct.

B.1 State Prediction

The filter predicts state according to a process model that is described by

$$x_k = Fx_{k-1} + w,$$

where x_k is the state vector of the process at step k, matrix F describes a linear prediction of the next state given the previous state, and w represents the process noise vector. A new observation vector z_k relates to the actual state through

$$z_k = Hx_k + v,$$

where matrix H converts a state vector into the measurement domain and v represents the measurement noise vector. The filter assumes that the process noise and the measurement noise are independent of each other and normally distributed with covariance matrices Q and R, respectively.

When tracking only one target, the Kalman filter defines the conditional probability density function of the state vector at time instant k as a multivariate normal distribution with mean $\bar{x}$ and covariance $\bar{P}$. At each time step, the filter predicts the new target position as

$$\bar{x}^{k+1} = F\hat{x}^k \qquad \text{and} \qquad \bar{P}^{k+1} = F\hat{P}^k F^T + Q^T, \tag{B.1}$$

where $\hat{x}$ and $\hat{P}$ are the estimates after the last sample was received.

For two-dimensional tracking applications with only slight changes in trajectory we can model the system as

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad x = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix},$$

where (p_x, p_y) represent a position and (v_x, v_y) a velocity vector. A larger process noise component captures the probability of changing directions or velocity.

B.2 Hypotheses Generation and Selection

The algorithm generates a set of hypotheses when new samples are received—one for each permutation of the sample set. A hypothesis represents a possible assignment of new samples to targets. It then calculates the likelihood for each hypothesis and selects the one with maximum likelihood.

The probability of hypothesis Ω_i at time k, given the set of measurements Z^k with cardinality M, is described by

$$P_i^k \equiv P(\Omega_i^k \mid Z^k) \approx \prod_{m=1}^{M} f(z_m) \tag{B.2}$$

where f is defined by the following equation (B.3). Based on the observation equation in the Kalman filter, the conditional probability density function of the observation vector z_k obeys a multivariate normal distribution

$$f(z^k \mid \bar{x}^k) = N(z^k - H\bar{x}^k, B), \tag{B.3}$$

where $B = H\bar{P}^k H^T + R$ and $N(x, P)$ denotes the normal distribution

$$N(x, P) = e^{-\frac{1}{2}x^T P^{-1} x} / \sqrt{(2\pi)^n \mid P \mid}.$$

Both x^k and P are calculated using the update equation at the prediction step. Equation (B.3) calculates how close a new observation lies to a predicted position; these values are then combined into the probability of each hypothesis.

After calculating the probability of each hypothesis, we choose the hypothesis j with the maximum probability and also calculate the log-likelihood ratio as follows.

$$\log \Lambda^k = \log \frac{P_i^k}{\sum_{i=1, i \neq j}^{I} P_i^k} \tag{B.4}$$

References

[1] http://bard.wr.usgs.gov/.

[2] http://jcp.org/en/jsr/detail?id=179/.

[3] http://pems.eecs.berkeley.edu/public/.

[4] http://tier.cs.berkeley.edu/wiki/home.

[5] http://www.cs.berkeley.edu/ honicky/nsmarts/.

[6] http://www.fcc.gov/bureaus/wireless/.

[7] http://www.privacyrights.org/ar/chrondatabreaches.htm.

[8] http://www.sensorplanet.org/.

[9] http://www.urban-atmospheres.net/.

[10] http://www.usatoday.com/tech/products/2008-02-23-1454383337.htm.

[11] Paramics v4.0 - microscopic traffic simulation system. `www.paramics-online.com`.

[12] TeleNav. http://www.telenav.net/, 2004.

[13] Inrix. http://www.inrix.com/, 2006.

[14] Intellione. http://www.intellione.com/, 2006.

[15] Participatory Urbanism. http://www.urban-atmospheres.net/ParticipatoryUrbanism/index.html , 2008.

[16] N. Adam, V. Janeja, and V. Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 576–583, 2004.

[17] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Symposium on Principles of Database Systems*, 2001.

[18] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[19] M. Allen, L. Girod, R. Newton, S. Madden, D. T. Blumstein, and D. Estrin. Voxnet: An interactive, rapidly-deployable acoustic monitoring platform. In *IPSN '08: Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 371–382, Washington, DC, USA, 2008. IEEE Computer Society.

[20] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Computing*, 7(5):275–286, 2003.

[21] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *17th International World Wide Web Conference*, pages 237–246, Beijing, China, 2008.

[22] M. BARBARO and T. Z. Jr. A face is exposed for aol searcher no. 4417749. `http://www.nytimes.com/2006/08/09/technology/09aol.html`.

[23] D. Beneventano, S. Bergamaschi, S. Lodi, and C. Sartori. Consistency checking in complex object database schemata with integrity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):576–598, 1998.

[24] A. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *IEEE PerSec*, 2004.

[25] C. Bettini, S. Mascetti, X. S. Wang, and S. Jajodia. Anonymity in location-based services: Towards a general framework. In *The 9th International Conference on Mobile Data Management (MDM'08)*, pages 69–76, 2007.

[26] C. Bettini, X. SeanWang, and S. Jajodia. Protecting privacy against location-based personal identification,. In *2nd VLDB Workshop SDM*, 2005.

[27] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[28] R. Cayford and T. Johnson. Operational parameters affecting the use of anonymous cell phone tracking for generating traffic information. *Transportation Research Board 82nd Annual Meeting*, 1(3):03–3865, 2003.

[29] R. Cayford and T. Johnson. Operational parameters affecting use of anonymous cell phone tracking for generating traffic information. *Institute of transportation studies for the 82th TRB Annual Meeting*, 1(3):03–3865, Jan 2003.

[30] D. Chaum. Untraceable electronic, mail return addresses, and digital pseudonyms. *Communications of the ACM*, 1981.

[31] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178, New York, NY, USA, 2006. ACM.

[32] A. Civilis and S. Pakalnis. Techniques for efficient road-network-based tracking of moving objects. *IEEE TKDE*, 17(5):698–712, 2005. Senior Member-Christian S. Jensen.

[33] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[34] L. Cranor, M. Langheinrich, M. Marchiori, and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification. W3C Recommendation, Apr. 2002.

[35] L. F. Cranor, P. Guduru, and M. Arjula. User interfaces for privacy agents. *ACM Trans. Comput.-Hum. Interact.*, 13(2):135–178, 2006.

[36] X. Dai, M. Ferman, and R. Roesser. A simulation evaluation of a real-time traffic information system using probe vehicles. In *Proceedings of the IEEE Intelligent Transportation Systems*, pages 475–480, 2003.

[37] J. Deng, R. Han, and S. Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *Proceedings of the IEEE/Create-Net SecureComm*, Athens, Greece, September 2005.

[38] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *2nd Workshop on Privacy Enhancing Technologies*, 2002.

[39] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, pages 303–320, 2004.

[40] J. Douceur. The sybil attack. `http://citeseer.ist.psu.edu/douceur02sybil.html`, 2002.

[41] 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.

[42] A. Escudero-Pascual, T. Holleboom, and S. Fischer-Hubner. Privacy of location data in mobile networks. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (Nordsec 2002)*, 2002.

[43] H. Federrath, A. Jerichow, and A. Pfitzmann. Mixes in mobile communication systems: Location management with privacy. In *Proceedings of the First International Workshop on Information Hiding*, pages 121–135, London, UK, 1996. Springer-Verlag.

[44] M. Ferman, D. Blumenfeld, and X. Dai. A simple analytical model of a probe-based traffic information system. In *Proceedings of the IEEE Intelligent Transportation Systems*, pages 263–268, 2003.

[45] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium*, pages 12–12, Berkeley, CA, USA, 2006. USENIX Association.

[46] A. Gal and V. Atluri. An authorization model for temporal data. In *Proceedings of the 7th ACM CCS*, pages 144–153, New York, NY, USA, 2000. ACM Press.

[47] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proceedings of the 25th IEEE ICDCS 2005*, pages 620–629, Washington, DC, USA, 2005.

[48] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: anonymizers are not necessary. In J. T.-L. Wang, editor, *SIGMOD Conference*, pages 121–132. ACM, 2008.

[49] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM (USA)*, 42(2):39–41, 1999.

[50] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[51] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the ACM MobiSys*, 2003.

[52] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless lan through disposable interface identifiers: a quantitative analysis. In *Proceedings of the 1st ACM WMASH*, pages 46–55. ACM Press, 2003.

[53] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *Proceedings of the Second International Conference on Security in Pervasive Computing*, 2005.

[54] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proceedings of IEEE/Create-Net SecureComm*, Athens, Greece, September 2005.

[55] B. Hoh and M. Gruteser. Computer ecology: Responding to mobile worms with location-based quarantine boundaries. In *International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks*, 2006.

[56] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing privacy preservation of anonymous location sampling techniques in traffic monitoring systems. In *Proceedings (Poster Session) of IEEE/Create-Net SecureComm 2006*, August 2006.

[57] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.

[58] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of ACM CCS 2007*, October 2007.

[59] Y.-C. Hu and H. J. Wang. Location privacy in wireless networks. In *Proceedings of the ACM SIGCOMM Asia Workshop 2005*, April 2005.

[60] 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 *4th ACM SenSys*, Boulder, CO, November 2006.

[61] T. Ishizaka, A. Fukuda, and S. Narupiti. Evaluation of probe vehicle system by using micro simulation model and cost analysis. *Journal of the Eastern Asia Society for Transportation Studies*, 6:2502–2514, 2005.

[62] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series. Prentice Hall, Englewood Cliffs, New Jersey, March 1988.

[63] T. Jiang, H. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th ACM MobiSys*, New York, NY, USA, 2007. ACM Press.

[64] N. JOHN SCHWARTZ. Cellphone tracking study shows we are creatures of habit. `http://www.nytimes.com/2008/06/05/science/05mobile.html?em`.

[65] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. Knowl. Data Eng.*, 19(12):1719–1733, 2007.

[66] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk. Enhancing source-location privacy in sensor network routing. In *Proceedings of the 25th IEEE ICDCS'05*, pages 599–608, Washington, DC, USA, 2005.

[67] J. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. In *Proceedings of WMASH*, pages 110–118, 2004.

[68] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz. AnonySense: Opportunistic and privacy-preserving context collection. In *The Sixth International Conference on Pervasive Computing (PERVASIVE), To appear*, May 2008.

[69] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. Random data perturbation techniques and privacy preserving data mining. In *IEEE ICDM*. IEEE Press, 2003.

[70] T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 211–225, Washington, DC, USA, 2005. IEEE Computer Society.

[71] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) (to appear)*, April 2008.

[72] J. Krumm. Inference attacks on location tracks. In *Proceedings of the Pervasive 2007*, May 2007.

[73] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *Ubicomp*, pages 243–260, 2006.

[74] 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*, pages 19–28, New York, NY, USA, 2006. ACM Press.

[75] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd VLDB'2006*, pages 763–774. VLDB Endowment, 2006.

[76] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset, Oct 2006.

[77] L. Pareschi, D. Riboni, and C. Bettini. Protecting users' anonymity in pervasive computing environments. *percom*, 0:11–19, 2008.

[78] M. A. Quddus, W. Y. Ochieng, L. Zhao, and R. B. Noland. A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7(3):157–167, 2003.

[79] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, Dec 1979.

[80] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1998.

[81] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki. Caravan: Providing location privacy for vanet. In *3rd workshop on Embedded Security in Cars (ESCAR2005)*, 2005.

[82] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *2nd Workshop on Privacy Enhancing Technologies*, 2002.

[83] H. Shin, V. Atluri, and J. Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *The 9th International Conference on Mobile Data Management (MDM'08)*, pages 73–80, Beijing, China, 2008.

[84] B. Smith, H. Zhang, M. Fontaine, and M. Green. Cell phone probes as an ATMS tool. Research Report UVACTS-15-5-79, June 2003.

[85] E. Snekkenes. Concepts for personal location privacy policies. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57, New York, NY, USA, 2001. ACM Press.

[86] A. Solanas and A. Martínez-Ballesté. A ttp-free protocol for location privacy in location-based services. *Comput. Commun.*, 31(6):1181–1191, 2008.

[87] I. I. C. staff. How much do you trust big brother? *IEEE Internet Computing*, 1(6):8–16, 1997.

[88] L. Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[89] K. P. Tang, P. Keyani, J. Fogarty, and J. I. Hong. Putting people in their place: an anonymous and privacy-sensitive approach to collecting sensed data in location-based applications. In *Proceedings of CHI '06*, pages 93–102, 2006.

[90] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *The 9th International Conference on Mobile Data Management (MDM'08)*, pages 65–72, Beijing, China, 2008.

[91] U. o. M. Transportation Studies Center. *Final Evaluation Report for the CAPITAL-ITS Operational Test and Demonstration Program*. Transportation Studies Center, University of Maryland, 1997.

[92] C. Troncoso, G. Danezis, E. Kosta, and B. Preneel. Pripayd: privacy friendly pay-as-you-drive insurance. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 99–107, New York, NY, USA, 2007. ACM.

[93] F. L. Wong, M. Lin, S. Nagaraja, I. Wassell, and F. Stajano. Evaluation framework of location privacy of wireless mobile systems with arbitrary beam pattern. In *CNSR '07: Proceedings of the Fifth Annual Conference on Communication Networks and Services Research*, pages 157–165, Washington, DC, USA, 2007. IEEE Computer Society.

[94] J. M. Wozencraft and I. M. Jacobs. *Principles of Communications Engineering*. John Wiley & Sons Inc, 1966.

[95] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *Proceedings of the 27th IEEE Infocom 2008*, pages 547–555, Phoenix, Arizona, USA, 2008.

[96] A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164, 1982.

[97] Y. Yim and R. Cayford. Investigation of vehicles as probes using global positioning system and cellular phone tracking: field operational test. California PATH Working Paper UCB-ITS-PWP-2001-9, Institute of Transportation Studies, University of California, Berkeley, 2001.

[98] J. Yoon, B. Noble, and M. Liu. Surface street traffic estimation. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 220–232, New York, NY, USA, 2007. ACM.

[99] M. Youssef, V. Atluri, and N. R. Adam. Preserving mobile customer privacy: an access control system for moving objects and customer profiles. In *Proceedings of the 6th MDM '05*, pages 67–76, New York, NY, USA, 2005. ACM Press.

Vita

Baik Hoh

1995.2 Graduated from Daejon Science High School, Daejon, Korea

1995.3-1999.2 B.S. in Electrical and Computer Engineering, KAIST, Daejon, Korea

1999.3-2001.2 M.S. in Electrical and Computer Engineering, KAIST, Daejon, Korea

2001.3-2003.7 Research Engineer, Turbotek Inc., Bundang, Korea

2003.8-2008.8 Graduate Assistant, Electrical and Computer Engineering, Rutgers University, Piscataway, NJ

2007.5-2008.1 Research Intern, Nokia Research Center, Palo Alto, CA