Mert D. Pesé*, Xiaoying Pu, and Kang G. Shin

SPy: Car Steering Reveals Your Trip Route!

Abstract: Vehicular data-collection platforms as part of Original Equipment Manufacturers' (OEMs') connected telematics services are on the rise in order to provide diverse connected services to the users. They also allow the collected data to be shared with third-parties upon users' permission. Under the current suggested permission model, we find these platforms leaking users' location information without explicitly obtaining users' permission. We analyze the accuracy of inferring a vehicle's location from seemingly benign steering wheel angle (SWA) traces, and show its impact on the driver's location privacy. By collecting and processing real-life SWA traces, we can infer the users' exact traveled routes with up to 71% accuracy, which is much higher than the state-of-the-art.

Keywords: Automotive Privacy, Location Privacy

DOI 10.2478/popets-2020-0022
Received 2019-08-31; revised 2019-12-15; accepted 2019-12-16.

1 Introduction

Modern vehicles are generating a massive amount of data and becoming more connected and autonomous. A recent CNN article [1] argued that data collected from autonomous and connected cars are becoming very valuable. By capitalizing on car data collected from everyday driving, vehicle OEMs wish to provide better service to their customers and create new businesses. A Q&A article from Associated Press [2] predicts cars equipped with data-collection platforms to rise up to 75% by 2020, while this figure was around 20% in 2016. For instance, BMW launched a new platform called `CarData` [3] in May 2017 which allows their approximately 8.5 million `ConnectedDrive` users to collect certain telematics data. This data is stored on BMW servers and processed by `ConnectedDrive` services. Upon explicit users' permission, BMW also allows the collected data to be shared with third-party service providers through a business-to-business (B2B) interface. Third-parties can submit apps to the OEM app store which are published after passing the OEM's review. The third-parties must explicitly request certain telematics data, such as the odometer reading or vehicle speed, which is then securely collected from the in-vehicle network and uploaded to the service providers' server which can then process the data in any way they desire. A concrete use-case will be presented later as part of the threat model.

The goal of this paper is to uncover a new threat model brought by the new vehicular data-collection platforms and address the associated privacy issues. In particular, we will show how an installed malicious third-party app from the OEM's app store can infer vehicle locations or a trip route merely using the steering wheel angle (SWA) traces. This SWA data is processed to derive the road curvatures of the traveled route and then matched against the curvatures of roads in a certain area which can be obtained from publicly available online map APIs.

This attack becomes feasible due to vulnerabilities in the design of data-collection platforms, especially the coarse-grained permission model which is partly caused by insufficient and lax privacy regulations for vehicular data-collection platforms, as well as users' comfort in sharing vehicular sensor data as shown in our survey in Sec. 3. An attacker can pose as a malicious third-party and submit an app to the app stores of car data-collection platforms with the goal of acquiring the vehicle's location at any point during the trip. Since privacy concerns have already been raised for data collection from vehicles [4], the Alliance of Automobile Manufacturers (AAMs) — consisting of nearly all North American OEMs — recently developed some privacy guidelines [5]. These guidelines require explicit user consent for collection of sensitive data, such as geolocation. As a result, the attacker will provide an app with seemingly harmless sensor permissions, such as SWA which users are more likely to grant permission to than GPS according to our sensitivity survey results.

Another reason for choosing SWA is the submission process with the OEM which may deny publication of the app if it is obviously over-privileged, i.e., if too many (sensitive) permissions are requested for the stated use-case of the app. BMW, for instance, states that an app submission from a third-party will be reviewed before publication in the app store. Furthermore, according to

***Corresponding Author: Mert D. Pesé:** The University of Michigan - Ann Arbor, E-mail: mpese@umich.edu
Xiaoying Pu: The University of Michigan - Ann Arbor, E-mail: xpu@umich.edu
Kang G. Shin: The University of Michigan - Ann Arbor, E-mail: kgshin@umich.edu

Table 1. Comparison of SPy with existing approaches

	Na16 [6]	**Mi15 [7]**	**Zh17 [8]**	**Ga14 [9]**	**De13 [10]**	**SPy**
Data Source	Phone IMU Sensors	Phone Power Consumption	Speed from OBD-II Device	Speed from OBD-II Device	Speed from GPS Tracking Unit	Vehicular Data Collection Platform
Reference Source	Maps	Prerecorded Power Profiles for Each Phone	Maps	Maps	Time Stamp + Speed + Distance Traveled	Maps
Pre-processing	**Easy**	Hard	**Easy**	**Easy**	Medium	**Easy**
#Apps in App Market	Android: 3.5M (12/17) iOS: 2.2M (01/17)	Android: 3.5M iOS: 2.2M	N/A	N/A	N/A	**BMW: 90 (Jan '18)**
Matching Method	Turn Angle Similarity + Curve Similarity + Travel Time Similarity	HMM	HMM, DFS	Elastic Pathing	DFS	Road Curvature Matching (RoCuMa)
No Starting Point Assumption	✓	✗	✗	✗	✗	✓
Accuracy of Estimating Entire Road	13-38%	45% (of full route)	70% in Top 30 Candidate Routes	14% (less than 250m error)	37%	**71%**

the aforementioned privacy regulations for vehicles, the use of SWA does not even require explicit permission at installation time of the app and would thus be available in any installed app without user consent.

Location privacy is a trending and well-researched area. It has been shown that location data can be used to accurately identify users by using a combination of their home and workplace which are the easiest identifiable locations in a GPS trace [11]. There are several related studies of inferring location on smartphones from sensor data [6, 7]. Their threat models exploit design vulnerabilities in mobile operating systems' permission model (i.e., using zero-permission data for location inference) whereas we uncover a new threat model in an unexplored but rapidly growing domain (i.e., connected/autonomous cars). Our work differs in multiple ways from mobile location privacy studies as shown in Table 1.

Specifically, we propose SPy (**S**teering **P**rivac**y**) that constitutes several important contributions:

- Discovery of a new threat model based on malicious third-party apps running on vehicular telematic platforms. A malicious third-party app from the OEM app store can be used to perform an *offline* attack to infer user location by collecting SWA traces which are available without explicit user consent. This is due to the weak design of current permission models as discussed in Sec. 4 and lack of awareness/sensitivity of drivers towards the collection of vehicular sensor data as shown in a survey in Sec. 3.
- Development of novel **Ro**ad **Cu**rvature **Ma**tching (RoCuMa). It takes the road curvature derived from the victim's SWA trace and matches it against a ground truth database of road curvatures obtained from publicly available maps if area is roughly known.
- Achieving much higher accuracy in inferring a victim's location trace than state-of-the-arts. Note that this accuracy only holds for areas with moderately curvy roads. We will discuss this limitation of RoCuMa in Sec. 7.1.

Table 2. Frequently-used input streams for data collection (adapted from [12])

- Oil level
- Avg. speed
- Throttle position
- Start/end odometer
- Distance
- Odometer
- Fuel consumption
- RPM
- Trip number
- Injector circuit malfunction
- VIN
- Battery level
- Max. speed
- Event threshold
- Outside temperature
- Tire pressure
- Check engine light on
- Timestamp
- Car weight
- Latitude
- Hard braking
- Steering wheel angle
- ABS
- Windshield/ wiper information
- Longitude
- Make/model/year
- Airbag status
- Accelerometer
- Ignition status
- DTC code detection
- Speed
- Fuel level
- Seat belt status
- Positional quality
- Volumetric sensors

2 Background

2.1 Data Collection from Vehicles

We briefly introduce how vehicular data is collected. Vehicular sensor data is collected from a set of Electronic Control Units (ECUs) of the vehicle. ECUs are usually interconnected with each other by an on-board communication bus, or in-vehicle network (IVN), with the Controller Area Network (CAN) being the most popular in current vehicles. CAN data can be collected from ECUs using aftermarket dongles which are plugged into the OBD-II port, such as ELM327 or OpenXC [13, 14].

In recent years, wireless connectivity in vehicles has been gaining popularity. According to [15], 250M vehicles will be connected to the Internet of Things (IoTs) by 2020. Existing connected vehicles' (CVs') functionalities comprise infotainment, safety, diagnostics efficiency, navigation and payments [16]. In the next phase of CVs — which is starting now — cars will connect to third-party services using a built-in data connection, introducing novel vehicular data-collection platforms, such as BMW CarData [3, 17]. Already 78M vehicles are connected to the web, with 98% of all new vehicles sold in the US and Europe are expected to have cellular connections by 2021 [18].

A recent Frost&Sullivan study [19] gives an overview of frequently-used input streams for vehicular data-collection systems as summarized in Table 2.

2.2 Privacy Regulations

The rise of car data-collection platforms also raises privacy concerns; the US Government Accountability Office (GAO) released a document in July 2017 addressing this issue [4]. 13 of 16 selected automakers in the GAO's review reported collecting, using and sharing data from connected vehicles such as geolocation. Furthermore, the Alliance of Automobile Manufacturers (AAM) recently developed the *Consumer Privacy Protection Principles: Privacy Principles for Vehicle Technologies and Services* ("Consumer Privacy Protection Principles") [5] which went into effect January 2, 2016. Among other principles, these voluntary guidelines encourage affirmative consent for sensitive data called "covered information" which are listed as:

- **Driver Behavior:** Information about how a person drives a vehicle. Concrete examples stated in that document are vehicle speed, seat belt use, and information about braking habits. Information used only for safety, diagnostics, warranty, maintenance, or compliance purposes are explicitly ruled out.
- **Geolocation:** Information about the precise geographic location of a vehicle.
- **Biometrics:** Information about an owner's or registered user's physical or biological characteristics that can be used to identify the person.

As a result, the aforementioned platforms for vehicular data collection have to ask users for their explicit consent before requesting this sort of data from them. These voluntary guidelines are a good starting point to address the issues with the permission models of data-collection platforms, but are currently not binding.

3 Sensitivity Survey

Since vehicular data-collection platforms are proliferating and a vast variety of sensor data can be collected, we are interested in how sensitive consumers are to this new technology. More specifically, we would like to study how comfortable users would be in sharing their vehicle data with their OEM or any third-party apps. To this end, we conducted a survey ($N = 100$) on Amazon Mechanical Turk (AMT). We obtained an approval from our university's IRB (Registration No. IRB00000245) to conduct this survey.

All participants — qualified for the Masters requirement — were paid \$1 per survey and were given a maximum of 30 minutes to complete the survey. The mean and median times to complete this survey were 7 minutes 54 seconds and 5 minutes 2 seconds, respectively. For further information about the time distribution, see Fig. 10. Accounting for the average completion time, our \$1 payments to the participants were above the federal minimum wage in the U.S. We included demographics questions in the survey, including gender, their country of residence, whether they work in a highly technical field and are familiar with car telematics. 61% of the participants were male (39% female) and 85% came from the United States (13% from India, 2% from Canada). Only 26% of the respondents self-identified as tech-savvy, but 39% were familiar with car telematics.

We chose 20 of the most frequently collected sensors (see Table 2) and asked the participants two questions (**Q1** and **Q2**). For better understanding of these 20 sensors by the readers of this paper, we included their descriptions and potential privacy implications in Appendix A. We introduced a five-point Likert scale for these 20 sensors ranging from 1 (*strongly disagree*) to 5 (*strongly agree*), as well as an option *Not familiar*, and asked them the following two questions:

Q1. How comfortable would you be of sharing the following data types with an **OEM**? If you are not familiar with a term/sensor, please choose "Not familiar".

Q2. How comfortable would you be with sharing the following data types with a **third-party application provider**? If you are not familiar with a term/sensor, please choose "Not familiar".

To analyze the survey results, we checked model assumptions and used a mixed-effect linear model with a fixed effect on data source (OEM vs. third-party) and random effects on individual participants and car sensor variables [20], as shown in Eq. 1. By using a random effect on the sensor variable, we can obtain an *estimate*, with 95% confidence interval (CI), of the average response for each sensor while pooling all estimates towards the global average response for regularization. By adopting this estimation approach, we avoid using multiple comparison correction on a large number of variables (20 sensors) if we had employed null hypothesis statistical testing [21]. The Likert scale responses are treated as numeric values from 1 to 5.

$$response \sim datasource + (1|sensor) + (1|participant) \quad (1)$$

The average response for sharing data with OEM was 3.63, 95% CI = $[3.33, 3.93]$, between "Neutral"(3) and "Comfortable"(4). The average response for third-party sharing was 3.12, 95% CI = $[3.06, 3.18]$. Participants might be more comfortable with sharing their car sensor data with OEMs than sharing with third-parties. The result for individual sensors, relative to the average for all responses, are shown in Fig. 1. The participants were least willing to share their GPS sensor data, while they are most comfortable with sharing the outside temperatures of their cars. Location-related parameters such as *GPS Location*, *Maximum Speed*, and *Current Speed* were the most sensitive, their response well below the average on the comfort scale. *Steering Wheel Angle* is observed to have a middling response with its CI overlapping with 0 (the average), meaning that participants were not particularly uncomfortable with sharing this sensor.

Overall, participants are uncomfortable with sharing their GPS locations, while they did not suspect *Steering Wheel Angle* for any harmful purpose such as possible location inference as stated in its description (see Appendix A). We note that speed information, which has been leveraged for location inference in related studies [8–10], was already a sensitive parameter (*Current Speed*, *Average Speed* and *Maximum Speed*). On the other hand, using *Steering Wheel Angle* readings to infer location is a more likely attack than previously conceived, and shows the feasibility of such an attack within our threat model.

4 Threat Model

In Sec. 2, we discussed existing privacy regulations in the automotive domain which are relatively lax due to their non-binding nature. In Sec. 3, we presented a user survey showing that participants were not very sensitive about sharing their *steering wheel angle* (SWA) readings with other parties. In order to fully understand the threat model and feasibility of the proposed privacy attack, one has to understand the permission model of a vehicular data-collection platform. In the following, we present a concrete example derived from BMW CarData [3].

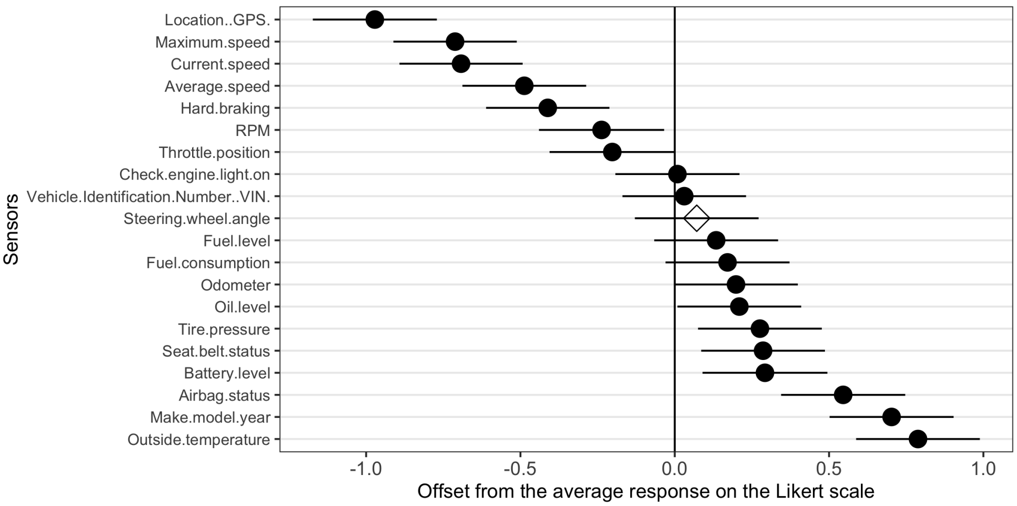


Fig. 1. Estimation of response for individual sensor reading from both OEM and third-parties (details in Equation 1). Error bars are 95% confidence intervals. The steering wheel angle is marked with a diamond.

The driver of a vehicle equipped with aforementioned data-collection platform wants to install a third-party app which makes use of the SWA sensor. The app has to be submitted to the OEM's app store by a third-party entity which passes an initial review by the OEM. The OEM has the right to revoke data access to the third-party whenever it detects a violation of its terms of service, i.e., the third-party using the data for other purposes than intended/stated [22]. Since the OEM does not have any influence on the data at the third-party [22], it is essential for the OEM to catch third-parties with malicious intents at the review stage, albeit it could only perform a superficial review due to lack of access to the source code which is a proprietary IP of the third-party entity. As a result, the chance of getting the app published in the OEM's app store would be high if the app is not "obviously" over-privileged (for instance, using GPS permission for a fuel-consumption tracking app) or requesting permission for sensors which the app is unrelated to.

An example workflow for data collection and distribution to third parties is shown in Fig. 2. Alice finds an interesting third-party app from the malicious third-party entity, Mallory, which she wants to install. BMW's CarData platform [3] gives Alice an overview of sensors the app requests for proper functioning together with the purpose of the app, akin to app permissions shown at installation time in a mobile phone app store. Suppose this app requests SWA readings (which we will later use to infer location) besides other sensors. Unlike the Android Play Store where users can selectively check permissions they want to give to the app, The OEM only allows users to completely agree or decline [22]. If agreed, the OEM transmits a copy of the collected telematics data to Mallory's server via their business-to-business (B2B) interface, but as previously mentioned, does not have any influence on what happens afterwards with the data. The OEM is committed to only accept trustworthy service providers as business partners for their platform, such as insurance companies. If Mallory has a valid use-case for their app using SWA readings, the OEM might approve Mallory as a trustworthy entity. Since the user and OEM do not have any technical control on what happens to the data after giving Mallory permission to run that application, she can use the SWA readings as she likes, including inferring location as we will show in this paper.

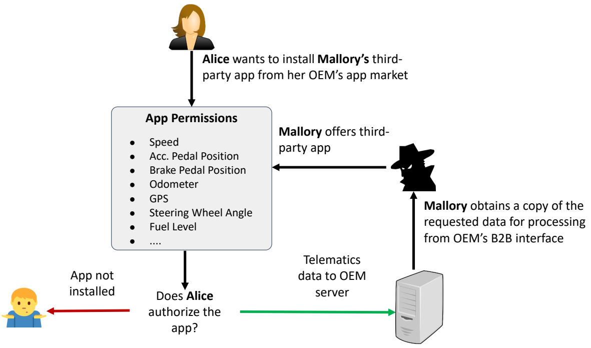

Alice wants to install Mallory's third-party app from her OEM's app market
App Permissions
Speed
Acc. Pedal Position
Brake Pedal Position
Odometer
GPS
Steering Wheel Angle
Fuel Level
....
Mallory offers third-party app
Mallory obtains a copy of the requested data for processing from OEM's B2B interface
Telematics data to OEM server
App not installed
Does Alice authorize the app?

Fig. 2. Example workflow of installing a third-party app in an OEM data-collection platform

The major reason for this risk/vulnerability, besides the relay of data to remote third-party servers just based on user consent, is the weak permission model of those data-collection platforms, due partly to lax regulation. Any sensors that are not covered by the three categories in the aforementioned guidelines [5] can be even regarded as *zero-permission*, comparable to the use of this concept in mobile phone operating systems. The SWA sensor is not clearly assigned to the list of "covered information". One might argue that SWA can be regarded as part of driving behavior information which is part of the "covered information" list. Although we agree and cannot explicitly rule out this sensor from the list, SWA could rather fly under the radar during the design of permission models and can thus be used as *zero-permission*. After all, the guidelines are merely voluntary and each OEM is free to design the permission model stricter or looser.

The feasibility of a location inference attack exploiting SWA is a combination of three factors: (i) Weak/insufficient architecture design by the OEM regarding third-party apps, (ii) users being comfortable sharing SWA data, and (iii) lax privacy regulation. Even if an OEM decided to classify SWA as part of "covered information", we believe that a location inference attack exploiting SWA would still happen with a high likelihood. This holds especially when compared to data collected from GPS or speed sensors. As shown in our survey results from Fig. 1, these two data sources are highly sensitive and the chance of a user installing an app with these permissions less likely. In fact, it has already been shown in literature [8–10] that location inference by using speed information is possible, albeit with less feasibility as discussed in this section and lower accuracy as shown below.

5 System Design

We present the system design of SPy which infers location from SWA sensor data. For this purpose, we need to estimate road curvature as an intermediate step toward location estimation using a novel technique, called RoCuMa (Road Curvature Matching). A deeper look into SWA readings — one of the most frequently collected data according to [19] — shows that these values are highly correlated to the road curvature. Fig. 3 (a) shows an example plot for SWA readings and the road curvature among a busy road in the mid-sized US city of Ann Arbor where all experiments have been conducted. The cross-correlation for all roads is found to be always above 0.8, and 0.93 for this specific example.

Since this looked promising, we continued exploration of the high correlation to infer location from SWA readings. Fig. 3 (b) shows a system block diagram. The first step is to derive road curvature from SWA readings collected by a malicious third-party app from the victim's vehicular data-collection platform. The ground truth road curvature can be calculated from publicly available sources, such as OpenStreetMap (OSM). By matching the road curvature from the attack with the ground truth curvature, we will infer the traveled route of the victim for its trip and thus its location.

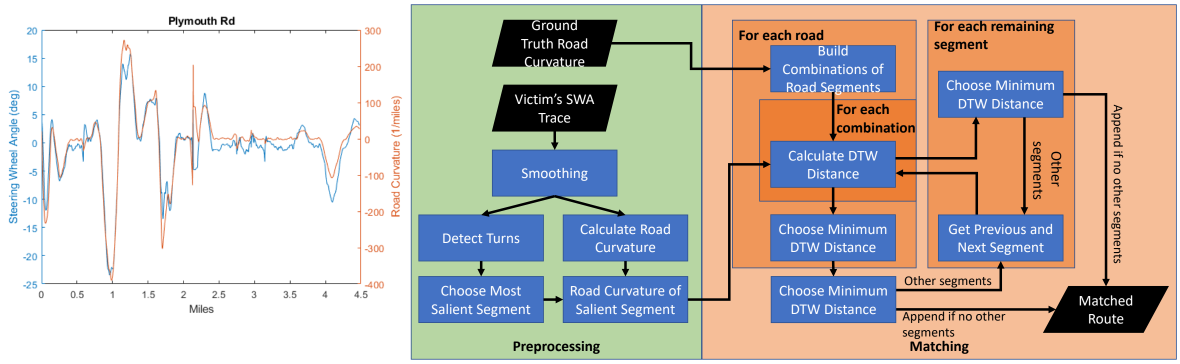
We will first (in Sec. 5.1) introduce the dataset used finding SWA traces for the attack. Then, we will detail how to build ground truth road curvature from OSM in Sec. 5.2. Finally, Sec. 5.3 and Sec. 5.4 cover the location-privacy attack. Specifically, we will first show how to derive road curvature from SWA data and then infer location from road curvatures (RoCuMa).

5.1 Input Data

Our data source is a dataset collected by researchers in our group using the OpenXC platform [14] during their daily driving. The entire dataset consists of 58 trips/attack traces collected from several drivers and vehicles, with 41 recorded in a 2016 Ford Explorer, 10 in a 2017 Lincoln MKZ, 5 in a 2017 Ford Escape and 1 each in a 2016 Ford Focus Hatchback and 2017 Ford Fiesta, respectively. An overview of vehicle sensor parameters available in the dataset is provided in Table 5 of Appendix B. The dataset includes GPS coordinates as well as SWA readings. The former are used in Sec. 6 as ground truth to compare against the derived location trace and thus evaluate the accuracy of SPy. The latter are used as attack traces. Finally, we use the OpenStreetMap (OSM) API to derive road curvature and build our ground truth road curvature database.

5.2 Building the Ground Truth Road Curvature Database

Our goal is to derive ground truth road curvature for all roads in a selected area. We exported a rectangular area around our test city Ann Arbor. The obtained file includes nodes that contain longitude and latitude values. OSM defines ways as an ordered list of nodes representing roads. Based on this representation, we apply spline


Plymouth Rd
Steering Wheel Angle (deg)
Road Curvature (1/miles)
Miles
20
15
10
5
0
-5
-10
-15
-20
-25
300
200
100
0
-100
-200
-300
-400
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Ground Truth Road Curvature
Victim's SWA Trace
Smoothing
Detect Turns
Calculate Road Curvature
Choose Most Salient Segment
Road Curvature of Salient Segment
Preprocessing
For each road
Build Combinations of Road Segments
For each combination
Calculate DTW Distance
Choose Minimum DTW Distance
Choose Minimum DTW Distance
For each remaining segment
Choose Minimum DTW Distance
Other segments
Get Previous and Next Segment
Append if no other segments
Other segments
Append if no other segments
Matched Route
Matching

(a) (b)

Fig. 3. (a) Smoothed SWA readings vs. road curvature to depict correlation (b) Block diagram of system design

interpolation to the list of nodes of each way since we want to calculate that road's curvature in the next step.

Given x and y as longitude and latitude (x' and x'' denoting their first and second derivatives), respectively, the signed road curvature k [23] is defined as

$$k = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}}. \tag{2}$$

The curvature data is stored for each road together with the SWA readings. An example for a stored road is given in Fig. 3(a). Both the SWA and curvature values are depicted over the mile marker of the road. Note that the definition of the start and end mile of a road are deliberate and that we store each road in both directions. A traveled path usually consists of multiple road sections of different roads, i.e., the driver will turn onto another road at some point during the trip. In order to consider this, we have to split the roads in the ground truth database into multiple *road segments*. We define a road segment as a road portion between two intersections. A road section, which depicts the road fragment between two turns, can consist of one or multiple road segments (since the driver does usually not turn at each intersection). In order to achieve this, we calculate the intersection points between roads from OSM. An intersection point can be a four-way or a three-way (also called *T-intersection*). The intersection points are then mapped to the mile marker of each road which is then split into multiple road segments between two intersections by a simple rectangular window. Our naming convention for the road curvature ground truth is G_x_y with x denoting the ID of the road and y the ID of the road segment on that road.

5.3 Road Curvature from SWA Readings

As shown in next subsection, location will be inferred from road curvature. Below we will briefly describe how road curvature can be calculated from SWA readings. In the previous subsection, we stored road curvature values for each road segment. We also have SWA readings from the 58 attack traces that we collected from multiple cars. Based on our motivation that these two variables are highly correlated, we tried to find the line of best fit between these. We obtained the best fit for a polynomial of first degree. All other polynomials of higher degree and sinusoidal regression models returned higher RMSE values. In order to build a robust and accurate model, we concatenated the data points of all SWA readings and their respective curvatures from the ground truth database.

As a result, the relationship between steering wheel angle (swa) and road curvature (curv) can be expressed through the following equation, using linear regression with Bisquare weights:

$$curv = 15.76 \cdot swa - 4.923 \tag{3}$$

Despite the attempt to find the most optimal parameters for the linear model, we just want to show the linear relationship between steering wheel angle and road curvature. The absolute values of scale and offset in Eq. (3) can be ignored in our map-matching technique RoCuMa (see Sec. 5.4) since the ground truth and attack curvatures are being normalized and the very small intersection of Eq. 3 barely affects the relation between SWA and curvature. As a result, even SWA readings themselves could be used as input since we show that only the temporal sequence of these values turn out to be relevant. Nevertheless, we will still use predicted road curvature values from SWA readings in order to stay as accurate as possible.

5.4 Location from Road Curvature

After obtaining the ground truth road curvature and learning the relation between SWA readings and road curvature, the victim's location can be inferred from their collected SWA readings. We call this novel map matching RoCuMa (Road Curvature Matching) which is detailed next.

5.4.1 Step 1: Pre-processing

According to our threat model, the victim installs a malicious third-party app on their connected vehicular data-collection platform and agrees to share their SWA readings with the third-party service provider. After the victim has completed their trip, the attacker has access to the entire SWA trace of the victim and can process this data offline. The SWA trace merely consists of SWA readings with the corresponding time stamps. In order to eliminate noise, the readings are smoothed by a moving average filter.

SWA readings offer numerous insights into the road geometry. For instance, turns can be easily detected by analyzing the trace. We want to automatically detect left and right turns along the victim's route so we can divide the entire trace into multiple road sections (not to be confused with our definition of road segments). Each section denotes a single road that the victim has traversed. Turns are detected by SWA values exceeding a specific threshold. The latter depends on the steering ratio for each individual car. Since all experiments from our dataset were conducted using the same vehicle model, the steering ratio and thus threshold were the same (90° in this case).

Since RoCuMa cannot detect U-turns or roundabouts, we are limited to conventional turns. At a turn event, a new road section is added and stored together with the type of turn (left or right). The general idea is to find an initial section and to match this against the entire ground truth data and then match the previous and following sections in a smaller search space. The most salient or distinct section is chosen as initial section. We define the most salient section as the curviest, i.e., the section with the maximum absolute SWA value. The reason for this choice is to increase the chance to match this section correctly against the entire ground truth which is the search space in this step. Since the majority of roads in our dataset have none to very small curvature (87.9%), the likelihood of matching a straight section correctly is very small. Matching this initial section correctly will also determine the accuracy of RoCuMa for the most part. Our evaluations in Sec. 6 show little error in matching the remaining sections due to a smaller search space.

5.4.2 Step 2: Matching

The initial section may consist of multiple road segments since drivers usually do not turn at every intersection. Nevertheless, the initial section is bounded by an integer number of road segments by their definition. We do not know how many road segments the initial section consists of, and hence we need to match it against all single road segments as well as combinations of multiple adjacent road segments.

Alg. 1 describes the matching process of the initial section against the entire search space in pseudocode. First, the initial section is matched against all single road segments in the ground truth database (in both directions). For quantifying the similarity between two segments, we use *Dynamic Time Warping* (DTW). This algorithm calculates the Euclidean distance between the curvature signals after stretching them to a common set of instants so that the distance is minimized. The reason for choice of DTW in our case is simple: the ground truth road curvature is stored with respect to location (mile marker) whereas the attack road curvature has been calculated from SWA readings with respect to time. With the knowledge of vehicle speed, we could have easily converted road curvature against time to road curvature against location, but our threat model does not assume the collection of the speed parameter. With certain options of the DTW algorithm, we will later see in Sec. 6 that this method works well for comparing two segments with each other without the knowledge of speed.

In Alg. 1, c denotes the attack road curvature we obtained from the SWA readings and s the road segment or combination of road segments (s_R being the road segment in reverse direction) which it is matched against. These curvatures have to be z-normalized first which is commonly used with the DTW algorithm. Since the amplitude of the calculated attack curvature differs from the amplitude of the curvatures in the ground truth and we are more interested in the structural similarities of these values, calculating the z-score is essential. This step converts all values in the curvature vector to a common scale with an average of zero and standard deviation of one with μ denoting, respectively, the mean and σ the standard deviation of the sample:

$$curv_{norm} = \text{ZSCORE}(curv) = \frac{curv - \mu}{\sigma}. \tag{4}$$

Due to normalization, the absolute value of the slope of the linear relationship we derived does not affect the matching process anymore as discussed before.

```
foreach $road \in roads$ do
    dist, dist_R ← $\emptyset$;
    foreach $road\_segment\ s \in G$ do
        score ← DTW(ZSCORE(c),ZSCORE(s));
        score_R ← DTW(ZSCORE(c),ZSCORE(s_R));
        dist(road).append(score);
        dist_R(road).append(score_R);
    end
    distM, distM_R ← $\emptyset$;
    for $k \in \{2, \ldots, N-1\}$ do
        sM ← all s succeeded by k segments;
        sM_R ← all s_R preceded by k segments;
        score ← DTW(ZSCORE(c),ZSCORE(sM));
        score_R ← DTW(ZSCORE(c),ZSCORE(sM_R));
        distM(road).append(score);
        distM_R(road).append(score_R);
    end
end
overall_min ← ARGMIN([dist, dist_R, distM,
  distM_R]);
first_segment, combo ← find_segment(overall_min);
```

Algorithm 1: Matching the initial section

Alg. 1 first calculates the DTW score for comparing the attack road curvature with each single road segment in both directions since we do not know which way the victim took. All these values are appended to an array with DTW scores. Then, the attack curvature (initial section) is matched against all possible combinations adjacent road segments, starting with two combined road segments up to $N-1$ combined road segments, where N denotes the number of road segments of a particular road. Thus, the total number of comparisons for matching the salient section is $\frac{N(N+1)}{2}$ and the computational complexity stands at $\mathcal{O}(N^2)$. Again, all the DTW scores are combined in an array in both directions. This procedure is repeated for each road in the ground truth database.

Finally, the minimum score is calculated for all these arrays and the minimum score is chosen which is also the most likely segment in the ground truth that our initial section matches to. Together with the minimum score, we also know the initial road segment and the number of road segments to follow this initial segment. We achieve the most crucial part in our matching process since we have to match the initial section against the entire ground truth. We append the matched segments to the output array of GPS coordinates of this particular road section.

After matching the initial road section, we have to check if the segmentation of the SWA trace from Step 1 returned other section preceding or succeeding the initial section. If there are more sections, we need to obtain the intersection of the initial section with the adjacent sections. Since intersection points are stored with the ground truth, we can easily see which roads the initial section intersects with. We distinguish between a four-way and a three-way intersection (T-intersection). Anyway, we know the type of turn (left or right) thanks to the characteristics we derived earlier from SWA traces. The only remaining challenge is to find the first segment of the new road section the victim turns to.

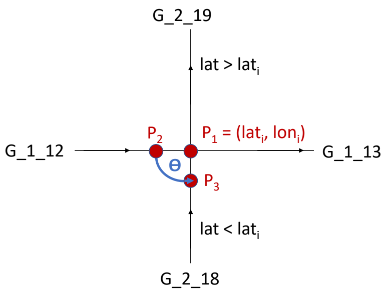


Fig. 4. Finding the first segment on a new road section

For an example four-way intersection, this procedure is depicted in Fig. 4. The turn angle Θ is defined as the four-quadrant inverse tangent with P_2 being a point on the segment of the road the victim is coming from (in this case the initial section) and P_3 being a point on the road section to turn on. P_1 is the intersection point. x and y are its latitude and longitude, respectively. This angle can be calculated by:

$$\begin{aligned}\Theta = tan^{-1}(P_2.y - P_1.y, P_2.x - P_1.x) \\ - tan^{-1}(P_3.y - P_1.y, P_3.x - P_1.x).\end{aligned} \tag{5}$$

After calculating this angle for both possible road segments of the intersecting road (in the T-intersection case, this would not be necessary if the victim has only one turn possibility), we compare the sign of the angle with the turn type and determine the first segment of the new road and the driving direction. Suppose the victim comes from G_1_12 (the next segment would be G_1_13) and turns right onto G_2_18. If the victim turned left, they would have ended up on G_2_19. In order to determine the first road segment on the new road (G_2_18), we use a GPS point P_2 on G_1_12 (which we already have matched), the intersection point P_1 and a point P_3 on G_2_18 and G_2_19 respectively using the trigonometric relationship from Eq. 5 to calculate the angle Θ. Based on the sign of that angle and the type of turn, we will determine that the first road segment of the new section will be G_2_18_R since the road will be traversed in the reverse direction.

Now, a simpler version of the same procedure from Alg. 1 can be repeated to match the previous and succeeding sections of the SWA trace. Since we already know the first road segment of the new road section, the search space will be much smaller. We only have to match p times where p is the number of remaining road segments from the new section ($p \leq N$). The minimum DTW score will yield the matched segments which we can append to the output. This process is repeated as long as there are sections in the SWA trace available. Once there is no section left, the output will consist of the GPS coordinates of the victim's SWA trace.

6 Evaluation

In what follows, we will evaluate multiple metrics of SPy to give a complete picture of the feasibility of privacy attack with SWA traces. We will cover the accuracy, memory footprint, complexity as well as applicability to other cities. Note that the latter is mostly a qualitative discussion since actual driving data in the other cities are not available.

6.1 Attack Trace Subset of a Single Road

First, we want to evaluate how well a subset of a road is matched to its corresponding positions. In this case, our ground truth database consists of only one road and the attack SWA trace represents part of this road.

We did not evaluate the performance of this testcase on a large number of attack traces since our goal is matching the SWA trace on multiple roads as detailed in the next subsection. In particular, we want to point out an issue of short traces we faced in order to improve Alg. 1. DTW comes with a width adjustment window ensuring that the warping compares sections of similar length and does not overfit. The effect of this parameter is shown in Fig. 5 where the blue and red lines/signals depict the attack and ground truth curvatures, respectively. The matching on the right side of this figure yields a smaller DTW distance which would make it an optimal candidate. Nevertheless, the matching on the left side is correct as one can see from the original signals on the top, but yields a higher DTW distance because of overfitting, i.e., the DTW algorithm with no width adjustment window set tries to perfectly align the signals which are similar as a temporal sequence and yields a smaller distance due to a smaller number of points.

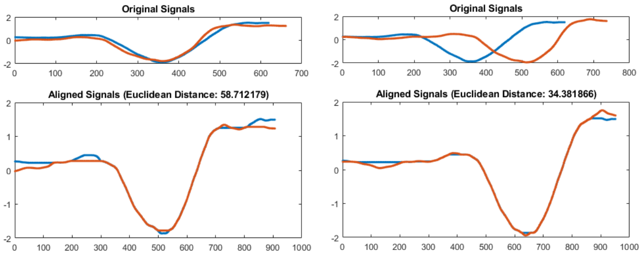


Fig. 5. Width adjustment window not set vs. 30

For a larger trace, this finding does not hold as the driver may travel at a higher speed so that the case depicted on the right might be true. We found through the analysis of multiple attacks that this observation mostly holds for less than 4 road segments. For more than 4 road segments, we do not set the width adjustment window in order not to affect results. Note that this threshold of 4 road segments does not guarantee a correct result and that an extensive analysis showed better results by setting this threshold.

6.2 Attack Trace on Multiple Roads

Next, our system is evaluated for the entire SWA traces that lie on multiple roads. We now evaluate the overall accuracy of this system, i.e., the percentage of correctly classified routes among our test set.

We built the ground truth database in a mid-sized US city Ann Arbor. The area covered by our ground truth database consists of 236 roads and 2776 road segments. For the attack traces, we used our own dataset introduced in Sec. 5.1 which only contains trips taken in Ann Arbor. Table 6 in Appendix B shows statistics of the 58 attack traces. Trip length distribution statistics including a histogram and CDF are also provided in the same appendix.

We used all recorded 58 trips from our dataset as different attack traces. As Table 6 shows, we use attack traces of different lengths and avoid only having short trips since we also want to evaluate the performance of RoCuMa on longer trips. An important pre-processing step was to manually discard the first and last one or two segments in each attack trace since they consisted of events, such as pulling out or into the driveway which would have made the matching process difficult. Note that this can be automated by a further signal analysis, but is not in the scope of this paper.

Eq. (6) defines the overall accuracy and Table 3 summarizes the results. We evaluate the full route match, i.e., how many trips are classified correctly over their full length.

$$overall\_accuracy = \frac{\#\ correctly\ matched\ attack\ traces}{\#\ all\ attack\ traces} \tag{6}$$

Table 3. Evaluation of overall accuracy with respect to trip length

	Maximum Trip Length (mi)	Full Route Match
All attacks	19.85	70.7%
20-percentile	1.12	72.7%
50-percentile	2.80	72.4%
75-percentile	5.03	69.8%
90-percentile	10.19	73.1%

Our system was able to match 41 of 58 attack traces, achieving 70.7% accuracy overall. Furthermore, we want to know how the accuracy is affected by the length of the attack trace. Fig. 6 summarizes our findings. For short trips up to 1.2 miles, the accuracy rises quickly up to 75%, but then experiences a small drop and fluctuates between 67% and 75%. Even for longer trips, the accuracy stays constant slightly above 70% and does not decrease significantly. We, therefore, conclude that the accuracy of RoCuMa is not greatly affected by the length of the trips. This conclusion can also be validated by the accuracy evaluation for percentiles in Table 3.

Finally, we also analyzed in how many cases we could match the initial (salient) section: 44 cases over the entire 58 attack traces. Thus, the critical part of RoCuMa is matching the initial section due to a very large search space. Only in 3 cases, the algorithm was unable to match the full trace despite obtaining the correct initial section. This explains why the accuracy does not drop significantly for longer trips. Once the initial section has been classified correctly, the chance of matching the remaining segments also correctly is high and thus does not affect the accuracy of the full route match.

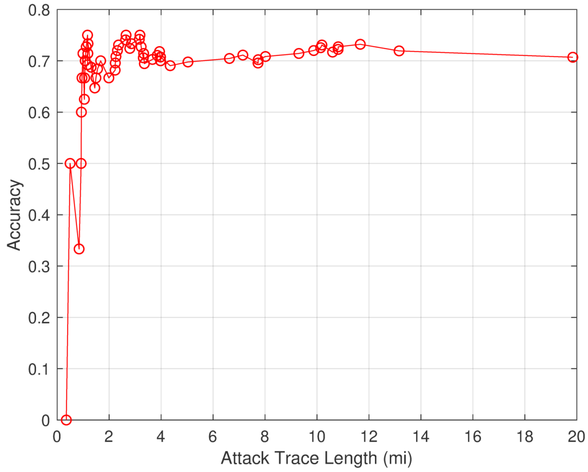


Fig. 6. Impact of trip length on overall accuracy

We analyzed the 3 mis-classified initial sections and found a common pattern why the salient section could not be matched. An example of this is depicted in Fig. 7.

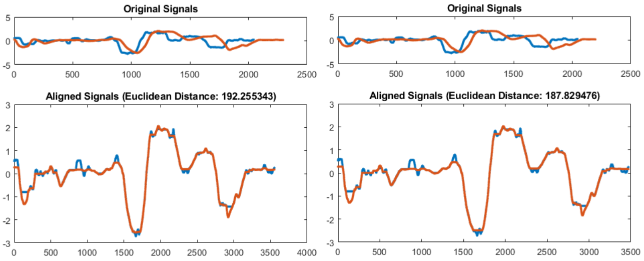


Fig. 7. Example of mis-classified salient section

The figure on the left shows the correctly matched initial section, but has a slightly higher DTW distance than the matched section on the right. The corresponding map area for this example is depicted in Fig. 8. Instead of matching until the correct intersection with *Road 2*, it matches up to the intersection. The reason for this is that the last part of the section until the false intersection is very straight, such as the road segment between the false and correct intersection. This additional road segment adds some distance in the DTW algorithm and leads to mis-classification. Salient sections with some curvature before the intersection with another road are all classified correctly.

6.3 Memory Footprint

An important metric to assess the feasibility of the proposed attack is the memory footprint of the ground

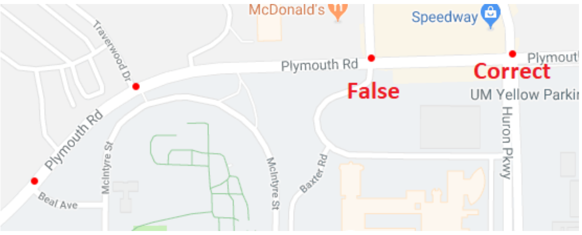


Fig. 8. The map area for mis-classified example

truth database. The malicious third-party has to possess a ground truth database of road curvatures to match the collected SWA trace. Road curvatures are stored for each road segment in a file with the format (curvature, latitude, longitude). The extended area of city Ann Arbor where we conducted experiments consists of approximately 540 miles of roads and its ground truth files take 29.8MB of storage. This translates to roughly 55.2kB of storage per mile and 10.6kB per road segment.

An attack is usually limited to a city level and only the ground truth of a city has to be stored. Nevertheless, an attacker might want to keep a ground truth database for a larger area which they can scale down to the area of interest later when mounting the attack. For instance, let us consider the Detroit metropolitan area of over 5 million inhabitants with 184,950 unique roads that subsumes Ann Arbor. Using linear extrapolation, we can estimate the required storage for this entire area to approximately 26 GB. It is reasonable for this metric and shows that the proposed attack again is feasible with regards to the memory footprint of the ground truth database.

6.4 Complexity and Computation Time

Since our system does not track the driver in real-time during their trip, but matches their SWA traces after the collection of entire trip data, the computation time is not a major issue for the performance of our system. Nevertheless, we wanted to see if the matching could be done in a reasonable time.

All attacks were implemented and run on a Windows 10 mobile workstation running MATLAB R2018a with an Intel Core i7-8650U CPU (quad-core using up to eight threads running at 1.90GHz) and 16GB of RAM. The maximum time an attack trace took was less than 19 min which is reasonable on a low-end setting for an offline attack. It is also worth mentioning that the DTW algorithm took more than 90% of the entire computation time for each of the 58 traces.

Furthermore, we would like to assess how this attack would scale with respect to time if the search space, i.e., the geographical area of the given ground truth, was larger than the mid-sized US city Ann Arbor we used.

As mentioned earlier, the main computation of this algorithm comes from matching the initial section against the entire ground truth database, incurring $\mathcal{O}(N^2)$ computation cost for N road segments. Matching the remaining sections will only cost $\mathcal{O}(N)$ and is thus negligible as our timing analysis shows: only 0.5% of the entire time is spent on average for matching the remaining sections while more than 99% is used for matching the initial section.

Scaling this attack in space has already been discussed in Sec. 6.3. In terms of time scalability, we can roughly estimate the required time with the knowledge of road segments in a city and the time spent in our test city. Table 4 summarizes the number of road segments in our reference city Ann Arbor and other selected cities. Given the worst-case time of 19 min for an attack in Ann Arbor, an attack in the entire geographical area of a major US city like Boston would take less than 4 hours. Under the same assumption, an attack in a major European city such as Munich would take slightly over 9 hours. Note again that for these experiments, we used a less powerful laptop and a malicious third-party is likely to have better equipment to run the attacks at a much faster speed.

As a result, even for larger cities around the globe, an offline attack can be considered feasible in terms of time. Finally, we also have to consider how scalable this attack is in different cities/areas with respect to the the curviness of the roads as well as the size of the ground truth database.

6.5 Applicability to Other Cities

Since RoCuMa relies heavily on curvature of the roads and matching the initial section is the crucial part in inferring the correct route, it will not work in areas which are on the grid, such as Manhattan. Nearly all roads there are straight and thus have close to zero curvature. Since all segments will look similar, the choice of the salient section will be unsuccessful. This drawback will also be discussed in Sec. 7. Since we want to justify the accuracy of the evaluation in the previous subsections, we need to introduce a measure of road curvature for the area where we conducted experiments, and compare it with other cities.

To this end, we selected eight cities in the US and Europe to show and discuss the curviness of their roads using an open-source algorithm for calculating the road curvature of maps obtained from OpenStreetMap [24]. The authors of this algorithm used a measure of road curvature in the range from 0 to over 20000 which is different from ours since they used a different approach and units to calculate the curvature. We processed the curvature of all roads in these cities that we wanted to analyze and visualized the distribution of length vs. curvature in Fig. 13 of Appendix C.

Then, we defined the average curvature index to quantify the curviness of the roads in that particular city. The following equation shows our rating formula, with segments being parts of all the roads in that city:

$$avg\_curv\_index = \frac{\sum_{i=1}^{\#segments} curv(i)\ length(i)}{\sum_{j=1}^{\#segments} length(j)} \qquad (7)$$

The average curvature indices for our selected cities are summarized in the Table 4, with the first row being our reference city Ann Arbor.

Table 4. Number of road segments and average curvature index for different cities

	# Road Segments	Avg. Curv. Index
Ann Arbor, MI	2776	207.82
Boston, MA	9539	195.25
San Francisco, CA	7515	158.73
Manhattan, NY	1920	92.51
Pittsburgh, PA	10692	248.61
Dublin, Ireland	12977	221.42
Ingolstadt, Germany	2338	225.17
Munich, Germany	15071	152.30

All 58 attack traces are located in a mid-sized US city Ann Arbor which is relatively curvy compared to most US cities on the grid such as San Francisco or Manhattan. European cities on the other hand have a higher curvature rating than Ann Arbor. As a result, our overall accuracy of 70.7% found in our experiments only holds for areas with a similar curvature rating. We do not have SWA data from trips in those cities, but the accuracy will likely decrease in cities with a lower curvature rating.

Table 4 also depicts the number of road segments in each city. It would be interesting to see how the accuracy scales to a larger or smaller area with more or less road segments, respectively. The main impact on accuracy is the success of matching the initial (salient) section as discussed earlier. As a result, the number of road segments does not necessarily affect the accuracy, but much more the similarity of highly curvy road segments (or lack thereof). If the initial section is mis-classified for another curvy section, the attack will not succeed. Similarity can be expressed by the standard deviation σ of curvature values. A larger σ results from more distinct and less similar road segments whereas a smaller σ represents more similar road segments.

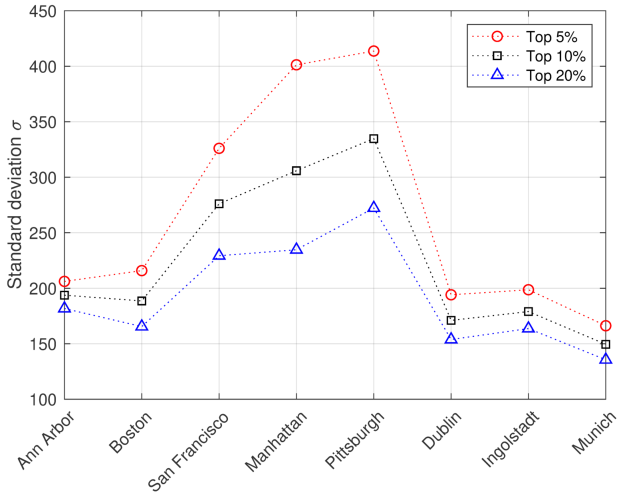


Fig. 9. Similarity comparison of curviest segments

To this end, we analyzed the curviest 5%, 10% and 20% of all road segments in the aforementioned eight cities and calculated the standard deviation σ for these intervals. Fig. 9 shows that San Francisco, Manhattan and Pittsburgh have more distinct road segments among their curviest ones than our reference city Ann Arbor. Together with the notion of curviness of these three cities, we can thus reason that an attack is more likely to succeed than in Ann Arbor, whereas Manhattan would still be difficult due to its geographical layout. Another relevant observation is that all European cities have a similar (Dublin, Ingolstadt) or slightly smaller (Munich) σ, and are thus equally or somewhat less distinct in their curviest segments. This can be explained by the higher number of curvy segments than, for instance, Manhattan. In the latter, there are only few curvy segments which are more distinct in their shape. So, the proposed attack would work equally well in Dublin and Ingolstadt, whereas the accuracy would drop in Munich, also partly due to the smaller average curvature.

Since all these comparisons are theoretical, we would like to see if these claims hold in a real setting. So, we decided to define a new "area" that is a subset of our reference city by only inspecting the performance of RoCuMa on 15 traces that were collected in that specific area (i.e., traces that start and end there). This specific part of the city has a higher average curvature index (268.73 instead of 207.82 for Ann Arbor) and is thus a reasonable choice for comparison with different city topologies. Furthermore, the number of road segments is only 550 and thus covers a small part of Ann Arbor. The mean trip length of this subset is 2.2 miles, compared to 4.28 miles for all 58 attack traces. RoCuMa was able to match 13 of 15 traces, yielding 86.7% accuracy, compared to 70.7% overall accuracy over the entire dataset. As a result, we saw that a geographical area with higher average curvature index can yield higher accuracy and thus we can experimentally validate our theoretical derivations.

7 Discussion

7.1 Limitations

First, we did not consider lane changes. According to the National Highway Traffic Safety Administration (NHTSA), an average lane change event has a change in steering wheel angle of 8.11 degrees [25]. We can ensure that all our traces contained lane changes. Since we smoothed the SWA data before processing it, these changes did not affect our performance. Nevertheless, lane-change events over a large number of lanes would definitely affect RoCuMa and have to be filtered out.

Second, a major point discussed in Sec. 6 was the variance of curvature of the roads in our dataset. Since we solely rely on road curvature, curvier roads in a city are advantageous for the accuracy of this algorithm. Many US cities were planned on the grid, i.e., they consist of straight roads with almost zero curvature. As a result, a victim traveling in Manhattan would not have to fear of getting their location information compromised whereas most European and smaller US cities contain curvier roads, making an attack much more feasible.

Third, we did not consider U-turns or roundabouts. Although we showed in Fig. 6 that U-turns can be detected, we did not evaluate any trips containing them.

Finally, this privacy attack assumes a rough knowledge of the city or limited geographical area. If our malicious third-party app can retrieve additional side-channel information, such as the Vehicle Identification Number (VIN), we could query a publicly available database such as *vehiclehistory.com* [26] to get information about where the car had been serviced recently and thus limit the location.

7.2 Comparison with Mobile World

Although SPy is similar to mobile device location inferences as summarized in Table 1, one needs to distinguish between automotive and mobile worlds. Our threat model is based on the vehicular ecosystem exploiting vulnerabilities in latest vehicular data collection and sharing platforms. All previous vehicular location inference studies relied on OBD-II dongles without sharing the data with any third-party developer. So the threat arising from the latter is very low.

Vehicular and mobile IMU sensors are highly correlated (e.g., speed–accelerometer, SWA–gyroscope) and it is possible to calculate the former from the latter as shown in [27]. For this experiment, they simultaneously collected SWA data from the CAN bus as well as gyroscope readings from mounted Google phones. Nevertheless, the SWA estimation from gyroscope readings resulted in errors. The mean error was reported to be 7.53°. Furthermore, the mean absolute error is not constant, but can deviate with respect to SWA. Vehicular sensors are more robust due to higher quality. For instance, a commercial steering angle sensor [28] offers an accuracy of 1.5°. As a result, the faultiness of gyroscope-based SWA estimations would affect the accuracy of the attack described in this paper. Vehicular sensors would be more effective in launching SPy's privacy attack.

Finally, the threat model requires explicit permission for speed data whereas the accelerometer is a *zero-permission* sensor on mobile phones. This makes the design of an algorithm leveraging speed data very cumbersome and the attacker must use less sensitive parameters such as SWA which has also been shown through our awareness survey in Sec. 3. Hence, previous work focusing on only speed or a combination thereof is unlikely to work within our threat model.

7.3 Countermeasures

Since its accuracy is evaluated to be relatively high, SPy provides a feasible way for malicious third-parties to infer their users' location without explicitly requesting them. OEMs will still have to be responsible for preserving their customers' privacy.

Two-layer privacy protection, both distorting (e.g., Laplace mechanism) and down-sampling the SWA data on OEM servers upon sharing with the third-party, might provide a promising approach. The amount of distortion is bounded by utility requirements of the apps since too much added noise on the data would make the third-party app unusable. Intuitively, reducing the sampling frequency of the original SWA trace will also reduce the accuracy of SPy reported in Sec. 6.2.

8 Related Work

8.1 Mobile Location Privacy

Location privacy is a well-researched area with several publications addressing what potential privacy threats are caused by location leakage and how those can be addressed or mitigated in their respective applications [29–31]. It has been shown that tracking user location is very privacy-sensitive since third-parties can extract various information from it and monetize the information. For instance, a third-party company tracking the eating behavior of a user in restaurants can sell this information to their health insurance company which adjusts their premium according to the healthiness of aforementioned behavior [32]. As mentioned before, the knowledge of both the home and workplace location which can be derived easily from a GPS trace can help identify the user with a high probability [11].

It has been shown that mobile apps using zero-permission (e.g., IMU) sensors alone can infer user location. Han *et al.* [33] used only accelerometer readings from smartphones in a vehicle to infer the starting location and trajectory of the traveled route to within a 200m radius of the true location. Narain *et al.* [6] inferred users' routes and locations using basic zero-permission IMU sensors, such as accelerometer, gyroscope and magnetometer on smartphones in moving cars. Their evaluation showed an accuracy of 30–60% in a real operational setting. In another case, Michalevsky *et al.* [7] showed that monitoring the aggregate power consumption on smartphones allowed to infer traveled routes. They used the changing power consumption of cellular radio on Android phones — which again does not require any permissions — as a function of the distance to cell towers to infer the location. Map matching in this case needed a lot of additional work as they had to provide a map with the signal strength of a specific cellular service provider as ground truth. Their accuracy accumulated up to 45% in an ideal case without background noise from other apps on the phone. Mosenia *et al.* [34] used several non-sensory and sensory data on phones such as air pressure, time zone and various IMU sensors to predict the user's location during four activities: walking, traveling on a train, driving, and traveling on a plane. It yields a high accuracy if the area is narrowed down, but comes with similar drawbacks of SPy such as curviness-dependency.

Other work focused on inferring location from vehicle speed data using OBD-II dongles [9, 10] or GPS tracking units installed in vehicles [8].

Furthermore, SPy distinguishes itself from others by addressing four key weaknesses in these related studies. First, the accuracy for deploying the attack in a real scenario is too low to be useful. Accuracy is defined as the percentage of the correct route being in a candidate set of possible inferred routes. Second, strong/unrealistic assumptions were made for their threat model, such as initial starting position of trip routes. Third, location attacks using phones are launched through a malicious app in mobile OS' app stores. Installing such an app is much less likely than in a vehicular app store due to the huge number of apps in the app store (see Table 1). Finally, some related work used other reference sources than map information to match their data input against. Scalability using publicly available map information is very high whereas methods with reference sources that had to be collected specifically for map matching will require serious additional effort and be less scalable. SPy overcomes all these weaknesses; see Table 1 for its comparison with related work.

8.2 Vehicular Data and Privacy

To this date, little has been done on automotive data privacy, especially about inferring user location from vehicular data because automotive data collection has never been a main focus of academic research, although it has always been possible to collect vehicular data for research through the OBD-II port. For instance, Enev *et al.* [35] showed in 2016 how driver fingerprinting based on data collected from the OBD-II connector can be done with up to 100% accuracy using readings from multiple sensors. They showed the feasibility of a privacy leakage, but did not discuss any countermeasures against this obvious privacy threat.

9 Conclusion

We have uncovered a new privacy threat model and shown that by solely using steering wheel angle (SWA) readings from the victim's car, it is possible to infer the victim's traveled routes with a relatively high accuracy using a novel technique called RoCuMa. Another contribution of this paper is an awareness survey about participants' sensitivity towards sharing certain vehicular sensor data with OEMs and third-parties.

Our work is an example of privacy leakage which results from giving third-party entities access to even seemingly benign sensors such as SWA. SPy shows that it is important to develop better permission models for vehicular data-collection systems such as in the mobile world and also deploy mitigation techniques to avoid malicious entities exploiting the drivers' privacy — especially considering the rapid growth of these systems and third-party applications running on them.

Acknowledgements

The work reported in this paper was supported in part by the US National Science Foundation under Grant CNS-1646130, and also by the Ford–University of Michigan Alliance Program.

References

[1] Matt McFarland. Your car's data may soon be more valuable than the car itself, February 2017. http://money.cnn.com/2017/02/07/technology/car-data-value/index.html.

[2] Tom Krischer and Dee-Ann Durbin. Q&a: The data your car collects and who can use it. https://apnews.com/31c018cdbc634d42a7c97d04774954b1, Sep 2016.

[3] Bmw group launches bmw cardata: new and innovative services for customers, safely and transparently. https://www.press.bmwgroup.com/global/article/detail/T0271366EN/bmw-group-launches-bmw-cardata:-new-and-innovative-services-for-customers-safely-and-transparently?language=en, May 2017.

[4] United States Government Accountability Office. VEHICLE DATA Industry and Federal Efforts Under Way , but NHTSA Needs to Define Its Role. Technical report, 2017.

[5] *Consumer Privacy Protection Principles PRIVACY PRINCIPLES FOR VEHICLE TECHNOLOGIES AND SERVICES*. ALLIANCE OF AUTOMOBILE MANUFACTURERS, INC, 2014.

[6] Sashank Narain, Triet D. Vo-Huu, Kenneth Block, and Guevara Noubir. Inferring User Routes and Locations Using Zero-Permission Mobile Sensors. *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, pages 397–413, 2016.

[7] Yan Michalevsky, Gabi Nakibly, Aaron Schulman, Gunaa Arumugam Veerapandian, and Dan Boneh. PowerSpy: Location Tracking using Mobile Device Power Analysis. *24th USENIX Security Symposium (USENIX Security 15)*, pages 785–800, 2015.

[8] Lu Zhou, Qingrong Chen, Zutian Luo, Haojin Zhu, and Cailian Chen. Speed-Based Location Tracking in Usage-Based Automotive Insurance. *Proceedings - International Conference on Distributed Computing Systems*, pages 2252–2257, 2017.

[9] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. Elastic pathing. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*, pages 975–986, New York, New York, USA, 2014. ACM Press.

[10] Rinku Dewri, Prasad Annadata, Wisam Eltarjaman, and Ramakrishna Thurimella. Inferring trip destinations from driving habits data. *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society - WPES '13*, pages 267–272, 2013.

[11] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. J. Bernheim Brush, and Yoshito Tobe, editors, *Pervasive Computing*, pages 390–397, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[12] Sarwant Singh. Are car companies going to profit from your driving data? https://www.forbes.com/sites/sarwantsingh/2017/11/06/are-car-companies-going-to-profit-from-your-driving-data/, Nov 2017.

[13] Obd - elm electronics. https://www.elmelectronics.com/products/ics/obd/, 2017.

[14] The openxc platform. http://openxcplatform.comOpenXC, 2017.

[15] Gartner says by 2020, a quarter billion connected vehicles will enable new in-vehicle services and automated driving capabilities. https://www.gartner.com/newsroom/id/2970017.

[16] Lynn Walford. Definition of connected car – what is the connected car? defined. http://www.autoconnectedcar.com/definition-of-connected-car-what-is-the-connected-car-defined/, Jul 2018.

[17] Bmw cardata. https://youtu.be/pcLsGapVkRY?t=770, Jan 2018.

[18] Pymnts. Who controls data in web-connected vehicles? https://www.pymnts.com/innovation/2018/data-sharing-smart-cars-privacy/, Jun 2018.

[19] Automotive data monetisation pricing and business models. http://www.frost.com/c/10046/sublib/frost-content.do?sheetName=report-overview&sheetGroup=MD48-01-00-00-00&viewName=virtual-brochure&repid=MD48-01-00-00-00, 2017.

[20] Ron Fricker. Linear Regression Analysis for Survey Data. Retrieved from http://faculty.nps.edu/rdfricke/MCOTEA_Docs/Lecture%2015%20-%20Linear%20Regression%20Analysis%20for%20Survey%20Data.pdf.

[21] Dong Kyu Lee. Alternatives to p value: confidence interval and effect size. *Korean journal of anesthesiology*, 69(6):555–562, 2016.

[22] Bmw connecteddrive customer portal – connecting to your bmw digitally. https://www.bmw-connecteddrive.co.uk/app/index.html#/portal/faq-and-support?section=cardata.

[23] Curvature, Sep 2019.

[24] Adam Franco. Road curvature - find twisty roads. http://kml.roadcurvature.com/, 2016.

[25] Suzanne E Lee, Erik CB Olsen, and Walter W Wierwille. A comprehensive examination of naturalistic lane-changes. Technical report, 2004.

[26] Research any vehicle in seconds. https://www.vehiclehistory.com/.

[27] Luyang Liu, Hongyu Li, Jian Liu, Cagdas Karatas, Yan Wang, Marco Gruteser, Yingying Chen, and Richard P Martin. Bigroad: Scaling road data acquisition for dependable self-driving. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 371–384. ACM, 2017.

[28] Steering angle sensor for automotive applications. http://www.methode.com/Documents/TechnicalLibrary/Steering_Angle_Sensor_Data_Sheet.pdf, 2007.

[29] Kassem Fawaz and Kang G Shin. Location privacy protection for smartphone users. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 239–250. ACM, 2014.

[30] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[31] Bill Schilit, Jason Hong, and Marco Gruteser. Wireless location privacy protection. *Computer*, 36(12):135–137, 2003.

[32] Dan Tynan. Why location privacy is important. https://www.itworld.com/article/2752981/mobile/why-location-privacy-is-important.html, Jun 2010.

[33] Jun Han, Emmanuel Owusu, Le T Nguyen, Adrian Perrig, and Joy Zhang. Accomplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–9. IEEE, 2012.

[34] Arsalan Mosenia, Xiaoliang Dai, Prateek Mittal, and Niraj Jha. Pinme: Tracking a smartphone user around the world. *IEEE Transactions on Multi-Scale Computing Systems*, 2017.

[35] Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno. Automobile Driver Fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2016(1), 2016.

[36] What is a vehicle identification number (vin)? https://www.autocheck.com/vehiclehistory/autocheck/en/vinbasics, 2018.

[37] P. Handel, I. Skog, J. Wahlstrom, F. Bonawiede, R. Welch, J. Ohlsson, and M. Ohlsson. Insurance telematics: Opportunities and challenges with the smartphone solution. *IEEE Intelligent Transportation Systems Magazine*, 6(4):57–70, winter 2014.

[38] I. Skog and P. Handel. Indirect instantaneous car-fuel consumption measurements. *IEEE Transactions on Instrumentation and Measurement*, 63(12):3190–3198, Dec 2014.

[39] David Tracy. How electronic throttle control works. https://jalopnik.com/how-electronic-throttle-control-works-499966101, May 2013.

[40] Transmission ratio rpm calculator. http://spicerparts.com/calculators/transmission-ratio-rpm-calculator, 2018.

[41] Philip Reed. What your check engine light is telling you. https://www.edmunds.com/car-care/what-your-check-engine-light-is-telling-you.html, May 2011.

[42] Openxc data set. http://openxcplatform.com/about/data-set.html.

A Survey

The descriptions and privacy implications of the 20 sensors we used in the survey are listed in the following.

Odometer: This is your current odometer reading which you can also see on your dashboard. It tells you how many miles you traveled. For each trip, the receiving party (OEM or 3rd party) can calculate how long you drove.

Vehicle Identification Number (VIN): This is a unique identifier which is associated to your vehicle. This number is also engraved into your windshield (readable from the outside of your car). There are vehicle history services in several countries that help potential car owners use VINs to find vehicles that are defective or have been written off. The receiving party can thus see how many owners that vehicle had, if it had been in accidents or even service records (some garages submit to the record database) [36].

Outside temperature: This is the outside temperature at the location where you are driving your car. It is usually the same as displayed on the dashboard of your car. Theoretically, the receiving party can reconstruct a route if the trip is long and there is a lot of changes in the outside temperature during this trip. This is very cumbersome and potentially not precise though.

Location (GPS): This is the location where your car is right now. It consists of a pair of two numerical values (latitude and longitude) which can pinpoint you pretty exactly where your car is right now. Depending on how frequently it is collected, the receiving party can track all your routes.

Current speed: This is the speed your vehicle is currently driving. The receiving party can reconstruct your driving behavior by analyzing the acceleration or deceleration (braking) or even infer your location [8–10, 37].

Average speed: The average speed is usually calculated over a trip. A trip can be the time between starting your engine and turning it off. The receiving party can only see a single mean value of your speed over that trip.

Maximum speed: This is the maximum speed you reach during a trip. The receiving party can only see a single value of your speed over that trip.

Fuel consumption: This is how much fuel your car is consuming at that moment. It can be also displayed on your dashboard. The receiving party can potentially analyze your driving behavior since sudden acceleration can be inferred [38].

Throttle position: The throttle is a device controlling the flow of fuel or power to an engine. Since the throttle goes up with acceleration, it is possible to make conclusions about the speed of the vehicle [39].

RPM: The RPM (revolutions per minute) is the amount the engine rotates the crankshafts per minute. The higher the speed, the higher the RPM before the transmission shifts and the RPM goes down again in the next gear. There is a certain correlation between RPM and speed and thus, the receiving party might infer your speeding and acceleration behavior [37, 40].

Steering wheel angle: This is a value which denotes the current position of your steering wheel. For instance, the steering wheel angle is 0° while driving straight and goes up to 90° when you are making a right turn. The receiving party can infer of how aggressively your turns are [37] and it might even be possible to calculate your rough location from it.

Airbag status: This indicates if your airbags are functioning and especially if they should deploy in case of an accident for a front seat passenger. The receiving party can infer if a front seat passenger is traveling with you.

Seat belt status: This indicates which seat belts are fastened in the car. In the case of an accident, it is possible to say if the driver had his seat belt fastened or not (Well, it might be too late for them anyway...). Furthermore, the receiving party can know who wears their seat belt when.

Battery level: This is basically monitoring your battery health. The receiving party can tell if you need to replace your battery.

Tire pressure: This sensor is collecting the pressure you have in each of your tires. In your dashboard, you also have an indicator light which shows up when at least one of your tires is deflated or inflated and not in the correct range to operate.

Hard braking: This indicates the amount of deceleration. If the receiving party is an insurance company for instance, this might look bad for your driving behavior and you might get a more expensive premium [37].

Make/model/year: Self-explanatory. For instance, 2008 Hyundai Elantra. This can be also read out from the VIN which has been explained above.

Fuel level: This is the amount of fuel you have left in your tank. Usually it is calculated as a percentage of the full tank. The receiving party might infer when and how frequently you stop at gas stations since this will result in a spike of fuel level.

Check engine light on: This indicates if there might be something wrong with some part of your vehicle since the sensors report abnormal data. You can also see this light indicator on your dashboard. There might be multiple reasons for it: In a severe case, it might be your engine misfiring (and you want to get your car checked out) or it can just be a loose gas cap. The receiving party can have more information about the specific error and inform you how to proceed [41].

Oil level: This indicates when your next oil change is due. You can also see this light indicator on your dashboard if you should change your oil. The receiving party can monitor how frequently you change your oil and if you are neglecting oil changes until the last second since it can have some negative effects on your engine with time.

Fig. 10 depicts the distribution of times the participants took to complete the survey.

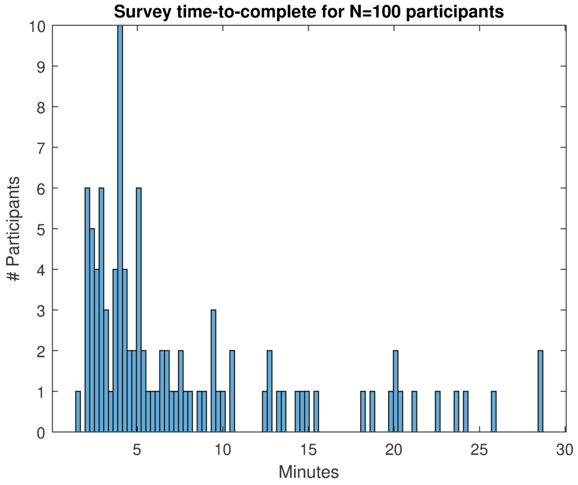


Fig. 10. Histogram of Survey Time-to-Complete

B Dataset Statistics

This appendix includes statistics of the dataset we used in our experiments.

Table 5 summarizes the parameters that are available through the OpenXC platform [42].

Sensor	Units	Frequency
Vehicle speed	km/h	10 Hz
Acceleration pedal	%	10 Hz
Steering wheel angle	deg	10 Hz
Brake pedal	Boolean	1 Hz
Parking brake	Boolean	1 Hz
Torque	Nm	10 Hz
Gear position	Categorical	1 Hz
Engine speed	RPM	10 Hz
Odometer	km	10 Hz
Ignition status	Categorical	1 Hz
Fuel level	%	2 Hz
Fuel consumption	l	10 Hz
Light status	Boolean	1 Hz
Door status	Boolean	1 Hz
Windshield wiper status	Boolean	1 Hz
GPS Latitude	deg	1 Hz
GPS Longitude	deg	1 Hz

Table 5. OpenXC dataset parameters

Table 6 shows statistics of the 58 attack traces.

Table 6. Statistics of used attack traces in our dataset

	Attack traces from dataset (N=58)
Mean	4.28 mi
Median	2.83 mi
Minimum	0.35 mi
Maximum	19.85 mi
Standard Deviation	4.03 mi

Fig. 11 and Fig. 12 depict the histogram and CDF of the attack trace lengths respectively.

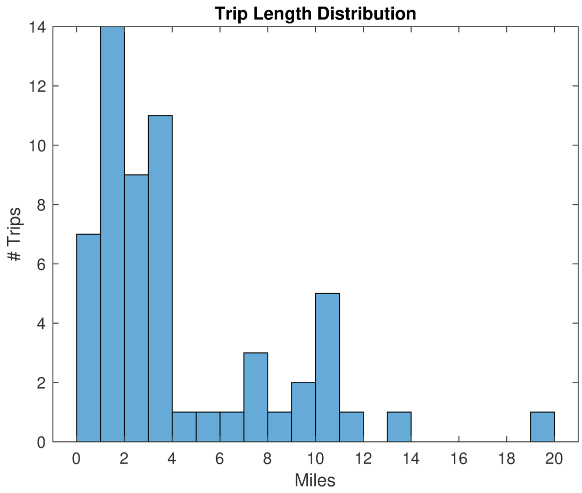


Fig. 11. Histogram of trip length distribution

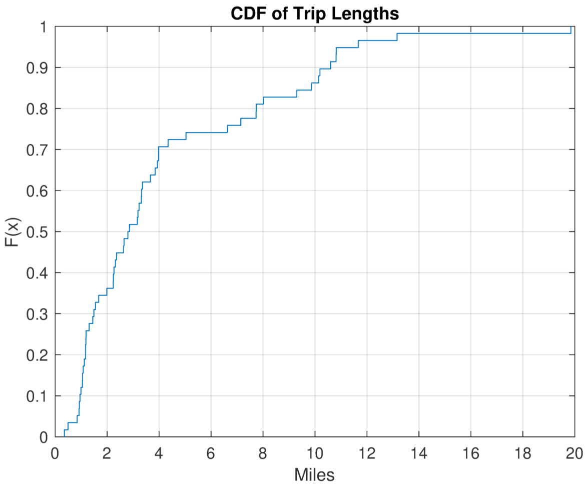


Fig. 12. CDF of trip length distribution

C Road Curvatures in Different Cities

As explained earlier in Sec. 6.5, Fig.13 in this appendix shows the curvature values with their corresponding length of segments (in miles) for the selected eight cities in the US and Europe.

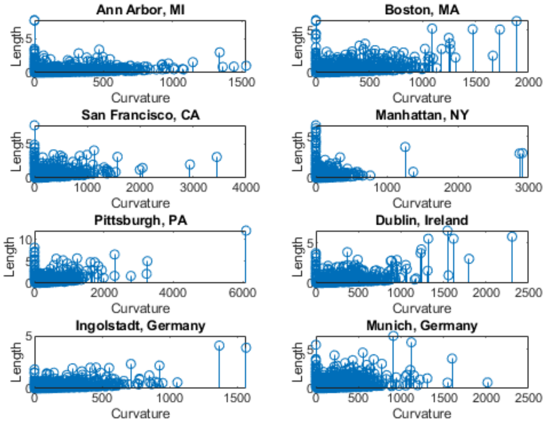


Fig. 13. Curvature values with length of segments for 8 selected cities