Privacy Enabled Noise Free Data Collection in Vehicular Networks

Firstname Lastname

***Abstract*—Preserving the privacy of a user who participates in data collection is very important. Adding noise to the collected data can preserve the privacy but often leads to decrease in utility. If data is collected without adding noise then there would be no decrease in utility. In future, lots of vehicles would be able to communicate with each other and would serve as a potential source of collecting large amounts of location based data. In this paper we look into the aspect how noise free data can be collected in a privacy preserved manner in a vehicular network. We preserve privacy of the user by introducing temporal and spatial variations using Random Delays and Indirections and collect the data in a noise free form. We take advantage of the mobility of the vehicles so that the adversary is unable to localize the users who collected or uploaded the data samples. We rely on vehicle and network simulators for trying different experiments and validating our approaches.**

I. Introduction

Many networked users and devices are interested in participating in distributed sensing and data collection for the purpose of betterment of human society or for earning some monetary rewards. For example, a car driver can report the current state of traffic, speed of vehicle, etc which would be beneficial to build a dynamic congestion map of a city. This information could further be used for mitigation purposes like routing the vehicles through less congested areas of a city thereby avoiding grave traffic congestion situations in a city.

However, these participating nodes are concerned about their privacy and do not want the data collected by them to be associated with their identities. In many cases, a participating node reports its identity, its location, and its measurement (could be radio measurements, car velocity measurements, etc) to a central coordinator which then makes this data available for different applications. Such a data collection system does not preserve the location privacy of the participating users. If an adversary can correlate the data collected by a user with their identity then the adversary can know a lot of things about a user like driving patterns, places visited frequently and can even potentially track them. This is a serious privacy issue and users would not participate in data collection if there are no privacy guarantees. To preserve privacy of users researchers have used the concept of pseudonym ID's [3], group pseudonym and their extensions [8] but all of them have drawbacks as an adversary can still correlate the data collected with individual users. Another approach to preserve privacy is to add noise to the collected data [4] but this often leads to decrease in utility.

Advisor: Dipl.–Ing. Firstname Lastname, Lehrstuhl für Nachrichtentechnik, TUM, WS 2050/2051.

In future more and more vehicles would be able to communicate with each other and upload the data to a central controller. Vehicular networks can serve as a potential source of collecting large amounts of location based data. This kind of data has a wealth of information and can be used in a variety of applications like route planning, travel demand modeling, city infrastructure modeling, emissions and air pollution modeling, etc. The protocol used to make the vehicles communicate with each other is Dedicated Short Range Communication (DSRC). It operates in the 5.9GHz band and its range is 1000 meters [5]. However, the range depends on the type of application and typically lies between 100-300 meters [2]. According to the US standards, vehicles participating in V2V (Vehicle to Vehicle) safety applications belong to Class C vehicles which have a communication range of 400 meters [2].

In this paper we envision a novel approach to collect data in a privacy preserved manner without adding noise to the data samples in case of vehicular networks where the participating nodes are mobile. We focus on noise free form of data collection so that there is no decrease in utility. To solve this, we take advantage of the mobility of vehicles as they move at a fast pace. We introduce temporal and spatial variations using the concept of Random Delays and Indirections in the data collection process itself. In a nutshell, our approach is as follows - the data collector node instead of directly uploading the data to the central controller has an option of forwarding the data to the central controller or to one of its neighboring node with some probability. This is the indirection probability. For each transmission a delay is introduced which is chosen randomly depending on the type of application. The data is finally uploaded to the central controller which in our case is the adversary by some other node after many intermediate delays and hops. The central controller cannot associate the data sample with the vehicle which is currently at the location where the data was collected. This is because the vehicle which collected the data would have moved to a new location of which the adversary has no idea. This approach gives us a two fold benefit of preserving the privacy and getting the data samples in a noise free form.

We demonstrate privacy in terms of the node which collected the data and the node which uploaded the data. Privacy is said to be lost if an adversary can localize a node and can correlate the data sample collected at that location with the node. We model the effect of introducing random delays and indirections on privacy. We consider different traffic scenarios for our experiments and show for what values of random delays and indirections we get maximum privacy.

Talk about experiments and results

For the safety applications the data needs to reach the

neighboring vehicles as soon as possible to avoid any mishap. Therefore we cannot do much in terms of introducing random delays and indirections as the safety of the people is of prime importance and privacy takes a backseat. We focus on the subset of applications where it is fine even if the data reaches to another vehicle or to a central controller after some delay. Some of these applications are - emissions and air pollution modeling, city infrastructure modeling, validating transportation data from other sources, **[add other applications]**

The rest of the paper is organized as follows. Section II discusses the adversary model, Section III contains our methodology of random delays and indirections for preserving privacy. Section IV talks about the privacy metrics. Section V contains details about our implementation and setup of simulations. It also discusses the caveats associated with different ways to cause indirections and different values of random delays. Section VI presents the evaluation and results of our experiments in detail. Section VII talks about related work and finally Section VIII presents future work and conclusion.

II. Adversary Model

We assume that the central coordinator/server is the adversary. The adversary can read all the contents of the data which are uploaded to the server. The adversary also knows which particular user uploaded the data and the time at which the data was originally collected and uploaded to the server. In short the adversary knows the following-

(X, Y, t_0, t_{server}, U, some measurements)

- X,Y - latitude, longitude where data was collected. In the rest of the paper tuple of (X,Y) will be represented by Z.
- t_0 - time at which data was collected.
- t_{server} - time at which data was uploaded to the server.
- U - Unique ID of the vehicle which uploads the data.
- Some measurements- application specific data like RSS values, speed of the vehicle, etc.

The adversary does not knows about the intermediate hops that happen because of the indirections. The adversary is unaware of the random delays which are being added at each hop, it can only calculate the total delay (t_{server}-t_0) when the data is uploaded to it.

III. Methodology

A. Basic Approach

As shown in Fig.1(a) the Data Source (S_0) collects data, say its velocity at time t_0 is v. S_0 uploads its location, collected data and the time at which data was collected to the central controller (adversary). The central controller knows that the data was uploaded by S_0 which is also the data collector in this case. Thus the privacy is S_0 is lost as the adversary can correlate the data uploaded with the unique ID of the source, Id1 in this case.

B. Indirections

Indirections are used to introduce spatial variations. Each node has an option of uploading the data to the server with some probability or forward the data to one of its neighboring node. Let $p(I)$ be the probability of indirections. If $p(I) = 1$ then the node always uploads the data to the server. If $p(I) = 0.2$ then the node uploads the data to the server with probability 0.2 and with probability $1 - p(I)$ it forwards the data to one of its neighbors. Thus the data would reach the server after multiple hops.

Fig.1(b) shows that Id1 is the data source and it collected data at time t_0. It forwards the data to one of its neighbors Id3 due to the indirections probability. Id3 forwards the data which it received from Id1 to Id2. Finally Id2 uploads to the Data Collector which is the central controller or adversary in our case. Thus the data reaches the central controller after multiple hops and the adversary cannot associate this data with the node which uploaded the data. However, each transmission in DSRC is of 100 ms **(VERIFY)**. Therefore the total delay after which the data reaches the central controller is 300 ms (as 3 hops have happened). The distance traveled by the data source i.e Id1 in 300 ms is very less (5.36 meters if traveling at 40 mph) or we can say it is virtually at the same location where it collected the data and the adversary can still localize Id1. Thus privacy is not preserved by only introducing indirections.

C. Random Delay

Random Delays are used to introduce temporal variations. The basic idea is not to send the data directly to the neighboring node rather introduce some delay before sending the data as shown in Fig.1(c). Id1 is the data source and it collects data at time t_0. Id1 then transmits the data to Id3 after introducing some random delay. Id3 does the same and transmits the data to Id2. The data is finally uploaded to the central controller by Id2. For each message hop the delay is randomly chosen between 'x' and 'y' seconds. Because of this, when the data reaches the server the total delay is the cumulative sum of the individual Random Delays which are added at each hop. This total delay give the original data collector, Id1 in this case to travel away from its original position in any direction. Thus the adversary has no idea of who or where the original data collector is. The adversary only knows the original location and time where the data was collected (Section II), but does not know who collected the data.

D. Problem Using Random Delay And Indirections

As the adversary is unaware of the location of the data collector, the adversary would try to localize the data uploader, say U. We assume that all the vehicles are able to communicate with each other. Privacy is said to be lost if the server can localize U.

The server adopts the following strategy to localize U-

- Server calculates the time difference ($t_{server} - t_0$). This is the delay after which the data collected by the source node (S_0) at location Z and time t_0 reaches the server.
- The server calculates the distance that can be traveled in the above time difference at 'v' mph. (Server can find the value of 'v' based on the location where data was collected).

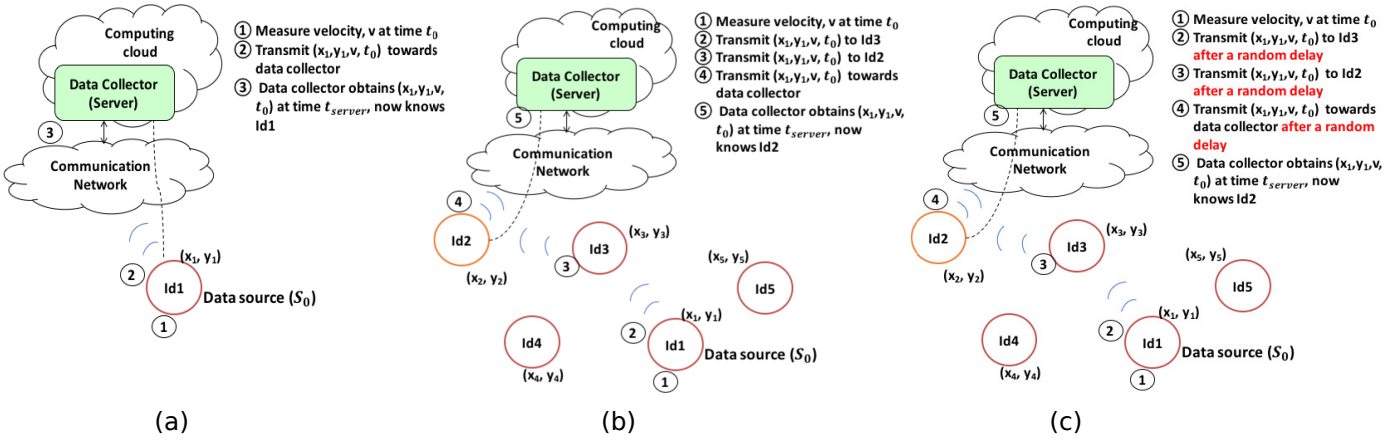

Figure 1. Different Approaches of Data Collection

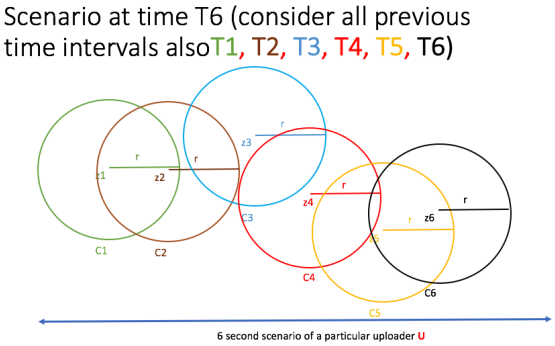

Figure 2. Scenario after 6 data uploads by a particular uploader U

- Distance traveled by a vehicle is $r_0 = v * (t_{server} - t_0)$. We also need to take the range of DSRC protocol into consideration to calculate the value of 'r_0'. If the vehicles are moving in both the directions along a road network then the concept of random walk needs to be taken into consideration as the message can be transmitted in either direction of the traffic. Let us say that 'r' represents the distance traveled by the message packet after taking into consideration speed of vehicle, range of DSRC protocol and the concept of random walk.
- At time t_{server} the data uploader U can be anywhere in a circle of radius r centered at (Z).
- After 1 upload, the server knows that U can be anywhere inside the circle of radius r centered at (Z) and it can be any one of 'k' vehicles which are inside the above circle. If the area of the circle and 'k' are large then we can say that privacy is preserved of the uploader U.

But if the same uploader keeps uploading the data then the scenario changes. For each upload the server draws a circle as shown in Fig.2. When the server receives i^{th} data sample from the same uploader U it draws a circle C_i centered at Z_i. Based on the speed of the uploader and how frequently data samples are being uploaded the circles might intersect also.

Based on the speed of the uploader U and the number of times the upload was made by the same uploader (U) we can calculate the distance traveled by U in t seconds. Let this

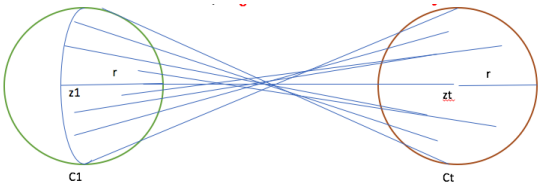

Figure 3. Set of possible points in C_1 which are at a distance of D_U^t from circle C_t.

distance be D_U^t. At t^{th} second, circle C_t would be formed. So the uploader U could have traveled only D_U^t distance from circle C_t to C_1.

To calculate the speed of U when it uploads the i^{th} data sample, the adversary would take into consideration all the road segments which are inside circle C_i. The adversary would take note of the speed limits of the different roads which are inside C_i. The adversary now has a distribution of speeds and can take the value which has maximum probability. This is the speed of the uploader U.

Fig.3 shows the set of possible points in C_1 where the uploader U could have been if at present uploader U is in circle C_t. All the lines shown in the Fig.3 are of distance D_U^t. For simplicity, these lines are shown as straight lines but in reality these lines would lie on the road segments. These lines should pass through all the intermediate circles $(C_2, C_3, C_4 ... C_{(t-1)})$ because the uploader crossed these circles in the past. The general idea is that as U uploads more and more data (more circles will be formed), the number of possible points to reach a previous circle (C_1) keeps on decreasing. This can potentially be a privacy leak as the server would know where the uploader was in the past and can correlate the data received from that location in the past. In this way the adversary can know who the data collector was at any particular time and combine it with the data.

We aim to find for how long an uploader U can safely upload the data i.e. the adversary cannot localize U. As uploader U keeps on uploading the data the set of possible points where U could have been keeps on decreasing. After sufficient time say $(t + m)$ the set of possible points would be very less or below the privacy level set by the uploader

U then it means that the adversary has localized uploader U. The uploader would stop uploading the data at time $(t + m)$ so that it does not looses it's privacy. We would set the values of Indirection probability and Random Delays in such a way that the uploaded can uploaded the data for a longer period of time without compromising on its privacy.

IV. Privacy Evaluation Metrics

- **K-anonymity:** This is the most widely used privacy metric. It means that an individual can be any one of the 'k' people in a set [10]. The larger the value of 'k' is, more is the privacy guaranteed. In our problem 'k' is the number of different vehicles, so the uploader can be any one of the 'k' vehicles.
- **Area of region of interest:** With continuous uploads the area of consideration for the adversary keeps on decreasing. Based on the data received the adversary draws a circle where an uploader can be. Thus, after many uploads the area of the region becomes small thereby causing privacy issues.

Each user can define his/her own privacy level using the parameter ϵ. Lower the value of ϵ more is the privacy level. Based on the value of ϵ the value of 'k' in **k-anonymity** and **area of region of interest** also changes.

V. Experimental Setup

A. SIMULATIONS

We used simulators to simulate the urban mobility environments where the vehicle move along the road segments, obey traffic rules, etc. We also need to make the vehicles communicate with each other. The data needs to be sent to the neighboring nodes which are in the range of the vehicle. We follow DSRC protocol which is designed for making the vehicles communicate with each other. Next we need a framework to make the network and vehicular simulator talk to each other. We are using SUMO for vehicular simulation, OMNeT++ for network simulation and VEINS framework to make vehicular communication possible.

SUMO: SUMO stands for Simulation of Urban MObility[6]. It is used for vehicular simulations. It is equipped with a large variety of rich features, some of them are supporting a large number of fleet vehicles, traffic lights simulations, map generation, support for Open Street Maps, etc. **It also has support for different car folloiwing models like Krauss, Wiedemann and lane changing models.**

OMNeT++: OMNeT++ is an event based network simulator [1]. It is used for modeling both wired and wireless communication networks, protocol modeling, queueing networks,etc. OMNet++ provides infrastructure and tools for writing simulations.

VEINS: Veins is a framework used for vehicular network simulation [9]. It interacts with OMNeT++ and SUMO in parallel which are connected via TCP socket. As the vehicles move in SUMO the updates of their positions are made in the OMNet++ simulator as movement of nodes.

To test our approach of Random Delays and Indirections we set up a custom map as sown in Fig.4. Each road segment

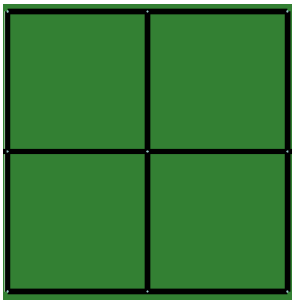

Figure 4. Area under consideration

is 500 meters in length and has 3 road lanes on each side. Thus the total area under consideration is 1km X 1km. To make the simulations realistic, traffic lights are present at the intersections. The car following Krauss model [7] with some modifications which is present in SUMO is used. In this model vehicles are allowed to travel as fast as possible while maintaining proper safety between them. The maximum speed of the vehicle is set to $25m/s$, acceleration is $0.8m/s^2$, deceleration is set to $4.5m/s^2$, the driver imperfection parameter sigma is set to 0.5. SUMO's default lane changing model **cite** is used for switching between the lanes. The transmission distance is set to 150 meters which means only those vehicles which are at a distance less than 150 meters will receive the message. **should we emphasize that our transmission range is low and we can increase it also?**

The indirection probability (ID) is set to 0.28 and random delays (RD) are chosen uniformly between 4 to 10 seconds. However, we would like to give the Data Source sufficient time to move away from it's current location before the data reaches the central controller. To achieve this we force the collected data sample to remain in the network for at least 15 seconds before it can be uploaded to the central controller. Thus, the indirection probability (ID) varies as following-

$$ID = \begin{cases} 0 & \text{Message lifetime} \leq 15 \\ 0.28 & \text{Message lifetime} > 15 \end{cases}$$

where Message Lifetime is the difference between current time and message creation time.

Depending on application type we can also increase the Indirection Probability (ID) if a particular message packet stays in the network for a large period of time. Choosing ID is very tricky because if we set the value of ID very low then the data sample might not reach the server and would just keep hopping in the network. If we set the ID very high then the data sample might reach the server in 1-2 hops thereby affecting privacy.

With the above set of parameters we ran a simulation for 12 minutes. New messages were generated every second and all the vehicles are able to communicate with each other. Around 20-25 vehicles were present during the course of simulation entering and exiting the map at different times and traveling on different routes. To study the privacy, let us assume that the adversary tries to track a particular vehicle. In our simulation this vehicle is named '1-id '. This particular vehicles stays in the road network as shown in Fig.4 for the complete simulation

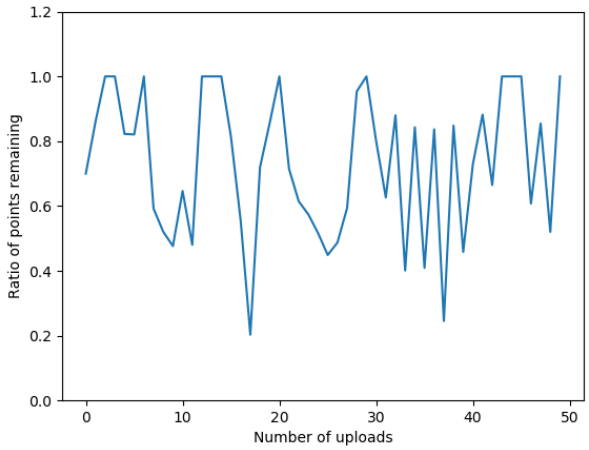

Figure 5. Ratio of remaining points

duration. During this time duration 1-id uploaded 50 data samples.

Algorithm 1 Adversary approach

procedure ADVERSARY
  uploads ← Upload number corresponding to a particular uploader
  V_S ← velocity of data source at location Z
  V_U ← velocity of uploader (drawn from a distribution of speeds based on speed limits of the roads in the vicinity of Z)
  t ← Current time − Previous message upload time
  D_U^{i-1} ← Distance traveled by uploader 'U' between i^{th} and $(i-1)^{th}$ upload i.e. between two successive uploads
  $C_{uploads}$ ←Circle of radius 'r' based on V_S
  $K_{initial}$ ← Number of possible points in circle ($C_{uploads}$)
  K_{common} ←$K_{initial}$
  for $i = uploads, i >= 2; i--$ **do**
    Calculate D_U^i
    $PossiblePoints^{i-1}$ = Set of possible points in (C_{i-1}) which are at a distance of D_U^i from (K_{common}^i)
    K_{common}^{i-1} be the number of points in set $PossiblePoints^{i-1}$
  Ratio = $K_{common}/K_{initial}$
  if $Ratio <= threshold$ **then**
    Localized Uploader U
    break

The adversary adopts the approach as described in Algorithm 1. Based on the total delay after which the message reaches the adversary, it draws a circle of radius 'r'as explained in Section III(D). The set of points which are inside this circle are referred as $K_{initial}$. In our case, after 50 uploads the adversary would have drawn 50 circles and have 50 values of $K_{initial}$ corresponding to each circle. Based on delay between upload 50 and 49, adversary would select a subset of points in C_{49} which are at a distance of D_U^{49} from the set of points in C_{50}. These set of points are stored in K_{common}^{49}. From the set of points in K_{common}^{49} adversary finds the set of those points in C_{48} which are at a distance of D_U^{48} and stores them in K_{common}^{48}. This process continues till K_{common}^{1} is reached. The adversary then calculates the ratio between K_{common} and $K_{initial}$. If the $Ratio$ is lower than a $threshold$ then we can say that the adversary has localized uploader U. For our case Fig.5 shows the ratio for 50 uploads made by the same uploader. From the figure we can see that the minimum value of ratio is 0.2, maximum value is 1,the mean and median value are 0.73 and 0.76 respectively. Corresponding to the minimum value the set of points are 95, where each point is 5 m apart. Thus the uploader can be in one of these 95 points.

VI. Evaluations and Results

Result goes here

VII. Related Work

related Work goes here

VIII. Conclusion

Conclusion goes here

References

[1] OMNET++ simulator. https://www.omnetpp.org. Accessed: 2018-02-08.
[2] Fan Bai and Hariharan Krishnan. Reliability analysis of dsrc wireless communication for vehicle safety applications. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 355–362. IEEE, 2006.
[3] Daniel Da Silva, Tracy Ann Kosa, Steve Marsh, and Khalil El-Khatib. Examining privacy in vehicular ad-hoc networks. In *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 105–110. ACM, 2012.
[4] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.
[5] Jinhua Guo and Nathan Balon. Vehicular ad hoc networks and dedicated short-range communication. *University of Michigan*, 2006.
[6] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.
[7] Stefan Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, 1998.
[8] Krishna Sampigethaya, Mingyan Li, Leping Huang, and Radha Poovendran. Amoeba: Robust location privacy scheme for vanet. *IEEE Journal on Selected Areas in communications*, 25(8), 2007.
[9] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011.
[10] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.