VTL Zone-Aware Path Cloaking Algorithm

Bin Zan Peng Hao* Marco Gruteser XueGang Ban*
WINLAB, Rutgers University
671 Route 1 South, North Brunswick, NJ 08902-3390
{zanb, gruteser}@winlab.rutgers.edu
*Rensselaer Polytechnic Institute
*110 8th St. Troy, NY 12180-3590
*{haop, banx}@rpi.edu

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

I. Introduction

Location traces obtained through the Global Positioning System (GPS) are a promising source for extracting many types of transportation data. GPS traces from taxi fleets, delivery trucks, or mobile phones, for example, are already used to infer traffic congestion on highways [1]. This concept of estimating traffic system states from location traces generated by mobile sensors in individual vehicles, promises substantially lower costs, since it does not depend on infrastructure installed along roadways [2]. Within this concept, many other transportation applications are possible. Throughout this paper, we will consider one motivating application which raises novel privacy challenges. This application focuses on state/performance estimation of signalized road intersections such as estimating real time delays, arrival volumes, and vehicle queue lengths. This has been a long-standing challenge in the transportation community especially when wide-area arterial networks are considered.

Collecting location traces raises many privacy concerns as discussed in the literature [3], [4]. At first glance, this can be addressed through established anonymization techniques or the techniques of changing pseudonyms since this particular application does not depend on vehicle identities. Recognizing that naively anonymized (i.e., simply omitting names, vehicle identifiers, etc.) location traces can often easily be re-identified, researchers have proposed several solutions [5], [6]. Spatial cloaking based on k-anonymity [7], [8] was not designed for a high update rate and also tends to modify the location traces so substantially that it cannot meet the accuracy requirements of such transportation applications. A Mix zone [9], [10] defines a particular area where locations cannot be revealed so that an adversary cannot trace the movements of vehicles across these mix zones. This works well when the traffic density is high, but provides no guarantees when there is less traffic than anticipated. Furthermore, most mix zone and related research [11], [12] have focused on finding the best location and size of zones for protecting privacy. However, our target application also introduces strict requirements on this location: data is only of interest around intersections. The uncertainty-aware path cloaking algorithm [4] also segments traces but was explicitly designed to achieve a defined level of privacy under all traffic conditions. It achieves this by filtering out points from the location traces whenever vehicles are trackable for a longer period of time, thus it provides no control of where data is omitted. The challenge in our motivating application lies in the requirement that location traces are needed only across intersections–exactly the area where mix zones are often placed to hide information. While there are some safety critical applications that require data virtually everywhere, for most applications, data should only be provided in the limited areas where it is needed. This is also consistent with the basic principle of minimizing the information leakage.

In this paper, we develop a zone-aware privacy algorithm to filter location traces, which still takes into account traffic density and uncertainty. This allows the algorithm to release location traces only in the intersection zones where data is needed by the application, yet still offer a fixed degree of privacy independent of traffic density. This is, to our knowledge, the first algorithm that combines these two aspects. More specifically, the algorithm seeks to achieve unlink-ability between released traces from any two different zones. We refer to the zones as VTL zones in the remainder of the paper, since their locations can be marked by two or more Virtual Trip Lines (VTL) [1]. To be able to adjust the filtering to traffic density, the algorithm needs to be aware of all vehicles' location traces. While we will simply refer to a proxy server with access to these traces, we note that there are multiple different usage scenarios for such an algorithm. Even if no proxy server exists, the algorithm could be useful to anonymized data before it is stored or before it is transmitted

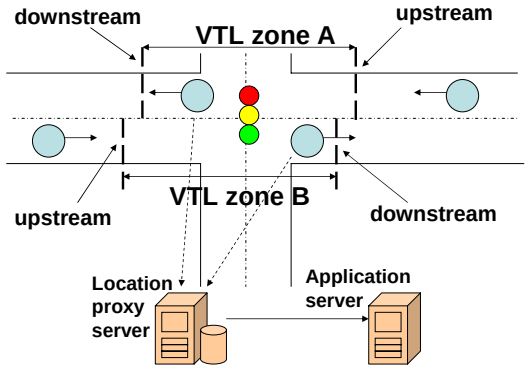

Fig. 1: System architecture

to another party.

The rest of the paper is organized as follows. In Section II, we describe our system, adversary model. We describe the main zone-aware path cloaking algorithm in section III. In section IV, we evaluate the performance through simulation. Section V discusses the distributed solution and concludes in section VI.

II. VTL Zones and Adversary Model

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

A. VTL Zone

A virtual trip line (VTL) zone is an area of the road network between two virtual trip lines. Virtual trip lines were originally introduced to allow an application to specify where nodes (e.g., vehicles) should provide location updates. The nodes would send a location update only when they cross this virtual trip line. The VTL zone extends this concept to a continuous part of the road network. In this model, mobile nodes will provide their entire location trace between two virtual trip lines. Figure 1 illustrates this concept with an intersection example. In this example, the intersection performance monitoring application benefits from receiving a longer location trace before the traffic light and requires less data after cars have passed this traffic light. For this reason, the application deploys separate VTL zones A and B for the two lanes of the road. Not shown in this example are additional trip lines that could be deployed on the crossing road.

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

B. System Model

Similar to the assumption in [4], we consider a model where all location updates from individual vehicles are collected by a centralized server. The confidentiality and authenticity of the transmission between vehicles and the server can be guaranteed with cryptographic methods. Since the applications we consider do not require any identity information, this server will remove all identifiers, for example, user IDs or license plate numbers. It will also further sanitize the location records through a cloaking algorithm (described in section III) to enhance their anonymity before it passes the location traces to applications.

At first sight, anonymization and cloaking of data may seem unnecessary in this scenario. If we consider the server trusted, then the application could be directly executed on it with the raw data. If it is not trustworthy, then an adversary could still get access to the data before it passes through the anonymization steps. In practice, however, we believe that this centralized anonymization approach is still a very useful model since the issues tend to be more subtle. First, the location update data often needs to be archived to allow extraction of long-term trends and facilitate debugging and development of improved application algorithms. An organization that collects data, may also want to share it with outside researchers or third-party application providers. Such archiving and sharing increases data breach risks and legal compliance costs, which can be alleviated through centralized anonymization [1].

C. Adversary Model

The objective of the adversary is to track the vehicles in the road network and eventually re-identify them, which compromises the location privacy of the drivers. The longer an adversary can track a vehicle, the higher the chance of re-identification and the higher the probability that the trace will contain sensitive location information. We assume that the adversary has gained access to a dataset that has passed through the anonymization steps. Access may have been obtained through insiders, subpoenas, or remote systems compromises, for example. We also assume that the adversary has an understanding of road traffic flows in the area of interest.

We further assume that VTL zone are small enough so (e.g., covering only one intersection) so that the trace from one VTL zone alone is not a privacy concern. However, if the adversary can link traces from multiple VTL zones to the same vehicle, it can reconstruct longer potentially sensitive trajectories.

Since the adversary cannot verify which guesses were correct, such guessing has little value if the confidence in such guesses is small (i.e., uncertainty is high). We measure uncertainty through the entropy $H = -\sum p_i \log p_i$, where p_i is the probability that two GPS traces from two VTL zones belong to vehicle i. Lower values of H indicate more certainty or lower privacy. We will refine this concept in the following algorithm description.

III. The VTL Zone-Aware Cloaking Algorithm

The goal of this cloaking algorithm is to eliminate linkability between any VTL zone pairs *regardless of traffic density* while keeping as much trace data as possible to be used

[1]Such centralized anonymization has already been adopted by Google [13], which indicates a commercial need for such technologies.

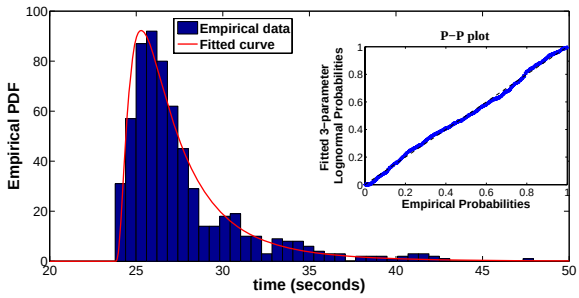

Fig. 2: A 3-parameter log-normal distribution fits the empirical data well.

by the applications. This means, that the adversary should be unable to track an individual user through multiple VTL zones even if the traffic density becomes very low. It also means that when increased traffic density leads to a naturally higher level of privacy, the algorithm should filter less information.

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

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

A. Travel Time Likelihood

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

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

[2]Similar observations are also shown in [14]. Here, we emphasize using 3-parameter log-normal to describe the travel time distribution instead of general log-normal distribution.

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

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

where θ is the threshold parameter, σ is the shape parameter and ζ is the scale parameter. The 3-parameter log-normal distribution can be fitted by Least Square Estimation (LSE) as shown in [15] in which the cumulative distribution function of 3-parameter log-normal

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

is used as input function for LSE:

$$\hat{b}_{opt} = \underset{b \in B}{\operatorname{argmin}}\, S(\hat{t}_1, ..., \hat{t}_n | b) \quad (3)$$

$$S = \sum_{i=1}^{n} w_i.(F_T(\hat{t}_i|b) - v_i)^2 \quad (4)$$

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

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

B. Path Likelihood

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

$$\rho_{a\to b} = \frac{\sum_{d\in D} k^d_{a\to b}}{\sum_{d\in D} k^d_a} \quad (5)$$

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

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

$$\rho_{a\to b} = \beta * \rho_{new} + (1-\beta) * \rho_{old} \quad (6)$$

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

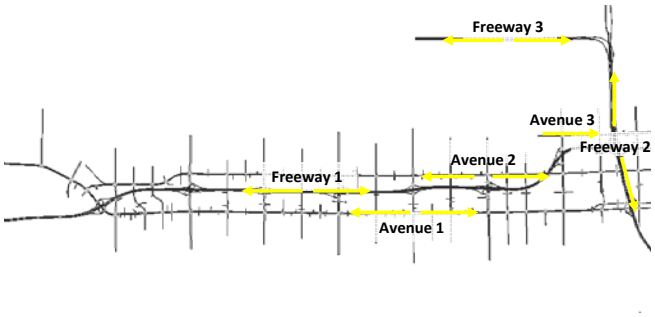

Fig. 3: On this map, total 102 VTL zones are deployed for the MSTP application [2].

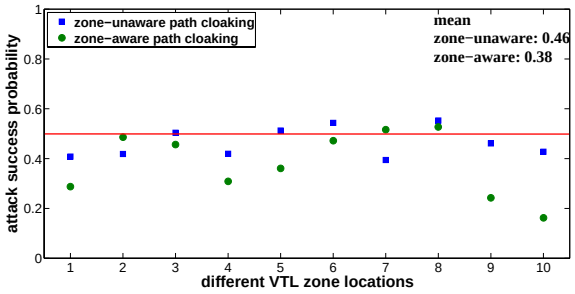

Fig. 4: The adversary successfully identify the trace at almost same probability under two privacy algorithms.

C. Trace Release

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

$$H = -\sum_{v \in V \setminus c} \sum_{d \in D_v} p^d_{v \to c} \log p^d_{v \to c} \tag{7}$$

where V is the whole VTL zone set, D_v are the vehicles which have their latest disclosed data trace in VTL zone v. As shown in equation (8), $p^d_{v \to c}$ is the probability that the trace in c is generated by the same vehicle as trace d in zone v (after normalizing)

$$p^d_{v \to c} = \frac{\rho_{v \to c} * p_T(t^d_{v \to c})}{\sum_{v' \in V \setminus c} \sum_{d' \in D_{v'}} \rho_{v' \to c} * p_T(t^{d'}_{v' \to c})} \tag{8}$$

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

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

$$H_m = H - \log m \tag{9}$$

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

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

IV. Simulation and Evaluations

In this section, we evaluate the proposed VTL zone-aware path cloaking algorithm through simulation. The tracking uncertainty α is set to be 0.95[3]. We compare the proposed VTL zone-aware path cloaking algorithm with a zone-unaware path cloaking algorithm from Hoh et al. [4]. First we compare both algorithms based on the privacy level can be achieved under same adversary model. Next, we compare the data quality obtained through different privacy algorithms. And finally, from analyzing the simulation results, we also show computation overhead can be largely reduced by setting proper time threshold.

A. Traffic Simulator

The simulation is based on the traffic data generated from Liu and Jabari's Paramics Traffic Simulation model [17]. As shown in Fig.3, timestamped location traces are collected from a sub-network of the SR41 corridor located in the city of Fresno, CA. The corridor comprises a stretch of the SR41 freeway and three parallel arterials, with a total of over 90 signalized intersection and 15 ramp metering controllers. It is approximately 16 miles in length and 4 miles in width. Overall, the network includes 20 arterials and 3 freeways. We marked VTL zones on the three avenues shown in Fig.3 for all signalized intersections, which yields a total of 102 VTL zones. Data are collected for about 1 hour. We have implemented the algorithms in Java (except for the extensions described by equations 6 and 9). On one Intel(R) Xeon(R) 2.66GHz core, the simulation with one data set (full density of 1 hour data) takes about 13 minutes to run (without optimization).

B. Privacy Results

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

The target of that adversary is to identify the records from the next directly connected VTL zone. As shown in Fig.4, two

[3]To see the impact of different uncertainty levels, please refer to our paper [16].

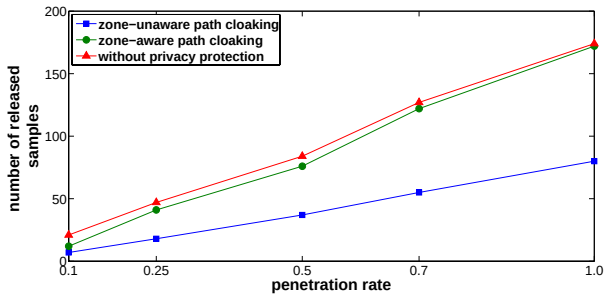

Fig. 5: The proposed zone-aware algorithm releases more samples than zone-unaware algorithm does.

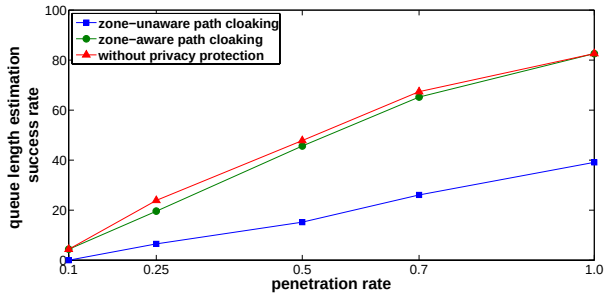

Fig. 6: The proposed zone-aware algorithm which has more sample being released results in better performance on a traffic monitoring model (MSTP [2]).

schemes results in similar attack success probability over ten different VTL zones. In terms of mean value, the zone-aware algorithm is slightly better.[4]

C. Data Quality Results

Fig.5 shows that the proposed algorithm releases more samples than the zone-unaware algorithm does. Note, that we only count location samples inside the VTL zones, which matter to the application. Overall, the amount of released location samples is much closer to the ideal case, labeled 'without privacy protection' in the figure, where no samples are suppressed. The penetration rate indicates the density of the traffic condition. For penetration rate less than 1, we generate the simulation data by random keeping vehicles based on the rate from the original (penetration rate=1.0) data.

The increased amount of available data also leads to improved traffic monitoring application performance, at least in the real time traffic signal queue length estimation application [2] we study here. As shown in Fig.6, the success rate[5] is high and close to ideal with the results from the proposed scheme. Due to the smaller number of released samples, the application performs less well with data cloaked by the zone-unaware algorithm. For example, at full penetration rate (1.0),

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

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

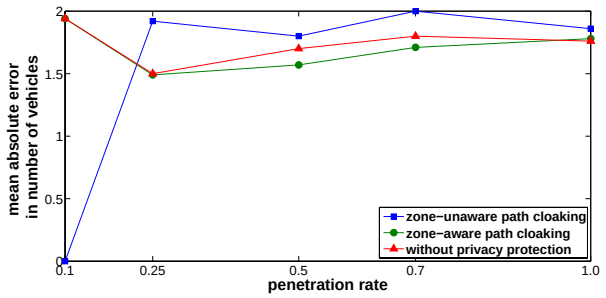

Fig. 7: The output from the proposed zone-aware algorithm results in small mean absolute error than the output from the zone-unaware algorithm.

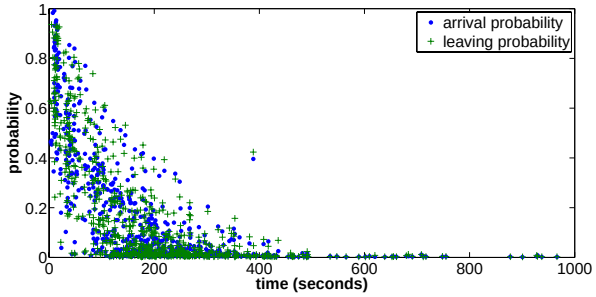

Fig. 8: When the mean traveling time between two VTL zones is larger than 15 minutes (900 seconds), the arrival probability is less than 0.01 and the leaving probability is less than 0.013.

the proposed zone-aware algorithm increases the success rate from 39% to 82% over the zone-unaware algorithm. We also observed a slight improvement in the mean absolute error with the zone-aware algorithm as shown in Fig.7. Surprisingly, this occasionally result in slight improvements even over the ideal uncloaked data set, likely because the privacy algorithm also removed outliers that affected the application.

D. The Impact of Travel Time Threshold

As shown in Fig.8, when the mean travel time between two zones is large, the arrival probability and the leaving probability are both very small. Here the arrival probability is defined as the conditional probability that a vehicle leaving from zone A will arrive at zone B while the leaving probability is defined as the conditional probability that a vehicle arriving at zone B is from zone A. Thus, in our cloaking algorithm, we do not consider zone pairs with more than 15 min mean travel time between them. This reduces the zone pairs that need to be evaluated by more than 90%—from 10302 to 917 pairs—and therefore significantly reduced the computational complexity.

Further study even shows that the ratio of disregarded extra samples is less than 0.05 when ignore all the cases that have more than 5 minutes travel time between two zones. However, the computational complexity is reduced by more than 80% as shown in Fig. 9. On the other side, it is easy to see that ignoring zone pairs that have large distance has positive impact

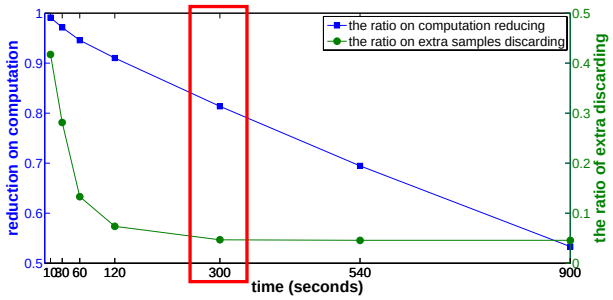

Fig. 9: Computation overhead can be saved by more than 80% if we ignore zone pairs that have travel time more than 5 minutes.

on privacy.

V. Discussion

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

Distributed Algorithms. The proposed solution assumes that the cloaking algorithm can be run in a centralized location as shown in Fig.1. It is also possible to remove the centralized location proxy server, and distribute the proposed algorithm on to the vehicles. For example, the application server could be used to compute the travel time distribution based on collected samples. It could also calculate the path likelihood. Every individual vehicle can locally estimate the overall probability and the entropy value to decide if it should release the sample to the server or not. Two issues need to be carefully considered. First is privacy, as many necessary information are needed from the application server for a individual mobile node to compute its entropy, the directly query process itself might introduce privacy leakage. The second issue is the computation and communication overhead introduced by the local computing. One way to address the privacy concern is by exploiting short-range communications between vehicles, as used for example in the geocache protocol [18]. Furthermore, by optimizing the time threshold as discussed in the last section, computation and communication overhead introduced by distributed computing can also be reduced.

VI. Conclusion

We have presented a virtual trip line zone-based path cloaking algorithm. Our proposed algorithm can reduce linkability between any VTL zone pair and minimize the number of trace samples that have to be removed to preserve location privacy. Simulation results show that the proposed algorithm significantly outperforms a zone-unaware cloaking algorithm in all kinds of traffic densities, and it increases the success rate of MSTP model, the targeted application, from 39% to 82% under full penetration rate. Note, our current work does not consider any background knowledge known by the adversary which may increase the ability of adversary to break privacy protection. We plan to study this issue in the near future.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[18] B. Zan, T. Sun, M. Gruteser, and Y. Zhang, "The boomerang protocol: Tieing data to geographic locations in mobile disconnected networks," in *Proceedings of IEEE International Conference on Mobile Data Management*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 258–263.