Privacy Aware Traffic Monitoring

Hairuo Xie, Lars Kulik and Egemen Tanin

***Abstract*—Traffic monitoring systems are vital for safety and traffic optimization. However, these systems may compromise the privacy of drivers once they track the position of each driver with a high degree of temporal precision. In this paper, we argue that aggregated data can protect location privacy while providing accurate information for traffic monitoring. We identify a range of aggregate query types. Our proposed *Privacy Aware Monitoring System (PAMS)* works as an aggregate query processor that protects the location privacy of drivers as it anonymizes the IDs of cars. Our experiments show that PAMS answers queries with high accuracy and efficiency.**

***Index Terms*—Privacy, Spatial data structures, Road transportation, Road vehicle location monitoring, Road vehicle identification**

I. Introduction

DUE to the growing complexity of modern transportation networks, traffic monitoring systems (TMSs) have received strong attention in many countries. TMSs collect statistical and real-time data to keep travelers safe and traffic efficient. For example, the ROMANSE project [1] uses various technologies to collect traffic data. The public can access this data, e.g., via the Internet, to query the traffic status in a certain area.

Protecting the location privacy of drivers is an important issue in TMSs. Failure to protect location privacy may cause location-based spam and threats to personal safety [2]. One major privacy concern stems from the identification of cars. In some systems, people can access raw data, such as real-time images from road cameras [3], enabling an adversary to determine the identities of vehicles. This problem is amplified if ID information is given to end users.

We argue that a TMS should not give end users access to information that can be used to identify vehicles. Instead, the system should only reveal *aggregated data*, i.e., summarized information from a number of locations for more than one car. For our system, aggregated data is a simple numeric value, e.g., traffic volume in an area. We call a query that asks for aggregated data an *aggregate query*. A common aggregate query in a TMS is: *what is the average daily traffic on a road measured over one year?* An adversary cannot extract personal information, such as the IDs of vehicles, from aggregated data. For stronger privacy protection, a TMS should not collect the true identities of vehicles. We propose a Privacy Aware Monitoring System (PAMS) that solves a range of aggregate queries without the need of true identities. Instead, PAMS collects *short IDs* that cannot be linked to full IDs during monitoring. The use of such *artificial IDs* has been recognized as an approach to protect privacy in a TMS [4].

Our system is built on spatial histograms that keep summarized information, e.g., counts of cars, at adjacent sensing locations. In our previous work, we proposed the *Distributed Euler Histogram (DEH)*, which uses simple counts to answer aggregate queries [5]. Although for total privacy protection DEH is a good choice with certain queries, it may not achieve good accuracy for some of the queries shown in this paper. This motivated us to design an extension, *Euler Histogram based on Short ID (EHSID)*, which still stores counts but bundles the counts with partial ID information.

This paper has three main contributions. First, we develop a range of aggregate queries to monitor traffic more accurately at a finer level of detail. Second, we introduce DEH-based data structures for TMS, as they allow a high degree of privacy in traffic monitoring. Third, we introduce PAMS, which is based on DEH-based data structures to answer aggregate queries with a high degree of accuracy and efficiency.

The remainder of this paper is organized as follows. The related work is described in Section II. We introduce aggregate query types for PAMS in Section III, the DEH in Section IV, and PAMS in Section V. Section VI shows the experimental results of DEH and PAMS. We conclude in Section VII.

II. Related Work

A. Safeguarding Privacy for RFID Technologies in TMS

Identification of vehicles is a frequent task in transportation systems, which is often based on image processing [6]. Our proposed system reidentifies vehicles using partial information from *Radio Frequency IDentification (RFID) tags*. Many applications in transportation networks use RFID technology. The Houston TranStar Automatic Vehicle Identification (AVI) TMS collects traffic data, such as average travel time and speed, based on *transponder tags* [3]. *Electronic Toll Collection (ETC) systems* also use RFID tags to recognize vehicles [7] and adjusts fees depending on the actual usage of roads [8]. By analyzing the waveform from RFIDs carried by passengers, the distance between bus stations and passengers can be estimated and the public transport services can be improved [9]. As RFID tags are promiscuous, i.e., simple tags reply to any reader that interrogates them, RFID tags may expose the details of vehicles and drivers. Thus, we require techniques to protect privacy in RFID-based systems. A *blocker tag* protects privacy by simulating a part of the spectrum of RFIDs [10]. Whenever a RFID reader accesses the tags within the spectrum, the blocker tag responds to the reader and causes the reader to stall. Thus, the actual RFIDs cannot be singulated and the privacy of those RFIDs is protected. Other research suggests to use *pseudonyms* as the temporary IDs of vehicles in traffic

Manuscript received

Hairuo Xie, Lars Kulik and Egemen Tanin are with NICTA Victoria Laboratory and the Department of Computer Science and Software Engineering, University of Melbourne, Victoria 3010, Australia (email: hrx, lars, egemen@csse.unimelb.edu.au). We thank Muhammad Umer for his review of this paper. This work is partially funded by ARC Grant DP0880215.

monitoring [11]. Hon et al. proposed an architecture that stores real identities separately from other data [12]. Langheinrich et al. introduced a method that uses multiple miniature RFID tags to distribute the original RFID [13]. However, to the best of our knowledge, there is a lack of research focusing on safeguarding privacy for RFID in TMS. We propose a TMS that prevents the disclosure of true identities associated with RFID tags while enabling a large range of new query types.

B. Privacy Protection in Sensor Networks

Similar to many TMSs, our system uses sensor network technologies to collect traffic data. Insufficient protection of privacy in sensor networks allows an adversary to monitor objects anonymously and remotely [14]. Although strong authentication and encryption aim to prevent eavesdropping of communication [15], they do not directly address the privacy issue of collected, personal data. Most research follows two directions: *data cloaking* and the use of *pseudonyms*.

Data cloaking can be achieved by removing details or adding perturbation to the data. Gruteser et al. showed how to protect location privacy by cloaking the location information [16]. In their approach, location information is blurred when counts of objects at a certain location do not reach a predefined anonymity level. In this way, the privacy of objects at that location is protected but the system can still give accurate counts of objects in large areas covering the location. Data cloaking is not applicable in our scenario as it may introduce substantial inaccuracy to the answers of our queries.

In the case that precise data must be transmitted through a network, using pseudonyms can help to protect privacy. For example, Misra et al. developed two approaches that allow anonymous communication between sensor nodes using pseudonyms [17]; Ouyang et al. showed the usage of pseudonyms that are generated from keyed hash functions [18]. Different to our approach, these works do not address aggregate queries.

III. Aggregate Query Types for PAMS

In this paper, we focus on the most common measure of road usage: traffic volume, which is often defined as the count of vehicles that pass through a road section or a position during a period [4]. Despite its broad usage, this way of measuring traffic volume is less suitable for certain applications.

Traffic volume is frequently used for estimating recreation activities. However, based on a study of the United States Forest Service, traffic volume usually provides the least precise information among a range of data sources [19]. This is partially due to the fact that traffic volume often contains redundant counts from returning entries of the same visitors. The proper adjustment of the counts is difficult because the variation of traffic conditions can be high between locations and seasons. Another study also shows the unreliability of traffic volume for estimating recreation visits [20]. The authors compared the traffic volume data from two systems. One system counts all incoming vehicles to the recreation area. The other system only counts short-term visitor cars. The study found that the gap in the results is tremendous between these systems and neither of them is reliable.

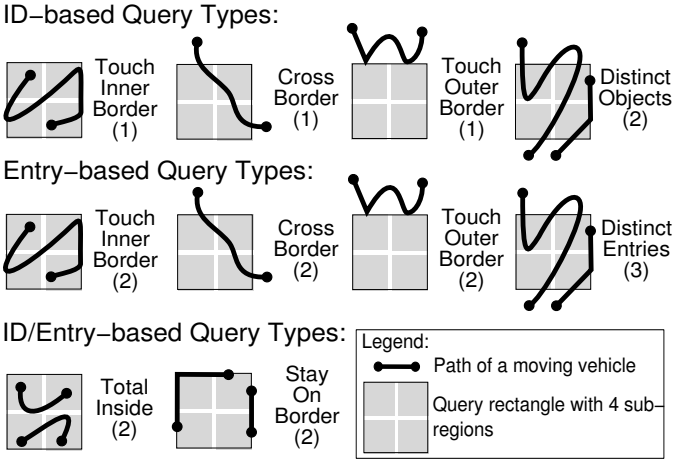


Fig. 1. Queries solved by PAMS.

Although traffic volume can be often represented by the counts of all vehicles, counts of vehicles with certain trip characteristics are more adequate for some applications. For example, Wilmshurst et al. collected traffic volume of objects that traveled along the road, crossed the road, entered the road and left the road [21]. Based on this data, installation of pedestrian facilities was proposed to improve the protection of pedestrians and vehicles. In another approach, counts of trips to and from a downtown area are collected to estimate travel patterns [22]. Since existing TMSs do not automatically collect trip information, transportation authorities use surveys as the primary source for trip-based traffic volume [23]. However, this method has many limitations, such as undercounting of trips [24] and data collection costs.

In order to collect the traffic volume mentioned above, a TMS has to reidentify vehicles. Reidentification avoids redundant counts from returning entries and irrelevant vehicles. We show that full ID information is not necessary to reidentify vehicles. Instead, our system PAMS only collects *short IDs* that have no link to the real IDs. This still allows the TMS to reidentify vehicles while protecting the privacy of drivers.

We distinguish three types of queries that ask for traffic volume. These query types result from the complete set of spatial relations between a line and a region [25], where the line represents the path of a vehicle and the region the query region. We generalize basic vehicular movement patterns from these relations. Each query asks for the counts with respect to a particular movement pattern in the query region. Figure 1 shows the queries. Each subfigure has a query rectangle consisting of 4 sub-regions. The traffic in each sub-region is monitored independently. The number under the name of each query is the answer for that query, given the path of moving object in the corresponding subfigure.

A. Using the Aggregate Queries in Traffic Monitoring

Aggregate queries can be independent or combined queries. For example, a road authority may require *the number of entries to a suburb in the last hour* by issuing a *Distinct*

Entries query with the suburb as the query region. For a combined query assume a scenario where traffic is slow on the main beltway around a city. There is no obvious reason for the slowdown such as an accident. Since the beltway could be used as an intermediate path by many vehicles, the road authority is interested in the aggregated traffic data in the last 30 minutes: *how many cars touched the beltway (boundary of the query region) from the outside*, *how many cars touched the road from inside*, *how many cars crossed the road* and *how many cars stayed on the road*. The authority could send the corresponding queries described in the following sections to investigate the main reason for the slowdown. We detail the query types in the subsequent sections.

B. ID-based Query Types

ID-based queries collect the volume of unique vehicles. We assume that each vehicle has a unique ID encoded in its RFID tag. Hence, counting unique IDs is equivalent to counting unique vehicles. An ID should only be counted once during aggregation. The *Touch Inner Border* and *Touch Outer Border* queries count unique vehicles that touch the query region's boundary from the inside and the outside, respectively. The *Cross Border* query collects the volume of unique vehicles that cross the boundary. The *Distinct Objects* query refers to the volume of unique vehicles that have been detected in the query region, including its boundary.

C. Entry-based Query Types

Entry-based queries ask for the volume of trips to an area. We define an entry as a connected set of points on the path of a moving vehicle. Although a vehicle has only one entry to the whole region, it may have multiple entries to a constrained query region, where its path is divided into multiple sub-paths. Entry-based queries count the entries even if the entries are from the same vehicle. In comparison to the previous type, the answer for the entry-based *Touch Inner Border* query in Figure 1 is 2 because the vehicle touches the boundary twice. Similarly, we observe the differences for *Cross Border* and *Touch Outer Border* queries. This query type also contains a query named *Distinct Entries*, which computes the volume of entries to the query region, including its boundary. As we can see from the last sub-figure, the two vehicle paths are divided into three sub-paths, which leads to 3 distinct entries.

D. ID/Entry-based Query Types

For vehicles with certain movement patterns, counting unique IDs is equivalent to counting entries. We identify two ID/Entry-based queries: the *Total Inside* query counts the unique vehicles or entries that stay in the interior of the query rectangle, whereas the *Stay On Border* query asks for the count of unique vehicles or entries that travel along the border.

IV. Distributed Euler Histograms (DEHs)

The underlying data structure of PAMS is inspired by Euler Histograms (EHs), which were initially designed to count the number of rectangular objects in multi-dimensional space

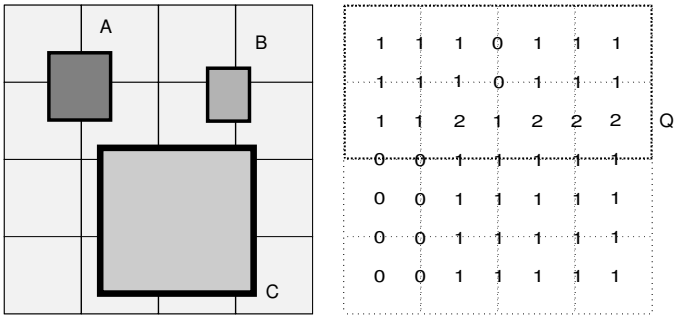


Fig. 2. Left: a 4 by 4 grid with 3 rectangles; right: the resulting EH and a query rectangle.

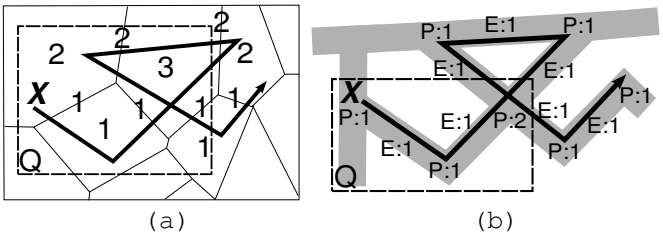


Fig. 3. Distributed Euler Histograms. In each subfigure, Q is the query rectangle and X is the path of a moving vehicle. Initial occurrence of a vehicle is its first entry to a region.

[26]. The histograms are constructed as follows. The space is partitioned into a grid. A count is maintained for each *face*, *edge* and *vertex* in the grid. For a rectangular object, the count for each corresponding face, edge or vertex is increased by 1. An example EH is given in Figure 2: 3 rectangular objects are mapped onto a regular space partition, a 4 by 4 grid, and the corresponding EH is given on the right part of the figure. From the EH the total number of distinct objects T in a given query region is determined by the formula $T = F - E + V$, where F is the sum of face counts overlapped by the (object) rectangles, E is the sum of edge counts intersecting the rectangles, and V is the sum of vertex counts enclosed by the rectangles. In Figure 2, we obtain for the query region Q: $T = 3$, which is the exact number of distinct objects intersecting Q. Although EHs can be easily built in a centralized database. they may not be suitable for wireless sensor networks. Hence, we created Distributed Euler Histograms (DEHs).

DEHs were developed to solve aggregate queries on moving objects in wireless sensor networks and can also be used for traffic monitoring. We created two variations of the DEH: a *Space-based DEH* (SDEH) and a *Graph-based DEH* (GDEH) (Figure 3a and 3b respectively).

For tracking unconstrained movements of vehicles, such as movement of a jeep on grassland, an SDEH can be used [5]. To construct an SDEH, the geographic space is divided into a Voronoi diagram. Each cell in the diagram contains a sensor responsible for detecting movements in that cell (sensors are not shown in figure). A SDEH keeps face counts and edge counts. A face count is kept for the interior of each cell and an edge count is kept for each edge between two adjacent cells. A count is the total number of detections on a face or an edge and will be incremented when a sensor begins to track

a moving vehicle, even if the vehicle had been encountered earlier. The SDEH correctly solves *Distinct Entries* queries (Section III-C) by the formula: $F-E$. F is the total count from faces that overlap with the query rectangle. E is the total edge counts between those faces. We assume that a vehicle always starts and stops on faces. Thus, the total face counts for a vehicle are the total edge counts plus 1, enabling to keep track of an entry. In Figure 3a, the total number F of faces overlapping with Q is 7 and the total count of the edges between those relevant faces E is 5. Hence, the number of entries is $7-5=2$.

For tracking movements on a constrained network, such as the movement of cars on a road network, we propose a GDEH. Its major difference to an SDEH is that points have the role of faces: instead of keeping face counts, we count the traffic at points, e.g., an intersection, a road end or a road check point. A point that is relevant to a query must be contained in the query rectangle. Any road segment between a pair of points is an edge. We assume that vehicles can only enter or leave the road network at points. This is true for the cars moving in a freeway system. Similar to an SDEH, this assumption ensures that the total point count for a single path is the total edge count plus 1. If a vehicle starts a trip on an edge, the closest point could be used as an approximation. In Figure 3b, the road network is represented as the grey-colored graph. The total point count (denoted by Ps) in Q is 4. The sum of edge counts (denoted by Es) between those points is 2, which leads to the number of entries: $P-E=4-2=2$.

DEHs ensure the privacy of vehicles as no ID is needed. DEHs are also efficient since each sensor only stores and sends a fixed number of simple counts, which consume much less bandwidth than sending a large amount of full IDs. However, DEHs cannot achieve high accuracy for queries (Section VI-B) that require reidentification of vehicles. Thus, we design an extension: an Euler Histogram based on Short ID (EHSID).

V. Privacy Aware Monitoring System (PAMS)

We distinguish between *absolute privacy* and *relative privacy*. Absolute privacy protects the true identities of users, whereas relative privacy is related to the probability that a user can be reidentified. The goal of our system is to protect the absolute privacy of a driver. To achieve a high accuracy level, PAMS needs to reidentify vehicles. Identification means revealing the full ID, whereas reidentification is simply used to differentiate vehicles.

PAMS collects and aggregates *short IDs* of vehicles. When a sensor detects a vehicle, it reads certain bits from the vehicle's full ID based on a random pattern. The random pattern is centrally issued by the system and is periodically updated for all sensors at the same time. This series of bits is the vehicle's short ID. For example, assuming the pattern is $1,2,5,7$. Then a full ID 11001001 will be converted to a short ID 1110. We assume that vehicles are equipped with active transponders, which have sufficient computational power to use asymmetric cryptography [27] to protect communication.

To evaluate the absolute privacy gained by our system, we define a privacy metric based on *k-anonymity* [28], which

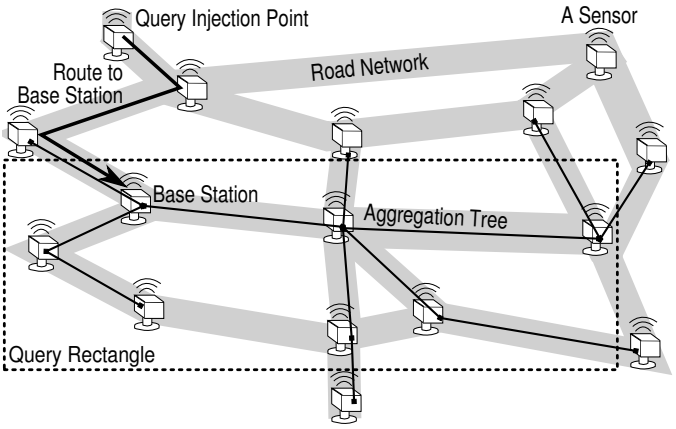


Fig. 4. Structure of PAMS.

requires that an object cannot be differentiated from other $k-1$ objects. We define k as:

$$k = 2^{l_F - l_S} \tag{1}$$

where l_F is the length of full ID and l_S is the length of short ID. And we define our privacy metric P as:

$$P = 1 - \frac{1}{k} = 1 - 2^{l_S - l_F} \tag{2}$$

Based on this definition, using full IDs leads to no privacy, i.e., $P=0\%$. P cannot reach 100% as no system, including PAMS, can achieve total privacy. The best case, i.e., the highest value for P, is using short IDs of length 0, which can be achieved in DEH. The absolute privacy level is high when the length of short IDs is significantly lower than the length of full IDs. For example, if the full IDs are 64-bits and short IDs are 32-bits, then P is 99.99999998%. The privacy level is also directly related to the overheads of PAMS. Higher number of bits in short IDs leads to higher costs and lower privacy level. In terms of communication and storage costs, the ideal length of short IDs should be as small as possible. However, as shown in our experimental results (Section VI-C1), an insufficient length for short IDs has a negative impact on accuracy. Hence, a balance between privacy, accuracy and efficiency is needed.

PAMS periodically update the random pattern for generating short IDs. The period needs to be properly set. Using a short ID for a long time, e.g., a week, increases the risk that an adversary could try to link the short ID against other information, e.g., that a car is often parked overnight at an owner's house. On the other hand, replacing short IDs too frequently reduces the accuracy, because the system cannot reidentify vehicles correctly.

We give an overview of PAMS in Figure 4. Sensors such as RFID readers that can read and write RFID tags are installed at every endpoint and intersection of a road network. Those sensors maintain the EHSID, which we detail in the following section. A query can be injected at any sensor node, called a *query injection point*. The query can be injected outside or inside a query region; it is forwarded to a node, *base station*, inside the query region. To build a route from the query injection point to the base station, we use a method derived

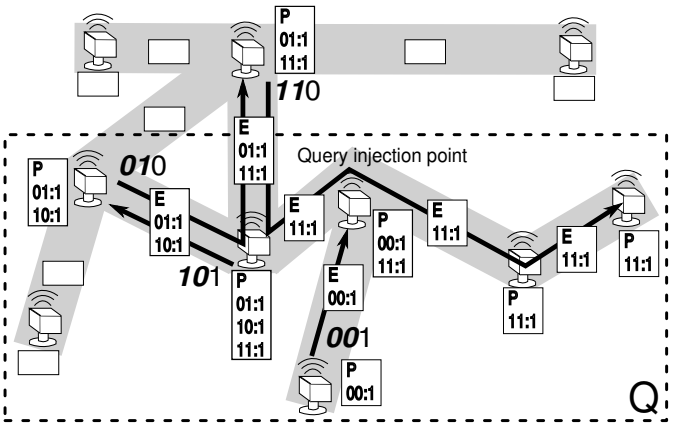


Fig. 5. EHSID with 4 paths of vehicles and a query rectangle Q.

from GPSR [29]. We assume that the query injection point and the query region are dynamically given. It is inefficient to maintain routing information for possible routes in our networks. A routing strategy, similar to GPSR, is used. PAMS uses a simplified version of the greedy forwarding method of GPSR. We assume that a sensor can directly communicate with its 1-hop neighbors and knows its own location and those of its immediate neighbors. We also assume that the deployment of sensors has a sufficient density to preserve connectivity. Upon receiving a query, a sensor finds the next node on the route by checking which neighbor is the closest one to the query region. The query will be forwarded to that node.

Dissemination of queries and aggregation of results are based on a data collection method known as TAG [30]. A tree topology is an essential part of TAG. Tree-based aggregation is good for efficiency as sensors do not need to communicate with a data center directly. In PAMS, the root of the tree is the base station that receives the query in the query region. The tree topology is generated by flooding. At the first step, the root broadcasts the query to its immediate neighbors, which form the first level of the tree. Then, all first-level nodes broadcast the query to their immediate neighbors who form the second level. A node will ignore the query if it is already in the tree. This progress will continue until the query reaches the nodes that are outside the query region. Once the tree is built, results are aggregated from the bottom to the root. Before sending aggregated data to the parent node, a sensor waits for a sufficient period of time to receive data from its child nodes.

PAMS is compatible with the existing TMSs and ETCs. It can share the communication network with current systems. The RFID readers in ETCs can also be used by PAMS. Two additions are needed to build PAMS upon existing infrastructures. First, base stations need to be deployed in the regions that are often queried. Second, basic routing and aggregation protocols need to be installed at the sensors.

A. Euler Histogram based on Short ID (EHSID)

Similar to a GDEH, an EHSID is built on a network. At every point and edge, we store a list of short IDs together with the count of vehicles. We show a simple example in Figure 5 with 4 cars moving in a road network. For simplicity, a full ID has 3 bits and its corresponding short ID is its 2-bit prefix. We label the points with P and edges with E. The aggregated data at a point is kept by the sensor installed at that point, whereas the data for an edge is kept by the two sensors that are the endpoints of that edge. Each sensor is responsible for tracking the vehicles moving *toward* it along the edge. An item in a list is shown as $X{:}Y$ where X is a short ID and Y is the count for the short ID.

TABLE I
POINT CATEGORIES IN EHSID

Point Category	Relative Position to Query Region
Just Outside (JO)	Located outside and has a one-hop neighbor inside the query region.
On Border (OB)	Located inside and has a one-hop neighbor outside the query region.
Just Inside (JI)	Located inside and no one-hop neighbor is JO but has an OB one-hop neighbor.
Completely Inside (CI)	Located inside but is not OB or JI.

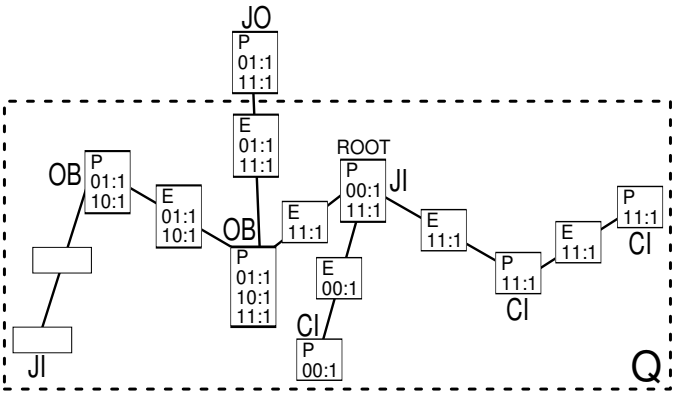


Fig. 6. Aggregation tree, point categories and data stored in EHSID based on the settings in Figure 5.

The spatial relationship between the path of a vehicle and the boundary of a query region determines if the vehicle's movement contributes to the result for the query. For example, if a vehicle should be counted for an ID-based *Touch Inner Border* query, it appears on and inside the border but not outside the query region. We categorize the points based on their spatial relationships to the boundary of the query region. The data of each category is aggregated independently. Table I shows our point categories. Figure 6 shows the topology of the aggregation tree and the categories of points based on the settings in Figure 5. The root of the tree is the query injection point inside the query region in Figure 5.

We do not consider points that are located further than JO points outside the query region. Additional points do not contribute to our queries and will only consume a considerable amount of bandwidth.

Three steps are needed to answer a query.

1) Aggregating data from points and edges: Every node in the aggregation tree builds datasets for the aggregated data from its children and itself. We denote P_{jo}, P_{ob}, P_{ji} and P_{ci} as the aggregated datasets for points of category JO, OB, JI and CI, respectively. We denote the aggregated datasets for edges as $E_{x,y}$ where x is the point category at one end and

TABLE II
FIND RELEVANT SHORT IDS FOR A QUERY

Query	Relevant IDs on Points	Meaningful Edge Datasets
Touch Inner Border (ID-based or Entry-based)	$(P_{ob} \cap P_{ji}) - P_{jo}$	$E_{ob,ob}$, $E_{ji,ji}$, $E_{ob,ji}$
Cross Border (ID-based or Entry-based)	$P_{jo} \cap P_{ob} \cap P_{ji}$	$E_{jo,jo}$, $E_{ob,ob}$, $E_{ji,ji}$, $E_{jo,ob}$, $E_{ob,ji}$
Touch Outer Border (ID-based or Entry-based)	$(P_{jo} \cap P_{ob}) - P_{ji}$	$E_{jo,jo}$, $E_{ob,ob}$, $E_{jo,ob}$
Distinct Objects (ID-based) and Distinct Entries (Entry-based)	$P_{ob} \cup P_{ji} \cup P_{ci}$	$E_{ob,ob}$, $E_{ji,ji}$, $E_{ci,ci}$, $E_{ob,ji}$, $E_{ji,ci}$
Total Inside (ID/Entry-based)	$(P_{ji} \cup P_{ci}) - P_{ob}$	$E_{ji,ji}$, $E_{ci,ci}$, $E_{ji,ci}$
Stay On Border (ID/Entry-based)	$(P_{ob} - P_{jo}) \cap (P_{ob} - P_{ji})$	$E_{ob,ob}$

y is the point category at the other end. Similar to the data collected by individual sensors, an aggregated dataset is a list of short IDs and their counts. In every aggregated dataset at a node, the count for a short ID is the sum of the counts for that ID from the same datasets in the sub-tree, which is rooted at the node. Take the settings in Figure 6 as an example. When aggregation is done at the root, P_{jo} has two items, 01:1 and 11:1. P_{ob} contains three items, 01:2, 10:2 and 11:1. P_{ji} contains two items, 00:1 and 11:1. P_{ci} has two items, 00:1 and 11:2. Similarly, $E_{jo,ob}$ stores the data on edges between JO points and OB points, which is 01:1 and 11:1. $E_{jo,jo}$ is empty as there is only one JO point. $E_{ob,ob}$ has two items, 01:1 and 10:1.

2) Finding short IDs relevant to a query: A short ID of a vehicle in the aggregated dataset may not be relevant to a query if path of the vehicle does not fulfill the definition of the query. As movements of vehicles are highly random, PAMS needs to find out the relevant short IDs before solving a query. To do this, we use the formulas listed in Table II. Note that the set operations in Table II are only on the short IDs in the point datasets. The counts for short IDs are not needed in this step. Take the same example as above: the relevant short IDs to a *Touch Outer Border* query in Figure 6 can be extracted by the formula $(P_{jo} \cap P_{ob}) - P_{ji} = (\{01,11\} \cap \{01,10,11\}) - \{00,11\} = \{01,11\} - \{00,11\} = \{01\}$. Hence, there is only one relevant short ID, 01.

3) Computing the final result: Methods for computing the final results are different for ID-based queries and Entry-based queries. Entry-based queries require an additional step. For ID-based queries, EHSID only needs to compute the cardinality of the set of short IDs extracted from the second step. As in the previous example, the answer for the ID-based *Touch Outer Border* query is 1.

For Entry-based queries, EHSID uses a method similar to the solution for *Distinct Entries* queries in GDEH. First, we compute the total point count, P, from the relevant IDs extracted in the previous step. Using the settings in Figure 6, the total count for the short ID 01 in JO and OB points is 3, hence, $P = 3$. Then, we sum up the counts for the extracted short IDs from the edge datasets. Here, we do not consider the edge datasets linking to the points that are deducted in the previous step. The meaningful edge datasets for each query are listed in Table II. We denote the total edge count as E. For the same example as above, we sum up the counts for ID 01 in the relevant edge datasets, $E_{jo,jo}$, $E_{jo,ob}$, and $E_{ob,ob}$, and we obtain $E = 2$. The final result can be computed as $P - E$. In this example, the result for Entry-based *Touch Outer Border* query is $P - E = 3 - 2 = 1$.

For ID/Entry-based queries, we can use the method for ID-based queries or Entry-based queries. As a relevant ID has only one entry, any method has the same results.

B. Estimation of Accuracy

Although many factors affect the accuracy of an answer to a query, the main factor is the number of short IDs involved in the query, in particular the short IDs of vehicles that are not shared by other vehicles. This is because reidentification is an essential part of PAMS. To correctly reidentify a vehicle, the short ID of a vehicle should be different from all other vehicles. A high probability that a vehicle can be reidentified leads to a high accuracy in answering a query. So the ratio between the number of exclusively-used short IDs and the total number of full IDs shows the accuracy level.

Let us assume that the number of vehicles, i.e., full IDs, in the query region is N_F and the length of a short ID is l bits. For the n^{th} full ID, the probability that the corresponding short ID is different to the short IDs of all former $n-1$ full IDs is:

$$P_n = \left(\frac{2^l - 1}{2^l}\right)^{n-1} \tag{3}$$

Then, we can estimate the total number of exclusively-used short IDs as:

$$U = \sum_{n=1}^{N_F} P_n \tag{4}$$

Finally, we give our accuracy metric as:

$$A = \frac{U}{N_F} \tag{5}$$

For example, if 500 full IDs (vehicles) appeared in the query region and PAMS uses 9-bit short IDs, then $A = 63.9\%$. As shown in the experimental results (Section VI-C1), this metric estimates accuracy well. System administrators can use the metric to configure the system, such as setting the proper length of short IDs to achieve a certain accuracy level for a certain area.

VI. EXPERIMENTS

A. Experimental Setup

We build our experimental environment using the J-Sim simulator [31]. We use the Network-based Generator of Moving Objects [32] to create moving object paths for vehicles in the road network of Melbourne, Australia. A virtual sensor is placed at every end point and intersection on the road network. We maintain the EHSID in the virtual sensors. To evaluate the performance of PAMS, we maintain two other data structures.

First, we maintain an *Euler Histogram based on Full ID (EHFID)*, which is similar to EHSID except that full IDs are used. We set the length of full IDs to 64 bits, which is a common setting in EPCs [33]. Since EHFID correctly

TABLE III
EXPERIMENTAL SETTINGS

No. of Bits	Perc. of QRS	No. of Sensors	No. of Vehicles	Max. No. of Trips	% of 2-trip Vehicles
7-16	10	1000	1000	2	10
12	10-100	1000	1000	2	10
12	10	100-1000	1000	2	10
12	10	1000	100-1000	2	10
12	10	1000	1000	1-10	Fixed
12	10	1000	1000	2	10-100

reidentifies vehicles, it always achieves 100% accuracy for all queries. We use it as the benchmark for the accuracy level achieved by other approaches.

Second, we maintain *Full ID Aggregation (FIA)* that only keeps full IDs at points, which is a simple method that can be used in any TMS. Compared to EHFID, FIA does not keep any data on edges and the counts for IDs on points. This means that FIA resides between EHSID and EHFID in terms of privacy. For ID-based and ID/Entry-based queries, FIA uses the same solutions as EHFID. For Entry-based queries, FIA uses the method for ID-based queries as an approximation.

The design goals of PAMS are good privacy protection, high accuracy and high efficiency. The key challenge is to achieve a good balance between them. Since PAMS achieves the maximum protection of privacy by using short IDs, we focus on accuracy and efficiency in our experiments.

Our experimental settings are shown in Table III. *Number of Bits* is the length of short ID in bits. *Percentage of QRS* is the ratio between the Query Rectangle Size (QRS) and the size of whole deployment area. *Number of Sensors* is the number of sensors in the whole network. *Number of Vehicles* is the number of vehicles in the network. *Maximum Number of Trips* is the highest number of trips made by a vehicle. A trip is defined as the shortest path between two randomly selected points in the road network. *Percentage of 2-trip Vehicles* is the ratio between 2-trip vehicles and all vehicles.

For each setup (a row in Table III), we generate 100 query rectangles at random positions. We solve all queries for each query rectangle using all approaches. Based on the average accuracy for each query, we compute the overall accuracy, i.e., the average accuracy for all queries. We also compute the average storage costs and average communication costs. The storage cost is the number of bits stored in the whole area. The communication cost is the number of bits transmitted for solving a query.

B. Results on GDEH

Before presenting the results for PAMS, we look at results of GDEH. This experiment is conducted in the same environment for testing PAMS. The simulation uses 1000 moving vehicles and 1000 sensors. 10% of the vehicles have two trips in the network. Other vehicles have only one trip. We change the percentage of QRS from 10% to 100%. Figure 7 shows communication costs of GDEH and FIA. When QRS covers 10% of the whole area, GDEH uses 66442 bits for solving a query in average while FIA needs 590313 bits. At 100% QRS, GDEH only needs 511488 bits, which is still smaller than the costs of FIA at 10% QRS. We also observe a decrease of FIA's

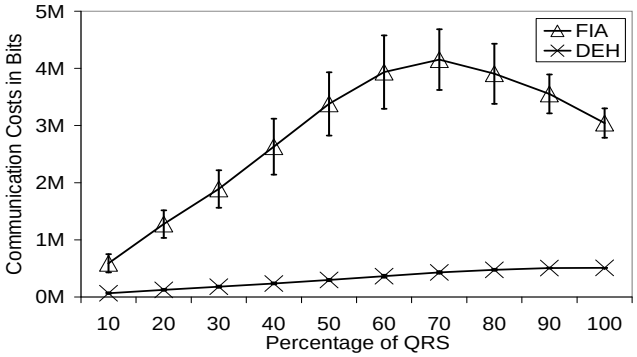


Fig. 7. Communication costs of DEH and FIA.

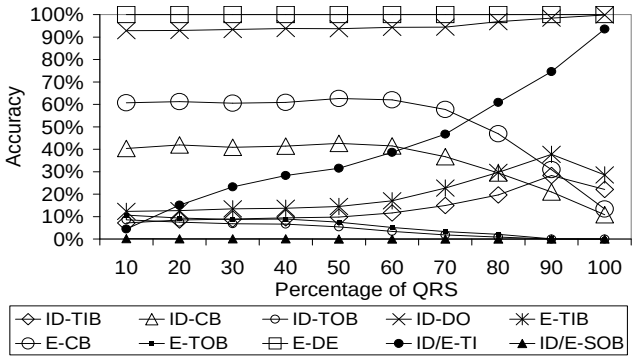


Fig. 8. Accuracy of GDEH for 10 different queries.

communication costs after 70% QRS, which results from the fact that some datasets used in FIA become smaller after that point. For example, if QRS is 100% of the whole area, all datasets relating to the JO points are empty as no JO points exist. Different to FIA, a sensor in GDEH only needs to report one face count and one edge count during aggregation. This explains the efficiency in GDEH's communication costs. Its storage costs (not shown) have a similar behavior.

The accuracy of GDEH for each query is shown in Figure 8. ID-based queries are labeled with prefix "ID" while Entry-based queries are labeled with prefix "E". The rest two with prefix "ID/E" are ID/Entry-based queries. As we can see from the figure, GDEH performs well for *Distinct Entries* (E-DE) queries and *Distinct Objects* (ID-DO) queries. If the system only needs to solve these queries, GDEH is a perfect solution in terms of privacy, accuracy and efficiency. For other queries, however, GDEH does not achieve high accuracy because it cannot avoid counts from irrelevant vehicles. For example, GDEH cannot deduct the counts from vehicles that crossed the border when answering ID-based *Touch Inner Border* queries. Interestingly, the accuracy of ID/Entry-based *Total Inside* (ID/E-TI) queries increases as QRS grows, because more paths of moving vehicles are entirely contained in the larger query rectangles. Motivated by these results, we developed EHSID, with the goal to provide good accuracy levels for all queries in a privacy aware environment.

C. Results on PAMS

1) Number of Bits: This parameter affects the accuracy of PAMS. If the length of short ID is too small, some vehicles

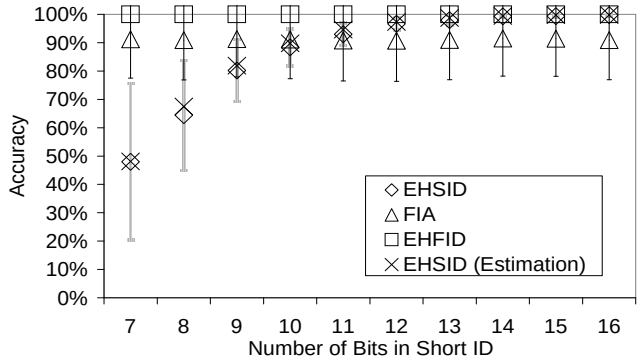


Fig. 9. Overall accuracy of EHSID, FIA, and EHFID.

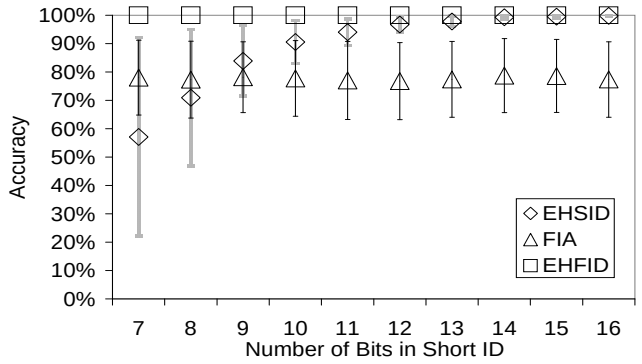


Fig. 10. Accuracy of EHSID, FIA, and EHFID for Entry-based queries.

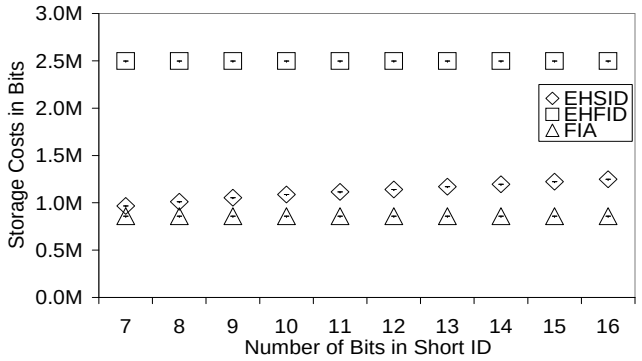


Fig. 11. Storage costs of EHSID, EHFID and FIA.

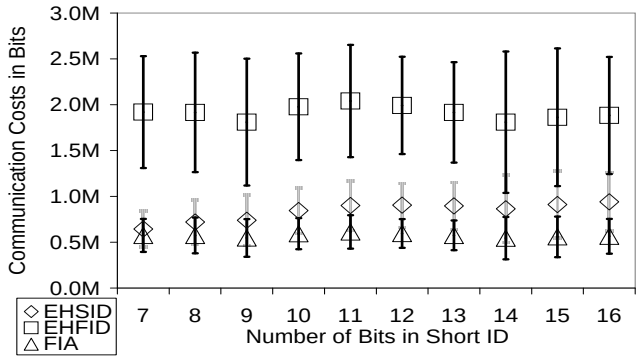


Fig. 12. Communication costs of EHSID, EHFID and FIA.

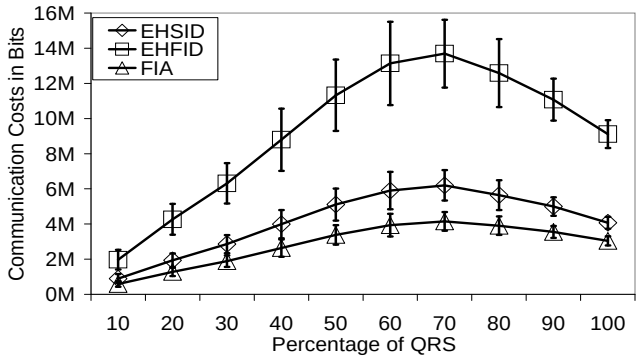


Fig. 13. Communication costs of EHSID, EHFID and FIA.

may share the same short ID. This causes errors in the answers (Figure 9). The results show that the overall accuracy of EHSID drops from 99.7% at 16-bits to 48% at 7-bits. Unlike EHSID, FIA always achieves 90% overall accuracy. However, it does not perform well for the Entry-based queries as counts of IDs are not maintained (Figure 10). As shown in the chart, FIA remains at 77% accuracy while EHSID achieves 90% and above accuracy when short IDs are 10-bits or longer. In Figure 9, we also show the estimated accuracy, which is computed from the formulas given in Section V-B. The parameter N_F used in those formulas is the average number of full IDs in the query region given all queries. Our results show a good match between the estimated accuracy and the real accuracy. We select 12-bits as the default length of short ID for the following experiments, as it is the shortest length that ensures at least an 95% overall accuracy (Figure 9).

This parameter also affects the privacy level and the overheads in PAMS. Based on our privacy metric described in Section V, the absolute privacy achieved by PAMS drops from $1-6.9e^{-18}$ to $1-3.6e^{-15}$ when the length of short ID increases from 7-bits to 16-bits. However, even at the worst case, PAMS still attains nearly a 100% absolute privacy. For storage costs, Figure 11 shows that the average storage overhead of our system increases from 96716 bits to 1249104 bits. The storage costs of PAMS (EHSID) are slightly higher than FIA but are much lower than EHFID. Similar to the storage costs, we also observe a growth of communication overheads (from 644514 bits to 941025 bits) when the length of short ID increases from 7-bits to 16-bits (Figure 12).

2) Percentage of QRS: The ratio between QRS and the size of deployment area has possible significant effects on the communication costs of PAMS. When QRS is small, only a small portion of the sensors are covered. Thus, the amount of data transmitted during aggregation is small. As QRS grows, all point datasets and edge datasets expand, which incurs higher communication costs. However, when QRS keeps growing after a certain point, the communication costs of all approaches drop. This is due to the similar reason that causes the decrease of FIA's communication costs in earlier experiments (Section VI-B). As shown in Figure 13, the costs of EHSID increase from 897152 bits at 10% QRS to 6197925 bits at 70% QRS then back to 4069118 bits at 100% QRS. We also observed that EHSID consumes much less bandwidth than EHFID. However, FIA is a little more efficient than EHSID as FIA does not need to keep counts for IDs and the data on edges.

3) Number of Sensors: The number of sensors affects PAMS' storage costs. As shown in Figure 14, the storage costs of EHSID increase from 589820 bits to 1193588 bits

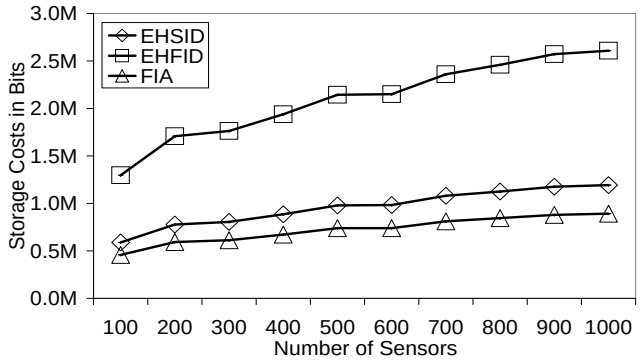


Fig. 14. Storage costs of EHSID, EHFID and FIA.

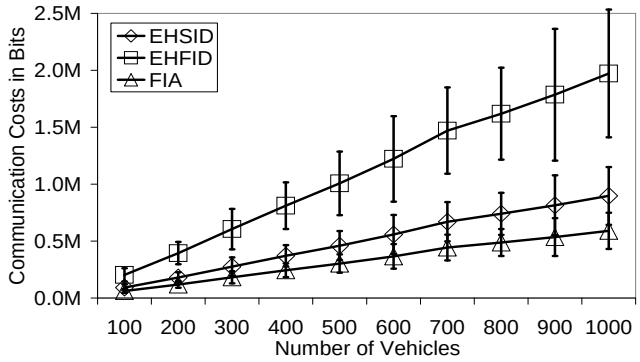


Fig. 15. Communication costs of EHSID, EHFID and FIA.

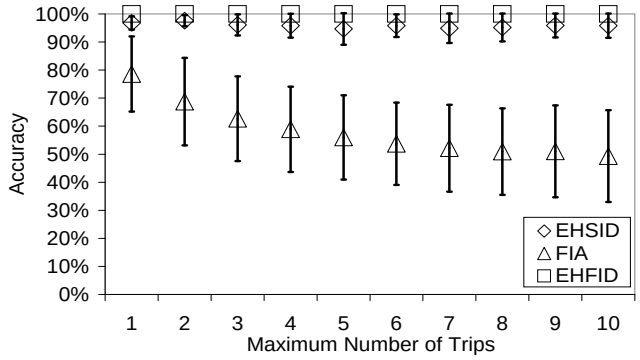


Fig. 16. Accuracy of EHSID, FIA, and EHFID for Entry-based queries.

when the number of sensors increases from 100 to 1000. This is understandable as a vehicle will visit more sensors if the deployment of sensors is denser. We also noticed that EHSID is more efficient than EHFID, due to its use of short IDs. FIA is more efficient than EHSID for the same reason as in the previous experiment.

4) Number of Vehicles: As more vehicles move in the network, PAMS needs higher communication costs to answer a query because more short IDs and counts need to be transmitted. As our result shows (Figure 15), the communication costs of EHSID increase from 92756 bits to 897152 bits when this parameter increases from 100 to 1000. We also observed similar changes in EHFID and FIA. Our results show that EHSID needs to transmit much less data than EHFID. The communication costs of EHSID and FIA are comparable.

5) Maximum Number of Trips: As in real life, drivers may make multiple trips during a day. Certain types of vehicles have more trips than others. For example, cabs usually have

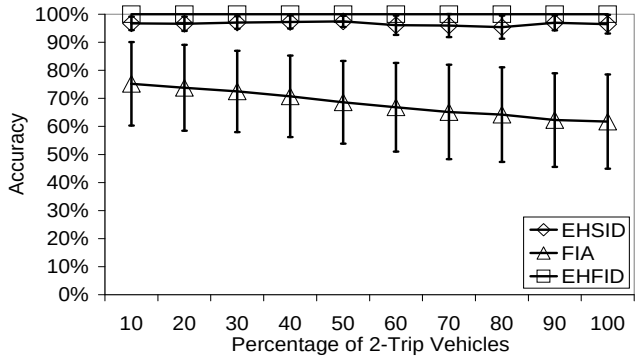


Fig. 17. Accuracy of EHSID, FIA, and EHFID for Entry-based queries.

more trips than commuter cars that only travel between home and office. We expect that the number of trips has a significant effect on the number of entries to a region, because a vehicle may enter the same region during new trips. In this experiment, the maximum number of trips per vehicle varies from 1 to 10. For each setting, the number of vehicles that made a certain number of trips is uniformly distributed. For example, when the maximum number of trips is 3, 33% of the vehicles have 1 trip, 33% of them have 2 trips, and the remaining vehicles have 3 trips. For a multi-trip travel, the destination of the previous trip is the origin of the next trip. We designed these settings to simulate the complex traffic conditions in a dense area, e.g., the downtown area of a city, where various types of vehicles may appear. The result shows that PAMS significantly outperforms the pure ID-based approach, FIA (Figure 16) for Entry-based queries. The accuracy of PAMS (EHSID) stays at 96%. At the same time, the accuracy of FIA drops from 78.6% to 49.3%. PAMS performs better because EHSID collects the counts of IDs that are important for solving Entry-based queries. FIA cannot achieve this and therefore performs worse.

6) Percentage of 2-trip Vehicles: We control the percentage of 2-trip vehicles, e.g., commuters. All vehicles, except the 2-trip vehicles, have only 1 trip. For Entry-based queries, PAMS (EHSID) is significantly more accurate than FIA (Figure 17). Our result shows that EHSID always achieves high accuracy, 97%, while FIA performs worse when more vehicles have 2 trips. FIA never reaches 80% accuracy and only achieves 61.7% accuracy when all vehicles have 2 trips.

VII. Conclusion

We introduced PAMS, a TMS that protects the privacy of vehicles in three aspects. First, the system only solves aggregate queries that hide ID information to end users. Second, it only collects short IDs in processing that have no linkage to full IDs. Third, short IDs are refreshed periodically to further reduce the privacy risk.

PAMS has advantages over the existing systems in solving a range of queries that ask the traffic volume of unique vehicles and the traffic volume of vehicles with certain trip characteristics. Our experiments show that PAMS achieves good accuracy levels. For Entry-based queries, PAMS is more accurate than a system that only collects full IDs. Furthermore, PAMS achieves a good efficiency level due to its use of short

IDs. In summary, PAMS achieves a good balance between privacy, accuracy and efficiency for traffic monitoring.

Our current work addresses queries on traffic volume counts. An immediate direction for future work is to explore queries involving other types of data, such as travel time and speed. Also, to formalize the effects of traffic network settings on the performance of PAMS, a further study is required. This will help field engineers to fine tune a given deployment. As using the data from probe vehicles is becoming popular in TMSs, improving the privacy protection for probe vehicles is an important research direction. PAMS protects the absolute privacy of drivers but needs reidentification of vehicles. Exploring approaches that can protect both absolute and relative privacy is a future research topic.

REFERENCES

[1] Hampshire County Council, Southampton City Council and Portsmouth City Council, "ROad MANagement System for Europe (ROMANSE)," http://www.romanse.org.uk/.

[2] M. Duckham and L. Kulik, "Location privacy and location-aware computing," in *Dynamic & Mobile GIS: Investigating Change in Space and Time*, 2006, pp. 35–51.

[3] Houston TranStar Consortium, "Houston TranStar AVI traffic monitoring system," http://traffic.houstontranstar.org/.

[4] Federal Highway Administration of US Department of Transportation, "Introduction to traffic monitoring," http://www.fhwa.dot.gov/ohim/tmguide/tmg2.htm, 2001.

[5] H. Xie, E. Tanin, and L. Kulik, "Distributed histogram for processing aggregate data from moving objects," in *MDM*, 2007, pp. 152–157.

[6] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, pp. 42–53, 2004.

[7] E-ZPass Customer Service Center, "E-ZPass," http://www.ezpass.com/.

[8] Singapore Land Transport Authority , "Electronic Road Pricing System," http://www.lta.gov.sg/.

[9] A. Sugiura and Y. Yukizaki, "Investigation of calculation/distance measurement method using spread-spectrum communications system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 2, pp. 130–135, 2002.

[10] A. Juels, R. L. Rivest, and M. Szydlo, "The blocker tag: selective blocking of RFID tags for consumer privacy," in *CCS*, 2003, pp. 103–111.

[11] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security and Privacy*, vol. 02, no. 3, pp. 49–55, 2004.

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

[13] M. Langheinrich and R. Marti, "RFID privacy using spatially distributed shared secrets," in *UCS*, 2007, pp. 1–16.

[14] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Computer*, vol. 36, no. 10, pp. 103–105, 2003.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[16] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *HOTOS*, 2003, pp. 28–28.

[17] S. Misra and G. Xue, "Efficient anonymity schemes for clustered wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 1/2, pp. 50–63, 2006.

[18] Y. Ouyang, Z. Le, Y. Xu, N. Triandopoulos, S. Zhang, J. Ford, and F. Makedon, "Providing anonymity in wireless sensor networks," *ICPS*, pp. 145–148, 2007.

[19] D. B. English, S. M. Kocisa, J. R. Arnolda, S. J. Zarnocha, and L. Warren, "The effectiveness of visitation proxy variables in improving recreation use estimates for the usda forest service," *Journal for Nature Conservation*, vol. 11, pp. 332–338, 2003.

[20] H. K. Cordell and G. T. Green, "An evaluation of traffic counts used for estimating recreation visitation: a case study of Jekyll Island State Park, Georgia," http://warnell.forestry.uga.edu/, Internet Research Information Series, Tech. Rep., March 2008.

[21] L. Wilmshurst, "Hagley/ferrymead community board report: Main road redcliffs - pedestrian facilities," http://archived.ccc.govt.nz//, 2001.

[22] Dillon Consulting Limited, "Kingston Transportation Master Plan Final Report," http://www.cityofkingston.ca/, 2004.

[23] Federal Highway Administration of US Department of Transportation, "National household travel survey," http://nhts.ornl.gov/, 2008.

[24] J.-P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*. Routledge, 2006.

[25] M. J. Egenhofer and J. R. Herring, "Categorizing binary topological relations between regions, lines, and points in geographic databases," National Center for Geographic Information and Analysis, Tech. Rep. 90-12.

[26] R. Beigel and E. Tanin, "The geometry of browsing," in *LATIN*, 1998, pp. 331–340.

[27] W. Stallings, *Cryptography and Network Security*. Prentice Hall, 2006.

[28] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

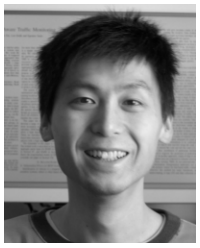
[29] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *MobiCom*, 2000, pp. 243–254.

[30] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.

[31] H.-Y. Tyan, "Design, realization and evaluation of a component-based compositional software architecture for network simulation," Ph.D. dissertation, 2002, Ohio State University.

[32] T. Brinkhoff, "A framework for generating network-based moving objects," *Geoinformatica*, vol. 6, no. 2, pp. 153–180, 2002.

[33] EPCglobal, *EPC Tag Data Standard (TDS)*, http://www.epcglobalinc.org/, 2006.



Hairuo Xie is a Ph.D. student in the Department of Computer Science and Software Engineering, the University of Melbourne. He received his Master of IT degree from the University of Melbourne in 2005. His research focuses on distributed data structures and aggregate query processing in sensor networks.

Lars Kulik received his PhD from the University of Hamburg, Germany, in 2002. He is a senior lecturer in the Department of Computer Science and Software Engineering at the University of Melbourne. His research focuses on efficient algorithms for moving objects and traffic applications, methods for safeguarding location privacy, information dissemination and aggregation algorithms in sensor networks, spatial algorithms in pervasive computing environments, and robust algorithms that cope with imperfection, especially in mobile computing.

Egemen Tanin is a senior lecturer in the Department of Computer Science and Software Engineering, the University of Melbourne. He received his PhD degree in computer science from the University of Maryland, College Park, Maryland, where he also held a postdoctoral research associate position from 2001 until 2003. His areas of research include spatial data management and database visualization.