Privacy-Preserving Multi-Point Traffic Volume Measurement Through Vehicle-to-Infrastructure Communications

Yian Zhou, *Member, IEEE*, Shigang Chen, *Senior Member, IEEE*, You Zhou, *Student Member, IEEE*, Min Chen, *Student Member, IEEE*, and Qingjun Xiao, *Member, IEEE*

***Abstract*—Traffic volume measurement is critical in vehicular networks. Existing research on traffic volume measurement mainly focuses on single-point traffic statistics. In this paper, we switch our view from *single-point* to *multi-point* and study the important problem of privacy-preserving multi-point traffic volume measurement in vehicular cyber-physical systems (VCPSs), which complements the state of the art. While embracing automatic traffic data collection, which the VCPS provides through vehicle-to-infrastructure communications, we also need to accept the accompanying challenges: First, the privacy of all participating vehicles should be preserved as an inherent requirement of a VCPS; second, the measurement scheme should be efficient enough to fit today's large-scale vehicular networks. In this paper, we start from a novel scheme that measures traffic volume between two arbitrary points (locations) through variable-length bit array masking. Then, we extend the idea of variable-length bit array masking to address the more challenging problem of three-point traffic measurement and present a general framework to measure traffic among three or more locations. We also perform extensive simulations to demonstrate the superior performance, applicability, and scalability of our schemes.**

***Index Terms*—Privacy, traffic measurement, vehicular networks.**

I. Introduction

TRAFFIC volume measurement is critical in vehicular networks and transportation engineering. In general, traffic volume statistics can be summarized into two categories: "*single-point*" statistics and "*multi-point*" statistics. Existing research on traffic volume measurement mainly focuses on *single-point* traffic statistics such as annual average daily traffic, which estimate the number of vehicles passing a specific point (geographical location) during some measurement period, and various predication models [1]–[4] have been proposed to measure them using data recorded by roadside units (RSUs). *Multi-point* traffic statistics, by contrast, describe the number of vehicles traveling through multiple points (geographical locations) during a measurement period. Multi-point traffic statistics provide essential input to a variety of studies, such as estimating traffic link flow distribution for investment plan, calculating road exposure rates for safety analysis, and characterizing turning movements at intersections for signal timing determination [5]. In this paper, we switch our view from *single-point* to *multi-point* and study the important problem of privacy-preserving multi-point traffic measurement in vehicular cyber-physical systems (VCPSs), which complements the state of the art. Our goal is to utilize the VCPS for automatic traffic data collection through vehicle-to-infrastructure communications and measure multi-point traffic while preserving vehicles' privacy.

Greatly advanced by new technologies in vehicular communications and networking [6]–[10], the VCPS has emerged as one of the most promising research areas in road networks. It integrates wireless communications and on-board computers into transportation systems to enhance road safety and improve driving experience [11], [12]. In particular, the IEEE has standardized dedicated short-range communications (DSRC) under IEEE 802.11p [13], which supports transmitting/receiving messages between vehicles and RSUs. A great advantage that the VCPS provides is automatic traffic data collection: Each vehicle simply transmits its ID as it passes each RSU. From the IDs collected by all RSUs, we can easily figure out the multi-point traffic data. However, this straightforward approach leads to serious privacy breaching as it also tracks the entire moving history of vehicles. As more and more people are concerned about their privacy, any traffic measurement scheme to be deployed in the VCPS should consider travelers' privacy as its top priority. The transportation authorities from different countries have put forward a number of principles to protect travelers' privacy. An example is the "anonymity by design" principle required by IntelliDrive from the United States Department of Transportation [14]. Keeping the privacy requirement in mind, it is clearly not acceptable to have the vehicles report their unique identifiers. Other permanently or temporarily fixed numbers also bare the potential of giving away the vehicles' trajectory. Therefore, the challenge is to design a measurement scheme in which a

Manuscript received April 2, 2015; revised July 15, 2015 and September 1, 2015; accepted September 21, 2015. Date of publication October 7, 2015; date of current version December 14, 2015. This work was supported by the National Science Foundation under Grant CNS-1409797. The review of this paper was coordinated by Guest Editors.

Y. Zhou, S. Chen, Y. Zhou, and M. Chen are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yian@cise.ufl.edu; sgchen@cise.ufl.edu; youzhou@cise.ufl.edu; min@cise.ufl.edu).

Q. Xiao is with the Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China (e-mail: csqjxiao@seu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2015.2487985

0018-9545 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

vehicle never transmits any unique identifier or any fixed number for privacy protection, with the random and de-identified information that the vehicle submits still able to support the traffic measurement among multiple different locations.

However, limited research work exists in the literature to address this problem. The most related studies are that of Lou and Yin [15] and our previous work [16]–[19]. The work of Lou and Yin tries to infer "two-point" statistics from "single-point" data, but the high computation overhead limits its practicability. More importantly, it is not designed for multiple points. Google announced providing real-time traffic data service in Google maps [20], but their approach cannot assure the vehicle's privacy since it uses the Global Positioning System and Wi-Fi in phones to track locations [21]. Our previous work [16] utilizes an encryption method to preserve vehicles' privacy and measures two-point traffic based on the encrypted vehicle IDs. The computation efficiency is improved to $O(n_x n_y)$ for each pair of RSUs, where n_x and n_y denote the number of vehicles passing them, respectively. This is better than the work in [15], but the overhead is still too high for today's large-scale road networks. Motivated by the works in [22] and [23], we propose a new approach in [17], which further improves the computation efficiency to $O(n_x + n_y)$ through the design of fixed-length bit arrays. However, the paper makes an unrealistic assumption about traffic similarity and uses bit arrays of equal length at different RSUs to encode the passing vehicles, such that the bit arrays from two RSUs can be bitwise compared to extract a statistical result for two-point traffic volume. The scheme works great when all RSUs observe comparable numbers of vehicles. However, in reality, the traffic volume at different RSUs greatly varies. For example, according to the 2012 yearly traffic volume report from the New York State Department of Transportation [24], major intersections in New York have hundreds of thousands of cars passing by every day, whereas light-traffic intersections only have a few hundred cars passing by during the same period. Considering this more realistic situation where different RSUs observe varied traffic, the performance in [17] dramatically decreases in terms of both vehicle privacy and measurement accuracy, which, therefore, limits its practicability. As a continuous effort in improving efficiency, privacy, and accuracy, we design variable-length bit array masking in [19] to remove the similar traffic assumption. However, it only handles two-point traffic.

This journal paper makes a significant new advance to handle multi-point traffic measurement, which extends and generalizes over our previous two-point scheme [19]. As far as we know, it is the first study of the privacy-preserving multi-point traffic measurement problem that measures the traffic passing through multiple locations. We propose novel solutions based on variable-length bit arrays for privacy-preserving multi-point traffic measurement, which tackle the efficiency, privacy, accuracy, and generalization problems encountered by all previous solutions. We begin by introducing our two-point traffic measurement scheme, then extend our idea of variable-length bit array masking to address the more challenging three-point traffic measurement problem, and eventually present a general multi-point traffic measurement framework to measure traffic volume among more than two points (locations). We demonstrate the superior performance, applicability, and scalability of our solutions through mathematical proof and extensive simulations.

II. Problem Statement

A. Problem Definition

We consider a VCPS involving three groups of entities: vehicles, RSUs, and a central server, with the latter two forming the infrastructure. Vehicles and RSUs each has a unique ID and is equipped with computing and communication capabilities. Vehicles can communicate with RSUs in real time via DSRC [13]. RSUs are connected to the central server through wired or wireless means, and they report information collected from vehicles to the central server periodically.

Given any d locations where RSUs are installed, we define the set of vehicles that pass all the d locations during some measurement period T as a d-point traffic flow. We want to measure the number of vehicles in the flow, which is called the d-point traffic volume. For example, the two-point traffic volume among a set of two RSUs $\{R_x, R_y\}$ measures the number of vehicles passing by both R_x and R_y, whereas the three-point traffic volume among a set of three RSUs $\{R_x, R_y, R_z\}$ describes the number of vehicles passing by all three RSUs, i.e., R_x, R_y, and R_z. The problem is to measure the d-point traffic volume $(d > 1)$ while protecting vehicles' privacy. To achieve privacy protection, we need a solution in which a vehicle never transmits any unique identifier or any permanently or temporarily fixed data. Ideally, the information transmitted by the vehicles to the RSUs looks totally random, out of which neither the identity nor the trajectory of any vehicle can be pried with high probability. One typical application scenario is to measure multi-point traffic in a city with a typical measurement period of a day, where RSUs may be deployed at any interested locations in the city.

B. Threat Model

We assume that RSUs are semi-trusted: On the one hand, all RSUs are from trustworthy authorities, which can be enforced by authentication based on public key infrastructure, and RSUs will not be compromised. Vehicles can use the public key certificate broadcasted by RSUs, which they obtained from the trusted third parties, to verify the RSUs. On the other hand, the authorities may exploit the information collected by RSUs to track individual vehicles when they need to do so. For instance, if a vehicle transmits any unique identifier upon each query, that identifier can be used for tracking purposes.

Note that there are also other ways to track a vehicle, for example, tailgating the vehicle or setting cameras near RSUs to take photos and using image processing to recognize it. These methods are beyond the scope of this paper. In this paper, we focus on preventing automatic tracking caused by the traffic measurement scheme itself.

We also assume that a special medium access control (MAC) protocol such as SpoofMAC [25] is applied to support privacy preservation such that the MAC address of a vehicle is not fixed. Vehicles may pick a MAC address randomly from a large

space for one-time use when needed. Through this, vehicles can report information to RSUs for traffic flow measurement without revealing their true identities.

C. Performance Metrics

We consider three performance metrics to evaluate a traffic measurement scheme: computation overhead, measurement accuracy, and preserved privacy.

1) Computation Overhead: It includes the computation overhead for each vehicle per RSU en route, and for each RSU per passing vehicle, and for the central server to measure the multi-point traffic volume among an arbitrary set of RSUs.

2) Measurement Accuracy: Let n_c be the actual multi-point traffic volume among a set of RSUs and $\hat{n}_c$ be the estimator for n_c. We measure the accuracy of a multi-point traffic measurement scheme by evaluating the bias and standard deviation of $\hat{n}_c/n_c$. Clearly, a good measurement scheme should have close-to-zero bias and relatively small standard deviation.

3) Preserved Privacy: The essence of privacy preservation in multi-point transportation traffic measurement is to give the adversary only a limited chance of identifying partially or fully any trajectory of any vehicle. Accordingly, we quantify the privacy of a scheme through a parameter p that satisfies the following requirement: The probability for any "trace" of any vehicle to not be identified must be at least p, where a trace of a vehicle is a pair of RSUs it has passed by. A larger value of p means better privacy. Intuitively, a scheme with $p = 0.9$ is better than a scheme with $p = 0.5$ in terms of privacy because the latter gives the adversary a better chance to link traces of a vehicle to obtain its trajectory since it allows the traces to be identified with a higher probability, i.e., $1 - p$.

Note that our privacy definition agrees with the privacy requirements as proposed in [26] and [27]. In [26], different privacy metrics [27], [28] are surveyed to characterize the vehicles' privacy level. In contrast to the anonymity set analytical models [27], which vary as the traffic conditions change, it is easier to judge the privacy level of a traffic measurement scheme through a single quantitative metric of probability that fits the global system and applies to various traffic conditions and scenarios. In [28], the overall probability for an adversary to follow a vehicle from the origin to the destination (OD data) with an entropy perspective is considered. However, we believe that stronger privacy, which considers the probability for the trajectory of a vehicle (as opposed to the narrower OD data) not to be identified by any adversary, is desirable for VCPSs. For example, the identity of a vehicle may be revealed at some location (not necessarily at the origin or the destination of its trip), e.g., through a photograph triggered by the vehicle rushing a red light or by a police car stopping the vehicle. These circumstances are not covered by the privacy definition in [28] but are captured by ours.

III. Two-Point Traffic Measurement

Here, we introduce our privacy-preserving two-point traffic measurement scheme in [19], which is designed based on variable-length bit arrays, a novel "unfolding" technique, and a formally derived MLE estimator. We first describe the two measurement phases in the proposed scheme and then analyze its performance.

A. Online Coding Phase

Our two-point scheme consists of two phases: online coding phase for storing de-identified vehicle information in bit arrays of RSUs and offline decoding phase for measuring two-point traffic volume between two arbitrary RSUs based on the reported bit arrays. In the following, we explain the first phase.

Each RSU R_x maintains a counter n_x, which keeps track of the total number of passing vehicles during the current measurement period. R_x also maintains a bit array B_x with length m_x to mask vehicle identities. We require the lengths of all bit arrays to be a power of 2, i.e., m_x must be in the form of 2^k, to facilitate the comparisons of varied-length bit arrays (more explanation later). We set the value of m_x to be $m_x = 2^{\lceil \log_2(\bar{n}_x \times \bar{f}) \rceil}$, where $\bar{n}_x$ is the expected traffic at R_x during the measurement period based on history average traffic at the same location and the same time, and $\bar{f}$ is a system-wide parameter whose value affects the tradeoff between measurement accuracy and level of privacy. Clearly, m_x is the smallest integer that is a power of 2 and no less than $\bar{n}_x \times \bar{f}$. At the beginning of each measurement period, n_x and all bits in B_x are set to zeros.

Each vehicle v has a logical bit array LB_v, which consists of s bits randomly selected from an imaginary array B_* whose size m_* is equal to that of the largest bit array among all RSUs, where $s \ll m_*$. The indexes of these bits in B_* are $H(v \oplus K_v \oplus X[0]), \ldots, H(v \oplus K_v \oplus X[s-1])$, where $\oplus$ is the bitwise XOR, $H(\ldots)$ is a hash function whose range is $[0, m_*)$, X is an integer array of randomly chosen constants to arbitrarily alter the hash result, and K_v is the private key of v to protect its privacy.

Table I lists some frequently used notations in this paper. Given the notations and data structures, online coding works as follows. RSUs broadcast queries in preset intervals (e.g., once a second), ensuring that each passing vehicle receives at least one query and, at the same time, giving enough time for the vehicle to reply. Collisions can be resolved through a well-established carrier-sense multiple access or time-division multiple access protocol, which are not the focus of this paper. Every query that an RSU sends out includes the RSU's RID, its public key certificate, and the size of its bit array. Suppose a vehicle, whose ID is v, receives a query from an RSU, whose ID is R_x and the bit array size is m_x. It first verifies the certificate to authenticate the RSU. After verifying that R_x is from trustworthy authority, v will randomly select a bit from its logical bit array LB_v by computing an index $b = H(v \oplus K_v \oplus X[H(R_x \oplus t) \bmod s])$, where t is the current time stamp. Then, v generates an index b_x in the range of $[0, m_x)$ corresponding to b, where $b_x = b \bmod m_x$, and sends b_x to R_x. Upon receiving the index b_x, R_x will first increase its counter n_x by 1 and then set the b_xth bit in B_x to 1. Therefore, the overall effect that v produces on R_x is

$$n_x = n_x + 1 \tag{1}$$

$$B_x\left[H\left(v \oplus K_v \oplus X\left[H(R_x \oplus t) \bmod s\right]\right) \bmod m_x\right] = 1. \tag{2}$$

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Meaning
R, R_x, R_y, R_z	RSUs
n_x, n_y, n_z	single-point traffic volume of RSU R_x, R_y, R_z, respectively
n_{xy}, n_{xz}, n_{yz}	two-point traffic volume between two RSUs, R_x and R_y, R_x and R_z, R_y and R_z, respectively
n_{xyz}	three-point traffic volume among three RSUs R_x, R_y, and R_z
$\hat{n}_{xy}, \hat{n}_{xz}, \hat{n}_{yz}$	MLE estimator of the two-point traffic volume n_{xy}, n_{xz}, n_{yz}, respectively
$\hat{n}_{xyz}$	MLE estimator of the three-point traffic volume n_{xyz}
B_x, B_y, B_z	bit arrays of RSUs R_x, R_y, R_z, respectively
B_{xy}, B_{xz}, B_{yz}	"unfolding" and "bitwise OR" result of two bit arrays, B_x and B_y, B_x and B_z, B_y and B_z, respectively
B_{xyz}	"unfolding" and "bitwise OR" result of three bit arrays, B_x and B_y and B_z
m_x, m_y, m_z	sizes of bit arrays B_x, B_y, B_z, respectively
m_{xy}, m_{xz}, m_{yz}	sizes of bit arrays B_{xy}, B_{xz}, B_{yz}, respectively
m_{xyz}	size of the bit array B_{xyz}
U_x, U_y, U_z	number of zeros in bit arrays B_x, B_y, B_z, respectively
U_{xy}, U_{xz}, U_{yz}	number of zeros in bit arrays B_{xy}, B_{xz}, B_{yz}, respectively
U_{xyz}	number of zeros in the bit array B_{xyz}
V_x, V_y, V_z	ratio of zeros in bit arrays B_x, B_y, B_z, respectively
V_{xy}, V_{xz}, V_{yz}	ratio of zeros in bit arrays B_{xy}, B_{xz}, B_{yz}, respectively
V_{xyz}	ratio of zeros in the bit array B_{xyz}
LB_v	the logical bit array of vehicle v
s	size of the logical bit array of every vehicle
f, f_x, f_y, f_z	single-point load factor, ratio of an RSU's bit array size over its traffic volume, e.g., $f_x = \frac{m_x}{n_x}$
$\bar{f}$	fixed load factor for all RSUs in our design
m	fixed bit array size for all RSUs in [17], i.e., $m_i = m, \forall R_i$

Note that the same vehicle may transmit different bit indexes at two RSUs. The probability for this to happen is $1 - (1/s)$, which is larger when the size s of LB_v is larger. Different vehicles may send the same index because their logical bit arrays share bits from B_x. As any vehicle does not have to transmit any fixed number in support of traffic measurement, we improve privacy protection. This is true even when there is a single vehicle passing through two RSUs.

B. Offline Decoding Phase

At the end of each measurement period, all RSUs will send their counters and bit arrays to the central server, which first updates the history average single-point traffic volume for the RSUs to take into account the traffic data in the current measurement period and then measures the two-point traffic volume between two arbitrary RSUs based on the reported counters and bit arrays.

Suppose the set of vehicles that pass RSU R_x (R_y) is denoted as S_x (S_y) with cardinality $|S_x| = n_x$ ($|S_y| = n_y$). Clearly, the set of vehicles that pass both RSUs R_x and R_y is $S_x \cap S_y$. Denote its cardinality as n_{xy}, i.e., $|S_x \cap S_y| = n_{xy}$, which is the value that we want to measure. Denote the size of the bit array B_x (B_y) stored in RSU R_x (R_y) as m_x (m_y). Without loss of generality, we assume that $m_x \leq m_y$. Given the counters n_x and n_y, and bit arrays B_x and B_y, the server measures n_{xy} as follows.

First, our previous work [17] shows that when two bit arrays have the same length, we are able to combine them through bitwise OR and produce a good estimate for the two-point traffic volume. Now, we have to deal with two bit arrays of different lengths. To combine the information of the two arrays through bitwise OR, the central server expands the smaller bit array B_x to the same size of B_y through a process called "unfolding," which is simply duplicating B_x multiple times until it reaches the size of B_y. Because m_x and m_y are both powers of 2 and $m_x \leq m_y$, it will always be true that m_y is divisible by m_x, which means that we can unfold B_x to the size of B_y by duplicating B_x for m_y/m_x times. (When we derive the new formula for estimating the two-point traffic volume, we will mathematically account for the impact of duplication.) The "unfolded" bit array of B_x is denoted as B_x^u. Specifically

$$B_x^u[i] = B_x[i \bmod m_x] \quad \forall\, i \in [0, m_y). \tag{3}$$

Second, the server takes a bitwise OR operation on B_x^u and B_y to obtain a new bit array B_{xy}, i.e.,

$$B_{xy}[i] = B_x^u[i] \vee B_y[i] \quad \forall\, i \in [0, m_y). \tag{4}$$

The bitwise OR operation is granted since the two bit arrays, i.e., B_x^u and B_y, are of the same size. Through requiring the size of all bit arrays to be a power of 2, we facilitate the comparison of varied-length bit arrays: The overall computation overhead to compare B_x and B_y is just $O(m_y)$, in contrast to $O(m_x \times m_y)$ without the "power of 2" requirement.

Finally, given B_{xy}, B_x (B_x^u), and B_y, the central server uses the following formula to estimate the two-point traffic volume between R_x and R_y:

$$\hat{n}_{xy} = \frac{\ln(V_{xy}) - \ln(V_x) - \ln(V_y)}{\ln\left(1 - \frac{s-1}{s} \times \frac{1}{m_y}\right) - \ln\left(1 - \frac{1}{m_y}\right)} \tag{5}$$

where V_{xy}, V_x, and V_y are random variables that represent the fraction of zero bits in B_{xy}, B_x, and B_y, correspondingly. Their values can be easily found by counting the number of zeros in B_{xy}, B_x, and B_y, which are denoted by U_{xy}, U_x, and U_y, respectively, and dividing them by the bit array size m_y, m_x, and m_y. That is, $V_{xy} = U_{xy}/m_y$, $V_x = U_x/m_x$, and $V_y = U_y/m_y$. Note that the fraction of zero bits in B_x^u is the same as B_x.

C. Performance Analysis

1) Measurement Accuracy: In [19], we have demonstrated that $\hat{n}_{xy}$ is an MLE estimator of n_{xy} and mathematically analyzed the measurement accuracy of our two-point scheme through the bias and standard deviation of $\hat{n}_{xy}/n_{xy}$. We also perform extensive simulations, which show that our scheme can indeed achieve very accurate measurement results. See [19] for detailed derivation, proof, and analysis.

2) Computation Overhead: Clearly, the computation overhead for the vehicles and RSUs of this scheme are comparable to that in [17]. In this scheme, when a vehicle v passes an RSU R_x, vehicle v only needs to compute two hashes to obtain an index of a random bit, and RSU R_x only needs to set one bit in its bit array B_x, as described in Section III-A. Hence, the computation overhead for each vehicle per RSU as well as for each RSU per vehicle are both $O(1)$.

As for the central server, the task it performs is a little bit more complicated than that in [17], but the computation overhead is comparable. First, the server unfolds the smaller bit array B_x to B_x^u, which has the same size as B_y. This operation costs $O(m_y)$ time. Second, it performs a bitwise OR over two m_y-bit arrays, i.e., B_x^u and B_y, to create a new bit array B_{xy} of size m_y, which also costs $O(m_y)$ time. Finally, the server counts the number of zeros in B_x, B_y, and B_{xy}, which takes $O(m_y)$ time as well. Therefore, the overall computation overhead for the server to measure the traffic volume between a pair of RSUs, i.e., R_x and R_y, is $O(m_y)$, where m_y is the size of the larger bit array of the two RSUs. Since [17] assumes that $m_x = m_y = m$ and its computation overhead for the server is $O(m)$, one can see that this scheme indeed achieves comparable computation overhead as in [17].

3) Preserved Privacy: In [19], we have analyzed the privacy of our two-point scheme through mathematical derivation and numerical analysis, which demonstrate that our scheme well preserves vehicles' privacy. Here, we directly give the formula for the privacy p of this scheme and refer interested readers to [19] for detailed derivation and analysis. Thus

$$p = \frac{1}{1 - P(\bar{A})} \times \left[\left(1 - \frac{1}{m_x}\right)^{n_{xy}} - \left(1 - \frac{1}{m_x}\right)^{n_x}\right] \times \left[\left(1 - \frac{1}{m_y}\right)^{n_{xy}} - \left(1 - \frac{1}{m_y}\right)^{n_y}\right] \quad (6)$$

where $P(\bar{A})$ is given in

$$P(\bar{A}) = \left(1 - \frac{1}{m_x}\right)^{n_x} \times C_1^{n_{xy}} + \left(1 - \frac{1}{m_y}\right)^{n_y} - \left(1 - \frac{1}{m_x}\right)^{n_x} \left(1 - \frac{1}{m_y}\right)^{n_y} \times C_2^{n_{xy}} \quad (7)$$

and C_1 and C_2 are both constants with values

$$C_1 = \frac{1}{s} \times \frac{1 - \frac{1}{m_y}}{1 - \frac{1}{m_x}} + \left(1 - \frac{1}{s}\right) \quad (8)$$

$$C_2 = \frac{1}{s} \times \frac{1}{1 - \frac{1}{m_x}} + \left(1 - \frac{1}{s}\right). \quad (9)$$

Note that if we set $m_x = m_y = m$ in (6), we get the same formula as in [17]. This is natural, since [17] is just a special case of this scheme.

IV. Multi-Point Traffic Measurement

A. From Two-Point to Multi-Point

In the previous section, we have presented our privacy-preserving two-point traffic measurement scheme, which can easily fit today's large-scale road networks. To serve for a broader spectrum of applications in transportation engineering, we are motivated to generalize our design to address the more challenging problem of multi-point traffic measurement.

Here, we will show how to extend our idea of variable-length bit array masking to address three-point traffic measurement, which observes the potential of further generalization to solve multi-point traffic measurement. Intuitively, if we can unfold two bit arrays to obtain statistical results related to the two-point traffic volume, we should also be able to unfold three or more bit arrays to get a statistical estimator for the multi-point traffic volume. The measurement process should be similar: Vehicles report random indexes from their logical bit arrays to mark RSUs' varied-length bit arrays, and the central server performs unfolding and bitwise OR operations on three or more bit arrays to obtain statistical results related to the multi-point traffic volume. If an MLE estimator can also be mathematically derived from those statistical results, it will be easy for the central server to compute the multi-point traffic volume.

In the remaining part of this section, we follow the above thinking to develop our privacy-preserving three-point traffic measurement scheme. We first explain the two measurement phases, validate the MLE estimator used to measure three-point traffic volume, and then analyze its performance. Finally, we generalize our two-point and three-point schemes and present a general framework for multi-point traffic measurement.

B. Privacy-Preserving Three-Point Traffic Measurement

1) Online Coding Phase: The online coding phase of our three-point scheme is exactly the same as our two-point scheme. Each RSU R_x maintains a counter n_x to record the total number of passing vehicles and a bit array B_x with length $m_x = 2^{\lceil \log_2(\bar{n}_x \times \bar{f}) \rceil}$ to collect vehicles' "masked" data, where $\bar{n}_x$ is the expected traffic volume in R_x, and $\bar{f}$ is a system-wide load factor, whose value is the same for all RSUs. At the beginning, n_x and all bits in B_x are set to zeros. For privacy protection, each vehicle v also has a logical bit array LB_v consisting of s bits randomly selected from an imaginary array B_* whose size m_* is equal to that of the largest bit array among all RSUs, where $s \ll m_*$. The bit indexes in B_* are $H(v \oplus K_v \oplus X[0]), \ldots, H(v \oplus K_v \oplus X[s-1])$. Some frequently used notations can be found in Table I.

Vehicles and RSUs cooperate to automatically collect "masked" traffic data. When a vehicle v receives a query from an RSU R_x, whose bit array is B_x with size m_x, it first verifies R_x. Once R_x is authenticated, v randomly selects a bit from LB_v by computing an index $b = H(v \oplus K_v \oplus X[H(R_x \oplus t) \bmod s])$, where t is the current time stamp, then generates an

index $b_x = b \bmod m_x$ in the range of $[0, m_x)$, and finally sends b_x to R_x. Upon receiving index b_x, R_x increases its counter n_x by 1 and sets the b_xth bit in B_x to 1.

2) Offline Decoding Phase: At the end of each measurement period, all RSUs will send their counters and bit arrays to the central server, which first updates the history single-point traffic data for the RSUs to take into account the current measurement period. Then, the server will measure the three-point traffic volume among three arbitrary RSUs based on the reported counters and bit arrays, which incurs a little bit more work than the two-point scheme (due to the third involving RSU). However, the measurement process is similar, and the computation overhead is also comparable to the two-point case.

We first define some notations (also summarized in Table I). We denote the set of vehicles passing RSUs R_x, R_y, and R_z as S_x, S_y, and S_z with cardinality $|S_x| = n_x$, $|S_y| = n_y$, and $|S_z| = n_z$, respectively. The set of vehicles that pass the set of three RSUs $\{R_x, R_y, R_z\}$ is $S_x \cap S_y \cap S_z$. Denote its cardinality as n_{xyz}, i.e., $n_{xyz} = |S_x \cap S_y \cap S_z|$, which is the value that we want to measure. The set of vehicles passing both R_x and R_y is $S_x \cap S_y$, whose size is denoted as n_{xy}, i.e., $n_{xy} = |S_x \cap S_y|$. Similarly, we have $n_{xz} = |S_x \cap S_z|$, $n_{yz} = |S_y \cap S_z|$. In addition, we denote the size of the bit arrays B_x, B_y, and B_z stored in RSUs R_x, R_y, and R_z as m_x, m_y, and m_z, respectively. Without loss of generality, we assume that $m_x \le m_y \le m_z$.

Given above notations, the central server measures n_{xyz} by performing the four steps of unfolding and bitwise OR operations below and then computing the MLE estimator in (14).

Step 1: The server unfolds B_x to the same size of B_y and takes a bitwise OR operation on the unfolded bit array and B_y to obtain a new bit array B_{xy} of size m_y. More specifically

$$B_{xy}[i] = B_x[i \bmod m_x] \vee B_y[i] \quad \forall\, i \in [0, m_y). \tag{10}$$

Step 2: The server unfolds B_x to the same size of B_z and takes a bitwise OR operation on the unfolded bit array and B_z to obtain a new bit array B_{xz} of size m_z. More specifically

$$B_{xz}[i] = B_x[i \bmod m_x] \vee B_z[i] \quad \forall\, i \in [0, m_z). \tag{11}$$

Step 3: The server unfolds B_y to the same size of B_z and takes a bitwise OR operation on the unfolded bit array and B_z to obtain a new bit array B_{yz} of size m_z. More specifically

$$B_{yz}[i] = B_y[i \bmod m_y] \vee B_z[i] \quad \forall\, i \in [0, m_z). \tag{12}$$

Step 4: The server unfolds B_x and B_y to the same size of B_z and takes a bitwise OR operation on the two unfolded bit arrays and B_z to obtain a new bit array B_{xyz} of size m_z. More specifically

$$B_{xyz}[i] = B_x[i \bmod m_x] \vee B_y[i \bmod m_y] \vee B_z[i] \quad \forall\, i \in [0, m_z). \tag{13}$$

Finally, given B_x, B_y, B_z, B_{xy}, B_{xz}, B_{yz}, and B_{xyz}, the MLE formula that the central server uses to estimate the three-point traffic volume of RSUs R_x, R_y, and R_z is

$$\hat{n}_{xyz} = \frac{W}{\ln\left(1 - \frac{1}{m_z}\right) + \ln(C_3) - \ln(C_4) - 2\ln(C_5)} \tag{14}$$

where W is a function of zero ratios in the bit arrays, i.e.,

$$W = \ln V_{xyz} + \ln V_x + \ln V_y + \ln V_z - \ln V_{xy} - \ln V_{xz} - \ln V_{yz} \tag{15}$$

and C_3, C_4, and C_5 are constants whose values are

$$C_3 = \frac{1}{s} \times \left(1 - \frac{s-1}{s} \times \frac{1}{m_z}\right) + \left(1 - \frac{1}{s}\right)\left(1 - \frac{1}{m_y}\right)\left(1 - \frac{s-2}{s} \times \frac{1}{m_z}\right) \tag{16}$$

$$C_4 = 1 - \frac{s-1}{s} \times \frac{1}{m_y} \tag{17}$$

$$C_5 = 1 - \frac{s-1}{s} \times \frac{1}{m_z}. \tag{18}$$

In (15), V_{xyz}, V_x, V_y, V_z, V_{xy}, V_{xz}, and V_{yz} are random variables that represent the fraction of zero bits in B_{xyz}, B_x, B_y, B_z, B_{xy}, B_{xz}, and B_{yz}, correspondingly. Their values can be easily found by counting the number of zeros in the bit arrays, which are denoted by U_{xyz}, U_x, U_y, U_z, U_{xy}, U_{xz}, and U_{yz}, respectively, and dividing them by the corresponding bit array size. For example, $V_{xyz} = U_{xyz}/m_z$, $V_x = U_x/m_x$, and $V_{xy} = U_{xy}/m_y$.

3) Derivation of the MLE Estimator $\hat{n}_{xyz}$: Now, we follow the MLE method to derive $\hat{n}_{xyz}$ given by (14). The derivation process is similar to the two-point scheme [19]: We first derive the probability $q(n_{xyz})$ for an arbitrary bit in B_{xyz} to be "0" and use $q(n_{xyz})$ to establish the likelihood function $\mathcal{L}$ to observe U_{xyz} "0" bits in B_{xyz}. Finally, maximizing $\mathcal{L}$ with respect to n_{xyz} will give the MLE estimator, i.e., $\hat{n}_{xyz}$.

Consider an arbitrary bit b in B_{xyz}. Let A_b be the event that the bth bit in B_{xyz} remains "0," then $q(n_{xyz})$ is the probability for A_b to occur. Note that the set of all vehicles passing R_x and/or R_y and/or R_z (i.e., $S_x \cup S_y \cup S_z$) can be partitioned into seven sets: $S_x \cap S_y \cap S_z$, $S_x \cap S_y - S_z$, $S_x \cap S_z - S_y$, $S_y \cap S_z - S_x$, $S_x - S_y - S_z$, $S_y - S_x - S_z$, and $S_z - S_x - S_y$. Consider the vehicles in each partition. Clearly, event A_b is equivalent to the combination of the following seven events.

I) *Event H_1: For vehicles passing R_x, R_y, and R_z* (i.e., *in set $S_x \cap S_y \cap S_z$*), *none of them have chosen bit* $(b \bmod m_x)$ *in B_x or bit* $(b \bmod m_y)$ *in B_y or bit b in B_z*. Otherwise, bit b in B_{xyz} will be "1" according to (13). There are n_{xyz} vehicles in the set $S_x \cap S_y \cap S_z$, and Fig. 1 shows the decision tree for each individual car $v \in S_x \cap S_y \cap S_z$ to not set those bits. For R_x, v should choose $b_1 \bmod m_x \neq b \bmod m_x$, and the probability is clearly $1 - (1/m_x)$ (root node in Fig. 1).

Given its selection of b_1 in RSU R_x, v has two choices in R_y: First, as shown in the left node of the second level in Fig. 1, with a probability of $1/s$, v selects the same bit b_1 in R_y (hence will not set bit $b \bmod m_y$ in B_y); second, as shown in the right node

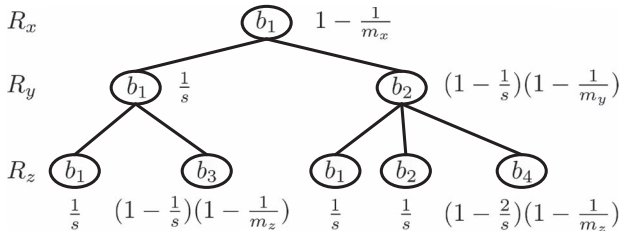


Fig. 1. Decision tree for an arbitrary bit b in B_{xyz} to remain "0" after a car v passing by all three RSUs R_x, R_y, and R_z (i.e., $v \in S_x \cap S_y \cap S_z$) sets bits in the three bit arrays (B_x, B_y, and B_z). The number inside each node represents the index that v chooses for the corresponding RSU, and the math formula next to the node represents the probability for v to choose that index, given the condition that all ancestor nodes have been chosen.

of the same level, with a probability of $1-(1/s)$, v chooses a separate bit b_2 randomly from its logical bit array LB_v, and the conditional probability for $b_2 \bmod m_y \neq b \bmod m_y$ is $1-(1/m_y)$.

Now, we examine the choices for v to not set bit b in B_z of RSU R_z given its previous selections at R_x and R_y (the five nodes at the bottom level of Fig. 1). Under its first choice at R_y, to not set bit b in B_z, v can either choose the same bit b_1 with a probability of $1/s$ (node #1) or select a separate bit b_3 randomly from LB_v with a probability of $1-(1/s)$, and the conditional probability for $b_3 \neq b$ is $1-(1/m_z)$ (node #2). Under its second choice at R_y, v can have three choices to not set bit b in B_z: 1) With a probability of $1/s$, v chooses b_1 in R_z (node #3); 2) with a probability of $1/s$, v chooses b_2 in R_z (node #4); 3) with a probability of $1-(2/s)$, v chooses a separate bit b_4 randomly from LB_v, and the conditional probability for $b_4 \neq b$ is $1-(1/m_z)$ (node #5).

Note that the probabilities in the given analysis are all conditional probabilities given that the ancestor nodes have been chosen. To sum up, the probability of H_1 is

$$\begin{aligned} Q_1 &= \left\{\left(1-\frac{1}{m_x}\right)\left[\frac{1}{s}\times\left(1-\frac{s-1}{s}\times\frac{1}{m_z}\right)\right.\right. \\ &\quad \left.\left.+\left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_y}\right)\left(1-\frac{s-2}{s}\times\frac{1}{m_z}\right)\right]\right\}^{n_{xyz}} \\ &= \left(1-\frac{1}{m_x}\right)^{n_{xyz}} C_3^{n_{xyz}}. \end{aligned} \tag{19}$$

II) *Event* H_2: *For vehicles passing only* R_x *and* R_y (i.e., *in set* $S_x \cap S_y - S_z$), *none of them have chosen bit* $(b \bmod m_x)$ *in* B_x *or bit* $(b \bmod m_y)$ *in* B_y. We analyze the probability of each individual vehicle to not set those two bits at R_x and R_y, which is exactly the same as that for Event E_1 of our two-point analysis in [19]. Since there are $n_{xy}-n_{xyz}$ cars in the set $S_x \cap S_y - S_z$, the probability of H_2 is

$$\begin{aligned} Q_2 &= \left\{\left(1-\frac{1}{m_x}\right)\left[\frac{1}{s}+\left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_y}\right)\right]\right\}^{n_{xy}-n_{xyz}} \\ &= \left(1-\frac{1}{m_x}\right)^{n_{xy}-n_{xyz}} C_4^{n_{xy}-n_{xyz}}. \end{aligned} \tag{20}$$

III) *Event* H_3: *For vehicles passing only* R_x *and* R_z (i.e., *in set* $S_x \cap S_z - S_y$), *none of them have chosen bit* $(b \bmod m_x)$ *in* B_x *or bit* b *in* B_z. There are $n_{xz}-n_{xyz}$ cars in the set $S_x \cap S_z - S_y$. Similar to H_2, we get the probability of H_3 as

$$\begin{aligned} Q_3 &= \left\{\left(1-\frac{1}{m_x}\right)\left[\frac{1}{s}+\left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_z}\right)\right]\right\}^{n_{xz}-n_{xyz}} \\ &= \left(1-\frac{1}{m_x}\right)^{n_{xz}-n_{xyz}} C_5^{n_{xz}-n_{xyz}}. \end{aligned} \tag{21}$$

IV) *Event* H_4: *For vehicles passing only* R_y *and* R_z (i.e., *in set* $S_y \cap S_z - S_x$), *none of them have chosen bit* $(b \bmod m_y)$ *in* B_y *or bit* b *in* B_z. There are $n_{yz}-n_{xyz}$ cars in the set $S_y \cap S_z - S_x$. Similar to H_2, we get the probability of H_4 as

$$\begin{aligned} Q_4 &= \left\{\left(1-\frac{1}{m_y}\right)\left[\frac{1}{s}+\left(1-\frac{1}{s}\right)\left(1-\frac{1}{m_z}\right)\right]\right\}^{n_{yz}-n_{xyz}} \\ &= \left(1-\frac{1}{m_y}\right)^{n_{yz}-n_{xyz}} C_5^{n_{yz}-n_{xyz}}. \end{aligned} \tag{22}$$

V) *Event* H_5: *For vehicles passing only* R_x (i.e., *in set* $S_x - S_y - S_z$), *none of them have chosen bit* $(b \bmod m_x)$ *in* B_x. There are $n_x - n_{xy} - n_{xz} + n_{xyz}$ cars in the set $S_x - S_y - S_z$, and each of them has a probability of $1-(1/m_x)$ to not set bit $(b \bmod m_x)$ in B_x. Therefore, the probability of H_5 is

$$Q_5 = \left(1-\frac{1}{m_x}\right)^{n_x-n_{xy}-n_{xz}+n_{xyz}}. \tag{23}$$

VI) *Event* H_6: *For vehicles passing only* R_y (i.e., *in set* $S_y - S_x - S_z$), *none of them have chosen bit* $(b \bmod m_y)$ *in* B_y. There are $n_y - n_{xy} - n_{yz} + n_{xyz}$ cars in the set $S_y - S_x - S_z$, and each of them has a probability of $1-(1/m_y)$ to not set bit $(b \bmod m_y)$ in B_y. Hence, the probability of H_6 is

$$Q_6 = \left(1-\frac{1}{m_y}\right)^{n_y-n_{xy}-n_{yz}+n_{xyz}}. \tag{24}$$

VII) *Event* H_7: *For vehicles passing only* R_z (i.e., *in set* $S_z - S_x - S_y$), *none of them have chosen bit* b *in* B_z. There are $n_z - n_{xz} - n_{yz} + n_{xyz}$ cars in the set $S_z - S_x - S_y$, and each of them has a probability of $1-(1/m_z)$ to not set bit b in B_z. Therefore, the probability of H_7 is

$$Q_7 = \left(1-\frac{1}{m_z}\right)^{n_z-n_{xz}-n_{yz}+n_{xyz}}. \tag{25}$$

Combining the given analysis, we obtain the probability $q(n_{xyz})$ for bit b in B_{xyz} to remain "0" as

$$\begin{aligned} q(n_{xyz}) &= Q_1 \times Q_2 \times Q_3 \times Q_4 \times Q_5 \times Q_6 \times Q_7 \\ &= C_3^{n_{xyz}} \times C_4^{n_{xy}-n_{xyz}} \times C_5^{n_{xz}+n_{yz}-2n_{xyz}} \\ &\quad \times \left(1-\frac{1}{m_x}\right)^{n_x} \times \left(1-\frac{1}{m_y}\right)^{n_y-n_{xy}} \\ &\quad \times \left(1-\frac{1}{m_z}\right)^{n_z-n_{xz}-n_{yz}+n_{xyz}}. \end{aligned} \tag{26}$$

Similar to our two-point analysis in [19], we know that for any bit in B_z, the probability for it to remain "0" after n_z vehicles, each choosing a random bit from B_z, is

$$q(n_z) = \left(1 - \frac{1}{m_z}\right)^{n_z} \tag{27}$$

and the expected values for V_z and V_{xyz} are

$$E(V_z) = E\left(\frac{U_z}{m_z}\right) = \frac{m_z \times q(n_z)}{m_z} = q(n_z) \tag{28}$$

$$E(V_{xyz}) = E\left(\frac{U_{xyz}}{m_z}\right) = \frac{m_z \times q(n_{xyz})}{m_z} = q(n_{xyz}). \tag{29}$$

From [19], we also get

$$\left(\frac{1 - \frac{s-1}{s} \times \frac{1}{m_y}}{1 - \frac{1}{m_y}}\right)^{n_{xy}} = \frac{V_{xy}}{V_x \times V_y}. \tag{30}$$

Similarly, we have

$$\left(\frac{1 - \frac{s-1}{s} \times \frac{1}{m_z}}{1 - \frac{1}{m_z}}\right)^{n_{xz}} = \frac{V_{xz}}{V_x \times V_z} \tag{31}$$

$$\left(\frac{1 - \frac{s-1}{s} \times \frac{1}{m_z}}{1 - \frac{1}{m_z}}\right)^{n_{yz}} = \frac{V_{yz}}{V_y \times V_z}. \tag{32}$$

Substituting [19, eqs. (10)–(13)] as well as (27)–(32) to (26) and replacing $E(V_x)$, $E(V_y)$, $E(V_z)$, and $E(V_{xyz})$ with their instance values V_x, V_y, V_z, and V_{xyz}, respectively, we have

$$V_{xyz} = \frac{V_{xy} \times V_{xz} \times V_{yz}}{V_x \times V_y \times V_z} \times \left[\frac{\left(1 - \frac{1}{m_z}\right) \times C_3}{C_4 \times C_5^2}\right]^{n_{xyz}}. \tag{33}$$

Finally, solving (33) gives the MLE estimator $\hat{n}_{xyz}$, as described in (14).

4) Computation Overhead: Note that the online coding phase works exactly the same as our two-point scheme; hence, the computation overhead for the vehicles and RSUs of our three-point scheme is exactly the same as the two-point scheme. For both schemes, when a vehicle v passes an RSU R_x, v only needs to compute two hashes to obtain an index of a random bit, and R_x only needs to set one bit in its bit array B_x. Hence, the computation overhead for each vehicle per RSU as well as for each RSU per passing vehicle are both $O(1)$.

Our three-point and two-point schemes diverge from the offline decoding phase, where the central server performs a little bit more task for three-point traffic measurement: It takes four "unfolding" and bitwise OR operations instead of one. Similar to our two-point analysis, in our three-point scheme, the "unfolding" and bitwise OR operation in step 1 costs $O(m_y)$ time, and steps 2, 3, and 4 each costs $O(m_z)$ time, leading to an overall computation overhead of $O(m_z)$, where m_z is the size of the largest bit array among the three RSUs. One can see that our three-point scheme is also very efficient.

5) Preserved Privacy: Since the way RSUs collect data from passing vehicles in our three-point scheme is no different from our two-point scheme, the preserved privacy is also the same. For both schemes, the privacy p, satisfying the requirement that the probability for any "trace" of any vehicle not to be identified must be at least p, is actually the conditional probability that states to what degree observing a same bit to be set in both bit arrays of two RSUs does not represent a common vehicle passing by both RSUs (a piece of a vehicle's trace). The reason is that the only information a vehicle v ever reports to an RSU is a bit index drawn from the same common pool uniformly at random, and the adversary can only attempt to identify the trace of a vehicle through the observation of the bits that are chosen by the vehicles to be set as "1" in both RSUs. Therefore, the preserved privacy of our three-point scheme is also given by (6), with the same outstanding conclusions as our two-point scheme [19].

C. Generalization to Multi-Point Traffic Measurement

We have proposed two schemes for privacy-preserving traffic measurement, which can efficiently measure the traffic volume among an arbitrary set of two or three RSUs. Below, we generalize our design to a multi-point traffic measurement framework for measuring traffic covering $d > 2$ locations and discuss its performance as d increases.

1) Framework: Similar to the two-point and three-point schemes, our general scheme to measure d-point traffic includes two phases: online coding phase for RSUs to collect deidentified vehicle information through varied-length bit arrays and offline decoding phase for the central server to compute d-point traffic among an arbitrary set of d RSUs based on the bit arrays. The online coding phase works exactly the same as our two-point and three-point schemes.

The offline decoding phase is also similar. At the end of each measurement period, all RSUs will send their counters and bit arrays to the central server. To compute the d-point traffic volume among an arbitrary set of d RSUs, $\{R_1, \ldots, R_d\}$, the central server will perform a series of "unfolding" and bitwise OR operations in between the bit arrays of the d RSUs to generate a series of statistical results (more specifically, the zero ratios of the resulting bit arrays) that are related to the d-point traffic volume. Again, if an MLE estimator can be derived based on these statistical results, the central server can easily compute the d-point traffic volume. Therefore, the key is to establish the relationship between the zero ratios of the bitwise ORed bit arrays and the d-point traffic volume.

Before deriving this relationship, we first define some notations. We denote the set of d RSUs as $\mathcal{S}_d$, i.e., $\mathcal{S}_d = \{R_1, \ldots, R_d\}$. Without loss of generality, we assume $m_1 \leq m_2 \leq \cdots \leq m_d$, where m_i is the size of the bit array B_i in R_i, $1 \leq i \leq d$. For an arbitrary set $\mathcal{S} \subset \mathcal{S}_d$ of RSUs, we unfold their bit arrays to the same size of the largest bit array among $\mathcal{S}$ and perform a bitwise OR over the unfolded bit arrays to obtain a new bit array $B_{\mathcal{S}}$, whose zero ratio is $V_{\mathcal{S}}$. Denote the set of vehicles passing all RSUs in $\mathcal{S}$ as $\mathcal{V}_{\mathcal{S}}$ with cardinality $\mathcal{N}_{\mathcal{S}} = |\mathcal{V}_{\mathcal{S}}|$. Clearly, we want to measure $\mathcal{N}_{\mathcal{S}_d}$.

Given an arbitrary bit b in $B_{\mathcal{S}}$, the probability for b to be "0" after an arbitrary vehicle $v \in \mathcal{V}_{\mathcal{S}}$ marks bits for all RSUs in $\mathcal{S}$ is denoted as $P_{\mathcal{S}}$. Similar to our two-point and three-point

schemes, we can derive the overall probability $q(\mathcal{N}_{\mathcal{S}_d})$ for an arbitrary bit b in $B_{\mathcal{S}_d}$ to be "0" after online coding as

$$q\left(\mathcal{N}_{\mathcal{S}_d}\right) = P_{\mathcal{S}_d}^{\mathcal{N}_{\mathcal{S}_d}} \times \prod_{1\le i\le d} P_{\mathcal{S}_d-\{R_i\}}^{\mathcal{N}_{\mathcal{S}_d-\{R_i\}}-\mathcal{N}_{\mathcal{S}_d}} \times \prod_{1\le i<j\le d} P_{\mathcal{S}_d-\{R_i,R_j\}}^{\mathcal{N}_{\mathcal{S}_d-\{R_i,R_j\}}-\mathcal{N}_{\mathcal{S}_d-\{R_i\}}-\mathcal{N}_{\mathcal{S}_d-\{R_j\}}+\mathcal{N}_{\mathcal{S}_d}} \times\cdots\times \prod_{1\le i\le d} P_{\{R_i\}}^{\mathcal{N}_{\{R_i\}}-\sum_{1\le j\le d}^{j\ne i}\mathcal{N}_{\{R_i,R_j\}}+\cdots+(-1)^{d-1}\mathcal{N}_{\mathcal{S}_d}} \tag{34}$$

where each term captures the probability for bit b in $B_{\mathcal{S}_d}$ to be "0" after the set of vehicles passing only l ($d \ge l \ge 1$) RSUs in $\mathcal{S}_d$ mark bits in the bit arrays, and the superscript in each term denotes the corresponding vehicle set cardinality derived from the inclusion–exclusion principle.

Given the above analysis, we present Algorithm 1 to iteratively derive the MLE estimator $\hat{\mathcal{N}}_{\mathcal{S}_d}$, whose correctness can be easily proved through mathematical induction, which we omit.

Algorithm 1 Iterative Algorithm to Derive the MLE Estimator $\hat{\mathcal{N}}_{\mathcal{S}_d}$

1: **Inputs**: d, P_1, P_2, P_3, $\{m_i\}_{1\le i\le d}$, $\{\mathcal{N}_{\{\mathcal{R}_i\}}\}_{1\le i\le d}$, $\hat{\mathcal{N}}_{\mathcal{S}_2}$
2: **Initialize**: $P_{\mathcal{S}_2} \leftarrow P_1$, $P_{\{\mathcal{R}_1\}} \leftarrow P_2$, $P_{\{\mathcal{R}_2\}} \leftarrow P_3$, $\mathcal{I}_{P_2} \leftarrow \{P_{\mathcal{S}_2}, P_{\{\mathcal{R}_1\}}, P_{\{\mathcal{R}_2\}}\}$
$\mathcal{N}_{\mathcal{S}_2} \leftarrow \hat{\mathcal{N}}_{\mathcal{S}_2}$, $\mathcal{I}_{\mathcal{N}_2} \leftarrow \{\mathcal{N}_{\mathcal{S}_2}, \mathcal{N}_{\{\mathcal{R}_1\}}, \mathcal{N}_{\{\mathcal{R}_2\}}\}$
3: **for** $j \leftarrow 2$ to $d-1$ **do**
4: **Step 1**: Use decision tree as Fig. 1 to obtain $P_{\mathcal{S}_{j+1}}$
5: **Step 2**: Use $P_{\mathcal{S}_{j+1}}$ and $\mathcal{I}_{P_j} = \{P_{\mathcal{S}_j}\} \bigcup \{P_{\mathcal{S}_j-\{R_i\}}\}_{1\le i\le j} \bigcup\cdots\bigcup \{P_{\{R_i\}}\}_{1\le i\le j}$ to update $\mathcal{I}_{P_{j+1}} = \{P_{\mathcal{S}_{j+1}}\} \bigcup \{P_{\mathcal{S}_{j+1}-\{R_i\}}\}_{1\le i\le j+1} \bigcup\cdots\bigcup \{P_{\{R_i\}}\}_{1\le i\le j+1}$
6: **Step 3**: Use $\mathcal{I}_{\mathcal{N}_j} = \{\mathcal{N}_{\mathcal{S}_j}\} \bigcup \{\mathcal{N}_{\mathcal{S}_j-\{R_i\}}\}_{1\le i\le j} \bigcup\cdots\bigcup \{\mathcal{N}_{\{R_i\}}\}_{1\le i\le j}$ to update $\mathcal{I}_{\mathcal{N}_{j+1}} - \{\mathcal{N}_{\mathcal{S}_{j+1}}\} = \{\mathcal{N}_{\mathcal{S}_{j+1}-\{R_i\}}\}_{1\le i\le j+1} \bigcup\cdots\bigcup \{\mathcal{N}_{\{R_i\}}\}_{1\le i\le j+1}$
7: **Step 4**: Use $\mathcal{I}_{P_{j+1}}$, $\mathcal{I}_{\mathcal{N}_{j+1}} - \{\mathcal{N}_{\mathcal{S}_{j+1}}\}$, and formula (34), and replace $q(\mathcal{N}_{\mathcal{S}_{j+1}}) = E(V_{\mathcal{S}_{j+1}})$ by its instance value $V_{\mathcal{S}_{j+1}}$, to get the MLE estimator $\hat{\mathcal{N}}_{\mathcal{S}_{j+1}} = \mathcal{F}_{j+1}(\{V^*_{\mathcal{S}_{j+1}}\})$
8: **Step 5**: $\mathcal{N}_{\mathcal{S}_{j+1}} \leftarrow \hat{\mathcal{N}}_{\mathcal{S}_{j+1}}$, $\mathcal{I}_{\mathcal{N}_{j+1}} \leftarrow \mathcal{I}_{\mathcal{N}_{j+1}} - \{\mathcal{N}_{\mathcal{S}_{j+1}}\} \bigcup \{\mathcal{N}_{\mathcal{S}_{j+1}}\}$
9: **end for**

In Algorithm 1, the inputs P_1, P_2, and P_3 are probability formulas given in [19, eqs. (6)–(8)], with the notations n_x, n_y, and n_c changed to $\mathcal{N}_{\{\mathcal{R}_1\}}$, $\mathcal{N}_{\{\mathcal{R}_2\}}$, and $\mathcal{N}_{\{\mathcal{R}_1,\mathcal{R}_2\}}$, respectively. We first initialize the probability set $\mathcal{I}_{P_2}$ and the vehicle cardinality set $\mathcal{I}_{\mathcal{N}_2}$ from the two-point derivation, which serves as the base case of our iterative algorithm. Then, the for-loop works iteratively, where the iteration j derives $\mathcal{I}_{P_{j+1}}$ and $\mathcal{I}_{\mathcal{N}_{j+1}}$ based on $\mathcal{I}_{P_j}$ and $\mathcal{I}_{\mathcal{N}_j}$ obtained from the previous iteration. Note that $\mathcal{I}_{\mathcal{N}_{j+1}}$ includes the MLE estimator $\hat{\mathcal{N}}_{\mathcal{S}_{j+1}}$ as a function $\mathcal{F}_{j+1}(\{V^*_{\mathcal{S}_{j+1}}\})$ of the zero ratios, where the set $\{V^*_{\mathcal{S}_{j+1}}\}$ contains the zero ratio $V_{\mathcal{S}}$ of $B_{\mathcal{S}}$ for all $\mathcal{S} \subset \mathcal{S}_{j+1}$, $\mathcal{S} \ne \emptyset$. Therefore, when the for-loop completes, we will obtain the MLE estimator $\hat{\mathcal{N}}_{\mathcal{S}_d}$ as a function $\mathcal{F}_d(\{V^*_{\mathcal{S}_d}\})$ of the zero ratios in the corresponding bitwise ORed bit arrays.

2) Discussion: We conclude with a quick discussion about the performance of our general d-point ($d > 1$) traffic measurement scheme. Clearly, since RSUs collect de-identified information from passing vehicles in the same way as our two-point and three-point schemes, the preserved privacy is also the same. Moreover, the computation overhead for vehicles and RSUs remains $O(1)$. However, as d increases, the computation overhead for the central server to measure d-point traffic exponentially grows. Given d bit arrays of d RSUs, the central server needs to perform unfolding and bitwise OR on every l ($2 \le l \le d$) bit arrays to generate $2^d - d - 1$ new bit arrays and compute the zero ratios in them and d original bit arrays, which costs an overall $O(2^d \times m_d)$ time.

In addition, as d increases, the measurement accuracy of our general scheme is expected to decrease. The reason is that, for each iteration j of the MLE derivation, an instance value of zero ratio $V_{\mathcal{S}_{j+1}}$ replaces its expected value $q(\mathcal{N}_{\mathcal{S}_{j+1}}) = E(V_{\mathcal{S}_{j+1}})$ to get the MLE estimator $\hat{\mathcal{N}}_{\mathcal{S}_{j+1}}$, which introduces a certain level of inaccuracy. This inaccuracy will accumulate as d increases. When d exceeds some value, e.g., 10, our d-point scheme may not work well as our current two-point and three-point schemes. However, in reality, the d-point traffic of interest usually has small values of d, such as 2 or 3. Therefore, our general scheme is still sufficient to serve for most applications.

V. Simulation Results

We perform simulations to evaluate the measurement accuracy of our solutions. In [19], we have compared our two-point scheme with two different settings, i.e., fixed bit array size m as in [17] versus fixed load factor f as in [19]. Hence, here, we focus on evaluating the measurement accuracy of our three-point scheme, also under the two different settings, i.e., fixed m versus fixed f. Note that if we set $m_x = m_y = m_z = m$ in (14), we can easily get the MLE formula for $\hat{n}_{xyz}$ under the setting of fixed bit array size m for all RSUs.

We conduct two sets of simulations. The first set is to observe the accuracy of our three-point scheme when the single-point traffic volume of three RSUs are comparable, which means that the two settings, i.e., fixed m and fixed f, are now equivalent. The simulations are controlled by the following parameters: n_x, n_y, n_z, n_{xyz}, s, and m (f). Their values are chosen as follows: $n_x = n_y = n_z = n$, where $n = 50\,000$, $100\,000$, or $500\,000$, and n_{xyz} varies from 0.01 n to 0.5 n, with a step size of 0.001 n; $s = 2, 5, 10$, and $m_x = m_y = m_z = m$ ($f_x = f_y = f_z = f$) is chosen to achieve the optimal privacy p according to (6).

Figs. 2–4 show our simulation results when $n = 50\,000$, $100\,000$, and $500\,000$, respectively. One can see that our three-point scheme is quite accurate under $s = 2$ (the measured three-point traffic volume $\hat{n}_{xyz}$ closely follows its real value n_{xyz} in the first plot of the three figures). With the increment of s, the measurement results slightly diverge from their real values (see the last plot of the three figures), which means larger

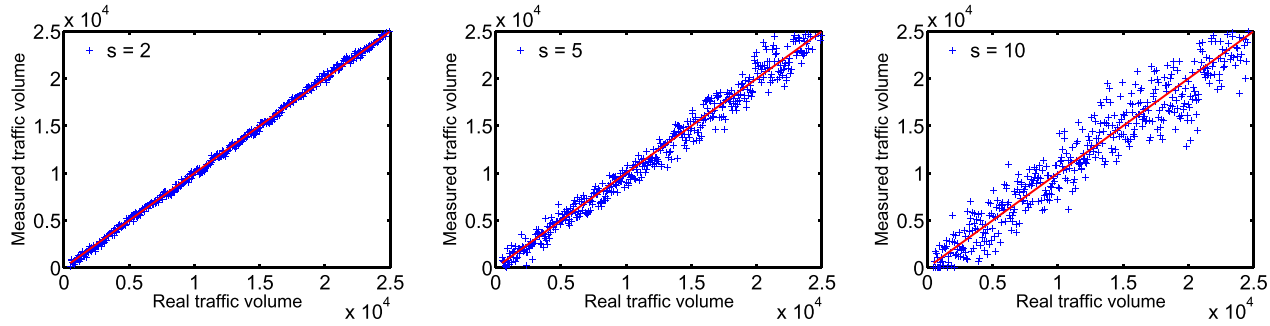


Fig. 2. Measurement accuracy with optimal privacy, $n_x = n_y = n_z = n = 50\,000$, and $n_{xyz} = [0.01n, 0.5n]$. The x-axis shows real three-point traffic volume, and the y-axis shows the measured three-point traffic volume. The three plots are controlled by s. *First Plot*: $s = 2$; *Second Plot*: $s = 5$; *Third Plot*: $s = 10$.

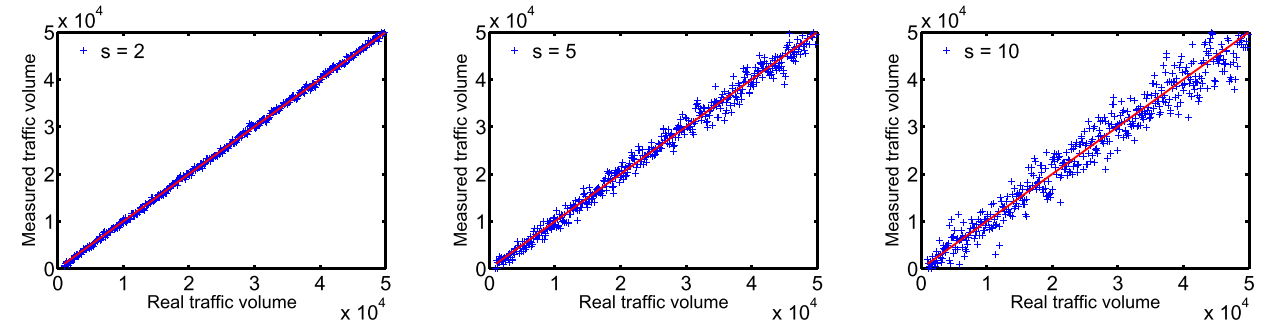


Fig. 3. Measurement accuracy with optimal privacy, $n_x = n_y = n_z = n = 100\,000$, and $n_{xyz} = [0.01n, 0.5n]$. The x-axis shows real three-point traffic volume, and the y-axis shows the measured three-point traffic volume. The three plots are controlled by s. *First Plot*: $s = 2$; *Second Plot*: $s = 5$; *Third Plot*: $s = 10$.

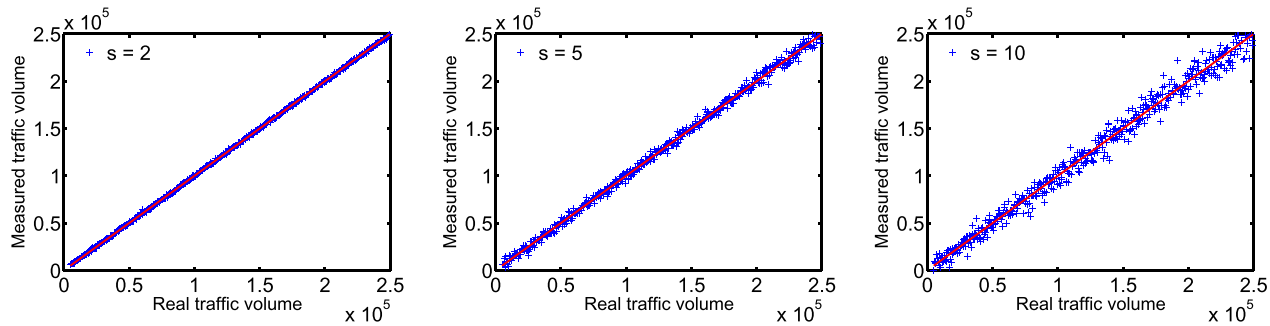


Fig. 4. Measurement accuracy with optimal privacy, $n_x = n_y = n_z = n = 500\,000$, and $n_{xyz} = [0.01n, 0.5n]$. The x-axis shows real three-point traffic volume, and the y-axis shows the measured three-point traffic volume. The three plots are controlled by s. *First Plot*: $s = 2$; *Second Plot*: $s = 5$; *Third Plot*: $s = 10$.

values of s will bring in less-accurate measurement results. This conclusion is similar to what we get from the two-point traffic measurement scheme in [17]. Intuitively, if a vehicle v has a larger logical bit array, the chance for it to report the same bit index to different RSUs decreases, which means the common information collected by different RSUs is reduced. Therefore, the accuracy will also be affected for both the two-point and the three-point measurement. One can also observe that the measurement accuracy of our three-point scheme improves along with the increment of n (compare each plot in Fig. 2 with Fig. 4), which is a natural phenomenon since our estimator is derived from the statistical MLE method.

The second set of simulations is to observe the measurement accuracy of our three-point scheme when the single-point traffic volume of three RSUs may differ. When RSUs' traffic volume is not the same, will the two settings, i.e., fixed m and fixed f, begin to show differences as we expected? If so, how will the gap between RSUs' single-point traffic volume influence the performance of our scheme under the two different settings? These are the questions to investigate.

Bearing these questions in mind, the second set of simulations is controlled by the following parameters: n_x, n_y, n_z, n_{xyz}, s, f, and m. Their values are chosen as follows: $n_x = 10\,000$, $n_z = n_y = n_x$ or $n_z = 4n_y = 16n_x$ or $n_z = 8n_y = 64n_x$, n_{xyz} varies from $0.01n_x$ to $0.5n_x$, with step size of $0.001n_x$. s is set to 2, 5, and 10. m is the fixed bit array size for all RSUs under the first setting, and f is the fixed load factor for all RSUs under the second setting. The values of m and f are chosen to guarantee minimum privacy of at least 0.5 under the two settings, respectively.

Figs. 5 and 6 show the simulation results for our three-point scheme with fixed bit array size m and fixed load factor

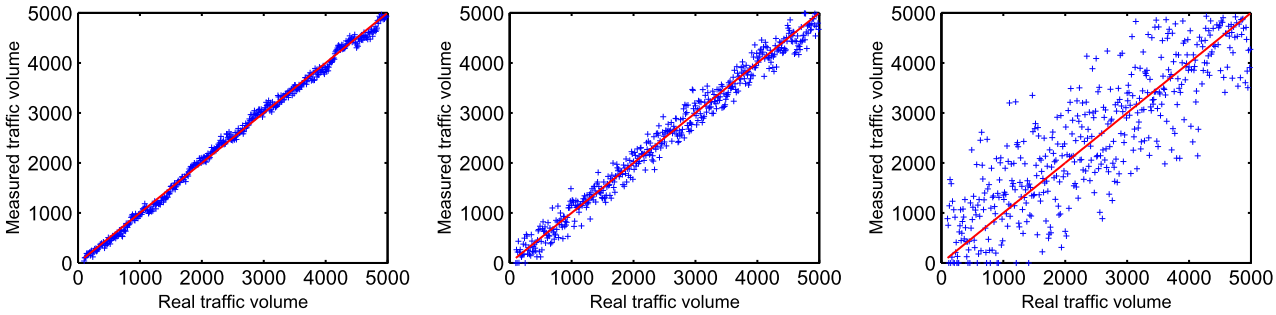


Fig. 5. Measurement accuracy of our scheme with fixed bit array size m. The x-axis shows real three-point traffic volume, and the y-axis shows measured three-point traffic volume. $s = 2$, $n_x = 10\,000$, and $n_{xyz} = [0.01n_x, 0.5n_x]$. The three plots are controlled by the ratio of n_y and n_z over n_x. *First Plot*: $n_z = n_y = n_x$; *Second Plot*: $n_z = 4n_y = 16n_x$; *Third Plot*: $n_z = 8n_y = 64n_x$.

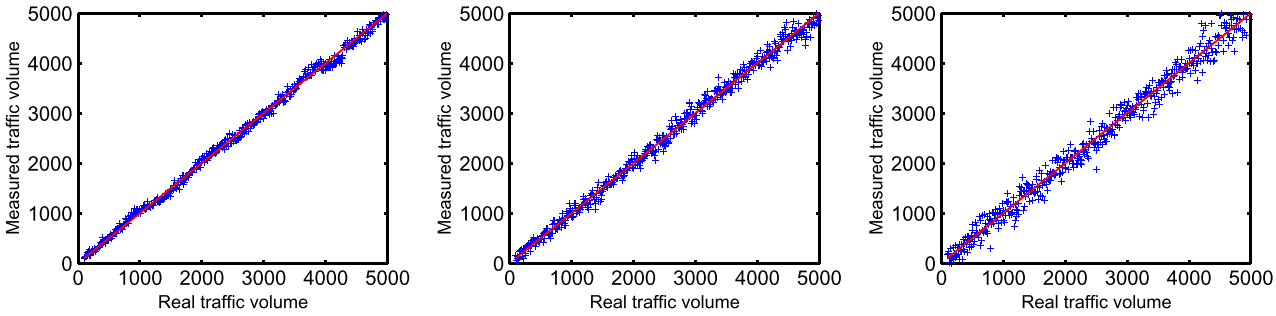


Fig. 6. Measurement accuracy of our scheme with fixed load factor f. The x-axis shows real three-point traffic volume, and the y-axis shows measured three-point traffic volume. $s = 2$, $n_x = 10\,000$, and $n_{xyz} = [0.01n_x, 0.5n_x]$. The three plots are controlled by the ratio of n_y and n_z over n_x. *First Plot*: $n_z = n_y = n_x$; *Second Plot*: $n_z = 4n_y = 16n_x$; *Third Plot*: $n_z = 8n_y = 64n_x$.

f, respectively, both under $s = 2$. Since the results for $s = 5$ and $s = 10$ are quite similar, we omit them. From the two figures, one can observe two key trends: 1) When the traffic volume of the three RSUs is comparable, i.e., $n_z = n_y = n_x$, our three-point scheme under the two settings, i.e., fixed m and fixed f, indeed achieves comparable accuracy (first plot in Figs. 5 and 6); 2) when the traffic volume varies for different RSUs, our three-point scheme achieves far better accuracy under the fixed f than under the fixed m, and the performance difference enlarges with the widening of the gap among the three RSUs' single-point traffic volume (the second and third plots in Figs. 5 and 6). The two trends observed from the measurement results of our three-point scheme also coincide with those shown in our two-point scheme.

VI. Conclusion

In this paper, we focused on privacy-preserving multi-point traffic measurement, which serves for a broad spectrum of applications in transportation engineering. As far as we know, this work is the first to study the measurement of traffic covering more than two locations. Through variable-length bit arrays, we combine the automatic traffic collection by VCPSs with a rigorous statistical MLE methodology, to propose two novel efficient schemes for two-point and three-point traffic measurement. Our schemes can protect vehicles' privacy and achieve sound measurement results. We also presented a general framework to measure traffic covering more than two locations. The proposed schemes have potential applications beyond vehicular networks, such as privacy-preserving traffic estimation in a subway system with tagged toll cards. It is also possible to use it for estimating the movement patterns of mobile users in a corporate wireless network.

References

[1] J. K. Eom, M. S. Park, T. Heo, and L. F. Huntsinger, "Improving the prediction of annual average daily traffic for nonfreeway facilities by applying a spatial statistical method," *J. Transp. Res. Board*, vol. 1968, pp. 20–29, 2006.

[2] M. C. Neto, Y. Jeong, M. K. Jeong, and L. D. Han, "AADT prediction using support vector regression with data-dependent parameters," *Expert Syst. Appl.*, vol. 36, pp. 2979–2986, Mar. 2009.

[3] B. Yang, Y. Wang, S. Wang, and Y. Bao, "Efficient local AADT estimation via SCAD variable selection based on regression models," in *Proc. Control Dec.*, 2011, pp. 1898–1902.

[4] I. Tsapakis, W. H. Schneider, and A. Nichols, "A Bayesian analysis of the effect of estimating annual average daily traffic for heavy-duty trucks using training and validation data-sets," *Transp. Plan. Technol.*, vol. 36, no. 2, pp. 201–217, 2013.

[5] "Traffic monitoring guide," U.S. Dept. Transp., Washington, DC, USA, 2013. [Online]. Available: http://www.fhwa.dot.gov/policyinformation/tmguide

[6] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[7] M. Pan, P. Li, and Y. Fang, "Cooperative communication aware link scheduling for cognitive vehicular ad-hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 760–768, May 2012.

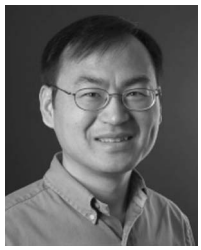
[8] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, Sep. 2010.

[9] Y. Zhu, Y. Wu, and B. Li, "Vehicular ad hoc networks and trajectory-based routing," *Internet Things*, vol. 9, pp. 143–167, 2014.

[10] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.
[11] J. Eriksson and H. Balakrishnan, "Cabernet: Vehicular content delivery using WiFi," in *Proc. MOBICOM*, 2008, pp. 199–210.
[12] U. Lee, J. Lee, J. Park, and M. Gerla, "FleaNet: A virtual market place on vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 344–355, Jan. 2010.
[13] Y. L. Morgan, "Notes on DSRC & WAVE standards suite," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 4, pp. 504–518, 4th Quart. 2010.
[14] [Online]. Available: http://www.transportation.gov
[15] Y. Lou and Y. Yin, "A decomposition scheme for estimating dynamic origin–destination flows on actuation-controlled signalized arterials," *Transp. Res. C, Emerging Technol.*, vol. 18, no. 5, pp. 643–655, Oct. 2010.
[16] Y. Zhou, S. Chen, Z. Mo, and Y. Yin, "Privacy preserving origin–destination flow measurement in vehicular cyber-physical systems," in *Proc. IEEE CPSNA*, 2013, pp. 32–37.
[17] Y. Zhou, Q. Xiao, Z. Mo, S. Chen, and Y. Yin, "Privacy-preserving point-to-point transportation traffic measurement through bit array masking in intelligent cyber-physical road systems," in *Proc. IEEE CPSCom*, 2013, pp. 826–833.
[18] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, "Privacy-preserving transportation traffic measurement in intelligent cyber-physical road systems," *IEEE Trans. Veh. Technol.*, to be published, DOI 10.1109/TVT.2015.2436395.
[19] Y. Zhou, S. Chen, Z. Mo, and Q. Xiao, "Point-to-point traffic volume measurement through variable-length bit array masking in vehicular cyber-physical systems," in *Proc. IEEE ICDCS*, 2015, pp. 51–60.
[20] Google map's time-in-traffic feature. [Online]. Available: http://mashable.com/2012/03/29/google-maps-traffic-data
[21] T. Jeske, "Floating car data from Smartphones: What Google and Waze know about you and how hackers can control traffic," in *Proc. BlackHat Europe*, 2013, pp. 1–12.
[22] M. Yoon, T. Li, S. Chen, and J. Kwon Peir, "Fit a spread estimator in small memory," in *Proc. INFOCOM*, 2009, pp. 504–512.
[23] T. Li, S. Chen, and Y. Qiao, "Origin–destination flow measurement in high-speed networks," in *Proc. INFOCOM*, 2012, pp. 2526–2530.
[24] "Traffic Volume Report," New York State Dept. Transp, New York, NY, USA, 2012. [Online]. Available: https://www.dot.ny.gov/divisions/engineering/technicalservices/highway-data-services/traffic-data
[25] SpoofMAC, "Spoof your MAC address," 2015. [Online]. Available: https://github.com/feross/SpoofMAC
[26] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym schemes in vehicular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, 1st Quart. 2015.
[27] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in VANETs," *IEEE Trans. Veh. Technol.*, 2011, pp. 86–96.
[28] Z. Ma, F. Kargl, and M. Weber, "Measuring long-term location privacy in vehicular communication systems," *Comput. Commun.*, vol. 33, no. 12, pp. 1414–1427, 2010.

Yian Zhou (M'15) received the B.S. degree in computer science and economics from Peking University, Beijing, China, in 2010. She is currently working toward the Ph.D. degree in computer and information science and engineering with the University of Florida, Gainesville, FL, USA.

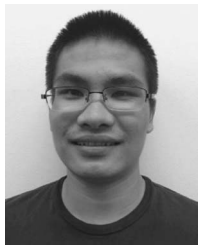
Her advisor is Dr. S. Chen. Her research interests include traffic flow measurement, cyber-physical systems, big network data, security and privacy, and cloud computing.



Shigang Chen (A'03–M'04–SM'12) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1993 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 1996 and 1999, respectively, all in computer science.

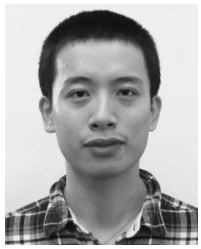
After graduation, he worked for Cisco Systems, San Jose, CA, USA, for three years before joining the University of Florida, Gainesville, FL, USA, in 2002, where he is currently a Professor with the Department of Computer and Information Science and Engineering. From 2002 to 2003, he served on the Technical Advisory Board for Protego Networks. He has published more than 130 peer-reviewed journal/conference papers. He is the holder of 12 U.S. patents. His research interests include computer networks, Internet security, wireless communications, and distributed computing.

Dr. Chen is an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKS, *Computer Networks*, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. From 2010 to 2013, he served on the Steering Committee of the IEEE International Workshop on Quality of Service. He received the IEEE Communications Society Best Tutorial Paper Award in 1999 and the National Science Foundation CAREER Award in 2007.



You Zhou (S'13) received the B.S. degree in electronic information engineering from the University of Science and Technology of China, Hefei, China, in 2013. He is currently working toward the Ph.D. degree in computer and information science and engineering with the University of Florida, Gainesville, FL, USA.

His advisor is Dr. S. Chen. His research interests include network security and privacy, big network data, and Internet of things.



Min Chen (S'14) received the B.E. degree in information security from the University of Science and Technology of China, Hefei, China, in 2011. He is currently working toward the Ph.D. degree with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA.

His advisor is Dr. S. Chen. His research interests include Internet of things, big network data, next-generation radio-frequency identification systems, and network security.

Qingjun Xiao (M'12) received the B.Sc. degree in computer science from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003; the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2007; and the Ph.D. degree from the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong, in 2011.

He is currently an Assistant Professor with Southeast University, Nanjing. His research interests include protocols and distributed algorithms in wireless sensor networks, radio-frequency identification systems, and network traffic measurement.

Dr. Xiao is a member of the IEEE Communications Society and the Association for Computing Machinery.