Privacy-Preserving Transportation Traffic Measurement in Intelligent Cyber-physical Road Systems

Yian Zhou, *Student Member, IEEE*, Zhen Mo, *Student Member, IEEE*, Qingjun Xiao, *Member, IEEE*, Shigang Chen, *Senior Member, IEEE*, and Yafeng Yin

***Abstract*—Traffic measurement is a critical function in transportation engineering. We consider privacy-preserving point-to-point traffic measurement in this paper. We measure the number of vehicles traveling from one geographical location to another by taking advantage of capabilities provided by the intelligent cyber-physical road systems (CPRSs) that enable automatic collection of traffic data. The challenge is to allow the collection of aggregate point-to-point data while preserving the privacy of individual vehicles. We propose a novel measurement scheme, which utilizes bit arrays to collect "masked" data and adopts maximum-likelihood estimation (MLE) to obtain the measurement result. Both mathematical proof and simulation demonstrate the practicality and scalability of our scheme.**

***Index Terms*—Cyber-physical systems, maximum-likelihood estimation (MLE), privacy, transportation traffic measurement.**

I. Introduction

New technologies in vehicular communications and networking [1]–[6] have greatly advanced the design of intelligent cyber-physical road systems (CPRSs). To fully realize the potential of such systems and improve the capacity of existing infrastructures, traffic measurement is a critical function in transportation engineering [7]. There are two categories of traffic statistics, i.e., "*point*" statistics and "*point-to-point*" statistics. Point statistics describe the number of vehicles traversing a specific *point* (location). Various prediction models have been proposed to estimate them [8]–[11]. Point-to-point statistics describe the number of vehicles traveling between two *points* (locations). They are essential inputs to a variety of studies, including estimation of traffic link flow distribution as part of investment plan and calculation of road exposure rates as part of safety analysis, etc. Although some point-to-point statistics may be inferred from point data [12], the practicality is limited by either high computation overhead or degraded measurement accuracy. As for direct measurement of "point-to-point" traffic, little work has been done particularly when drivers' location privacy is concerned.

This paper considers the important problem of privacy-preserving *point-to-point* transportation traffic measurement. The set of vehicles traveling from one geographical location to another is modeled as a traffic flow, and the flow size is the number of vehicles in the set. To enable automatic collection of traffic flow data, we take advantage of intelligent CPRSs, which integrate the latest technologies in wireless communications and on-board computer processing into transportation systems [13], [14]. In particular, IntelliDrive [15] from the U.S. Department of Transportation [16] envisions a nationwide system where vehicles communicate with roadside equipments (RSEs) in real time via dedicated short-range communications (DSRC). In CPRSs, vehicles may report their IDs to RSEs when they pass by, and this information can be used by the authority to measure traffic flows. However, if a vehicle keeps transmitting its unique identifier to RSEs, the information will enable others to track its entire moving history. As increasingly more people are concerned about their location privacy, the degree of privacy that a traffic measurement scheme preserves will directly affect its applicability.

To address the concerns of privacy, there are many issues that we need to consider. First of all, we need a criterion to tell what is good privacy and what is not. In this paper, we capture the essence of privacy in traffic flow measurement and quantify it as a probability that a potential tracker cannot identify any trace of any vehicle. Second, given this criterion, how can we preserve the optimal privacy? Apparently, the better the privacy, the more applicable the measurement scheme. Furthermore, to protect the privacy of vehicles, only randomized and de-identified information is collected. How can we achieve sound measurement accuracy based on information that looks totally random?

In this paper, we propose a novel scheme for privacy-preserving traffic flow measurement. It utilizes bit arrays to encode "masked" data sent from vehicles to RSEs and adopts maximum-likelihood estimation (MLE) to obtain measurement

Manuscript received October 10, 2014; revised March 26, 2015; accepted April 20, 2015. Date of publication May 21, 2015; date of current version May 12, 2016. This work was supported in part by the National Science Foundation under Grant CPS 0931969, Grant NeTS 1115548, and Grant NeTS 1409797 and in part by the Center for Multimodal Solutions for Congestion Mitigation under a grant sponsored by the U.S. Department of Transportation. The review of this paper was coordinated by Prof. Y. Zhang.

Y. Zhou, Z. Mo, and S. 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; zmo@cise.ufl.edu; sgchen@cise.ufl.edu).

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

Y. Yin is with the Department of Civil and Coastal Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: yafeng@ce.ufl.edu).

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.2015.2436395

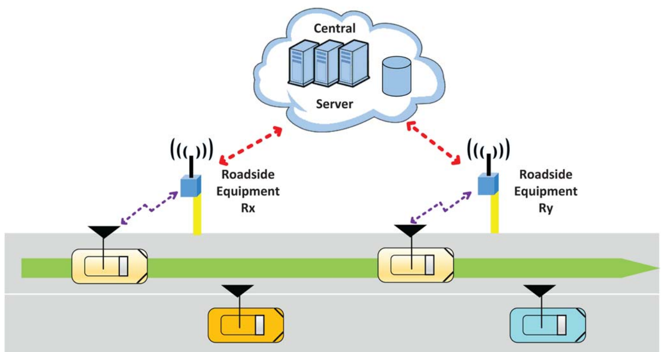


Fig. 1. Intelligent CPRS model.

results. The measurement accuracy and preserved privacy are analyzed through both mathematical proof and simulations, which demonstrate the applicability of our scheme.

The remainder of this paper is organized as follows. Section II gives the preliminaries. Section III presents our scheme and its analysis. Section IV shows simulation results. Section V summarizes related work. Section VI draws the conclusion.

II. Preliminaries

A. System Model

We consider an intelligent CPRS model, as shown in Fig. 1, which involves three types of entities, namely, vehicles, RSEs, and a central server. Each vehicle has a unique ID, i.e., its vehicle identification number. Each RSE also has its unique ID. Both vehicles and RSEs are equipped with computing and communication capabilities, e.g., on-board computer chips and communication modules. Vehicles communicate with RSEs in real time via DSRC [16]. RSEs are connected to the central server through wired or wireless means. They collect information from vehicles and transfer it to the central server on a periodical basis.

B. Problem Statement

We define a traffic flow between one RSE-equipped location and another RSE-equipped location as the set of vehicles traveling between the two locations during a measurement period. The size of the traffic flow is the number of vehicles in this set. Our problem is to measure the sizes of traffic flows in a road system between all pairs of locations where RSEs are installed while protecting vehicles' privacy. To achieve the privacy-preserving end, we need a solution in which a vehicle never transmits any fixed identifier. Ideally, the information transmitted by the vehicles to the RSEs looks totally random, out of which neither the identity nor the trajectory of any vehicle can be pried with high probability.

We also assume that a special medium access control (MAC) protocol 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.

C. Threat Model

We assume a semi-honest model for the RSEs. On the one hand, all RSEs are from trustworthy authorities, which can be enforced by authentication based on PKI. The vehicles can use the public key certificate broadcasted by RSEs, which they obtained from trusted third parties, to verify the RSEs. On the other hand, the authorities may exploit the information collected by RSEs to track individual vehicles when they need to do so. For instance, if a vehicle transmits any fixed identifier upon each query, that identifier can be used for tracking purposes.

Note that there are other ways to track a vehicle, for example, tailgating the vehicle or setting cameras near RSEs 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 automatical tracking caused by the traffic flow measurement scheme itself.

D. Performance Metrics

In this paper, we consider three performance metrics to evaluate a traffic flow measurement scheme, namely, measurement accuracy, computation overhead, and preserved privacy. They are defined in the following.

1) Measurement Accuracy: Let n_c be the real size of a traffic flow between a pair of locations and $\hat{n}_c$ be the corresponding measurement result. We specify the measurement accuracy through a parameter β such that the probability for n_c to fall into the interval $[\hat{n}_c \cdot (1-\beta), \hat{n}_c \cdot (1+\beta)]$ must be at least α, where α is a predetermined parameter in the range of [0, 1]. For a given probability α, a smaller value of β means better measurement results. For example, when $\alpha = 95\%$, a solution with $\beta = 0.05$ is more accurate than a solution with $\beta = 0.1$ because the former ensures that the measured traffic flow size has a probability of 95% to be within $\pm 5\%$ deviation from the real value, whereas the latter only ensures the measured result to be within $\pm 10\%$ deviation from the real value under the same probability.

2) Computation Overhead: We consider the computation overhead for vehicles, RSEs, and the central server. For vehicles, we measure the computation overhead for each vehicle per RSE en route. For RSEs, we measure the computation overhead for each RSE per passing vehicle. For the central server, we measure the computation overhead for it to measure the traffic flow size for a pair of RSEs.

3) Preserved Privacy: We capture the essence of privacy preservation in point-to-point transportation traffic measurement, which is allowing the tracker 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, which 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 RSEs 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 tracker 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., $1 - p$.

III. Privacy-Preserving Point-to-Point Transportation Traffic Measurement

Here, we present our novel scheme for privacy-preserving point-to-point transportation traffic measurement. There are two phases for each measurement period, namely, online coding and offline decoding. Online coding is an interaction between vehicles and RSEs to securely collect information for traffic flow measurement. Later in the offline decoding phase, the central server will use this information to compute traffic flow sizes. We first show the two measurement phases, and then evaluate our scheme with respect to the three performance metrics described in Section II-D.

A. Online Coding Phase

As presented in our previous work [17], in our scheme, each RSE R_x maintains a counter n_x, which keeps track of the total number of vehicles passing by during the current measurement period. R_x also maintains a bit array B_x with a fixed length m to mask vehicle identities. At the beginning of each measurement period, n_x and all the bits in B_x are set to zeros. In addition, each vehicle v has a logical bit array LB_v, which consists of s $(1 < s < m)$ bits randomly selected from B_x. The indices of these bits in B_x 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 whose purpose is to arbitrarily alter the hash result, and K_v is the private key of v to protect the privacy of its logical bit array.

The online coding phase is quite simple. RSEs broadcast queries in preset intervals (e.g., once a second), ensuring that each passing vehicle receives at least one query and meanwhile giving enough time for the vehicle to reply. Collisions can be resolved through well-established carrier sense multiple-access or time-division multiple-access protocols, which are not the focus of this paper. Every query that an RSE sends out includes the RSE's registered application provider identifier (RID) and its public key certificate. Suppose that a vehicle, whose ID is v, receives a query from an RSE, whose ID is R_x. The vehicle first verifies the certificate and then uses the RSE's public key to authenticate the RSE. After verifying that R_x is from a trustworthy authority, the vehicle v randomly selects a bit from its logical bit array LB_v by computing an index $b = H(v \oplus K_v \oplus X[H(R_x) \bmod s])$. The vehicle v then sends the resulting index b to the RSE R_x. Upon receiving the index b, R_x will first increase its counter n_x by 1 and then set the bth bit in B_x to 1

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

Note that the same vehicle may transmit different bit indices at two RSEs. The probability for this to happen is $1 - (1/s)$, which is larger when the size 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, we improve privacy protection. This is true even when there is a single vehicle passing through two RSEs.

B. Offline Decoding Phase

At the end of each measurement period, all RSEs will send their counters and bit arrays to the central server, which then performs the offline measurement. We employ the MLE [18] to measure the sizes of traffic flows based on the counters and bit arrays.

Suppose that the set of vehicles that pass RSE $R_x(R_y)$ 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 RSEs R_x and R_y is $S_x \cap S_y$. Denote its cardinality as n_c, which is the value that we want to measure. Furthermore, denote by S the subset of vehicles in $S_x \cap S_y$ that happen to set the same bit in B_x and B_y, where B_x and B_y are the bit arrays at R_x and R_y, respectively. Let n_o be the cardinality of S, i.e., $n_o = |S|$. Clearly, $S \subseteq S_x \cap S_y$ and $0 \le n_o \le n_c$. For any vehicle, it has the same probability $1/s$ to set any bit in its s-bit logical bit array. As a result, the probability for an arbitrary vehicle v from $S_x \cap S_y$ to select the same bit in both B_x and B_y is $s \times (1/s) \times (1/s) = 1/s$. Therefore, the number of such vehicles n_o is binomially distributed according to $B(n_c, 1/s)$. Accordingly, the probability for $n_o = z (0 \le z \le n_c)$ is

$$P(n_o = z) = \binom{n_c}{z} \left(\frac{1}{s}\right)^z \left(1 - \frac{1}{s}\right)^{n_c - z}. \tag{2}$$

Given the counters n_x and n_y and bit arrays B_x and B_y, we measure n_c as follows: First, take a bitwise AND of B_x and B_y and denote the resulting bit array as B_c. Namely

$$B_c[i] = B_x[i] \wedge B_y[i], \qquad \forall\, i \in [0, m-1]. \tag{3}$$

We can easily find out the number of 0's in B_c, denoted by U_c. In the following, we will analyze the probability for an arbitrary bit in B_c to remain "0" after the online coding phase and use it to establish the likelihood function for us to observe U_c "0" bits in B_c. Maximizing the likelihood function with respect to n_c will give the MLE estimate of n_c.

Clearly, the event for an arbitrary bit b in B_c to remain "0" after online coding is equivalent to the combination of the following two events.

1) *Event 1: None of the vehicles in S has chosen b at R_x and R_y.* If a vehicle $v \in S$ chooses b, then bit b in B_x and B_y are both set to "1" by v (hence, bit b in B_c is also "1"). Since each vehicle has probability $1/m$ to set bit b to "1", the probability for the vehicle not to choose bit b is $1 - (1/m)$. There are n_o vehicles in S. Therefore, the probability for the first event to happen is the following:

$$q_1 = \left(1 - \frac{1}{m}\right)^{n_o}. \tag{4}$$

2) *Event 2: Either none of the vehicles in $S_x - S$ has chosen b at R_x or none of the vehicles in $S_y - S$ has chosen b at R_y.* Otherwise, bit b in both B_x and B_y will be "1" (hence, bit b in B_c is "1"). The probability for bit b not chosen by any vehicle in $S_x - S$ is $(1 - (1/m))^{n_x - n_o}$,

and the probability for bit b not chosen by any vehicle in $S_y - S$ is $(1-(1/m))^{n_y-n_o}$. Therefore, the probability for the second event to happen is

$$q_2 = 1-\left(1-\left(1-\frac{1}{m}\right)^{n_x-n_o}\right)\times\left(1-\left(1-\frac{1}{m}\right)^{n_y-n_o}\right)$$
$$=\left(1-\frac{1}{m}\right)^{n_x-n_o}+\left(1-\frac{1}{m}\right)^{n_y-n_o}-\left(1-\frac{1}{m}\right)^{n_x+n_y-2n_o}. \tag{5}$$

Combining this analysis, the conditional probability for bit b in B_c to remain "0" given $n_o = z$ is $q_1 \times q_2$, i.e.,

$$q(n_c|n_o=z) = q_1 \times q_2$$
$$=\left(1-\frac{1}{m}\right)^{n_x}+\left(1-\frac{1}{m}\right)^{n_y}-\left(1-\frac{1}{m}\right)^{n_x+n_y-z}. \tag{6}$$

Given $q(n_c|n_o = z)$ and the distribution of n_o, the overall probability $q(n_c)$ for an arbitrary bit b in B_c to remain "0" is

$$q(n_c) = \sum_{z=0}^{n_c} q(n_c|n_o=z)\times P(n_o=z)$$
$$=\sum_{z=0}^{n_c} q(n_c|n_o=z)\times\binom{n_c}{z}\left(\frac{1}{s}\right)^z\left(1-\frac{1}{s}\right)^{n_c-z}$$
$$=\left(1-\frac{1}{m}\right)^{n_x}+\left(1-\frac{1}{m}\right)^{n_y}-\left(1-\frac{1}{m}\right)^{n_x+n_y}C^{n_c} \tag{7}$$

where C is a value determined by s and m only

$$C = \left(1-\frac{1}{s}\right)+\frac{1}{s}\times\frac{1}{1-\frac{1}{m}}. \tag{8}$$

Knowing that each bit in B_c has a probability $q(n_c)$ to remain "0", we can establish the likelihood function for us to observe U_c "0" bits in B_c (hence, $m - U_c$ "1" bits in B_c)

$$\mathcal{L} = (q(n_c))^{U_c}\times(1-q(n_c))^{m-U_c}. \tag{9}$$

The MLE estimate of n_c is the optimal value of n_c that maximizes the likelihood function in (9)

$$\hat{n_c} = \arg\max_{n_c}\{\mathcal{L}\}. \tag{10}$$

To find $\hat{n_c}$, we take a logarithm on both sides of (9)

$$\ln\mathcal{L} = U_c\times\ln q(n_c) + (m-U_c)\times\ln(1-q(n_c)). \tag{11}$$

Take the first-order derivative of (11), we have

$$\frac{d\ln\mathcal{L}}{dn_c} = \left(\frac{U_c}{q(n_c)}-\frac{m-U_c}{1-q(n_c)}\right)\times q'(n_c) \tag{12}$$

where $q'(n_c)$ can be computed from (7) as follows:

$$q'(n_c) = \frac{dq(n_c)}{dn_c}$$
$$= -\left(1-\frac{1}{m}\right)^{n_x+n_y}\times C^{n_c}\times\ln C. \tag{13}$$

To compute $\hat{n_c}$, we set the right side of (12) to 0

$$\left(\frac{U_c}{q(n_c)}-\frac{m-U_c}{1-q(n_c)}\right)\times q'(n_c) = 0. \tag{14}$$

Observe from (13) that $q'(n_c)$ cannot be 0 when $m > 1$ and $s > 1$. Therefore, we have

$$\frac{U_c}{q(n_c)}-\frac{m-U_c}{1-q(n_c)} = 0. \tag{15}$$

Substituting (7) into (15), we obtain the MLE estimator $\hat{n_c}$ of the desired traffic flow size n_c as follows:

$$\hat{n_c} = \frac{1}{\ln\left(1-\frac{1}{s}+\frac{1}{s}\times\frac{1}{1-\frac{1}{m}}\right)}\left\{-(n_x+n_y)\ln\left(1-\frac{1}{m}\right)\right.$$
$$\left.+\ln\left(\left(1-\frac{1}{m}\right)^{n_x}+\left(1-\frac{1}{m}\right)^{n_y}-\frac{U_c}{m}\right)\right\}. \tag{16}$$

C. Measurement Accuracy

In the subsequent sections, we discuss the performance of our scheme with respect to the three performance metrics described in Section II-D. We start with analyzing the measurement accuracy. The standard theory of MLE [19] says that when m, n_x, and n_y are large enough, the MLE estimator $\hat{n_c}$ approximately follows the normal distribution

$$\hat{n}_c \sim \text{Norm}\left(n_c, \frac{1}{\mathcal{I}(\hat{n}_c)}\right) \tag{17}$$

where $\mathcal{I}(\hat{n}_c)$ is the Fisher information on $\mathcal{L}$, which is defined as

$$\mathcal{I}(\hat{n}_c) = -E\left[\frac{d^2\ln\mathcal{L}}{dn_c^2}\right]. \tag{18}$$

We compute the second-order derivative of $\ln\mathcal{L}$ from (12)

$$\frac{d^2\ln\mathcal{L}}{dn_c^2} = \left(-\frac{U_c\cdot q'(n_c)}{q^2(n_c)}-\frac{(m-U_c)\cdot q'(n_c)}{(1-q(n_c))^2}\right)\cdot q'(n_c)$$
$$+\left(\frac{U_c}{q(n_c)}-\frac{m-U_c}{1-q(n_c)}\right)\cdot q'(n_c)\cdot\ln C \tag{19}$$

where C is given in (8), and $q'(n_c)$ is given in (13).

For an arbitrary bit b in B_c, it has the probability $q(n_c)$ to remain "0". U_c is the number of "0"s in B_c. Therefore, U_c follows a binomial distribution $B(m, q(n_c))$. Accordingly

$$E(U_c) = m\cdot q(n_c). \tag{20}$$

Substituting (19) and (20) to compute (18), we have

$$\mathcal{I}(\hat{n}_c) = \left(\frac{m \cdot q'(n_c)}{q(n_c)} + \frac{m \cdot q'(n_c)}{1 - q(n_c)}\right) \times q'(n_c) = \frac{m\,(q'(n_c))^2}{q(n_c)\,(1 - q(n_c))}. \quad (21)$$

According to (17), the variance of $\hat{n}_c$ is

$$Var(\hat{n}_c) = \frac{1}{\mathcal{I}(\hat{n}_c)} = \frac{q(n_c)\,(1 - q(n_c))}{m\,(q'(n_c))^2}. \quad (22)$$

Therefore, the confidence interval of our measurement is

$$\hat{n}_c \pm Z_\alpha \times \sqrt{\frac{q(n_c)\,(1 - q(n_c))}{m\,(q'(n_c))^2}} \quad (23)$$

where α is the confidence level, and Z_α is the α percentile for the standard Gaussian distribution [20]. For example, when $\alpha = 95\%$, $Z_\alpha = 1.6$.

D. Preserved Privacy

Next, we evaluate the preserved privacy of our measurement scheme. Note that, in our scheme, the only information that a vehicle v ever transmits to an RSE en route is an index of a bit b randomly selected from its s-bit logical bit array LB_v. From the tracker's point of view, it can only identify the trace of a vehicle passing by two RSEs R_x and R_y through the observation of the bits that are set to "1" in both B_x and B_y; these bits will be "1" in B_c. Therefore, the preserved privacy of our scheme is actually a conditional probability, which tells to what degree an observed "1" in B_c does not represent a common vehicle passing by both R_x and R_y. We derive this conditional probability in the following.

First, consider the probability for the tracker to observe an arbitrary bit b to be set to "1" in both its B_x and B_y (event A), i.e., $P(A)$. Obviously, the probability $P(A)$ is equal to 1 minus $q(n_c)$ given our analysis in Section III-B

$$P(A) = 1 - \left(1 - \frac{1}{m}\right)^{n_x} - \left(1 - \frac{1}{m}\right)^{n_y} + \left(1 - \frac{1}{m}\right)^{n_x + n_y} \times C^{n_c} \quad (24)$$

where C is given in (8).

Second, consider the conditional probability for such a bit b to not represent a common vehicle passing both R_x and R_y (event E), i.e., $P(E|A)$. This is the privacy p that we want to derive. Note that event E happens if and only if bit b in B_x is set only by vehicles passing only RSE R_x (i.e., in set $S_x - S_y$) and bit b in B_y is set only by vehicles passing only RSE R_y (i.e., in set $S_y - S_x$). Denote these two events as E_x and E_y, respectively. There are n_x (n_y) vehicles passing R_x (R_y), and n_c vehicles among them pass both R_x and R_y. Since each vehicle has a probability $1/m$ to set bit b to "1," the probability for E_x (E_y) to happen is

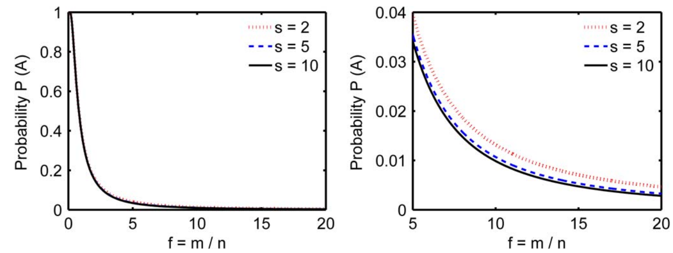


Fig. 2. $n_x = n_y = n = 50\,000$, and $n_c = 5000$. (Left plot) Probability $P(A)$ when m varies from $0.1n$ to $20n$, controlled by different $s = 2,\ 5,$ and 10. (Right plot) Zoom-in of the left plot when m varies from $5n$ to $20n$.

$$P(E_x) = \left(1 - \left(1 - \frac{1}{m}\right)^{n_x - n_c}\right) \times \left(1 - \frac{1}{m}\right)^{n_c} \quad (25)$$

$$P(E_y) = \left(1 - \left(1 - \frac{1}{m}\right)^{n_y - n_c}\right) \times \left(1 - \frac{1}{m}\right)^{n_c}. \quad (26)$$

Combining this analysis, we have the formula for the preserved privacy of our scheme, i.e.,

$$p = P(E|A) = \frac{P(E_x) \times P(E_y)}{P(A)} = \frac{\left(\left(1 - \frac{1}{m}\right)^{n_c} - \left(1 - \frac{1}{m}\right)^{n_x}\right) \times \left(\left(1 - \frac{1}{m}\right)^{n_c} - \left(1 - \frac{1}{m}\right)^{n_y}\right)}{P(A)} \quad (27)$$

where $P(A)$ is given in (24).

Observe that there are two parameters, i.e., s and m, that determine the value of $P(E|A)$. Among them, s only appears in the denominator $P(A)$, and it influences $P(E|A)$ through varying the value of $P(A)$. m influences both the denominator and the numerator. In the following, we first examine the influence of s on $P(A)$ (hence, on $P(E|A)$) and then analyze how m affects the value of $P(E|A)$.

1) Influence of s on $P(A)$: To examine how s affects $P(A)$, we take partial derivative of (24) with respect to s

$$\frac{\partial P(A)}{\partial s} = -\left(1 - \frac{1}{m}\right)^{n_x + n_y} \times \frac{n_c}{(m-1)s^2} C^{n_c - 1} \quad (28)$$

where C is given in (8). Clearly, $(\partial P(A)/\partial s) < 0$. Therefore, with the increment of s, the value of $P(A)$ decreases, and in turn, the value of $P(E|A)$ increases. In other words, privacy will be better with a larger value of s. The numerical results are shown in Fig. 2 where $n_x = n_y = n = 50\,000$; $n_c = 5000$; and $s = 2,\ 5,$ and 10, corresponding to three curves in each plot. Clearly, as s increases, the probability $P(A)$ decreases.

Another observation from the numerical results is that, when $s > 5$, the difference in probability $P(A)$ under different s becomes quite small. For instance, with $m \in [5n,\ 20n]$, the difference in $P(A)$ when $s = 5$ and $s = 10$ is smaller than 0.0005 (see the two lower curves in the right plot in Fig. 2). When $n > 10$, this difference becomes negligible. Therefore, when we analyze the effect of m on $P(E|A)$ in the following section and later when we set up the parameters for our simulations, we will only consider the cases of $s = 2,\ 5,$ and 10, with an established understanding that larger values of s will only make negligible difference.

2) Influence of m on $P(E|A)$: To examine the effect of m on $P(E|A)$, we take the partial derivative of (27) with respect to m and obtain the following:

$$\frac{\partial P(E|A)}{\partial m} = \frac{\frac{\partial P(E)}{\partial m} \times P(A) - \frac{\partial P(A)}{\partial m} \times P(E)}{P(A)^2} \tag{29}$$

where $P(E) = P(E_x) \times P(E_y)$. $P(E_x)$ and $P(E_y)$ are given in (25) and (26), respectively. Therefore, the partial derivative of $P(E)$ with respect to m is

$$\frac{\partial P(E)}{\partial m} = \frac{1}{m(m-1)}\left[(n_x+n_y)\left(1-\frac{1}{m}\right)^{n_x+n_y} + 2n_c\left(1-\frac{1}{m}\right)^{2n_c} - (n_c+n_x)\left(1-\frac{1}{m}\right)^{n_c+n_x} - (n_c+n_y)\left(1-\frac{1}{m}\right)^{n_c+n_y}\right]. \tag{30}$$

In addition, from (24), we can compute the derivative of $P(A)$ with respect to m

$$\frac{\partial P(A)}{\partial m} = \frac{1}{m^2}\left[-n_x\left(1-\frac{1}{m}\right)^{n_x-1} - n_y\left(1-\frac{1}{m}\right)^{n_y-1} + \left(1-\frac{1}{m}\right)^{n_x+n_y-2} \cdot C^{n_c} \cdot \left((n_x+n_y)\left(1-\frac{1}{m}\right) - \frac{n_c}{s \cdot C}\right)\right]. \tag{31}$$

We have proved that $(\partial P(A)/\partial m) < 0$, which means that $P(A)$ will decrease with the increment of m. In addition, $(\partial P(E)/\partial m)$ will be also negative when m exceeds a certain value, which means that $P(E)$ will also decrease with the increment of m afterward. Intuitively, increasing m gives each vehicle a smaller chance $1/m$ to set an arbitrary bit b. Hence, $P(E)$ and $P(A)$ also drop. The effect that m has on $P(E|A)$ is twofold: On one hand, the increment of m decreases the denominator $P(A)$, which improves the privacy; on the other hand, the increment of m decreases the numerator $P(E)$, which reduces the privacy. With the combination of the two effects, the partial derivative of $P(E|A)$ with respect to m can be positive, negative, or 0, according to (29). Therefore, given a value of s, we can choose an optimal m to achieve the best privacy. The optimal m is obtained by setting the right side of (29) to 0.

Fig. 3 shows the numerical results for the probability $P(E)$ and the preserved privacy $p = P(E|A)$ under different m when $n_x = n_y = n = 50\,000$; $n_c = 5000$; and $s = 2,\ 5,$ and 10. From the left plot, one can see that the three different values of s yield the same curve of $P(E)$ (or the three curves of $P(E)$ corresponding to $s = 2,\ 5,$ and 10 overlap completely). In other words, the value of s is irrelevant to the probability $P(E)$, which is consistent with our previous analysis. The value of m, on the other hand, has a clear impact on the value of $P(E)$. Specifically, there exists an optimal point where m^* produces a maximum value of $P(E)$. When $m < m^*$, the value of $P(E)$ dramatically increases with the increment of m. When $m > m^*$, the value of $P(E)$ decreases with a slower and slower pace. In the figure, $m^* = 0.39n$ results in an optimal value of $P(E) = 0.4856$. Recall from Fig. 2 that the value of $P(A)$ always decreases with the increment of m. Combining these results, we learn that, as m exceeds a certain value m^*, probabilities $P(E)$ and $P(A)$ will both drop if we further increase m, which is also consistent to our theoretic analysis.

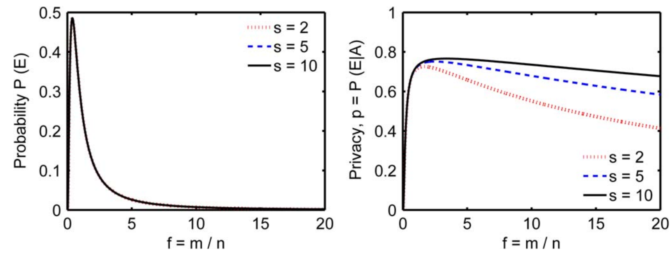


Fig. 3. $n_x = n_y = n = 50\,000$, and $n_c = 5000$. (Left) Probability $P(E)$ when m varies from $0.1n$ to $20n$ under different $s = 2,\ 5,$ or 10. (Right) Probability $P(E|A)$ when m varies from $0.1n$ to $20n$ under $s = 2,\ 5,$ or 10.

Finally, the right plot in Fig. 3 gives the combined effect of s and m on $P(E|A)$, the privacy of our scheme. The smallest value of $s = 2$ yields the bottom curve that represents the least privacy, whereas the largest value of $s = 10$ yields the top curve that represents the best privacy, which agrees with our previous analysis that a larger value of s brings better privacy. Clearly, in each curve, $P(E|A)$ first quickly increases and then slowly decreases with respect to m. There is an optimal value of m that gives the optimal privacy. For instance, $m = 3.6n$ gives the optimal privacy 0.7661 when $s = 10$. Another observation is that, when s is large (5 or 10), there always exists a smooth interval of m near its optimal point that can achieve near-optimal privacy. For example, when $s = 10$, the values of m in the interval $[3.6n,\ 11.2n]$ achieve privacy that is within 5% drop of the optimal privacy 0.7661. In practice, this smooth interval allows us to adjust the value of m to achieve better measurement results while preserving near-optimal privacy.

E. Computation Overhead

We conclude the discussion about the performance of our measurement scheme by a quick remark on the computation overhead incurred to each group of entities involved in the system. In our scheme, when a vehicle v passes an RSE R_x, the vehicle v only needs to compute two hashes to obtain an index of a random bit in its logical bit array LB_v, and the RSE R_x only needs to set one bit in its bit array B_x, as described in Section III-A. Therefore, the computation overhead for each vehicle per RSE and that for each RSE per vehicle are both $O(1)$. As for the central server, to compute the traffic flow size between a pair of locations, it only needs to perform a bitwise AND operation over two m-bit arrays, count the number of "0"s in the resulting bit array, and use formula (16) to compute the MLE estimator. Therefore, the computation overhead for the central server is $O(m)$.

TABLE I
VALUES FOR m TO ACHIEVE OPTIMAL p UNDER DIFFERENT s

s	2	5	10
optimal m	$1.7n$	$2.6n$	$3.6n$
optimal p	0.7258	0.7513	0.7661

IV. SIMULATION

Here, we evaluate the performance of our scheme through simulations. The simulation platform is a PC featured with an Intel Core i7-3770 CPU and 8-GB RAM. The simulations are performed under five system parameters, i.e., n_x, n_y, n_c, s, and m. For a pair of RSEs, R_x and R_y, n_x (n_y) is the number of vehicles passing by R_x (R_y). There are n_c vehicles passing both R_x and R_y, which means that the real traffic flow size is n_c. s is the number of bits in each vehicle's logical bit array, and m is the number of bits in each RSE's bit array. Our simulations consist of two parts. For each part, we first describe the settings of the system parameters and then report the simulation results and the analysis.

A. Measured Traffic Flow $\hat{n_c}$

We first measure traffic flows and observe how different parameters influence the gap between the measured flow sizes and the real sizes when the optimal privacy is preserved. We choose the five parameters as follows: $n_x = n_y = n = 50\,000$, 100 000, or 500 000, and n_c varies from $1\%n$ to $50\%n$, with a step size of $0.1\%n$; $s = 2,\ 5,$ and 10, and m is chosen to achieve the optimal privacy, as determined in Section III-D. Table I lists the values of the bit array size m to achieve the optimal privacy p under different values of s.

Figs. 4–6 show our simulation results when $n = 50\,000$, $100\,000$, and $500\,000$, respectively. For each figure, there are three plots, corresponding to the results of three sets of simulations controlled by parameter s, where $s = 2,\ 5,$ and 10. Each plot shows the measured traffic flow sizes $\hat{n_c}$ (y-axis) with respect to different real traffic flow sizes n_c (x-axis) under a given setting of n, s, and m, where m is chosen as described in Table I so that the optimal privacy is achieved. We also draw the equality line $y = x$ in each plot for reference. Clearly, the closer a point is to the equality line, the more accurate the measurement result.

From the three figures, one can see that our scheme is quite accurate because most of the points in all plots of the three figures lie closely to the equality line. In particular, given other parameters, our scheme produces almost perfect results when $s = 2$ (see the first plot in Figs. 4–6). When s becomes larger, there are slightly more points deviating from the equality line (see the third plot in Figs. 4–6), which indicates that larger values of s yield less accurate measurement results.

Recall that a larger value of s brings better privacy (see Table I). For example, the optimal privacy is 0.7661 when $s = 10$, better than the optimal privacy of 0.7258 when $s = 2$. This implies a tradeoff between the privacy and the accuracy. From Section III-D, we know when s is large, there always exists a smooth interval of m near its extreme point that can achieve comparable privacy as the optimal. For example, when $n_x = n_y = n = 50\,000$, $n_c = 5000$, and $s = 10$, the values of m within the interval $[3.6n,\ 11.2n]$ achieve privacy that is within just 5% drop of the optimal privacy 0.7661. In reality, one can choose a relatively large value for s (e.g., 5 or 10) and adjust the value of m to achieve better measurement results while still preserving comparable privacy as the optimal.

Finally, the measurement results are more accurate with larger values of n. There are fewer points deviating from the equality line $\hat{n_c} = n_c$ in the three plots in Fig. 6 than those in Fig. 4. This is also a natural phenomenon given that the result is measured through a statistical MLE estimator.

B. Measurement Bias and Relative Standard Error

Next, we study the measurement accuracy of the MLE estimator $\hat{n_c}$ in terms of bias and relative standard error. Similar to the previous part, there are three sets of simulations, corresponding to $n_x = n_y = n = 50\,000,\ 100\,000,$ and $500\,000$. For each set, there are three simulations controlled by different values of s, where $s = 2,\ 5,$ and 10. m is still chosen to achieve the optimal privacy p under each fixed s, as listed in Table I. We conduct 5000 independent runs for each simulation to observe statistical effects. For each run, we randomly choose a value for n_c from the range of $[0,\ 0.5n]$ and apply our scheme to obtain the corresponding value for $\hat{n_c}$. Now, we try to figure out the measurement bias $E(\hat{n_c} - n_c)$ and relative standard error $\sqrt{Var(\hat{n_c})}/n_c$ of our MLE estimator from the result of the 5000 independent runs of each simulation.

To better illustrate the simulation results, we divide the range of n_c, $[0,\ 0.5n]$, into 50 measurement scales, each of width $1\%n$; group the values of n_c and corresponding $\hat{n_c}$ from different runs into these 50 scales; and then numerically evaluate the measurement bias and relative standard error of $\hat{n_c}$ with respect to each scale of n_c. The simulation results are presented in Figs. 7–12, where the first three figures (see Figs. 7–9) show the measurement bias and the remaining three figures (see Figs. 10–12) show the relative standard error.

Figs. 7–9 show the measurement bias of $\hat{n_c}$ with respect to each scale of n_c under different values of n, where $n = 50\,000,\ 100\,000,$ and $500\,000$. Each figure consists of three plots, each corresponding to a fixed value of s, where $s = 2,\ 5,$ and 10. For each plot, the y-axis represents the measurement bias $E(\hat{n_c} - n_c)$, and the x-axis represents the mean value of n_c in each scale. The y-coordinate is within 2.5% of n, i.e., ranging from $-2.5\%n$ to $2.5\%n$. Note that the optimal privacy is always guaranteed for all simulations by setting m in accordance with s. In the figures, one can see that the measurement bias fluctuates around the zero-bias line for different scales of n_c. In addition, as observed from the three plots of each figure, under a fixed n, the measurement bias tends to fluctuate more often with higher amplitudes for larger values of s (e.g., compare the first plot in Figs. 7–9 with the third plot of the same figures), which implies that larger values of s will result in more $\hat{n_c}$ deviating from n_c and, in turn, yield less accurate measurement results. This observation agrees with our simulation results from the previous part. Furthermore, if we compare the plots from different figures (e.g., first plot of each figure), it is clear that, under the same value of s, increasing the value of n will reduce the fluctuation amplitudes of $\hat{n_c}$, which

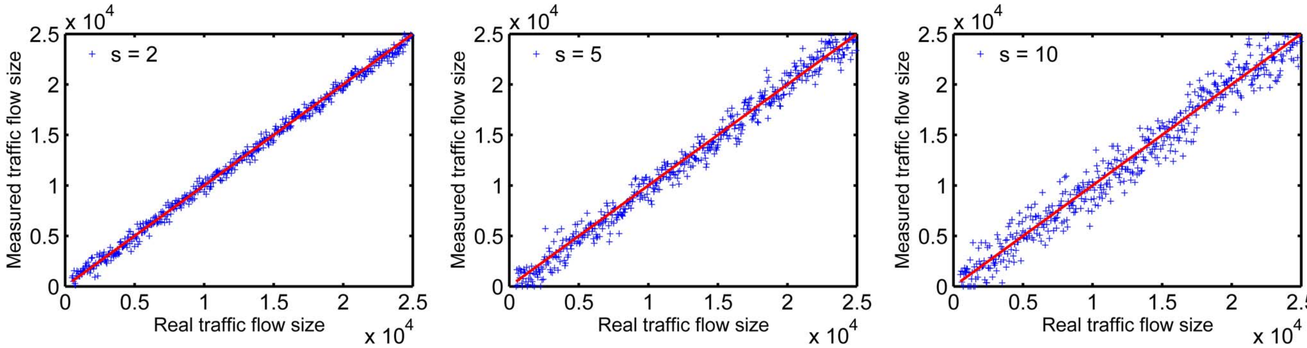


Fig. 4. Measurement accuracy with optimal privacy, $n_x = n_y = n = 50\,000$, and $n_c = [0.01n,\ 0.5n]$. The x-axis shows real traffic flow sizes, and the y-axis shows the corresponding measured traffic flow sizes. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

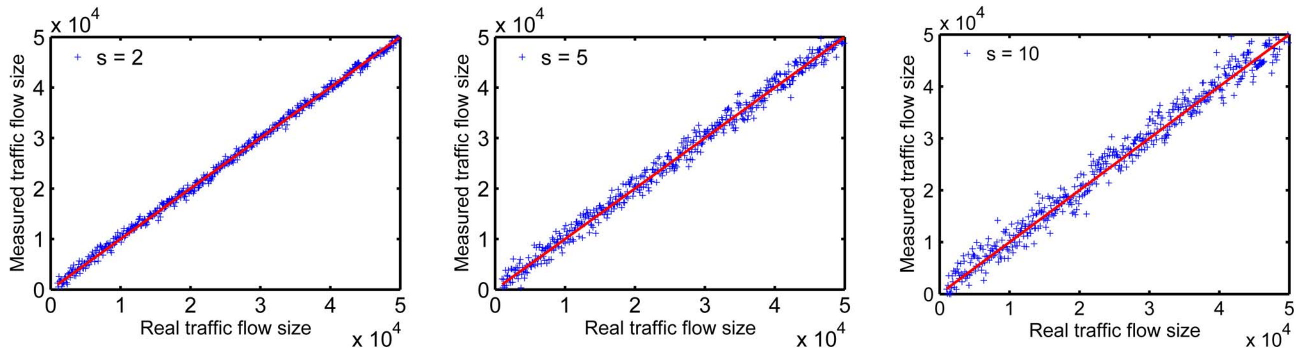


Fig. 5. Measurement accuracy with optimal privacy, $n_x = n_y = n = 100\,000$, and $n_c = [0.01n,\ 0.5n]$. The x-axis shows real traffic flow sizes, and the y-axis shows the corresponding measured traffic flow sizes. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

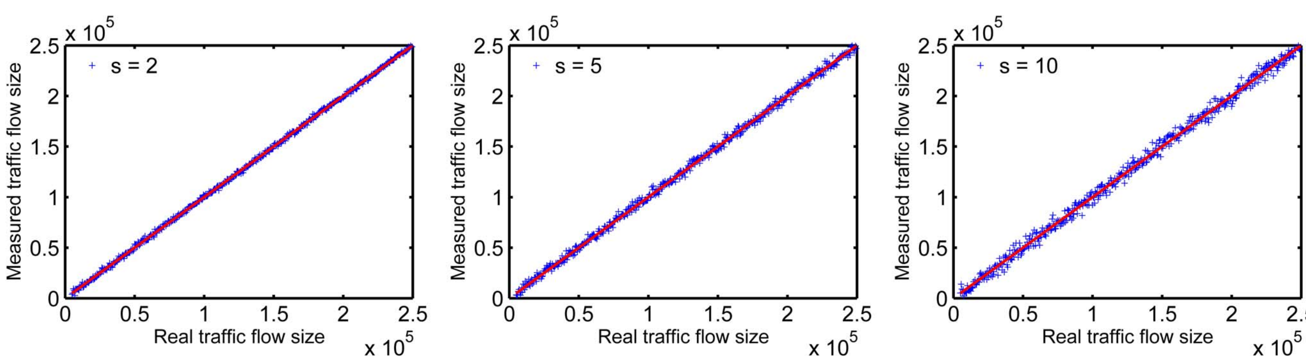


Fig. 6. Measurement accuracy with optimal privacy, $n_x = n_y = n = 500\,000$, and $n_c = [0.01n,\ 0.5n]$. The x-axis shows real traffic flow sizes, and the y-axis shows the corresponding measured traffic flow sizes. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

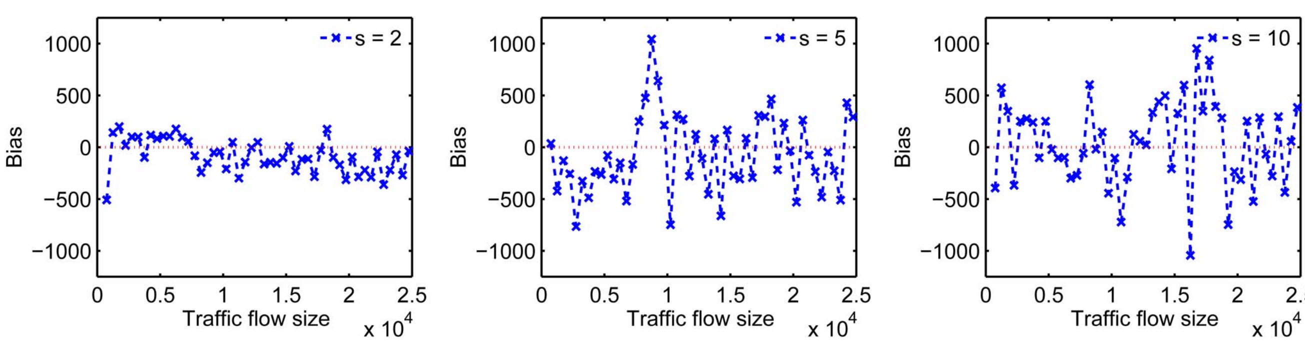


Fig. 7. Measurement bias with optimal privacy, $n_x = n_y = n = 50\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the corresponding measurement bias $E(\hat{n_c} - n_c)$. The y-coordinate is within 2.5% of n, i.e., $[-2.5\%n,\ 2.5\%n]$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

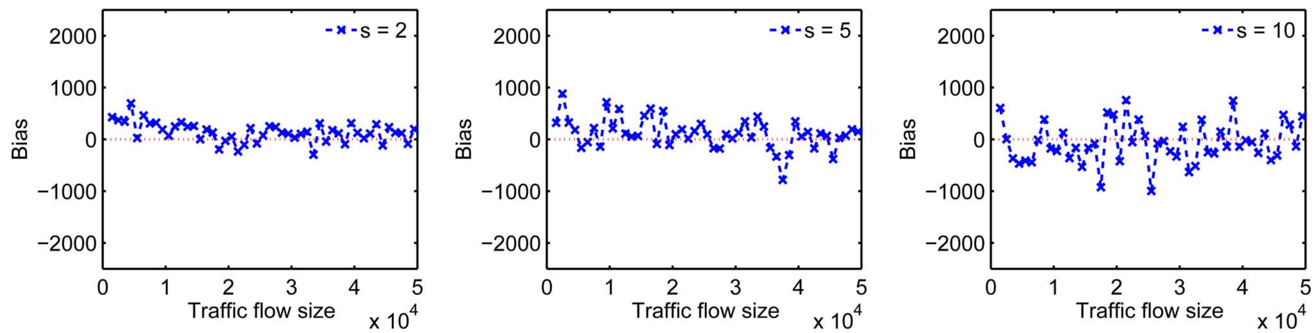


Fig. 8. Measurement bias with optimal privacy, $n_x = n_y = n = 100\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the corresponding measurement bias $E(\hat{n_c} - n_c)$. The y-coordinate is within 2.5% of n, i.e., $[-2.5\%n,\ 2.5\%n]$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

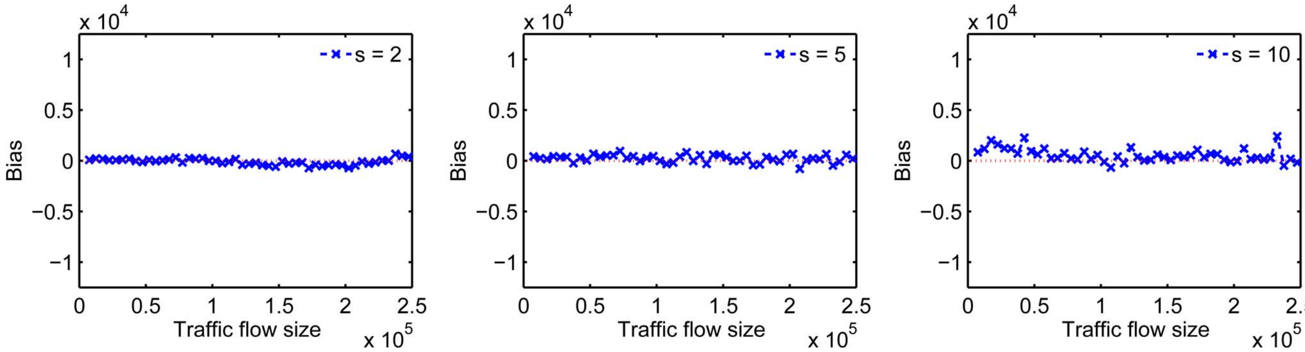


Fig. 9. Measurement bias with optimal privacy, $n_x = n_y = n = 500\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the corresponding measurement bias $E(\hat{n_c} - n_c)$. The y-coordinate is within 2.5% of n, i.e., $[-2.5\%n,\ 2.5\%n]$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

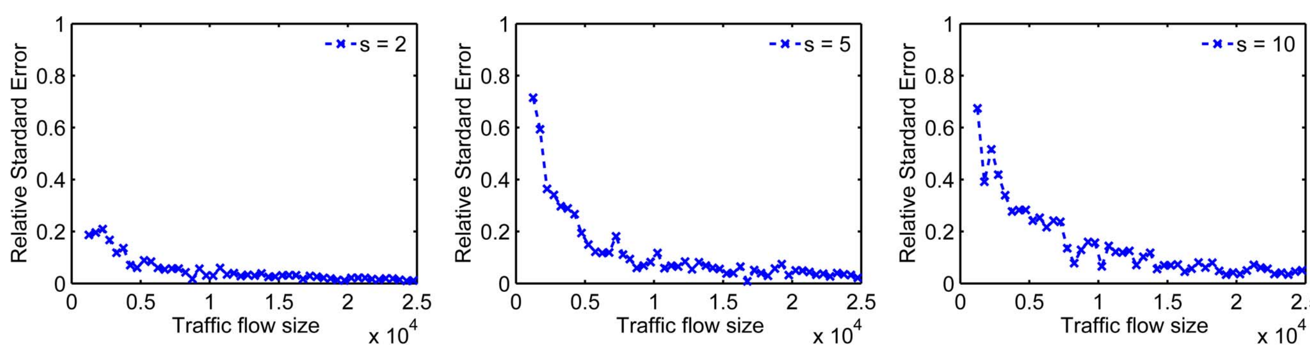


Fig. 10. Relative standard error with optimal privacy, $n_x = n_y = n = 50\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the relative standard error $\sqrt{Var(\hat{n_c})}/n_c$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

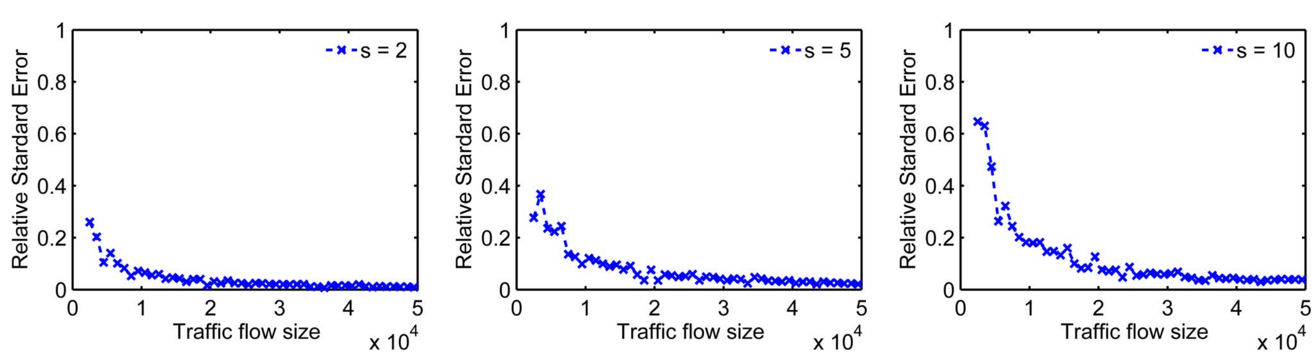


Fig. 11. Relative standard error with optimal privacy, $n_x = n_y = n = 100\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the relative standard error $\sqrt{Var(\hat{n_c})}/n_c$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

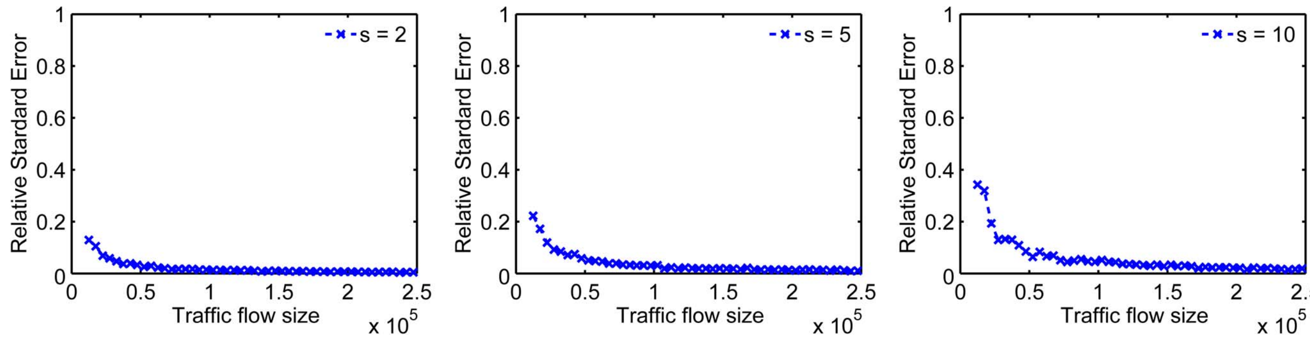


Fig. 12. Relative standard error with optimal privacy, $n_x = n_y = n = 500\,000$, and $n_c = [0,\ 0.5n]$. The x-axis shows scales of traffic flow sizes n_c, and the y-axis shows the relative standard error $\sqrt{Var(\hat{n_c})}/n_c$. The three plots are controlled by s. (First plot) $s = 2$. (Second plot) $s = 5$. (Third plot) $s = 10$.

means that our scheme will produce more stable and accurate measurement results for larger scale systems.

Figs. 10–12 show the relative standard error of $\hat{n_c}$ with respect to each scale of n_c under different values of n, where $n = 50\,000,\ 100\,000,$ and $500\,000$. There are also three plots in each figure, each corresponding to a fixed value of s, where $s = 2,\ 5,$ and 10. For each plot, the y-axis represents the relative standard error of $\hat{n_c}$, $\sqrt{Var(\hat{n_c})}/n_c$, and the x-axis represents the mean value of n_c in each scale. Still, optimal privacy is guaranteed through setting appropriate m. The major observation is that, given n, when s becomes larger, the relative standard error of $\hat{n_c}$ with respect to each scale of n_c also becomes larger. For instance, when $n = 50\,000$, the relative standard error of $\hat{n_c}$ is about 0.017 for the scale of n_c ranging from [8500, 9000] when $s = 2$, whereas its value reaches to about 0.13 when $s = 10$, almost eight times higher than the former value. Since the relative standard error for each scale of n_c becomes larger, the variance of $\hat{n_c}$ also becomes larger, which means that the measured traffic flow sizes will be more spread out from the real flow sizes. This observation also agrees with our previous simulation results, where there are relatively more points not close to the equality line for larger values of s under fixed n. Similarly, the variance becomes smaller when we increase the number n of vehicles. One can see that the relative standard errors are closer to 0 in Fig. 12 than those in Fig. 10, assuming that the same value of s is applied.

V. Related Work

A. Transportation Traffic Measurement

In the area of transportation traffic measurement, various prediction models have been proposed to measure "point" traffic statistics using data recorded by automatic traffic recorders installed at road sections, for example, the multiple linear regression model in [8], artificial neural network in [9], spatial statistical method in [10], and support vector regression in [11]. These solutions, although elegant, are not appropriate for "point-to-point" transportation traffic measurement. As stated in the introduction, "point-to-point" traffic measurement is also critical in traffic engineering. However, few research efforts exist in literature that focus on this problem while preserving the location privacy of individual vehicles in the meantime. The recent work in [12] tries to infer "point-to-point" statistics from "point" data, but the high computation overhead limits its practicability. Our previous work [21] utilizes an encryption method to preserve vehicles' location privacy and measures point-to-point traffic based on the encrypted vehicle IDs. The computation efficiency is improved to $O(n_x n_y)$ for each pair of RSEs, where n_x and n_y denote the number of vehicles passing them, respectively. This overhead is still too high for today's large-scale road networks. Although Google recently announced to provide real-time traffic data service in Google Maps [22], their approach cannot assure vehicle's privacy since it uses GPS and Wi-Fi in phones to track locations [23].

B. Network Traffic Measurement

Another branch of research that relates to (but is also significantly different from) ours is network traffic measurement, where researchers have proposed various methods for traffic flow measurement in the network environment, i.e., to measure the network traffic between two network routers. The solutions can be summarized into two categories. One is indirect estimation based on link load and network routing by employing statistical techniques [24], [25]. These methods cannot achieve high accuracy since their estimations are based on unknown traffic volume. The other is direct measurement by different counting methods [26], [27]. In particular, in [27], a bitmap-based counting method was developed for traffic flow measurement, which is most related to our work. However, all these solutions are not appropriate for our problem because they measure traffic in the network environment where the privacy of packets is not a concern, and counting can be done directly based on the packet IDs. In our problem, the privacy of vehicles is the major concern. Therefore, the solutions must incorporate randomization and de-identification techniques to protect vehicles' privacy and do counting based on information that looks totally random.

VI. Conclusion

In this paper, we have focused on privacy-preserving "point-to-point" transportation traffic monitoring in intelligent CPRSs. We formalize "point-to-point" traffic as traffic flows and quantify privacy as a probability. We propose a novel scheme that allows the collection of aggregate traffic flow data while preserving the privacy of individual vehicles. The proposed scheme utilizes bit arrays to collect "masked" data and adopts MLE to obtain the measurement result. Its feasibility and scalability are shown by mathematical proofs and simulations.

REFERENCES

[1] 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.

[2] 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.

[3] 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. Sys.*, vol. 21, no. 9, pp. 1227–1239, Sep., 2010.

[4] Y. Zhu, Y. Wu, and B. Li, “Vehicular ad hoc networks and trajectory-based routing,” in *Internet of Things*. Cham, Switzerland: Springer-Verlag, 2014, pp. 143–167.

[5] 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.

[6] R. Du *et al.*, “Effective urban traffic monitoring by vehicular sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 273–286, Jan. 2015.

[7] “Traffic Monitoring Guide,” U. S. Dept. Transp., Washington, DC, USA, 2013. [Online]. Available: http://www.fhwa.dot.gov/policyinformation/tmguide/tmg_2013/

[8] D. Mohamad, K. C. Sinha, T. Kuczek, and C. F. Scholer, “Annual average daily traffic prediction model for county roads,” *J. Transp. Res. Board*, vol. 1617, pp. 69–77, 1998.

[9] W. Lam and J. Xu, “Estimation of AADT from short period counts in Hong Kong—A comparison between neural network method and regression analysis,” *J. Adv. Transp.*, vol. 34, no. 2, pp. 249–268, 2000.

[10] 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.

[11] M. C. Neto, Y. Jeong, M. K. Jeong, and L. D. Han, “AADT prediction using support vector regression with data-dependent parameters,” *Exp. Syst. Appl.*, vol. 36, no. 2, pp. 2979–2986, Mar. 2009.

[12] Y. Lou and Y. Yin, “A decomposition scheme for estimating dynamic origin–destination flows on actuation-controlled signalized arterials,” *Transp. Res. C*, vol. 18, no. 5, pp. 643–655, Oct. 2010.

[13] J. Eriksson, H. Balakrishnan, and S. Madden, “Cabernet: Vehicular content delivery using WiFi,” *Proc. MobiCom*, 2008, pp. 199–210.

[14] 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.

[15] [Online]. Available: http://www.its.\ignorespacesdot.gov/press/2010/vii2intellidrive

[16] [Online]. Available: http://www.dot.gov/

[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] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Belmont, CA, USA, Duxbury, 2002.

[19] W. Newey and D. McFadden, “Large sample estimation and hypothesis testing,” in *Handbook of Econometrics*, vol. 4. Amsterdam, The Netherlands: Elsevier, 1994, pp. 2111–2245.

[20] W. Bryc, *The Normal Distribution: Characterizations With Applications*. Amsterdam, The Netherlands: Springer-Verlag, 1995.

[21] 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.

[22] Google Map’s Time-in-Traffic Feature. [Online]. Available: http://mashable.com/2012/03/29/google-maps-traffic-data/

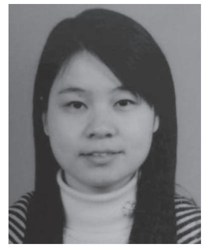
[23] T. Jeske, “Floating car data from smartphones: What Google and Waze know about you and how hackers can control traffic,” in *Proc. BlackHat Eur.*, 2013, pp. 1–12.

[24] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, “An information-theoretic approach to traffic matrix estimation,” in *Proc. SIGCOMM*, 2003, pp. 301–312.

[25] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, “Fast accurate computation of large-scale IP traffic matrices from link loads,” in *Proc. SIGMETRICS*, 2003, pp. 206–217.

[26] J. Cao, A. Chen, and T. Bu, “A quasi-likelihood approach for accurate traffic matrix estimation in a high speed network,” in *Proc. IEEE INFOCOM*, 2008, pp. 21–25.

[27] T. Li, S. Chen, and Y. Qiao, “Origin–destination flow measurement in high-speed networks,” *Proc. IEEE INFOCOM*, 2012, pp. 2526–2530.



Yian Zhou (S’14) received the B.S. degree in computer science and the B.S. degree in 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 Prof. S. Chen. Her research interests include traffic flow measurement, cyber-physical systems, big network data, security and privacy, and cloud computing.

Zhen Mo (S’14) received the B.E. degree in information security engineering and the M.E. degree in theory and new technology of electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively, and the Ph.D. degree in computer engineering from the University of Florida, Gainesville, FL, USA, in 2015.

He is currently with the Department of Computer and Information Science and Engineering, University of Florida. His research interests include network security and cloud computing 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 Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2011.

Currently, he is 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.

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

After graduation, he was with 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. He is the author or coauthor of more than 130 peer-reviewed journal/conference papers. He is the holder 12 U.S. patents. His research interests include computer networks, Internet security, wireless communications, and distributed computing.

Prof. Chen served on the Steering Committee of the IEEE/ACM International Symposium on Quality and Service (IWQoS) from 2010 to 2013 and on the technical advisory board of Protego Networks from 2002 to 2003. He is an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, *Computer Networks*, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He received the IEEE Communications Society Best Tutorial Paper Award in 1999 and the National Science Foundation Faculty Early Career Development (CAREER) Award in 2007.

Yafeng Yin received the B.E. degree in structural engineering, the B.E. degree in environmental engineering, and the M.S. degree in civil engineering from Tsinghua University, Beijing, China, in 1994, 1994, and 1996, respectively, and the Ph.D. degree in civil engineering from the University of Tokyo, Tokyo, Japan, in 2002.

Currently, he is an Associate Professor with the Department of Civil and Coastal Engineering, University of Florida, Gainesville, FL, USA. His research interests include transportation network modeling, highway traffic operations, transit planning and operations, infrastructure asset management, and assessments and evaluations of intelligent transportation systems technologies.