Point-to-Point Traffic Volume Measurement through Variable-Length Bit Array Masking in Vehicular Cyber-Physical Systems

Yian Zhou * Shigang Chen * Zhen Mo * Qingjun Xiao †
*Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL, USA
†Key Lab of Computer Network & Information Integration
Southeast University, Education Ministry, P. R. China

***Abstract*—In this paper, we consider an important problem of privacy-preserving point-to-point traffic volume measurement in vehicular cyber physical systems (VCPS), whose focus is utilizing VCPS to enable automatic traffic data collection, and measuring point-to-point traffic volume while preserving the location privacy of all participating vehicles. The novel scheme that we propose tackles the efficiency, privacy, and accuracy problems encountered by previous solutions. Its applicability is demonstrated through both mathematical and numerical analysis. The simulation results also show its superior performance.**

I. INTRODUCTION

Point-to-point traffic volume measurement is critical in transportation engineering. It estimates the number of vehicles traveling between two arbitrary *points* (geographical locations) in the road system during the measurement period, and provides essential input to a variety of transportation 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 [1]. In this paper, we consider the important problem of privacy-preserving point-to-point traffic volume measurement in vehicular cyber physical systems (VCPS), whose focus is utilizing VCPS to enable automatic traffic data collection, and measuring point-to-point traffic volume while preserving the location privacy of all participating vehicles.

VCPS has emerged as one of the most promising research areas in road networks. It integrates the latest technologies in wireless communications and on-board computer processing into transportation systems to enhance road safety and improve driving experience [2] [3]. In particular, IEEE has standardized Dedicated Short Range Communications (DSRC) under IEEE 802.11p [4], which supports transmitting/receiving messages between vehicles and roadside units (RSU). A great advantage that VCPS provides is automatic traffic data collection. For example, vehicles can report information to RSUs en route, and those information can be automatically collected and used for traffic volume measurement. However, challenges remain to be tackled before the beauty of VCPS can be fully appreciated by its large audience. As more and more people concern about their location privacy, any traffic measurement scheme that targets at being widely accepted and applied in VCPS should put travellers' privacy in the top priority. The transportation authorities from different countries also put forward a number of principles to protect drivers' privacy. For instance, the "anonymity by design" principle required by IntelliDrive [5] from USDOT [6] aims at privacy protection in the first place. Keeping the requirement of privacy preservation in mind, having the vehicles report their unique identifier such as their Vehicle Identification Numbers (VIN) is clearly not acceptable. Other permanently or temporarily fixed numbers also bare the potential of giving away the vehicles' moving trajectory. The challenge of addressing the privacy concerns of travellers while measuring point-to-point traffic volume opens the door to an interesting research problem: How to design a measurement scheme in which a vehicle never transmits any unique identifier (to protect their privacy from being pried via this unique ID), yet the random and de-identified information that the vehicle submits still supports the measurement of traffic volume between different locations?

This challenging problem has been partially solved by [7], [8] and [9]. [7] infers point-to-point statistics from point data, and [8] uses encryption method to preserve vehicle's privacy. However, their practicability are both limited by the high computation overhead. [9] utilizes bit arrays to collect "masked" data and is much more efficient. However, the solution assumes same traffic volume for all RSUs, and requires same length for all bit arrays in different RSUs, which greatly hurts its applicability. In reality, the traffic volume in different RSUs varies a lot. For example, according to the 2012 yearly traffic volume report from New York State Department of Transportation [10], major intersections in New York have hundreds of thousands of cars passing by every day, while light-traffic intersections

1063-6927/15 $31.00 © 2015 IEEE
DOI 10.1109/ICDCS.2015.14

only have a few hundreds of cars passing by during the same period. Considering this more realistic situation where traffic volume in different places actually differs a lot, the performance of [9] decreases dramatically in terms of both drivers' location privacy and measurement accuracy, which also limits its practicability.

To fit in the more realistic transportation model where different RSUs face different traffic volume, we propose a novel scheme for privacy-preserving point-to-point traffic volume measurement, which achieves better privacy for vehicles, more accurate measurement results, and comparable computation overhead with the previous best scheme. It utilizes variable-length bit arrays to encode traffic data reported by vehicles, and a novel "unfolding" technique to support traffic measurement based on those varied-length bit arrays. The measurement accuracy and preserved privacy are analyzed through both mathematical and numerical means, which demonstrate its applicability. The simulation results also show its superior performance.

II. PROBLEM STATEMENT

A. Problem Definition

We consider a vehicular cyber-physical system involving three groups of entities: vehicles, roadside units (RSU), and a central server. 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 [4]. RSUs are connected to the central server through wired or wireless means, and they report information collected from vehicles to the central server periodically.

The problem is given two arbitrary locations where RSUs are installed, to measure the number of vehicles that pass by both locations while protecting vehicles' location privacy. We need a solution in which a vehicle never transmits any unique identifier to protect their location privacy. 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.

We assume RSUs are semi-honest: On the one hand, all RSUs are from trustworthy authorities, which can be enforced by authentication based on PKI. The 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 purpose.

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 automatical tracking caused by the traffic measurement scheme itself.

We also assume that a special MAC protocol is applied to support privacy preservation such that the MAC address of a vehicle is not fixed. Vehicles may pick an MAC address randomly from a large space for one-time use when needed.

B. Performance Metrics

We consider three performance metrics: computation overhead, measurement accuracy, and preserved privacy.

1) Computation Overhead: It includes computation overhead for each vehicle per RSU en route, and for each RSU per passing vehicle, and for the central server to measure the traffic volume between a pair of RSUs.

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

3) Preserved Privacy: We use same privacy definition as [9], which is a probability p satisfying the requirement that the probability for any "trace" of any vehicle not to be identified must be at least p, where a trace of a vehicle is a pair of RSUs it has passed. Clearly, larger values of p mean better privacy because the tracker will have less chance to link traces of a vehicle to obtain its trajectory.

III. RELATED WORK

Research in transportation traffic measurement can be briefly summarized into two categories, measurement of *"point"* statistics and measurement of *"point-to-point"* statistics. In the past, the research focus is on estimation of *"point"* statistics such as annual average daily traffic (AADT), which tell the number of vehicles traversing a specific *point* (location), and various predication models [11], [12], [13], [14], [15], [16] have been proposed to measure them using data recorded by RSUs such as automatic traffic recorders (ATR) installed at road sections. For example, Mohamad et al. develop a multiple linear regression model which incorporates demographic variables to measure AADT for country roads in [11], and Lam et al. adopt artificial neural network to estimate AADT by using short period counts for urban areas in [12]. Other research efforts that belong to this category include the spatial statistical method proposed by Eom et al. in [13], the support vector regression model presented by Neto et al. in [14], the absolute deviation penalty procedure designed by Yang et al. in [15], and the regression and Bayesian based model derived by Tsapakis et al. in [16], etc.

Recently, the estimation of *"point-to-point"* traffic statistics start to draw attention of researchers, especially when the location privacy of travellers is involved. For example, Lou and Yin propose to infer point-to-point traffic statistics from point data [7], but its practicability is limited by high computation overhead. Our previous work [8] improves the measurement efficiency to $O(n^2)$ through encryption method, but it is still not enough for large-scale road networks. Google announced to provide real-time traffic data service in Google maps [17], but their approach cannot assure vehicle's privacy since it uses GPS and Wi-Fi in phones to track locations [18]. Motivated by [19] and [20] targeting at network measurement, we propose an efficient solution in [9] and [21], which utilize bit arrays to collect "masked" traffic data. To the best of our knowledge, the solution provided in [9] is the best state of art that addresses the privacy concerns of drivers and passengers while trying to measure the traffic volume between two arbitrary road locations. Assuming same traffic volume for all RSUs, the scheme uses same-length bit arrays, so they can be pairwise compared to obtain statistical results determined by the actual traffic volume, which can then be estimated by reversing the process. The scheme works perfectly when the traffic volume of all RSUs are comparable. However, when it comes to the more realistic situation where RSUs face different traffic volume, the performance of [9] decreases dramatically in terms of both location privacy of individual vehicles and measurement accuracy of aggregate traffic data, which limits its practicability. The scheme that we propose has comparable efficiency with [9] and furthermore, it can easily fit in the more realistic transportation model and achieve far better privacy and accuracy than [9].

IV. Novel Scheme of Privacy-Preserving Point-to-Point Traffic Measurement

A. Motivation

The solution in [9] uses same-length bit arrays to encode traffic data, and it only works when the traffic volume of all RSUs are comparable. When it comes to the more realistic situation where the number of cars passing by different RSUs actually varies a lot, its performance decreases dramatically. The problem lies in the great difficulty of determining an appropriate fixed bit array size, m. If a large m is chosen to accommodate the large traffic volume in major intersections, it will greatly hurt the privacy for the cars passing by light-traffic RSUs (will explain more later in Section VI). If a small m is chosen to provide relatively good privacy for all cars, the accuracy for measuring the traffic volume between heavy-traffic RSUs will be dramatically decreased (will explain more later in Section VII).

Can one achieve the goal of both obtaining sound measurement results and maintaining good privacy for all cars? The solution with fixed-length bit arrays is not applicable, so how about varied-length bit arrays? Intuitively, to maintain good privacy, light-traffic RSUs should have smaller bit arrays, and to achieve sound measurement results, heavy-traffic RSUs should have larger bit arrays. In other words, the sizes of bit arrays should be related to the traffic volume of corresponding RSUs. This motivates our design of a novel measurement scheme based on varied-length bit arrays. To enable comparison of bit arrays, we propose an "unfolding" technique, and require the size of all bit arrays to be power of 2, i.e., m must be in the form of 2^k.

B. Online Coding Phase

Our 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 traffic volume between RSUs based on the reported bit arrays. Now we explain the first phase.

In our scheme, 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 bit arrays to be power of 2, i.e., m_x must be in the form of 2^k, whose purpose will be explained later. In order for m_x to reflect the relation with the traffic volume in R_x, we determine its value as $m_x = 2^{ceil(\log_2(\bar{n}_x \times \bar{f}))}$, where $\bar{n}_x$ is the history average "point" traffic volume in R_x, $\bar{f}$ is a global parameter determined by history traffic data and same for all RSUs, and $ceil(t)$ is a function which returns the smallest integer that is not less than t. Clearly, m_x is the smallest integer that is 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.

In addition, each vehicle v has a logical bit array LB_v, which consists of s bits randomly selected from the largest bit array B_o among all RSUs, where $s \ll m_o$. The indices of these bits in B_o are $H(v \oplus K_v \oplus X[0])$, ..., $H(v \oplus K_v \oplus X[s-1])$, where $\oplus$ is the bitwise XOR, $H(...)$ is a hash function whose range is $[0, m_o)$, X is an integer array of randomly chosen constants to arbitrarily alter the hash result, and K_v is the private key of v whose purpose is to protect its privacy.

Given above notations and data structures, the online coding phase works as follows. RSUs broadcast queries in pre-set 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. 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 bit array size is m_x. It first verifies the certificate to authenticate the RSU. After verifying that R_x is from the trustworthy authority, v will randomly select a bit from its logical bit array LB_v by computing an index $b = H(v \oplus K_v \oplus X[H(R_x) \, mod \, s])$. Then v generates an index b_x in the range of $[0, m_x)$ corresponding to b by a modulus operation, $b_x = b \, mod \, m_x$, 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:

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

$$B_x[H(v \oplus K_v \oplus X[H(R_x) \, mod \, s]) \, mod \, m_x] = 1. \tag{2}$$

C. 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 "point" traffic volume for the RSUs to take into account the traffic data in the current measurement period, and then measures the "point-to-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 RSU R_x and R_y is $S_x \cap S_y$. Denote its cardinality as n_c, i.e., $|S_x \cap S_y| = n_c$, 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 $m_x \leq m_y$ (otherwise, change the role of R_x and R_y). Given the counters n_x and n_y, and bit arrays B_x and B_y, the server measures n_c as follows:

First, the central server expands the two bit arrays B_x and B_y to the same size. To enable traffic volume measurement based on different-sized bit arrays, we propose an ***"unfolding"*** technique, which is expanding the bit arrays to the same size by duplicating their content. Here we only need to expand the smaller bit array B_x to the size of the larger bit array B_y, since m_x and m_y are both power of 2 and $m_x \leq m_y$, and for any two numbers that are power of 2, the division result of the larger value over the smaller value must be an integer. Therefore, we can expand B_x by duplicating its content for $\frac{m_y}{m_x}$ times to create a new bit array of size m_y. We call it the "unfolded" bit array of B_x, and denote it as B_x^u. More specifically,

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

Second, the central server takes a bitwise OR of B_x^u and B_y, and the resulting bit array is denoted as B_c:

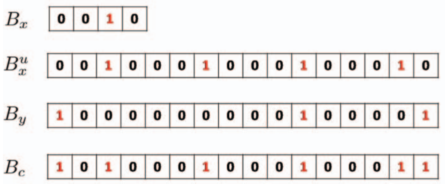
$$B_c[i] = B_x^u[i] \ \vee \ B_y[i], \ \forall i \in [0, m_y). \tag{4}$$

The bitwise OR operation is granted since the two bit arrays, B_x^u and B_y, are of the same size. Figure 1 shows an example of the unfolding and bitwise-OR operation. In this example, B_x is unfolded to B_x^u, and a bitwise-OR is performed on B_x^u and B_y to produce B_c.

Finally, **given B_c, B_x (B_x^u), and B_y, the central server uses the following formula to estimate the point-to-point traffic volume between R_x and R_y:**

$$\hat{n_c} = \frac{\ln(V_c) - \ln(V_x) - \ln(V_y)}{\ln(1 - \frac{s-1}{s} \times \frac{1}{m_y}) - \ln(1 - \frac{1}{m_y})} \tag{5}$$

where V_c, V_x, and V_y are random variables (R.V.) which represent the fraction of bits whose values are zeros in B_c, B_x, and B_y, correspondingly. Their values can be easily found by counting the number of zeros in B_c, B_x, and B_y, denoted by U_c, U_x, and U_y respectively (note U_c, U_x, and U_y are also R.V.s), and dividing them by the corresponding bit array size m_y, m_x, and m_y. That is, $V_c = \frac{U_c}{m_y}$, $V_x = \frac{U_x}{m_x}$, and $V_y = \frac{U_y}{m_y}$. Note that the fraction of zero bits in B_x^u is the same as B_x.



B_x	0	0	1	0												
B_x^u	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
B_y	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
B_c	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1

Fig. 1. An example of unfolding and bitwise-OR operation.

D. Derivation of the MLE Estimator $\hat{n_c}$

Now we follow the standard maximum likelihood estimate (MLE) method [22] to derive $\hat{n_c}$ given by (5). Its accuracy will be analyzed in Section V. We first derive the probability $q(n_c)$ for an arbitrary bit in B_c to be '0', and use $q(n_c)$ to establish the likelihood function $\mathcal{L}$ to observe U_c '0' bits in B_c. Finally, maximizing $\mathcal{L}$ with respect to n_c will lead to the MLE estimator, $\hat{n_c}$.

Consider an arbitrary bit b in B_c. Let A_b be the event that the bth bit in B_c remains '0', then $q(n_c)$ is the probability for A_b to occur. Since the set of all vehicles passing R_x and/or R_y (i.e., $S_x \cup S_y$) can be partitioned into three sets, $S_x \cap S_y$, $S_x - S_y$, and $S_y - S_x$, it is clear that event A_b is equivalent to the combination of the following three events:

(I) *Event E_1: For vehicles passing both R_x and R_y (i.e., in the set $S_x \cap S_y$), none of them have chosen bit $(b \, mod \, m_x)$ in B_x or bit b in B_y.* Otherwise, according to (3) and (4), bit b in B_c will be '1'. For any vehicle, it has the same probability $\frac{1}{s}$ to select any bit in its s-bit logical bit array. So the probability for an arbitrary vehicle v from $S_x \cap S_y$ to select the same bit from its logical bit array in both R_x and R_y is $s \times \frac{1}{s} \times \frac{1}{s} = \frac{1}{s}$. In other words, if v chooses $(b' \, mod \, m_x) \neq (b \, mod \, m_x)$ in R_x, it has a probability of $\frac{1}{s}$ to choose the same bit b' in R_y (hence will not set bit b in B_y), and probability of $1 - \frac{1}{s}$ to choose a separate bit b'' randomly from B_y. The probability for $b'' \neq b$ is $1 - \frac{1}{m_y}$, and the probability for v to choose $(b' \, mod \, m_x)$ in B_x is $1 - \frac{1}{m_x}$. There are n_c cars in set $S_x \cap S_y$, so the probability of E_1 is

$$\begin{aligned} P_1 &= \left\{ \left(1 - \frac{1}{m_x}\right) \left[\frac{1}{s} + \left(1 - \frac{1}{s}\right) \left(1 - \frac{1}{m_y}\right) \right] \right\}^{n_c} \\ &= \left(1 - \frac{1}{m_x}\right)^{n_c} \left(1 - \frac{s-1}{s} \times \frac{1}{m_y}\right)^{n_c} \end{aligned} \tag{6}$$

(II) *Event E_2: For vehicles passing only R_x (i.e., in the set $S_x - S_y$), none of them have chosen bit $(b \, mod \, m_x)$ in*

B_x. Otherwise, from (3), bit b in B_x^u is '1' (so bit b in B_c is '1'). Since each vehicle in $S_x - S_y$ has probability $\frac{1}{m_x}$ to set bit $(b\, mod\, m_x)$, and there are $n_x - n_c$ vehicles in $S_x - S_y$, the probability of E_2 is

$$P_2 = \left(1 - \frac{1}{m_x}\right)^{n_x - n_c}. \tag{7}$$

(III) *Event E_3: For vehicles passing only R_y (i.e., in the set $S_y - S_x$), none of them have chosen bit b in B_y.* Otherwise, bit b in B_y will be '1' (hence bit b in B_c is also '1'). Similarly, we can derive its probability as

$$P_3 = \left(1 - \frac{1}{m_y}\right)^{n_y - n_c}. \tag{8}$$

Combining above analysis, we can obtain the overall probability $q(n_c)$ for bit b in B_c to remain '0' as follows:

$$\begin{aligned} q(n_c) &= P_1 \times P_2 \times P_3 \\ &= \left(1 - \frac{1}{m_x}\right)^{n_x} \left(1 - \frac{1}{m_y}\right)^{n_y} \left(\frac{1 - \frac{s-1}{sm_y}}{1 - \frac{1}{m_y}}\right)^{n_c} \end{aligned} \tag{9}$$

Since the bits in any logical bit array are selected from the largest physical bit array uniformly at random, the vehicles in set S_x (S_y) have the same probability of $\frac{1}{m_x}$ ($\frac{1}{m_y}$) to choose any bit in B_x (B_y). For any bit in B_x (B_y), the probability for it to be '0' after n_x (n_y) vehicles each choosing a random bit from B_x (B_y) is

$$q(n_x) = \left(1 - \frac{1}{m_x}\right)^{n_x}, \tag{10}$$

$$q(n_y) = \left(1 - \frac{1}{m_y}\right)^{n_y}. \tag{11}$$

Therefore, the number of zero bits in B_x follows a binomial distribution $U_x \sim B(m_x, q(n_x)) = B(m_x, (1 - \frac{1}{m_x})^{n_x})$, while the number of zero bits in B_y follows another binomial distribution $U_y \sim B(m_y, q(n_y)) = B(m_y, (1 - \frac{1}{m_y})^{n_y})$. From the property of binomial distribution [22], and $V_x = \frac{U_x}{m_x}$ and $V_y = \frac{U_y}{m_y}$, the expected values for V_x and V_y are

$$E(V_x) = E\left(\frac{U_x}{m_x}\right) = \frac{m_x(1 - \frac{1}{m_x})^{n_x}}{m_x} = q(n_x), \tag{12}$$

$$E(V_y) = E\left(\frac{U_y}{m_y}\right) = \frac{m_y(1 - \frac{1}{m_y})^{n_y}}{m_y} = q(n_y). \tag{13}$$

Substituting (12) and (13) to (9), and replacing $E(V_x)$ and $E(V_y)$ by their instance values, V_x and V_y, we have the following instance value for $q(n_c)$:

$$q(n_c) = V_x \times V_y \times \left(\frac{1 - \frac{s-1}{s} \times \frac{1}{m_y}}{1 - \frac{1}{m_y}}\right)^{n_c}. \tag{14}$$

Given the probability for each bit in B_c to be '0' as $q(n_c)$, we can establish the likelihood function $\mathcal{L}$ for us to observe U_c '0' bits in B_c (so $m_y - U_c$ '1' bits in B_c):

$$\mathcal{L} = (q(n_c))^{U_c} \times (1 - q(n_c))^{m_y - U_c}. \tag{15}$$

The MLE estimator of n_c is the optimal value of n_c that maximizes the likelihood function in (15). To find $\hat{n_c}$, we take logarithm on both sides of (15), and then take the first order derivative to obtain:

$$\frac{d \ln \mathcal{L}}{dn_c} = \left(\frac{U_c}{q(n_c)} - \frac{m_y - U_c}{1 - q(n_c)}\right) \times q'(n_c), \tag{16}$$

where $q'(n_c)$ can be computed from (9) as follows:

$$q'(n_c) = q(n_c) \times \ln\left(\frac{1 - \frac{s-1}{s} \times \frac{1}{m_y}}{1 - \frac{1}{m_y}}\right). \tag{17}$$

Since $q'(n_c)$ cannot be 0 when $m_x > 1$, $m_y > 1$, and $s < m_y$, setting the right side of (16) to 0 gives

$$q(n_c) = \frac{U_c}{m_c} = V_c. \tag{18}$$

Substituting (18) to (14) and reordering the items, we obtain the MLE estimator $\hat{n_c}$ as described in (5).

E. Computation Overhead

We conclude this section by a discussion about the computation overhead of our scheme. We compare it with the best state of art [9]. The other two performance metrics will be analyzed in the following two sections.

Clearly, the computation overhead for the vehicles and RSUs of our novel scheme are comparable to [9]. In our novel scheme, when a vehicle v passes an RSU R_x, the vehicle v only needs to compute two hashes to obtain an index of a random bit, and the RSU R_x only needs to set 1 bit in its bit array B_x, as described in Section IV-B. So 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 [9], but the computation overhead is comparable. First, the server expands the smaller bit array B_x to B_x^u, which has the same size as B_y, by duplicating its content. This operation costs $O(m_y)$ time. Second, it performs a bitwise OR over two m_y-bit bit arrays, B_x^u and B_y, to create a new bit array B_c of size m_y, which also costs $O(m_y)$ time. Last, the server counts the number of zeros in B_x, B_y, and B_c, 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, 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 [9] assumes that $m_x = m_y = m$ and its computation overhead for the server is $O(m)$, one can see that our novel scheme indeed achieves comparable computation overhead as [9].

V. Analysis on Measurement Accuracy

In this section, we analyze the measurement accuracy of the MLE estimator $\hat{n_c}$ mathematically. According to (5), $\hat{n_c}$ involves three random variables V_c, V_x, and V_y. Therefore, we first study the mean and variance of V_c, V_x, and V_y, based on which we derive the formula for the bias and standard deviation of $\hat{n_c}$.

A. Mean and Variance of V_c, V_x, and V_y

The mean values of V_x and V_y are given in (12) and (13). Their variance can be computed from the variance of U_x and U_y. Since U_x and U_y each follows a binomial distribution as mentioned in Section IV-C, we have

$$Var(V_x) = \frac{Var(U_x)}{{m_x}^2} = \frac{q(n_x) \times (1 - q(n_x))}{m_x}, \tag{19}$$

$$Var(V_y) = \frac{Var(U_y)}{{m_y}^2} = \frac{q(n_y) \times (1 - q(n_y))}{m_y}, \tag{20}$$

where $q(n_x)$ and $q(n_y)$ are given by (10) and (11).

Since the probability for any bit in B_c to be '0' is $q(n_c)$, the number of zeros in B_c also follows a binomial distribution $U_c \sim B(m_c, q(n_c)) = B(m_y, q(n_c))$. Therefore, the mean of U_c is $m_y \times q(n_c)$ and the variance of U_c is $m_y \times q(n_c) \times (1 - q(n_c))$. Since $V_c = \frac{U_c}{m_y}$, the mean and variance of V_c can be derived accordingly:

$$E(V_c) = E\left(\frac{U_c}{m_y}\right) = \frac{m_y \times q(n_c)}{m_y} = q(n_c), \tag{21}$$

$$Var(V_c) = \frac{Var(U_c)}{{m_y}^2} = \frac{q(n_c) \times (1 - q(n_c))}{m_y}, \tag{22}$$

where $q(n_c)$ is given by (9).

B. Mean and Variance of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$

Now we derive the mean and variance of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$. First, we define a function $f(V) = \ln V$, and expand the function by its Taylor series about the mean value of V, denoted as $w = E(V)$, to obtain

$$\begin{aligned} f(V) &= f(w) + (V-w)f'(w) + \frac{1}{2}(V-w)^2 f''(w)... \\ &= \ln(w) + \frac{V-w}{w} - \frac{(V-w)^2}{2w^2}... \end{aligned} \tag{23}$$

To get the expected value of $\ln(V)$, we truncate (23) after the third term since expected value of the second term is 0, and the third term is the first nonzero bias:

$$E(\ln(V)) = \ln(w) - \frac{E((V-w)^2)}{2w^2} = \ln(w) - \frac{Var(V)}{2w^2}. \tag{24}$$

According to (24), we can compute the mean of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$ based on the mean and variance values of V_c, V_x, and V_y. Below are the results:

$$E(\ln(V_x)) = \ln(q(n_x)) - \frac{1}{2m_x} \times \frac{1 - q(n_x)}{q(n_x)}, \tag{25}$$

$$E(\ln(V_y)) = \ln(q(n_y)) - \frac{1}{2m_y} \times \frac{1 - q(n_y)}{q(n_y)}, \tag{26}$$

$$E(\ln(V_c)) = \ln(q(n_c)) - \frac{1}{2m_y} \times \frac{1 - q(n_c)}{q(n_c)}. \tag{27}$$

To get the variance, we truncate (23) after two terms:

$$Var(\ln(V)) = Var\left(\ln(w) + \frac{V-w}{w}\right) = \frac{Var(V)}{w^2}. \tag{28}$$

Again, according to (28), we can compute the variance of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$ based on the mean and variance values of V_c, V_x, and V_y. Below are the results:

$$Var(\ln(V_x)) = \frac{Var(V_x)}{(E(V_x))^2} = \frac{1}{m_x} \times \frac{1 - q(n_x)}{q(n_x)}, \tag{29}$$

$$Var(\ln(V_y)) = \frac{Var(V_y)}{(E(V_y))^2} = \frac{1}{m_y} \times \frac{1 - q(n_y)}{q(n_y)}, \tag{30}$$

$$Var(\ln(V_c)) = \frac{Var(V_c)}{(E(V_c))^2} = \frac{1}{m_y} \times \frac{1 - q(n_c)}{q(n_c)}. \tag{31}$$

C. Mean and Variance of $\hat{n_c}$

Based on the mean of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$ derived previously, we obtain the mean of $\hat{n_c}$:

$$E(\hat{n_c}) = \frac{E(\ln(V_c)) - E(\ln(V_x)) - E(\ln(V_y))}{\ln(1 - \frac{s-1}{s} \times \frac{1}{m_y}) - \ln(1 - \frac{1}{m_y})}, \tag{32}$$

where $E(\ln(V_c))$, $E(\ln(V_x))$, and $E(\ln(V_y))$ are given in (25), (26), and (27). The estimation bias is

$$Bias\left(\frac{\hat{n_c}}{n_c}\right) = E\left(\frac{\hat{n_c}}{n_c}\right) - 1 = \frac{E(\hat{n_c})}{n_c} - 1. \tag{33}$$

We can also derive the variance of $\hat{n_c}$ as

$$Var(\hat{n_c}) = \frac{C + D}{\left[\ln\left(1 - \frac{s-1}{s} \times \frac{1}{m_y}\right) - \ln\left(1 - \frac{1}{m_y}\right)\right]^2} \tag{34}$$

where $D = Var(\ln(V_c)) + Var(\ln(V_x)) + Var(\ln(V_y))$, and $C = -C_1 - C_2 + C_3$ with $C_1 = Cov(\ln(V_c), \ln(V_x))$, $C_2 = Cov(\ln(V_c), \ln(V_y))$, and $C_3 = Cov(\ln(V_x), \ln(V_y))$. The three covariance terms can be derived by expanding the Taylor series of $\ln(V_c)$, $\ln(V_x)$, and $\ln(V_y)$ about the mean values of V_c, V_x, and V_y. For example, C_1 is derived as

$$\begin{aligned} C_1 &= -E(\ln(V_c))E(\ln(V_x)) + E(\ln(V_c)\ln(V_x)) \\ &= -E(\ln(V_c))E(\ln(V_x)) - \ln(E(V_c))\ln(E(V_x)) \\ &+ \ln(E(V_c))E(\ln(V_x)) + \ln(E(V_x))E(\ln(V_c)) \end{aligned} \tag{35}$$

Substituting the formula of $E(V_c)$, $E(V_x)$, $E(\ln(V_c))$, and $E(\ln(V_x))$, which we have already computed, we can obtain $Cov(\ln(V_c), \ln(V_x))$. $Cov(\ln(V_c), \ln(V_y))$ and $Cov(\ln(V_x), \ln(V_y))$ can be derived similarly. After obtaining the covariances, we can compute the variance of $\hat{n_c}$ based on (34). Finally, given $Var(\hat{n_c})$, the standard deviation of $\frac{\hat{n_c}}{n_c}$ is computed as follows:

$$StdDev\left(\frac{\hat{n_c}}{n_c}\right) = \frac{\sqrt{Var(\hat{n_c})}}{n_c}. \tag{36}$$

VI. Analysis on Preserved Privacy

We evaluate the preserved privacy of our novel scheme using the same definition as [9], which is the conditional probability that tells 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. 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. Therefore, the tracker can only identify the trace of a common vehicle through the observation of the bits that are chosen by the vehicles to be set as '1' in both RSUs.

A. Derivation of the Preserved Privacy

First, consider the probability for an arbitrary bit, b, to be '1' in both B_x^u and B_y (event A), $P(A)$. Denote its complementary event as $\bar{A}$. Clearly, $P(A) = 1 - P(\bar{A})$. Denote by S the subset of vehicles in $S_x \cap S_y$ that happen to choose the same bit in its logical bit array at both R_x and R_y. Let n_s be the cardinality of S, i.e., $n_s = |S|$. Clearly, $S \subseteq S_x \cap S_y$ and $0 \leq n_s \leq n_c$. As we mentioned earlier, the probability for $v \in S_x \cap S_y$ to select the same bit at both R_x and B_y is $\frac{1}{s}$. Therefore, the number of such vehicles, n_s, is binomially distributed according to $B(n_c, \frac{1}{s})$. The probability for $n_s = z \, (0 \leq z \leq n_c)$ is

$$P(n_s = z) = \binom{n_c}{z}\left(\frac{1}{s}\right)^z \left(1 - \frac{1}{s}\right)^{n_c - z}. \tag{37}$$

Clearly, event $\bar{A}$ is equivalent to the combination of the following two events: (1) *Event E_4: None of the vehicles in S has chosen b at R_x and R_y.* Otherwise, bit $(b \, mod \, m_x)$ in B_x (hence bit b in B_x^u) and bit b in B_y are both set to '1'. Clearly, the probability of E_4 is

$$q_4 = \left(1 - \frac{1}{m_y}\right)^{n_s}. \tag{38}$$

(2) *Event E_5: Either none of the vehicles in $S_x - S$ has chosen $(b \, mod \, m_x)$ at R_x or none of the vehicles in $S_y - S$ has chosen b at R_y.* Otherwise, the two corresponding bits are both set to '1'. Clearly, the probability of E_5 is

$$q_5 = 1 - \left[1 - \left(1 - \frac{1}{m_x}\right)^{n_x - n_s}\right]\left[1 - \left(1 - \frac{1}{m_y}\right)^{n_y - n_s}\right] \tag{39}$$

Combining above analysis, the probability of event $\bar{A}$ is

$$\begin{aligned} P(\bar{A}) &= \sum_{z=0}^{n_c} q_4(n_s|n_s = z) q_5(n_s|n_s = z) P(n_s = z) \\ &= \left(1 - \frac{1}{m_x}\right)^{n_x} \times C_4^{n_c} + \left(1 - \frac{1}{m_y}\right)^{n_y} \\ &\quad - \left(1 - \frac{1}{m_x}\right)^{n_x}\left(1 - \frac{1}{m_y}\right)^{n_y} \times C_5^{n_c}, \end{aligned} \tag{40}$$

where C_4 and C_5 are both constants with values $C_4 = \frac{1}{s} \times \frac{1 - \frac{1}{m_y}}{1 - \frac{1}{m_x}} + \left(1 - \frac{1}{s}\right)$, and $C_5 = \frac{1}{s} \times \frac{1}{1 - \frac{1}{m_x}} + \left(1 - \frac{1}{s}\right)$.

Secondly, consider the conditional probability for such a bit, b, to not represent a common vehicle passing both R_x and R_y (event E), $P(E|A)$. Note that event E happens if and only if bit $(b \, mod \, m_x)$ in B_x (hence bit b in B_x^u) is set only by vehicles passing only RSU R_x (i.e., in $S_x - S_y$), and bit b in B_y is set only by vehicles passing only RSU R_y (i.e., in $S_y - S_x$). Denote these two events as E_x and E_y, respectively. We can easily derive their probability as:

$$P(E_x) = \left(1 - \left(1 - \frac{1}{m_x}\right)^{n_x - n_c}\right) \times \left(1 - \frac{1}{m_x}\right)^{n_c}, \tag{41}$$

$$P(E_y) = \left(1 - \left(1 - \frac{1}{m_y}\right)^{n_y - n_c}\right) \times \left(1 - \frac{1}{m_y}\right)^{n_c}. \tag{42}$$

Therefore, the preserved privacy of our novel scheme is:

$$\begin{aligned} p &= P(E|A) = \frac{P(E_x) \times P(E_y)}{P(A)} \\ &= \frac{1}{1 - P(\bar{A})} \times \left[\left(1 - \frac{1}{m_x}\right)^{n_c} - \left(1 - \frac{1}{m_x}\right)^{n_x}\right] \\ &\quad \times \left[\left(1 - \frac{1}{m_y}\right)^{n_c} - \left(1 - \frac{1}{m_y}\right)^{n_y}\right], \end{aligned} \tag{43}$$

where $P(\bar{A})$ is given in (40). Note that if we set $m_x = m_y = m$ in (43), we get the same formula as [9], which means that [9] is just a special case of our novel scheme.

B. Privacy Comparison with the Best State of Art

Note that the scheme in [9] works only if all bit arrays are of the same size. We have mentioned that if a large m is chosen to accommodate heavy-traffic RSUs, the privacy of cars passing light-traffic RSUs will be greatly hurt. Here we explain more through numerical analysis.

The first plot of Figure 2 shows the privacy p of [9] when m varies from $0.1n$ to $50n$, controlled by $s = 2, 5, 10$, where $n_x = n_y = n$. From the plot, one can see that the privacy of [9] is actually determined by the ratio f (called load factor) of m over n, and the optimal privacy is achieved at the optimal load factor f^* (approximately from 2 to 4). An important observation is that, when m is fixed, the privacy will vary a lot given different n (hence different f). If we choose a large m to accommodate RSUs with large n, say $n = n' = 500,000$, and $m = f_1 n' = 2n'$,

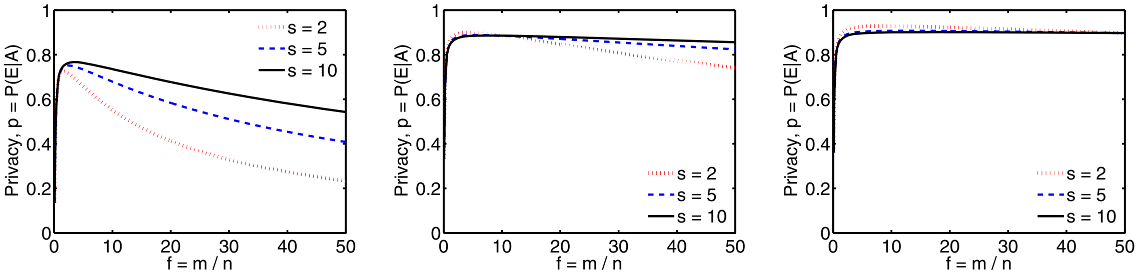


Fig. 2. Preserved privacy. *First Plot*: the privacy of both schemes with equal m; *Second Plot*: the privacy of our novel scheme when $n_y = 10n_x$; *Third Plot*: the privacy of our novel scheme with $n_y = 50n_x$.

then the privacy of cars passing RSUs with smaller n, say $n = n'' = \frac{n'}{25} = 20,000$, will be greatly hurt since the load factor for those RSUs will be $f_2 = 25f_1 = 50$ (see the rightmost point of the three curves). Specifically, the privacy suffers most for small values of s. For example, when $s = 2$, the privacy is only about 0.2. One can expect more drop in privacy for cars passing RSUs with less traffic. To guarantee a minimum privacy of cars regardless of RSUs, the value of m should be determined by the least traffic volume among all RSUs, n_{min}. For example, m should be no larger than $15n_{min}$ to guarantee a minimum privacy of 0.5 when $s = 2$. However, this brings another problem: the measurement accuracy for heavy-traffic RSUs will dramatically decrease (more on Section VII).

The problem of plummeted privacy in [9] originates from the fact that different RSUs have different traffic volume, and using same-length bit arrays will cause "unbalanced load factors". Below, we show that by using variable-length bit arrays so that their load factors are comparable, our novel scheme not only solves the plummeted privacy problem in [9], but also improves the optimal privacy when the traffic volume differs.

Figure 2 shows the privacy p of our novel scheme when the load factor varies from 0.1 to 50. Note that we use the same load factor $\bar{f}$ for all RSUs (so the lengths of bit arrays will vary given different traffic volume n at different RSUs). When the traffic volume of R_x and R_y are comparable, their bit arrays will have the same length, i.e., $m_x = m_y = m$, so the privacy formula for both schemes will be the same, resulting in the same graph as shown in the first plot of Figure 2. We stress that for our new scheme, since all RSUs use the same load factor $\bar{f}$, the privacy of all cars, regardless of the traffic volume of RSUs that they pass, will always be comparable as the optimal privacy if $\bar{f} = f^*$. For example, when $s = 5$, the privacy of the cars passing comparable-traffic RSUs will be more than 0.75. For RSUs with different traffic volume, the novel scheme has another advantage, which is improving the privacy of the cars passing those RSUs. The second and the third plot of Figure 2 show the privacy that our novel scheme preserves for cars passing RSUs with different traffic volume where $n_y = 10n_x$ and $n_y = 50n_x$, respectively. One can see that given $\bar{f} = f^*$, both plots show better optimal privacy than comparable-traffic RSUs. For instance, given $\bar{f} = 3$ when $s = 5$, the optimal privacy is 0.89 for $n_y = 10n_x$, and 0.91 for $n_y = 50n_x$, both greater than the optimal privacy of 0.75 for $n_x = n_y$. The improved privacy originates from the variable-length bit arrays. During the "unfolding" process, the content of B_x is duplicated to generate B_x^u. This effectively creates more common '1' bits in B_x^u and B_y that are not caused by common cars, thus adding one more level of "mask" for the traces of common vehicles.

VII. Simulation

We compare the performance of our novel scheme with the best state of art [9] through simulations. There are two sets of simulations: the first set of simulations considers a real Sioux Falls road network with known vehicle trip tables, while the second set considers a larger network with randomly generated traffic.

A. Simulation Results of the Sioux Falls Network

We first consider a real road network of Sioux Falls with known vehicle trip tables. First published by Lebranc etc. in [23], the Sioux Falls network has made its appearance in thousands of conference papers, journals and books (e.g., [24], [25], [26]). As illustrated by Figure 3, the Sioux Falls network contains 24 nodes (RSUs) with 76 arcs (road segments). In our simulations, we generate traffic according to the known vehicle trip table in [23] under the Sioux Falls network, and compute the daily point-to-point traffic volume between each pair of nodes using both the scheme in [9] and our novel scheme. The parameters for the two schemes are determined as follows. For both schemes, the number of bits in the logical bit array of each vehicle, s, is set to $2, 5, 10$ as [9]. $\bar{f}$ and m are chosen to guarantee a minimum privacy of at least 0.5. Recall that $\bar{f}$ is the fixed load factor for all bit arrays in our novel scheme, and m is the fixed bit array length in [9].

Table I shows the simulation results of eight typical node pairs in the Sioux Falls network of our novel scheme and the scheme in [9] under $s = 2$. Note that the unit for the traffic volume is thousands of vehicles/day. Also, since node 10 has the largest traffic volume among all 24 nodes, it is

TABLE I

Simulation results for the Sioux Falls network of our novel scheme and [9]. The unit for the traffic volume is thousands of vehicles/day. $R_y = 10$, $n_y = 451$. RSUs are sorted according to the traffic difference ratio against R_y, i.e., $d = \frac{n_y}{n_x}$. The error ratio for both scheme is defined as $r = \frac{|\hat{n_c} - n_c|}{n_c} \times 100\%$.

R_x	15	12	7	24	6	18	2	3
n_x	213	140	121	78	76	47	40	28
Traffic difference ratio, d	2.117	3.221	3.727	5.782	5.934	9.596	11.275	16.107
n_c	40	20	19	8	8	7	6	3
$\hat{n_c}$ (scheme in [9])	40.048	19.881	19.195	7.215	7.517	6.106	6.637	2.638
$\hat{n_c}$ (our novel scheme)	39.950	19.972	18.982	7.976	7.988	6.979	5.999	3.005
Error ratio, r (scheme in [9])	0.120%	0.595%	1.026%	9.813%	6.037%	12.771%	10.617%	12.067%
Error ratio, r (our novel scheme)	0.125%	0.140%	0.095%	0.300%	0.150%	0.300%	0.017%	0.167%

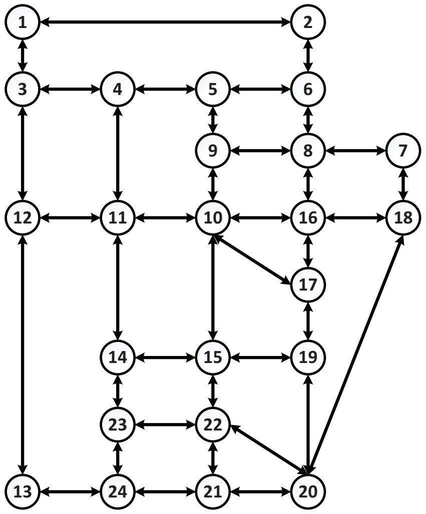


Fig. 3. Sioux Falls Network

chosen to be RSU R_y with $n_y = 451$. The other RSU R_x in each pair is randomly selected from the remaining nodes, and they are sorted according to their traffic difference ratio against R_y (i.e., $d = \frac{n_y}{n_x}$). The point-to-point traffic volume between each pair of R_x and R_y is measured by both our novel scheme and the scheme in [9], and the error ratio against the real traffic volume n_c, i.e., $r = \frac{|\hat{n_c} - n_c|}{n_c} \times 100\%$, is also calculated to better show the results. Clearly, the smaller the error ratio, the better the measurement result.

From Table I, one can see that when the traffic difference ratio d is small (i.e., the traffic volume of R_x and R_y are comparable), e.g., $n_y \approx 2n_x$ in the second column of Table I, both measurement schemes can achieve very accurate results (both around 0.1%). However, when the gap of traffic volume between two RSUs enlarges, the scheme in [9] starts to produce less and less accurate results. One can see that the error ratio r of [9] increases by an order of magnitude when the traffic difference ratio $d \approx 4$ (the fourth column of Table I), and over 2 orders of magnitude when $d \approx 16$ (the last column of Table I). On the other hand, our novel scheme remains accurate for all RSU pairs, with error ratio r constantly below 0.3%, which reflects its superior performance over [9].

B. Simulation Results of Randomly Generated Traffic

Next, we consider a larger network where the traffic is randomly generated. The simulations are controlled by six parameters, n_x, n_y, n_c, s, $\bar{f}$, and m. Their values are chosen as follows: $n_x = 10,000$, $n_y = n_x\,(10,000)$, $10n_x\,(100,000)$, or $50n_x\,(500,000)$, n_c varies from $0.01n_x$ to $0.5n_x$, with step size of $0.001n_x$. s is set to $2, 5, 10$, and $\bar{f}$ and m are chosen to guarantee a minimum privacy of at least 0.5.

Figure 4 shows the simulation results for [9], and Figure 5 shows the results for our novel scheme, both under $s = 2$. Since the simulations for $s = 5$ and $s = 10$ show similar results, here we omit them. For each figure, there are three plots, corresponding to the results of three groups of simulations controlled by n_y and n_x, where $n_y = n_x$, $n_y = 10n_x$, and $n_y = 50n_x$, respectively. Each plot shows the measured traffic volume $\hat{n_c}$ (y-axis) with respect to the real traffic volume n_c (x-axis). The equality line $y = x$ is also drawn for reference. Clearly, the closer a point is to the equality line, the better the measurement result. From the first plot of Figure 4 and 5, one can observe that both schemes achieve perfect performance when $n_y = n_x$. The reason for their comparable performance here is that our novel scheme is almost the same as [9] when $n_y = n_x = n$ (hence $m_y = m_x = m = \bar{f} \times n$). However, when RSUs with different traffic volume are involved, the measurement accuracy of [9] decreases dramatically. In particular, when $n_y = 50n_x$, the results of [9] are quite inaccurate (the measured results almost scatter everywhere in the third plot of Figure 4). On the contrary, our novel scheme stays accurate (the measured traffic volume closely follow their real values in Figure 5). This superior performance originates from our novel design of variable-length bit arrays and the “unfolding” technique, which eliminates the “unbalanced load factor” problem that [9] suffers.

VIII. Conclusion

In this paper, we design a novel scheme for privacy-preserving point-to-point traffic volume measurement in vehicular cyber physical systems, which achieves better privacy for vehicles, more accurate measurement results, and comparable computation overhead, compared with

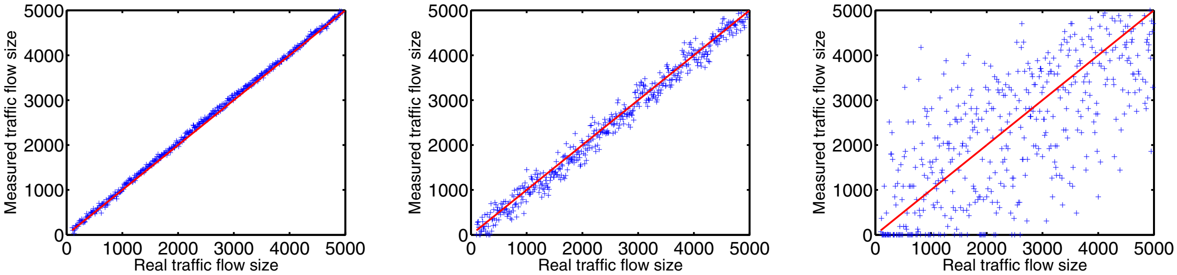


Fig. 4. Measurement accuracy of the scheme in [9]. The x-axis shows true traffic volume, and the y-axis shows measured traffic volume. $s = 2$, $n_x = 10,000$, $n_c = [0.01n_x, 0.5n_x]$. *First Plot*: $n_y = n_x$; *Second Plot*: $n_y = 10n_x$; *Third Plot*: $n_y = 50n_x$.

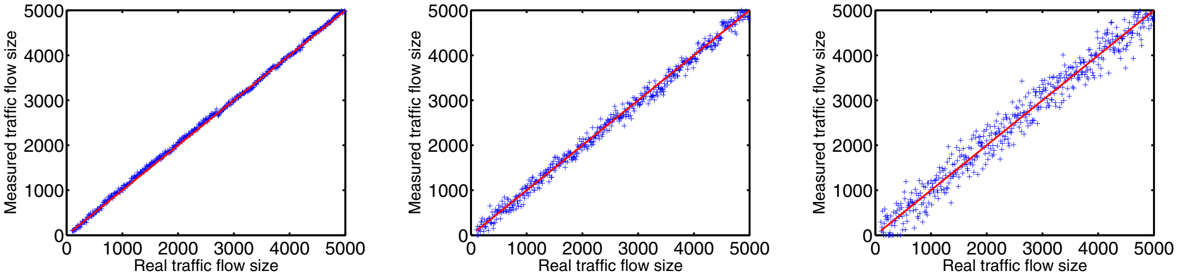


Fig. 5. Measurement accuracy of our novel scheme. The x-axis shows true traffic volume, and the y-axis shows measured traffic volume. $s = 2$, $n_x = 10,000$, $n_c = [0.01n_x, 0.5n_x]$. *First Plot*: $n_y = n_x$; *Second Plot*: $n_y = 10n_x$; *Third Plot*: $n_y = 50n_x$.

the previous best scheme. Its applicability and superior performance are demonstrated through mathematical and numerical analysis, and extensive simulation results.

IX. Acknowledgment

This work is supported in part by the National Science Foundation of United States under grants CPS 0931969, CNS-1115548 and CNS-1409797.

References

[1] USDOT, "Traffic Monitoring Guide," 2001. [Online]. Available: http://www.fhwa.dot.gov/ohim/tmguide/tmg3.htm

[2] J. Eriksson and H. Balakrishnan, "Cabernet: Vehicular Content Delivery Using WiFi," *Proc. of MOBICOM*, 2008.

[3] U. Lee, J. Lee, J. Park, and M. Gerla, "FleaNet: A Virtual Market Place on Vehicular Networks," *IEEE Trans. on Vehicular Technology*, 2010.

[4] Y. L. Morgan, "Notes on DSRC & WAVE Standards Suite," *IEEE Comm. Surveys & Tutorials*, 2010.

[5] [Online]. Available: http://www.its.dot.gov/press/2010/vii2intellidrive

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

[7] Y. Lou and Y. Yin, "A Decomposition Scheme for Estimating Dynamic Origin-destination Flows on Actuation-controlled Signalized Arterials," *Transportation Research Part C*, pp. 643–655, 2010.

[8] Y. Zhou, S. Chen, Z. Mo, and Y. Yin, "Privacy Preserving Origin-Destination Flow Measurement in Vehicular Cyber-Physical Systems," *Proc. of IEEE CPSNA*, pp. 32–37, 2013.

[9] 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," *Proc. of IEEE CPSCom*, pp. 826–833, 2013.

[10] NYSDOT, "Traffic Volume Report," 2012. [Online]. Available: https://www.dot.ny.gov/divisions/engineering/technical-services/highway-data-services/traffic-data

[11] D. Mohamad, K. C. Sinha, T. Kuczek, and C. F. Scholer, "Annual Average Daily Traffic Prediction Model for County Roads," *J. of the Transportation Research Board*, 1998.

[12] 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. of Advanced Transportation*, pp. 249–268, 2000.

[13] 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. of the Transportation Research Board*, pp. 20–29, 2006.

[14] M. C. Neto, Y. Jeong, M. K. Jeong, and L. D. Han, "AADT Prediction using Support Vector Regression with Data-Dependent Parameters," *Expert Systems with Applications*, vol. 36, pp. 2979–2986, March 2009.

[15] B. Yang, Y. Wang, S. Wang, and Y. Bao, "Efficient Local AADT Estimation via SCAD Variable Selection Based on Regression Models," *Control and Decision*, pp. 1898–1902, 2011.

[16] 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," *Transportation planning and technology*, pp. 201–217, March 2013.

[17] "Google map's time-in-traffic feature." [Online]. Available: http://mashable.com/2012/03/29/google-maps-traffic-data/

[18] T. Jeske, "Floating Car Data from Smartphones: What Google and Waze Know About You and How Hackers Can Control Traffic," *Proc. of the BlackHat Europe*, 2013.

[19] M. Yoon, T. Li, S. Chen, and J. kwon Peir, "Fit a Spread Estimator in Small Memory," *Proc. of INFOCOM*, pp. 504–512, April 2009.

[20] T. Li, S. Chen, and Y. Qiao, "Origin-Destination Flow Measurement in High-Speed Networks," *Proc. of INFOCOM*, pp. 2526–2530, 2012.

[21] Y. Zhou, Z. Mo, Q. Xiao, S. Chen, and Y. Yin, "Privacy-Preserving Transportation Traffic Measurement in Intelligent Cyber-Physical Road Systems," *IEEE Transactions on Vehicular Technology*, 2015.

[22] G. Casella and R. L. Berger, "Statistical Inference," *2nd edition, Duxbury Press*, 2002.

[23] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," *Transportation Research*, vol. 9, pp. 309–318, February 1975.

[24] M. C. Ferris and J. S. Pang, "Engineering and Economic Applications of Complementarity Problems," vol. 39, pp. 669–713, December 1997.

[25] H. Yang and H.-J. Huang, *Mathematical and Economic Theory of Road Pricing*. Elsevier, November 2005.

[26] S. H. Putman, *Integrated Urban Models Volume 2: New Research and Applications of Optimization and Dynamics*. Routledge Revival, April 2014, vol. 2.