Privacy-Preserving Traffic Flow Estimation for Road Networks

Elmahdi Bentafat, M. Mazhar Rathore, and Spiridon Bakiras
Division of Information and Computing Technology
College of Science and Engineering
Hamad Bin Khalifa University, Qatar
Email: {ebentafat, mrathore, sbakiras}@hbku.edu.qa

***Abstract*—Future intelligent transportation systems necessitate a fine-grained and accurate estimation of vehicular traffic flows across critical paths of the underlying road network. This task is relatively trivial if we are able to collect detailed trajectories from every moving vehicle throughout the day. Nevertheless, this approach compromises the location privacy of the vehicles and may be used to build accurate profiles of the corresponding individuals. To this end, this work introduces a privacy-preserving protocol that leverages roadside units (RSUs) to communicate with the passing vehicles, in order to construct encrypted Bloom filters stemming from the vehicle IDs. The aggregate Bloom filters are encrypted with a *threshold* cryptosystem and can only be decrypted by the transportation authority in collaboration with multiple trusted entities. As a result, the individual communications between the vehicles and the RSUs remain secret. The decrypted Bloom filters reveal the aggregate traffic information at each RSU, but may also serve as a means to compute an approximation of the traffic flow between any pair of RSUs, by simply estimating the number of common vehicles in their respective Bloom filters. We performed extensive simulation experiments with various configuration parameters and demonstrate that our protocol reduces the estimation error considerably when compared to the current state-of-the-art approaches. Furthermore, our implementation of the underlying cryptographic primitives illustrates the feasibility, practicality, and scalability of the system.**

***Index Terms*—vehicular traffic flow estimation, Bloom filters, privacy, homomorphic encryption**

I. Introduction

Traffic statistics facilitate transportation authorities in many use cases, including investment plans, signal time determination, road expansions, etc. Traffic statistics are measured in terms of single-point or multi-point traffic flows. Single-point traffic flow refers to the number of vehicles passing through a specific location, which can be measured by placing fixed sensors at roadsides, such as inductive loop detectors, wireless magnetometer sensors, road cameras, or microwave radar sensors. Single-point statistics are useful for measuring the average annual daily traffic (AADT) [1], [2], [3], [4]. On the other hand, multi-point traffic flow, sometimes referred to as origin-destination (O-D) or point-to-point flow, is defined as the total number of vehicles moving from one location to another. Multi-point traffic statistics may be deduced from single-point counters [5], however, there are serious concerns about the accuracy of such approaches, because they are oblivious to the identities of the underlying vehicles.

Consequently, to derive accurate traffic statistics, we need alternative methods for collecting data from moving vehicles. One approach is to use the drivers' smartphones [6], [7] or the GPS systems integrated in most vehicles [8], [9], in order to generate detailed object trajectories. Alternatively, recent advancements in vehicular communications and networking technologies have brought cyber physical systems (CPS) into road networks, where roadside equipment are used to collect traffic data [10]. The dedicated short-range communications (DSRC) protocol is standardized by the IEEE (802.11p) [11] and enables the direct communication between vehicles and roadside units (RSUs). In this scenario, generating accurate traffic statistics is trivial: each vehicle reports its ID to every RSU it encounters, with all the reports being aggregated to a centralized server (the transportation authority). Nevertheless, this approach violates the privacy of the vehicle owners and may reveal sensitive personal information, such as home and work locations, habits, etc.

To address these privacy concerns, Zhou et al. [12], [13] have proposed the use of a bit array as an alternative to individual vehicle IDs. In particular, every vehicle selects (in advance) a set of s bit locations that it may reveal to an RSU. When the vehicle identifies a new RSU, it randomly selects one of the s locations and sends it to the RSU as its identifier. Each RSU gradually aggregates the bit array data from all passing vehicles, by setting a bit to '1' if it is selected by at least one vehicle (non-selected bits are set to '0'). Point-to-point traffic flows are then constructed by comparing the bit arrays of the corresponding RSUs. While this approach improves the privacy compared to the trivial method, it still has several limitations. First, the bit information is exchanged in plaintext and is, thus, vulnerable to timing attacks. Indeed, any number of RSUs may collude to deduce the vehicles' secret information (the s bit locations), by correlating successive samples based on the driving distance between the RSUs. Second, for sufficient privacy, s should be relatively large (e.g., $s \geq 5$), which negatively affects the accuracy of the traffic flow statistics.

To this end, our work advances the state-of-the-art in two directions. Our major contribution is a novel method that summarizes the vehicle information at each RSU into an encrypted Bloom filter [14]. We employ a two-tiered approach where (i) the vehicles' Bloom filters are encrypted with a simple one-

time pad (OTP) cipher and (ii) the OTP keys are encrypted with a homomorphic *threshold* public key cryptosystem. The underlying cryptosystems make it infeasible for an adversary to decrypt the individual Bloom filters, thus enhancing significantly the vehicles' privacy. Each RSU aggregates the Bloom filters and OTP keys from multiple vehicles, and sends the corresponding ciphertexts to the transportation authority. Finally, the transportation authority engages multiple trusted parties to decrypt the aggregate OTP keys and retrieve the plaintext Bloom filters.

Our second contribution is a simple and accurate method for estimating the number of common vehicles in two distinct Bloom filters, which is an indication of the traffic flow volume between the two RSUs. We performed extensive simulation experiments and demonstrate that, compared to the current state-of-the-art approaches, our methods improve the accuracy of point-to-point traffic flow estimation by a large factor. In addition, our software implementation of the cryptographic primitives illustrates the feasibility and scalability of our approach.

II. Related Work

Early work on road network traffic flows focused on the prediction of the annual average daily traffic. To this end, a variety of machine learning models have been applied, including regression [2], [4], neural networks [3], support vector regression (SVR) [1], and regression with Bayesian analysis [15]. All these systems exploit the capabilities of roadside units for traffic data collection. On the other hand, Ref. [6], [9], [7] utilize data from mobile phones and GPS devices to extract origin-destination information. This is similar to the approach used by Google Maps and Waze to optimize their routing decisions [8].

Hoh et al. [16] highlighted the serious privacy threats in traffic monitoring systems, as they were able to locate 85% of the drivers' home locations from the collected data. This is also true for popular apps, like Google Maps and Waze, that use a static ID for each reporting client, even across different trips [8]. Therefore, to protect the privacy of trajectory data, Hoh and Gruteser [17] proposed a path perturbation algorithm for a centralized, trusted server, by employing path confusion. PADAVAN [18] is another scheme for anonymous data collection that allows anonymous data reporting, while avoiding fake submissions and linkage between submitted samples. Likewise, Rass et al. [19] introduced trajectory anonymization by deriving pseudonyms for trips and samples. Finally, Hoh et al. [20] proposed a distributed and privacy-preserving traffic monitoring system that utilizes virtual trip lines, i.e., geographical markers where the vehicle has to report its location.

The above-mentioned schemes introduce some level of privacy in traffic monitoring systems, but they are not well-suited towards estimating point-to-point traffic flows with high accuracy. To this end, several researchers applied intricate cryptographic protocols to protect the privacy of drivers and their trips. Förster et al. [21] proposed a distributed secret sharing algorithm, using location- and time-specific keys, and enforced k-anonymity of location data in a decentralized environment. Zhou et al. [22] measure origin-destination traffic flows using commutative one-way hash functions that are constructed from an RSA-like cryptosystem. Although the hash function hides the vehicle's ID from the RSU, it fails to protect the privacy of the trajectory when all the data is aggregated at the centralized server.

The current state-of-the-art approaches are due to Zhou et al. [12], [13], which provide privacy without the use of cryptographic techniques. Specifically, in their system, every receiver maintains a physical bit array of size m (similar to a Bloom filter), and each passing vehicle sets one of the bits to '1'. To avoid being identified across multiple receivers, the vehicle uses a logical bit array of size $s \ll m$, i.e., it can only set (randomly) one of s pre-selected bits at each receiver. The centralized server collects all the physical bit arrays and applies maximum-likelihood estimation (MLE) to measure the intensity of the traffic flow between any pair of receivers. While this approach is very efficient, it has several shortcomings. For sufficient privacy, s should be large, but this negatively affects the accuracy of the traffic flow estimation. Furthermore, the lack of encryption makes it possible to launch a variety of attacks, such as timing or direct observation, which may disclose the contents of a vehicle's logical bit array.

III. Preliminaries

A. System Model

We consider the cyber physical system depicted in Fig. 1, where roadside units (also called *receivers*) are installed at different geographical locations along the road network. Every vehicle and receiver is equipped with a computing unit, and is able to communicate with other devices through the IEEE 802.11p protocol. Whenever a vehicle comes in close proximity to an RSU (e.g., within 100–500m), it transmits an encrypted Bloom filter that is a representation of its unique identifier. (To enhance privacy, a vehicle may choose a different identifier at the start of a new trip.) The RSU blindly aggregates the individual Bloom filters and submits the resulting ciphertexts to the transportation authority. Note that we are interested in collecting *fine-grained* traffic statistics, so each RSU will produce a new Bloom filter when (i) a timer expires (e.g., every 2–5 minutes) or (ii) a threshold number of measurements is reached (e.g., 1000–2000 vehicles). To satisfy basic privacy requirements, an RSU will not submit a Bloom filter if the number of vehicles is below a lower threshold (e.g., 100–200 vehicles). Such fine-grained measurements allow for the collection of important traffic statistics, including point-to-point travel speed estimation.

Prior to system deployment, the transportation authority initializes a threshold Paillier cryptosystem [23] in collaboration with several third-party trusted entities. For example, these entities may include various consumer protection agencies and other non-profit organizations. The reason for employing a threshold cryptosystem is to prevent individual parties—especially the transportation authority that has access to all

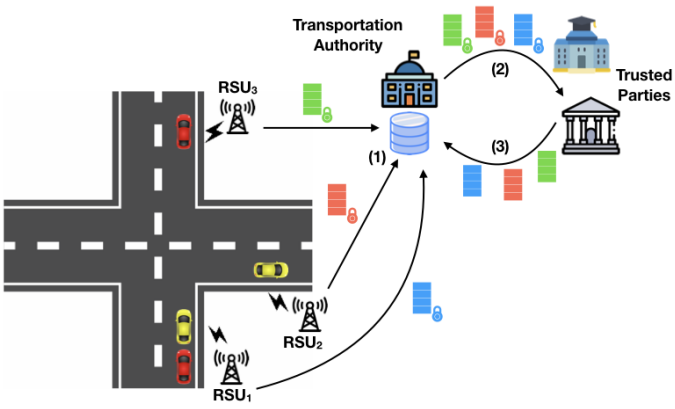


Fig. 1. System model

data through the RSUs—from decrypting the Bloom filters that are transmitted by the passing vehicles. Threshold decryption necessitates the collaboration of all involved parties, so a single honest player is sufficient for ensuring that only aggregate Bloom filters are being decrypted.

B. Threat Model

We assume that all entities in the aforementioned system model are semi-honest, i.e., they will execute all protocols correctly but will try to gain an advantage (in identifying vehicle-specific secret information) by examining the exchanged messages. We also allow collusions among the transportation authority and the RSUs, i.e., the adversary is given the full communication transcript of the underlying network. Furthermore, our methods do not require TLS-based communications to thwart eavesdropping attacks, because all transmitted information is encrypted. (Nevertheless, if we wish to authenticate the vehicles and/or receivers, we may utilize the TLS protocol with the existing public key infrastructure.) The only requirement with regards to privacy is that, out of the t trusted parties engaged in the threshold cryptosystem, at most $t-1$ of those can collude. Finally, we assume that the vehicles can remove all identifying information when communicating with a receiver, e.g., by spoofing the actual MAC address of their network interface card. While it is still possible to track a vehicle using road cameras or other devices, such attacks are out of the scope of this paper.

IV. Privacy-preserving Data Aggregation

Our solution comprises of three distinct phases: Bloom filter encryption, aggregation, and decryption. We describe all of them in detail in the following sections.

A. Bloom Filter Encryption

A Bloom filter [14] is a fast, memory-efficient, and probabilistic data structure that facilitates rapid searching. It is essentially an m-bit vector that is initialized with all its bits set to '0'. Prior to adding elements into the Bloom filter, we define k hash functions $H_1, H_2, \ldots, H_k$, where each hash function returns a value in the range $[0, m)$. To add an element v into the Bloom filter, we compute $H_i(v), \forall i \in \{1, 2, \ldots, k\}$ and set the corresponding bits in the bit vector to '1'. Unfortunately, aggregating Bloom filters with an additively homomorphic public key encryption scheme (such as Paillier) is infeasible, because the logical OR operation necessitates a fully homomorphic cryptosystem [24]. Instead, we may utilize *counting* Bloom filters [25] which are integer vectors that enable element deletions as well. In a counting Bloom filter, instead of setting the bits at the k vector positions, we simply increment the corresponding counters. However, this approach is vulnerable to correlation attacks, by inspecting the Bloom filters of two or more successive receivers. If their counters differ in only a small fraction of the m vector locations, an adversary may be able to deduce the k secret indexes of the non-common vehicles.

As a result, in this work, we employ a one-time pad cipher to encrypt the Bloom filter vectors. In particular, let $q = 2^\ell$ be a prime power and let $\mathbb{F}_q$ be the finite field of integers modulo q. For a vehicle v, its Bloom filter is represented as a vector $\mathbf{b} \in \mathbb{F}_q^m$, where $b_i, \forall i \in \{H_1(v), H_2(v), \ldots, H_k(v)\}$ is chosen uniformly at random from the range $[1, q)$. The remaining values are all set to zero. To encrypt its Bloom filter, the vehicle chooses a random vector $\mathbf{e} \leftarrow_{\$} \mathbb{F}_q^m$ and outputs the following ciphertext (in modulo q arithmetic).

$$\mathbf{c} = \mathbf{b} + \mathbf{e} \tag{1}$$

The next step is to devise an efficient method for vehicles to communicate the *aggregate* encryption keys to the transportation authority. For that purpose, we employ the additively homomorphic Paillier cryptosystem that is instantiated prior to system deployment. Under this cryptosystem, if $\mathsf{Enc}(a)$ and $\mathsf{Enc}(b)$ are the encryptions of messages a and b, respectively, we can blindly compute the encryption of message $(a + b)$ as $\mathsf{Enc}(a + b) = \mathsf{Enc}(a) \cdot \mathsf{Enc}(b)$. Assume now that the number of vehicles that may be summarized into a single Bloom filter is upper bounded by n. Then, the number of bits required to store a single Bloom filter entry is $\log n + \log q$. Based on the maximum message size that can be encrypted under Paillier (which depends on its RSA composite), we denote as l the max number of Bloom filter entries that can fit into a single Paillier ciphertext. Then, the vehicle will output the following ciphertext vector $\mathbf{r}$, where '|' denotes the concatenation operator.

$$\mathbf{r} = [\mathsf{Enc}(e_0|e_1|\ldots|e_{l-1}), \ldots, \mathsf{Enc}(\ldots|e_{m-2}|e_{m-1})]^\top \tag{2}$$

The size of vector $\mathbf{r}$ is $\lceil m/l \rceil$ and the elements e_i represent the elements of the key vector $\mathbf{e}$. The tuple $\langle \mathbf{c}, \mathbf{r} \rangle$ is the encrypted Bloom filter (i.e., identification) of that vehicle.

B. Bloom Filter Aggregation

When the vehicle encounters a new RSU, it will transmit its Bloom filter $\langle \mathbf{c}, \mathbf{r} \rangle$. After that, it will compute a re-randomized version of the Bloom filter, by choosing fresh random values for vectors $\mathbf{b}$ and $\mathbf{e}$. In this way, the vehicle can not be identified across multiple RSUs. Each RSU maintains, locally, an aggregate (encrypted) Bloom filter $\langle \mathbf{c}^A, \mathbf{r}^A \rangle$ that

summarizes the vehicles that have passed during the current measurement period. Once it receives a new sample from a passing vehicle, it updates the vectors as follows:

$$\mathbf{c}^A = \mathbf{c}^A + \mathbf{c} \tag{3}$$

$$\mathbf{r}^A = \mathbf{r}^A \odot \mathbf{r} \tag{4}$$

where $\odot$ denotes element-wise multiplication. When the current measurement period ends, the RSU will send $\langle \mathbf{c}^A, \mathbf{r}^A \rangle$ to the transportation authority, along with the duration of the measurement period (start and end time).

C. Bloom Filter Decryption

Decryption is a two-step process that involves the transportation authority and all the trusted third-parties. Once the transportation authority receives a new encrypted Bloom filter $\langle \mathbf{c}^A, \mathbf{r}^A \rangle$, it engages all t trusted parties to collectively decrypt the aggregate encryption key $\mathbf{e}^A = \sum_{i=1}^{n} \mathbf{e}^i$ from the ciphertext vector $\mathbf{r}^A$. This step entails the threshold decryption of $\lceil m/l \rceil$ Paillier ciphertexts. Next, it reduces (element-wise) $\mathbf{e}^A$ modulo q, and computes the plaintext of the aggregate Bloom filter as follows:

$$\mathbf{b}^A = \mathbf{c}^A - \mathbf{e}^A \tag{5}$$

It is important to note that, an adversary can not determine the number of vehicles that have set a certain bit, because the corresponding value is uniformly random in the range $[0, q)$. The downside of this approach is that certain Bloom filter entries that have been selected by *at least* two vehicles, may produce an incorrect value of zero (which signifies a '0' bit). However, the probability of that event is low. More specifically, let $P(i)$ be the probability that a certain Bloom filter entry is selected by exactly i out of n vehicles:

$$P(i) = \binom{nk}{i} \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{nk-i} \tag{6}$$

where m is the Bloom filter size and k is the number of hash functions. From this formula, we may compute the bit error probability P_{err} as follows:

$$P_{err} = [1 - P(0) - P(1)]\frac{1}{q} \tag{7}$$

As an example, if $n = 2000$, $m = 8000$, $k = 4$, and $q = 2^{10}$, the bit error probability is just 0.026%. As we will show in our simulation experiments, the effect of bit errors on the accuracy of our protocol is negligible.

D. Privacy Analysis

We define as a privacy breach the disclosure of a vehicle's secret Bloom filter. This may be accomplished by (i) performing ciphertext-only attacks on the underlying cryptosystems or (ii) analyzing Bloom filters from different receivers.

Regarding the first type of attack, we argue that it is infeasible because of the semantic security of the OTP and Paillier cryptosystems that render every message indistinguishable. Furthermore, according to our threat model stated in Section III-B, at least one of the trusted third-parties will not collude to decrypt individual Bloom filters. Instead, the only plaintext information available to the adversary is the aggregate encryption key $\mathbf{e}^A = \sum_{i=1}^{n} \mathbf{e}^i$ from the OTP ciphertexts of the n vehicles. That information alone is not sufficient to decrypt individual Bloom filters.

Indeed, let us consider a single Bloom filter entry j, and the n ciphertext values that are known by the adversary, namely $c_1, c_2, \ldots, c_n$. To retrieve the corresponding plaintext values b_i, the adversary must solve the following system of equations

$$c_i = b_i + e_i - s_i \cdot q, \forall i \in \{1, 2, \ldots, n\} \tag{8}$$

$$e_1 + e_2 + \ldots + e_n = e_j^A \tag{9}$$

Clearly, a unique solution does not exist, since there are $n+1$ equations with $3n$ unknowns. In fact, we can easily produce a solution for any combination of vehicles that have selected a non-zero value for that exact Bloom filter entry.

Therefore, the only viable attack vector for an adversary is to examine the Bloom filters from two different RSUs, whose Hamming distance is small. To this end, we consider the worst case scenario for our approach, which entails two almost identical datasets. In particular, consider the unlikely scenario where the adversary has knowledge (e.g., using external observations) that sets A and B (containing $n-1$ and n vehicles, respectively) are constructed such that all of A's vehicles are also present in B. (This is similar to the definition of differential privacy.) We can then determine the probability that the adversary can derive one or more of the k bit locations that identify the extra vehicle. Let P_i be the probability that we can identify exactly i out of k bits. This is equal to the probability that the i bits have been set by exactly one vehicle which, using Equation (6), can be written as:

$$P_i = P(1)^i = \left[\left(1 - \frac{1}{m}\right)^{nk} \frac{nk}{m-1}\right]^i \tag{10}$$

For instance, if $n = 2000$, $m = 8000$, and $k = 4$, the probability of recovering the vehicle's entire Bloom filter is just 1.8%.

V. Point-to-point Traffic Flow Estimation

Let us consider two sets A and B with cardinalities $n(A)$ and $n(B)$, respectively. According to basic set theory, the number of common elements, $n(A \cap B)$, is computed as

$$n(A \cap B) = n(A) + n(B) - n(A \cup B) \tag{11}$$

where $n(A \cup B)$ is the number of elements in their union. Assume now that the elements comprising A and B are unknown, but we do have their Bloom filter representations, BF_A and BF_B, in the form of m-bit vectors. Note that, if we compute the logical OR of BF_A and BF_B, we get the correct Bloom filter representation of $BF_{A \cup B}$. Therefore, to estimate the traffic flow between two receivers A and B, we need a formula that estimates the number of elements stored in a Bloom filter, based on the number of bits that are set.

To this end, Equation (6) can be used to estimate the cardinality of the underlying set, by measuring the fraction of bits that are '0'. More specifically, the probability that a bit is *not* set is given below:

$$P(0) = \left(1 - \frac{1}{m}\right)^{nk} \tag{12}$$

Solving for n gives us our estimate $\hat{n}$, which can be written as follows:

$$\hat{n} = \frac{\ln P(0)}{k \cdot \ln\left(1 - \frac{1}{m}\right)} \tag{13}$$

Therefore, given the Bloom filters from two distinct receivers A and B, the transportation authority will apply Equation (11) to estimate the volume of the traffic flow between them, where all set cardinalities are estimated using Equation (13).

VI. Simulation Experiments

A. Simulation Setup

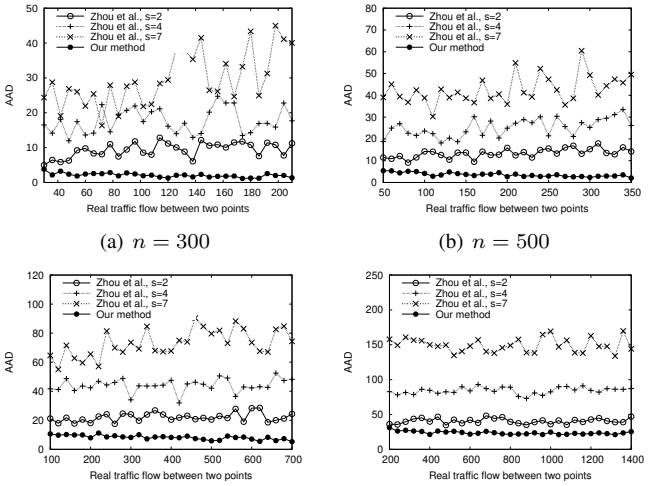
In this section, we evaluate the performance of our system in terms of accuracy and efficiency. To measure the accuracy, we simulated a pair of receivers, A and B, each holding a Bloom filter of size m that contains n entries (vehicles). We then vary the number of common vehicles passing through A and B to be in the range of 10%–70% of n, and measure the average absolute difference (AAD) between the real number of common vehicles and the estimated one. We performed each experiment 1000 times and compared our method against Zhou et al.'s state-of-the-art approach [13].

To measure the computational overhead, we implemented all cryptographic operations on a desktop machine with sixteen 3.0GHz CPUs and 64GB of memory. To simulate the limited computational capabilities of the vehicle and receiver, we employed a single CPU core to perform their tasks. On the other hand, the server process utilized all sixteen cores in a multi-threaded implementation. Our code was written in C++ and we leveraged the OpenSSL library[1] for arbitrary precision arithmetic operations. For sufficient security, the RSA modulus of the Paillier cryptosystem was set to 2048 bits, which produces ciphertexts of size 512 bytes.

B. Accuracy

Our basic motive is to design a system for fine-grained traffic flow estimation, so we considered a small number of vehicles ($n \leq 2000$) at each receiver. In this scenario, new Bloom filters will be generated at relatively short time intervals. Fig. 2 shows the AAD for different values of n, when $k = 4$ and $m = 8000$. For Zhu et al.'s method, we depict three different curves, i.e., for $s = 2, 4, 7$. Our approach clearly outperforms the competitors, especially for larger values of s. Note that, the value $s = 2$ is generally not recommended, due to insufficient privacy. When $s \geq 4$, our methods reduce the AAD by a factor of 3–8. In addition, our estimation error exhibits a significantly smaller variance across all settings.

[1]https://www.openssl.org/



(a) $n = 300$ (b) $n = 500$

(c) $n = 1000$ (d) $n = 2000$

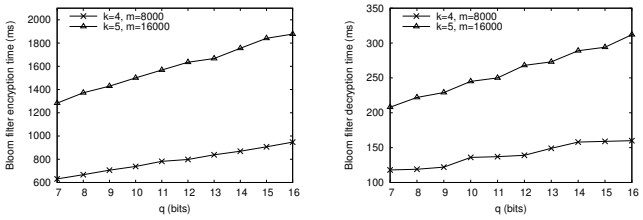
Fig. 2. AAD vs. real traffic flow ($k = 4, m = 8000, q = 2^7$)

C. Overhead

The main advantage of previous work [13] is the lack of cryptographic operations that result in a very efficient implementation. However, this has a negative impact on both the privacy of the vehicles and the accuracy of the traffic flow estimation. Therefore, to illustrate the feasibility of our approach, we need to investigate the overhead of the cryptographic operations in terms of both computation and communication costs. To this end, the two basic operations involved in our methods are the modular exponentiation (for Paillier encryption/decryption) and the modular multiplication (for Paillier homomorphic addition). In our software implementation, these operations cost, on average, 8 ms and 0.015 ms, respectively. Notice that the overhead of the OTP operations is negligible compared to the public key operations and is, thus, not measured in our results.

Fig. 3 shows the computational cost at the vehicle and the server, as a function of the bit-length of q. For the vehicle, the cost involves the encryption of the OTP keys into multiple Paillier ciphertexts. For $q = 2^7$ the cost is just 600 ms (for $m = 8000$), which is long enough for the vehicle to compute a new Bloom filter before reaching the next RSU. It is also possible for the vehicle to pre-compute (offline) several Bloom filters before the start of a new trip. Similarly, the cost at the server shows the time needed to decrypt a single aggregate Bloom filter (each trusted party will incur this cost). Here, the server application employs all sixteen CPU cores, so the cost is greatly reduced.

Finally, we investigate the scalability of our approach with regards to the vehicle throughput that can be supported at the RSUs. First, the computational cost involves only modular multiplications, which are considerably cheaper than exponentiations. As such, adding one vehicle to the aggregate Bloom filter entails just 1 ms of CPU time for $n = 2000$, $m = 8000$, and $q = 2^7$. At this rate, the receiver can process approximately 1000 vehicles/sec. On the other hand,



(a) Vehicle (b) Server

Fig. 3. Computational cost at vehicle and server ($n = 2000$)

the communication cost is the bottleneck with regards to the processing throughput. Indeed, the data rate of the DSRC protocol is between 6–27 Mbps, which can only support a limited number of Bloom filter transmissions within any given time period. As an example, when $n = 2000$, $m = 8000$, and $q = 2^7$, the size of a single Bloom filter is 43 KB. Fig. 4 shows the processing throughput as a function of the available bandwidth, which demonstrates that at 10 Mbps, the system is able to accommodate the load of a typical rush hour traffic.

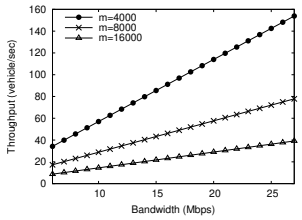


Fig. 4. Throughput vs. DSRC bandwidth ($n = 2000$, $q = 2^7$)

VII. Conclusions

We studied the problem of point-to-point traffic flow estimation in road networks. Our objective was to provide very accurate statistics, while at the same time protect the privacy of the underlying vehicles. To this end, we proposed a protocol based on vehicle-to-infrastructure communications, where vehicles transmit encrypted Bloom filters that are homomorphically aggregated at the roadside units. The aggregate Bloom filters are decrypted by the transportation authority, where the intensity of the traffic flow between two endpoints is measured by estimating the number of common vehicles in their Bloom filters. Our simulation results show that, compared to the current state-of-the-art, our methods improve the estimation accuracy by a large factor. In addition, our preliminary implementation demonstrates the feasibility and scalability of the system.

References

[1] M. Castro-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, no. 2, pp. 2979–2986, 2009.

[2] J. K. Eom, M. S. Park, T.-Y. Heo, and L. F. Huntsinger, "Improving the prediction of annual average daily traffic for nonfreeway facilities by applying a spatial statistical method," *Transportation Research Record*, vol. 1968, no. 1, pp. 20–29, 2006.

[3] W. H. Lam and J. Xu, "Estimation of AADT from short period counts in Hong Kong–A comparison between neural network method and regression analysis," *Journal of Advanced Transportation*, vol. 34, no. 2, pp. 249–268, 2000.

[4] D. Mohamad, K. C. Sinha, T. Kuczek, and C. F. Scholer, "Annual average daily traffic prediction model for county roads," *Transportation Research Record*, vol. 1617, no. 1, pp. 69–77, 1998.

[5] Y. Lou and Y. Yin, "A decomposition scheme for estimating dynamic origin–destination flows on actuation-controlled signalized arterials," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 643–655, 2010.

[6] N. Caceres, J. Wideberg, and F. Benitez, "Deriving origin–destination data from a mobile phone network," *IET Intelligent Transport Systems*, vol. 1, no. 1, pp. 15–26, 2007.

[7] J. White and I. Wells, "Extracting origin destination information from mobile phone data," in *Proc. International Conference on Road Transport Information and Control*, 2002.

[8] T. Jeske, "Floating car data from smartphones: What google and waze know about you and how hackers can control traffic," *Proc. of BlackHat Europe*, pp. 1–12, 2013.

[9] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transportation Research Record*, vol. 1855, no. 1, pp. 49–59, 2003.

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

[11] Y. L. Morgan, "Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 504–518, 2010.

[12] Y. Zhou, S. Chen, Y. Zhou, M. Chen, and Q. Xiao, "Privacy-preserving multi-point traffic volume measurement through vehicle-to-infrastructure communications," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5619–5630, 2015.

[13] 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*, vol. 65, no. 5, pp. 3749–3759, 2016.

[14] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[15] I. Tsapakis, W. H. Schneider IV, and A. P. 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*, vol. 36, no. 2, pp. 201–217, 2013.

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

[17] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proc. SecureComm*, 2005, pp. 194–205.

[18] A. Tomandl, D. Herrmann, and H. Federrath, "PADAVAN: privacy-aware data accumulation for vehicular ad-hoc networks," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2014, pp. 487–493.

[19] S. Rass, S. Fuchs, M. Schaffer, and K. Kyamakya, "How to protect privacy in floating car data systems," in *Proc. ACM International Workshop on Vehicular Ad Hoc Networksd (VANET)*, 2008, pp. 17–22.

[20] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2008, pp. 15–28.

[21] D. Förster, H. Löhr, and F. Kargl, "Decentralized enforcement of k-anonymity for location privacy using secret sharing," in *Proc. IEEE Vehicular Networking Conference (VNC)*, 2015, pp. 279–286.

[22] Y. Zhou, S. Chen, Z. Mo, and Y. Yin, "Privacy preserving origin-destination flow measurement in vehicular cyber-physical systems," in *Proc. IEEE International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, 2013, pp. 32–37.

[23] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of paillier's probabilistic public-key system," in *Proc. International Workshop on Practice and Theory in Public Key Cryptography (PKC)*, vol. 1992, 2001, pp. 119–136.

[24] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. ACM Symposium on Theory of Computing (STOC)*, vol. 9, no. 2009, 2009, pp. 169–178.

[25] L. Fan, P. Cao, J. M. Almeida, and A. Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 281–293, 2000.