User-Defined Privacy-Preserving Traffic Monitoring Against n-by-1 Jamming Attack

Li, Meng; Zhu, Liehuang; Zhang, Zijian; Lal, Chhagan; Conti, Mauro; Alazab, Mamoun

DOI
10.1109/TNET.2022.3157654

Publication date
2022

Document Version
Final published version

Published in
IEEE/ACM Transactions on Networking

User-Defined Privacy-Preserving Traffic Monitoring Against n-by-1 Jamming Attack

Meng Li, *Member, IEEE*, Liehuang Zhu, *Senior Member, IEEE*, Zijian Zhang, *Member, IEEE*, Chhagan Lal, *Member, IEEE*, Mauro Conti, *Fellow, IEEE*, *Senior Member, ACM*, and Mamoun Alazab, *Senior Member, IEEE*

***Abstract*— Traffic monitoring services collect traffic reports and respond to users' traffic queries. However, the reports and queries may reveal the user's identity and location. Although different anonymization techniques have been applied to protect user privacy, a new security threat arises, namely, *n-by-1 jamming attack*, in which an anonymous contributing driver impersonates n drivers and uploads n normal reports by using n reporting devices. Such an attack will mislead the traffic monitoring service provider and further degrade the service quality. Existing traffic monitoring services do not support customized queries, and private information retrieval techniques cannot be applied directly in traffic monitoring. We formally define the new attack and propose a traffic monitoring scheme TraJ to defend the attack and achieve user-defined location privacy. Specifically, we bridge anonymous contributing drivers without disclosing their speed set by using private set intersection. Each RSU collects time traffic reports and structures a weighted proximity graph to filter out malicious colluding drivers. We design a user-defined privacy-preserving query method by encoding complex road network. We leverage the uploading phase from private aggregation to collect traffic conditions and allow requesting drivers to dynamically and privately query traffic conditions. We provide a formal analysis of TraJ to prove its privacy and security properties. We also construct a prototype based on a real-world dataset and Android smartphones to demonstrate its feasibility and efficiency. A formal analysis demonstrates the privacy and security properties. Extensive experiments illustrate the performance and defense efficacy.**

***Index Terms*— Vehicular networks, traffic monitoring, security, privacy, edge computing, proximity graph.**

Manuscript received 29 May 2021; revised 16 December 2021 and 12 January 2022; accepted 5 March 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Y. Chen. Date of publication 23 March 2022; date of current version 17 October 2022. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62002094, in part by the Anhui Provincial Natural Science Foundation under Grant 2008085MF196, and in part by the EU LOCARD Project under Grant H2020-SU-SEC-2018-832735. *(Corresponding author: Liehuang Zhu.)*

Meng Li is with the Key Laboratory of Knowledge Engineering With Big Data (Hefei University of Technology), Ministry of Education, the Anhui Province Key Laboratory of Industry Safety and Emergency Technology, the Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), and the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: mengli@hfut.edu.cn).

Liehuang Zhu and Zijian Zhang are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: liehuangz@bit.edu.cn; zhangzijian@bit.edu.cn).

Chhagan Lal is with the Department of Intelligent Systems, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: c.lal@tudelft.nl).

Mauro Conti is with the Department of Mathematics and HIT Center, University of Padua, 35131 Padua, Italy, and also with the CyberSecurity Group, Department of Intelligent Systems, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: conti@math.unipd.it).

Mamoun Alazab is with the College of Engineering, IT and Environment, Charles Darwin University, Casuarina, NT 0810, Australia (e-mail: alazab.m@ieee.org).

Digital Object Identifier 10.1109/TNET.2022.3157654

I. INTRODUCTION

TRAFFIC monitoring in intelligent transportation systems [1], [2] has attracted increasing attention from both academia [3]–[6] and industry [7], [8]. A Traffic Monitoring Service Provider (TMSP) collects real-time traffic information from contributing drivers to provide traffic querying services to requesting drivers. With such traffic information, drivers can reduce travel time and fuel. In addition, the urban planning departments can further optimize future road construction plans. However, a forensic engineer extracted the data from a Chevrolet entertainment computer and determined that a modern vehicle could produce up to 25 gigabytes of data per hour from all sensors implemented in the vehicle [9]. If all the data are uploaded to the remote cloud for processing, the response time is inevitably long. Edge computing can be applied to address this problem. Since data are consistently produced at the edge of the vehicular network, processing the data and responding to drivers at the edge of the network would be more efficient [10]. For example, Li *et al.* [5] presented a privacy-preserving traffic monitoring scheme that utilizes edge nodes, i.e., road-side units (RSUs), to collect and process traffic information locally. This scheme addresses response time requirements, reduces network bandwidth usage, and ensures driver privacy are properly addressed.

Despite offering convenient and helpful service and promising market prospects, traffic monitoring services raise high privacy and security concerns [5], [11]. The TMSP collects a large amount of data (location and speed) from drivers [5] that contain sensitive information. By analyzing these data, the TMSP can acquire mobility patterns and extract more private information, such as home, religious places, and sexual inclinations [12].

There are **three motivations** behind this work. First, although traffic information can be collected in an anonymous and authenticated manner, it is still possible for malicious drivers to launch an n-by-1 jamming attack, as illustrated in Fig. 1. A malicious driver leverages n reporting devices (e.g., smartphones and tablet computers) to submit n normal traffic information reports to the TMSP. Afterward, the TMSP

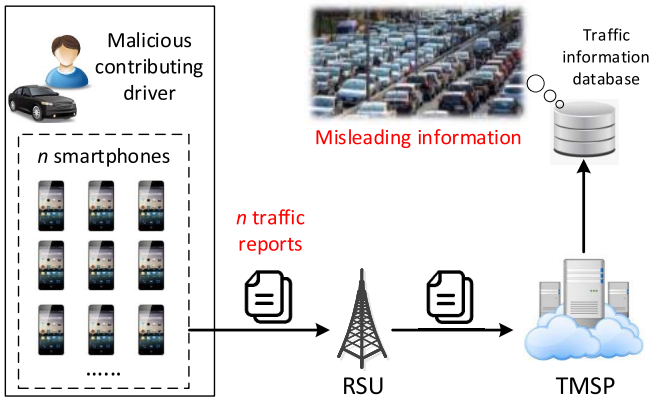


Fig. 1. Illustration of the n-by-1 jamming attack.

is tricked into believing a traffic jam on the driver's road. A news report states that an artist successfully tricked Google Maps [7] into believing a massive traffic jam on an empty street by slowly pulling a wagon with 99 cellphones on a road [13]. The **consequences** of this attack are severe. It causes a real impact in the physical world by rerouting vehicles which choose to avoid the reported (fake) traffic jam, disrupting the schedule of smart traffic lights, and even creating extra traffic jams on other roads. Second, drivers may leak their privacy when uploading/retrieving traffic information to/from the TMSP. For example, sending a traffic report has to include a real-time location, and sharing a sequence of real-time locations with the RHSP will expose the mobility pattern of a driver. Third, the TMSP should allow requesting drivers to control the granularity of how their location is protected when they are querying the traffic on certain roads. In other words, user-defined privacy [14]–[16] should be achieved.

To address the above issues, we propose TraJ, an innovative traffic monitoring scheme that supports user privacy, resists the n-by-1 jamming attack, and achieves user-defined location privacy. To achieve this goal, our work addresses the following **three technical challenges**: (1) How to defend against the new n-by-1 jamming attack when malicious drivers launch such an attack in an anonymous environment? Given the anonymization mechanism, the contributing drivers and their reports are made anonymous and unlinkable. This privacy feature, in turn, increases the possibility of users' misbehavior. Even if a trusted third party exists, it is still difficult for the TMSP to detect the attack. (2) How to design a customized and location privacy-preserving traffic query method for requesting drivers who have different choices of the queried roads in a complex road network without affecting efficiency? Requesting drivers query the traffic condition on a road segment, a road, or a route including several roads. Improper handling of the road network will result in low efficiency and bad user experience, let alone user-defined query. Meanwhile, we aim to protect location privacy based on the customized query, which requires a lightweight approach to balance efficiency and privacy. (3) How to allow requesting drivers to retrieve traffic information privately from the CS while they are not the traffic provider when using Private Information Retrieval (PIR)? At first glance, it is appropriate that we bring mature PIR techniques into traffic monitoring services to achieve privacy-preserving traffic uploading and requesting. But in PIR, it is the same client who uploads the indices to the server in the offline phase and then retrieves some bits in the online phase [17]–[19]. However, in the traffic monitoring service, i.e., crowdsensing platform, requesting drivers are not contributing drivers. If we ask the contributing drivers to upload traffic conditions via traditional PIR, the requesting drivers will not know which index to query.

To tackle the technical challenges, we offer the following **technical contributions**:

- We enhance the security model of traffic monitoring by considering the n-by-1 jamming attack from colluding contributing drivers that is possible in real life. Specifically, we give a formal definition of the new attack.
- We propose a traffic monitoring scheme TraJ to defend the n-by-1 jamming attack and achieve user-defined location privacy. For the first goal, we bridge anonymous contributing drivers without disclosing their speed set by using Private Set Intersection (PSI) [20]. Each RSU collects real-time traffic reports and structures a weighted proximity graph [21]. By analyzing the mass propagation in the graph and comparing features between normal case and abnormal case, we detect the n-by-1 jamming attack, thus solving the first challenge. For the second goal, we design a user-defined privacy-preserving query method. We organize the complex road network into a tree-structured hierarchy and encode roads to create flexible road choices, thus solving the second challenge. Next, we leverage the uploading phase from private aggregation [19] to collect traffic conditions. Requesting drivers can dynamically choose a road code according to their required level of location privacy without communication with the contributing drivers, thus solving the third challenge.
- We provide a formal analysis of TraJ to demonstrate its privacy and security properties. We also construct a prototype based on a real-world dataset and Android smartphones to demonstrate its feasibility and efficiency.

The rest of this paper is organized as follows. We review the related work in Section 2. Section 3 formalizes the problem. Section 4 introduces the necessary preliminaries. In section 5, we present the details of TraJ, followed by the privacy and security analysis in Section 6 and performance evaluation in 7, respectively. We provide some discussions in Section 8 and conclude the paper in Section 9.

II. Related Work

Hoh *et al.* [3] proposed a traffic monitoring scheme based on a GPS-enabled cellular phone platform utilizing the concept of virtual trip lines and a cloaking technique. A virtual trip line is a locational marker that notifies vehicles where to provide traffic information. The idea behind this approach is that traffic information on certain locations is more useful while some are more privacy-sensitive. Temporal cloaking is used to aggregate several location updates based on the identifiers of trip lines. However, they did not consider energy consumption and the gathered traffic information is not accurate because certain locations are omitted. The provided location k-anonymity lacks formal security proof.

Basudan *et al.* [4] proposed a road surface condition monitoring scheme called CLASC, which is based on a certificateless aggregate signcryption protocol. A mobile sensor reports a road event containing time, location, and signals. The event is processed into a secure road event report by the signcrypt function in the proposed protocol. Then, an aggregator collects local secure events to perform aggregation and batch verification on ciphertexts. If the aggregated result is valid, it will be forwarded to an RSU which unsigncrypts ciphertexts to obtain original events. However, the sensitive information is revealed to the RSU.

Li *et al.* [5] presented a privacy-preserving traffic monitoring scheme PAM to protect driver privacy and defend the false reporting attack. A contributing driver first undergoes a WiFi challenge handshake with drivers nearby to obtain proof of location and speed. Then, she/he forms an encrypted traffic report based on the BBS group signature [22]. The report is forwarded to a local fog node for verification. After collecting enough reports, the RSU forms a weighted proximity graph to screen malicious drivers by propagating trust values and compare each node's trust value with a threshold. However, PAM suffers from a high computational cost and did not consider the n-by-1 jamming attack.

Wang *et al.* [6] presented a cloud-based road condition monitoring scheme RCoM. They focus on three issues: authorized reporting, privacy-preserving monitoring, and source authentication. A vehicle interacts with an RSU to receive a token. Using the token, the vehicle submits an encrypted report to the server. The server first verifies the soundness of the reports and divides them into different equivalence groups by using the equality test. If the number of one group exceeds a threshold, the server notifies the root authority of an emergency case. Finally, the root authority decrypts and verifies the reported road condition. However, they did not consider the n-by-1 jamming or collusion.

The promotion over existing work in this paper is that TraJ resists the n-by-1 jamming attack in an anonymous environment, enables contributing drivers to privately handshake with nearby contributing drivers and allows requesting users to query traffic conditions on customized choice of roads without disclosing their location privacy.

III. Problem Statement

A. System Model

The TraJ system model consists of five types of entities: a trusted third party (TTP), a TMSP, contributing drivers, requesting drivers, and RSUs. It is displayed in Fig. 2 and key notations are explained in Table I.

TTP monitors the real-time road traffic with the help of the TMSP, so that it could make a timely response to emergency cases. It also initializes the TraJ system by generating public parameters and cryptographic keys for the TMSP, RSUs, and drivers. The TTP can be the Department of Transportation in a real-world implementation. It divides time into a sequence of time epochs. It also divides the managed roads into different road segments for each RSU. In this way, drivers will be able to transform their location into a road segment.

TMSP is a traffic data center with significant communicating and computing capabilities. It is engaged by the

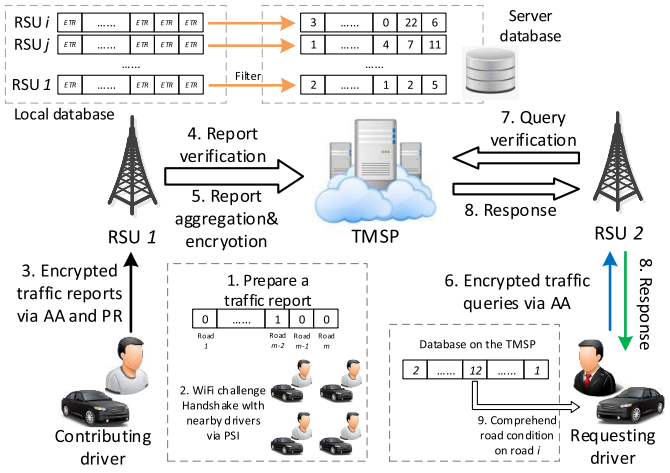


Fig. 2. Traffic uploading, traffic processing, and traffic querying.

TTP to collect/provide real-time traffic information from/to contributing/requesting drivers. At the very beginning, the TTP assigns a region covering a set of m roads and an instruction of indicators to each RSU. The instruction states that each index in the array corresponds to a specific road and the traffic indicator is a string of $\lceil log_2 w \rceil$ bits where w is the total number of traffic conditions. To improve monitoring accuracy, we could partition a road into several sections. In this work, we use $w = 4$ types of conditions: uncrowded (00), slightly congested (01), congested (10), and highly congested (11).

Contributing driver is the one holding real-time and local traffic information to be submitted. She/he first pairs with nearby drivers by undergoing a WiFi challenge handshake automatically and periodically. Then, the driver prepares a traffic report, i.e., an array of m bits. Each index in the array is either 0 or 1, where 0 means the driver is not on this road corresponding to the index and 1 otherwise. Next, the contributing driver uploads an encrypted traffic report to the TMSP via a local RSU. Each contributing driver is only allowed to submit one traffic report in one time epoch due to three reasons: first, the service provider periodically updates the traffic map, which needs a time epoch; second, normally, a driver can only submit one report at a time through manual operations; third, the duration of one time epoch can be adjusted according to an application, such as one second and five seconds. The incentive for drivers to upload their locations is the same as the one in other crowdsensing services, that is, sharing their own data to other users and expecting to receive requested data in return.

Requesting driver is the one expecting local or faraway traffic information. She/he submits an encrypted traffic query to a local RSU and awaits a traffic response. Then the requesting driver can choose an optimal route based on the returned traffic information. Each requesting driver is only allowed to submit one traffic query in one time epoch because we need to prevent malicious requesting drivers from submitting multiple queries in one time epoch to consume too many resources of the TMSP. The current formation requires requesting drivers to consecutively submit queries over a period of time to obtain the real-time information because the traffic condition is time-varying, and we need to guarantee that the returned traffic

conditions are fresh and useful to requesting drivers. The dynamic nature of traffic condition originates from several reasons, such as wide moving jam, butterfly effect, and road bottleneck [23]. For example, the traffic jam from the butterfly effect happens when a seemingly small disturbance in the normal flow of traffic, such as a vehicle suddenly changing lanes, results in a sequence of events that causes other vehicles to slow down. It is reported that a road traffic collision caused six miles of congestion and at least 60 minute-delay on the M6 northbound in Lancashire [24].

RSU is a road-side edge node with sufficient communicating and computing resources. It is deployed by the TMSP or the mobile network operator and is distributed in different regions. Each RSU collects real-time traffic reports and pairings from contributing drivers in its coverage region and acts as a frontline guardian for the system. It broadcasts the number of roads for the vector used in traffic uploading. It sends authenticated, verified, and filtered traffic reports to the TMSP and updates the received array stored in a local database. Meanwhile, each RSU responds to requesting drivers' traffic queries by searching in the local array or asking the TMSP for assistance if necessary.

TABLE I
KEY NOTATIONS IN TRAJ

Notation	Definition
CD, RD, RU	Contributing driver, requesting driver, RSU
n	Maximum number of devices by adversary
N	Maximum number of drivers in a region
m	Maximum number of road segments in RSU area
w	Total number of traffic conditions
q, g, e	Big prime, group generator, bilinear pairing
(x_1, x_2, x_3)	Service secret key
(X_1, X_2, X_3)	Service public key
E, A	Encoder, analyser
H, l	Hash function, bit length of $H(\cdot)$
C, D	Linear error-correcting code, decode function
S, sk	Signature of TTP, secret key of driver
$priK$, $pubK$	Private key, public key of RSU and TMSP
Arr	Local array in an RSU
TR, ETR	Traffic report, encrypted traffic report
pID	Pseudo-name of a contributing driver
DS, ds	Driving speed set, driving speed
$\hat{S}$, Tk	Blinded signature, login token
L	Maximum number of collected reports for an RSU
ATR	Aggregated and encrypted traffic report
TC, TQ	Traffic condition, traffic query

B. Threat Model

Privacy and security threats are raised from external and internal attackers for the traffic monitoring system. An external adversary eavesdrops on communication channels and captures transmitting messages to violate drivers' privacy. It can also launch the impersonation attack, message tampering attack, and replay attack.

The TTP is a fully trusted entity, and it cannot be breached by adversaries. The assumption of a perfectly secure TTP, as a common practice, is widely acknowledged in privacy-preserving schemes, especially vehicular schemes, such as: Trusted Server in W^3-$tess$ [25], Crypto Provider in pRide [26], Trusted Authority in NRS/TRS [27], Authority in lpRide [28], and Trusted Authority in PAM [5]. We follow this assumption, however, we also that the untrusted model for the TTP is an interesting research direction.

The TMSP and distributed RSUs are honest-but-curious. It means that they can intentionally try to recover the contents from the already collected data to infer drivers' privacy, namely their identity and location [29], [30].

Most contributing/requesting drivers are honest-but-curious and will send traffic reports/queries truthfully. A small part of contributing drivers is malicious. They can launch multiple uploading attack [31] and n-by-1 jamming attack. The multiple uploading attack refers to a contributing driver submitting multiple traffic reports at the same location by using one device/account within one time epoch. The immediate upload of multiple traffic reports at the same location is treated as an attack because it will trick the service provider and RSU into believing that there is a possible traffic jam at the driver's location. The n-by-1 jamming attack refers to a contributing driver submitting n traffic reports by using n devices/accounts within one time epoch. Specifically, the reports include abnormal reports and normal reports. The former refers to a malicious contributing driver submitting false traffic reports. The normal traffic report from such a driver is more concealing as it usually does not raise an alarm. Compared to other attacks, the strong demand for the n-by-1 jamming attack is to create a fake traffic jam. Now we give its formal definition.

Definition 1 (n-by-1 Jamming Attack): Given a traffic condition set $\mathcal{TC} = \{tc_1, tc_2, \ldots, tc_w\}$, a contributing driver set $\mathcal{CD} = \{cd_1, cd_2, \ldots, cd_N\}$, a reporting device set $\mathcal{RD} = \{rd_1, rd_2, \ldots, rd_n\}$, and an adversary $\mathcal{A}$, we define a monitoring function $F : \mathcal{A} \cup \mathcal{CD} \rightarrow \mathcal{TC}$. Normally, we have $F(\mathcal{CD}) \rightarrow \mathcal{TC}$, but the n-by-1 jamming attack makes $F(\mathcal{A}) \rightarrow \mathcal{TC}\backslash\{tc_1\}$ where tc_1 corresponds to the lowest congested traffic condition, i.e., no traffic jam. Note that $\mathcal{TC}\backslash\{tc_1\}$ is the function output of the monitoring function, and it is also the attack result set of the adversary. It means the adversary can behave arbitrarily when submitting traffic reports. We exclude the tc_1 because it is not considered an attack.

To defend this attack, we design a new mechanism F' satisfying

$$|\Pr[F(\mathcal{CD}) \rightarrow \mathcal{TC}] - \Pr[F'(\mathcal{A}) \rightarrow \mathcal{TC}]| \leq \text{negl}(n).$$

The **motivations** for the malicious driver launching the n-by-1 jamming attack are three-fold. First, it could be a random sabotage on the traffic monitoring service from a mischievous user. Second, it could be the intention to deliberately create a virtual traffic jam to clear the path for the adversary itself. Another incentive arises from a business competitor's agenda to disrupt an opponent's traffic monitoring service and attract users. A collusion attack is considered between malicious drivers who initiate the n-by-1 jamming attack. Moreover, a small part of requesting drivers are malicious, and they can launch multiple querying attacks, where a requesting driver submits multiple traffic queries within one time epoch.

A concrete transportation service example on this crowdsensing service: Say some contributing drivers, including Alice, are driving on road Park Avenue. They bridge with each other via WiFi challenge handshake to generate and

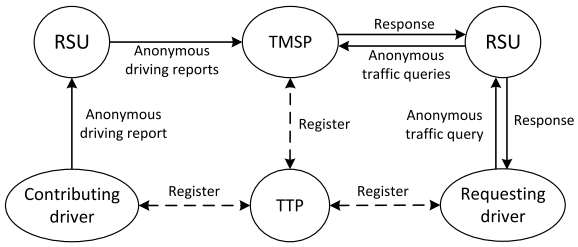


Fig. 3. Information flow chart.

send anonymous but bridged traffic reports to a local RSU. The RSU collects local reports within its coverage area, authenticates them, and establishes a weighted proximity graph of the drivers by using the handshakes between drivers and handshake numbers. After propagating mass values in the graph, the RSU filters out drivers with abnormal mass values and sends the normal reports to the TMSP. A requesting driver Bob is about to drive through the Park Avenue. To find out whether the road ahead is congested, he sends an anonymous traffic query to the RSU. The RSU authenticates the query and retrieves a query result from the TMSP for Bob. We show the information flow chart in Fig. 3.

C. Design Objectives

Privacy. (1) Identity privacy. When drivers participate in the system, their real identities must be hidden from other drivers, RSUs, the TMSP, and external entities when sending a traffic report or traffic query. (2) Data privacy. Contributing drivers' traffic reports must be hidden from other entities. (3) Location privacy. Contributing drivers' locations are protected from other entities. Furthermore, user-defined location privacy should be achieved in traffic querying, i.e., requesting drivers' locations are protected such that they can control the granularity of how their locations are preserved. (4) Unlinkability. Given two encrypted traffic reports/queries, any adversaries cannot decide whether they are produced from the same contributing/requesting driver.

Security. (1) Drivers' identities are authenticated. (2) The traffic monitoring system should resist the multiple requesting/querying attack, i.e., only one traffic report/query is allowed to be sent under one anonymous credential at one time. (3) The system should resist the n-by-1 jamming attack, i.e., the situation where an adversary sends n traffic reports by n reporting devices should be detected with a high probability.

Efficiency. The proposed scheme should achieve good efficiency with respect to computational costs and communication overhead for the driver, RSU, and TMSP.

IV. Preliminaries

A. One-Time and Anonymous Subscription

The one-time and anonymous subscription scheme mainly includes three phases: Setup: Given a security parameter 1^k, an authority selects a secret key, computes a corresponding public key, and initializes a login state. Registration: An entity sends a zero-knowledge proof of knowledge (ZKPoK) [33] with non-interactability [34] on two random numbers to the authority and receives a signature. The entity verifies the signature and obtains a secret key. Login: An entity sends a blinded signature, a login token, and a ZKPoK to the authority. The authority verifies the signature, the token, and to authorize the login.

B. Private Aggregation

The private aggregation protocol is an n-user secure protocol in the shuffled model that consists of a randomized encoder E and an analyzer A. In the protocol, each user i has an input a_i and encodes it to $\mathsf{E}(a_i)$. E can be random according to the private randomness of i. A shuffler receives n encodings $EC = (\mathsf{E}(a_1), \mathsf{E}(a_2), \ldots, \mathsf{E}(a_n))$ and applies a uniformly random permutation π on the n encodings. The permuted is sent to the analyzer which outputs $\mathsf{A}(EC)$. Specifically, we are concerned with the aggregation function and $\mathsf{A}(EC) = \sum_{i=1}^{n} a_i$.

C. Private Set Intersection

The private set intersection (PSI) is an two-user secure protocol in an interactive model that consists of a sender and a receiver. The sender chooses a value a and the receiver chooses a value vector $A_s = (a_1, a_2, \ldots, a_t)$. The sender and the receiver execute t instances of an oblivious transfer extension protocol of OOS [35], then the sender obtains a value vector $B = (b_1, b_2, \ldots, b_t)$. The sender computes t values $H(b_1 \oplus \mathsf{C}(00\ldots0) \wedge a), H(b_2 \oplus \mathsf{C}(00\ldots1) \wedge a), \ldots, H(b_t \oplus \mathsf{C}(11\ldots1) \wedge a)$ where C is a linear error-correcting code and H is a secure hash function. The receiver obtains C containing t values $C = (c_1, c_2, \ldots, c_t)$ where $c_i = b_i \oplus \mathsf{C}(d_i) \wedge a$ and d_i is a choice value. Then the sender sends $H(\mathsf{D}(B, x) \oplus \mathsf{C}(x) \wedge a)$ for each item x to the receiver which compares them to $H(\mathsf{D}(C, y))$ for each item y to determine whether there is an intersection where $\mathsf{D}(D, x) = x$ and the D has a homomorphic property.

D. Edge Computing

Edge computing [10] extends a part of the cloud's functions to the network edge by collecting and processing local data using edge nodes. An edge node has enough computing and communicating capabilities to respond to local queries from users. Edge computing offers several advantages, such as geo-distribution, location awareness, and low latency. Edge computing has been adopted in vehicular networks [5] where RSUs act as edge nodes to process traffic reports and requests to deliver local traffic monitoring services. The edge nodes are deployed by the service provider or the mobile network operator. We are facing new security and privacy threats as the edge nodes are not fully trusted. We need to pay attention to the potential risks incurred by the adoption of edge computing. In this work, we utilize edge computing to ease the burden of the TMSP and locally process traffic reports/queries in a privacy-preserving way.

V. Proposed Scheme

A. Overview

At a high level, TraJ consists of five phases: system initialization, entity registration, traffic uploading, traffic processing,

and traffic requesting. In system initialization, the TTP generates all the system parameters for anonymous authentication, WiFi handshake challenge, private aggregation, and private set intersection. In entity registration, contributing drivers, requesting driver, RSUs, and the TMSP register to the TTP and obtain corresponding keys. In traffic uploading, each contributing driver pairs with nearby drivers via a WiFi handshake challenges, and uploads an encrypted traffic report to a local RSU. In traffic processing, the RSU verifies the driver identity and data integrity, filters jamming reports, and sends an aggregated and encrypted traffic report to the TMSP. The TMSP constructs a tree-structured traffic map, codes each road according to its location in the map, and then publishes all the codes. In traffic querying, each requesting driver sends an anonymous traffic query to an RSU. The RSU verifies the driver identity and data integrity, retrieves a traffic response from the local database or the TMSP, and returns it to the driver.

We provide an overview of the TraJ by using Algorithm 1. It includes inputs and outputs, and the different steps that are carried out during the execution of TraJ.

Algorithm 1 TraJ

Input: 1^k, 1^{k^1}, 1^{k^2}, rd, cd, P_S, P_E, $\bar{P}$.
Output: $p.p$, sk, SK, $\mathcal{HKP}$, gsk^{rd}, gsk^{rd}, $(\mathsf{sk}^{ru}, \mathsf{pk}^{ru})$, $(\mathsf{sk}^{sp}, \mathsf{pk}^{sp})$, NQ, DR, $\mathcal{LDB}$, TQ, $\mathcal{TDB}$, ID.
/*System Initialization*/
1. TA generates public parameters $p.p.$, secret keys sk, SK, and $\mathcal{HKP}$ given security parameters 1^k, 1^{k^1}, 1^{k^2};
/*Entity Registration*/
2. rd obtains a group secret key gsk^{rd};
3. cd obtains a group secret key gsk^{rd};
4. ru registers to obtain a key pair $(\mathsf{sk}^{ru}, \mathsf{pk}^{ru})$;
5. NSP registers to obtain a key pair $(\mathsf{sk}^{sp}, \mathsf{pk}^{sp})$;
/*Traffic Uploading*/
6. rd sends a navigation query NQ to an RSU from start point P_S and endpoint P_S;
/*Traffic Processing*/
7. RSU verifies NQ, broadcasts a traffic task;
8. cd sends RSU a driving report DR from location $\bar{P}$;
9. RSU verifies and filters a set of $\{DR\}s$, stores the rest in local databaes $\mathcal{LDB}$ and sends them to NSP;
/*Traffic Querying*/
10. rd sends TQ to the RSU;
11. RSU searches $\mathcal{LDB}$ (requests NSP if needed) and returns a result to rd;
12. rd receives the traffic query result;
13. NSP verifies the received reports and forms a traffic congestion database $\mathcal{TDB}$;
14. Return $p.p.$, sk, SK, $\mathcal{HKP}$, gsk^{rd}, gsk^{rd}, $(\mathsf{sk}^{ru}, \mathsf{pk}^{ru})$, $(\mathsf{sk}^{sp}, \mathsf{pk}^{sp})$, NQ, DR, $\mathcal{LDB}$, TQ, $\mathcal{TDB}$, ID.

B. System Initialization

First, given a security parameter 1^k, the TTP generates a bilinear group $\mathbb{G} = \langle g \rangle$ of prime order q with a target group $\mathbb{G}_T$, a bilinear pairing $e(\cdot,\cdot)$, and $g_T = e(g,g)$. It selects $x_1, x_2, x_3 \leftarrow \mathbb{Z}_q$ and computes $X_1 = g^{x_1}$, $X_2 = g^{x_2}$, and $X_3 = g^{x_3}$. The TTP sets the service secret key $SK = (x_1, x_2, x_3)$ and the service public key $PK = (q, \mathbb{G}, \mathbb{G}_T, g, X_1, X_2, X_3)$. It also sets an empty login state $\lambda = (\{\})$. Second, the TTP chooses ElGamal encryption as the encoder $\mathsf{E} : \mathcal{X} \rightarrow \mathcal{Y}^m$ [36] and an analyzer $\mathsf{A} : \mathcal{Y}^{nm} \rightarrow \mathcal{Z}$. Then, the TTP chooses a random hash function $H : \{0,1\}^* \rightarrow \{0,1\}^l$, a linear error-correcting code C, a homomorphic decode function D, and a number t.

To be able to securely query traffic information for requesting drivers, we propose dividing the traffic region into a tree with nodes and edges. Each node is an intersection and each edge is a road connecting two intersections. We use one region in Washington as an example. In the figure, the traffic area is divided into a tree with the root node being the intersection I_r in the centre. $I_1, \ldots, I_6$ are the six intersections closest to the I_r from different directions which makes them the child nodes of I_r. The edges between I_r and $I_1, \ldots, I_6$ are labeled with $0, 1, \ldots, 5$. As the roads extend, more child nodes are located at more intersections and new edges are labeled with different codes. The coding mechanism is straightforward, i.e., appending $0, 1, \ldots$ to the code of the father edge as the new edge code. For example, the two edges bridging the two child nodes of I_1 are marked with 00 and 01. After all the edges are coded, a set of codes $\mathcal{C}$ are generated. Finally, the TTP publishes public parameters $p.p. = (q, g, \mathbb{G}, \mathbb{G}_T, e, X_1, X_2, X_3, \mathsf{E}, \mathsf{D}, H, \mathsf{C}, \mathsf{D}, t, \mathcal{C})$.

C. Entity Registration

Each contributing driver has a unique identifier CD_i, which can be a license plate, a mobile number, and social insurance number. CD_i registers to the TTP as follows. CD_i randomly selects $r_1, r_2 \leftarrow \mathbb{Z}_q$, computes a ZKPoK$\{(r_1, r_2)|M = g^{r_1} X_3^{r_2}$, and sends the proof to the TTP. The TTP verifies the proof and halts if it fails, otherwise proceed. It randomly selects $r_3 \leftarrow \mathbb{Z}_q^*$, computes a signature

$$S_i = (S_i^1, S_i^2, S_i^3, S_i^4) = (g^{r_3}, (S_i^1)^{x_2}, X_3^{r_3 x_2}, (S_i^1)^{x_1} M^{r_3 x_1 x_2}),$$

and returns S_i to CD_i. CD_i verifies S_i by checking whether

$$S_i^1 \neq 1, \quad e(g, S_i^2) = e(X_2, S_i^1), \quad e(g, S_i^3) = e(X_3, S_i^2),$$
$$e(g, S_i^4) = e(X_1, S_i^1) e(X_1, S_i^2)^{r_1} e(X_1, S_i^3)^{r_2}.$$

CD_i sets a secret key $sk_i = (S_i, r_1, r_2)$ if the verification passes.

A requesting driver RD_j registers to the TTP and receives a similar secret key sk_j. An RSU RU_o registers to the TTP and receives a pair of private key and public key $(priK_o, pubK_o)$, and a region covering a set of m road segments. RU_o also initializes an empty array Arr_o in its local database. The TMSP registers to the TTP and receives a pair of private key and public key $(priK, pubK)$.

D. Traffic Uploading

An RSU RU_o broadcasts an instruction message in its coverage region to inform corresponding drivers of which roads are in this region and their indexes. After receiving the message, a contributing driver CD_i forms a traffic report $TR_i = (0, \ldots, 1, \ldots, 0)$ on m roads where 1 is put at the

index that corresponds to CD_i's road. CD_i encrypts TR_i as ETR_i by using E on the m values in TR_i and random numbers.

Then, CD_i randomly generates a pseudo-name pID_i and forms a set of driving speed DS_i to include her/his current speed which we assume varies a little in a range. The purpose of the set of driving speed is to help complete handshake among contributing drivers by checking whether two drivers have the same speed through PSI. To complete a WiFi challenge handshake with another driver CD_j, CD_i chooses a value a and CD_j chooses a value vector $A_j = (a_1, a_2, \ldots, a_t)$. CD_i and CD_j execute t instances of oblivious transfers [35], CD_i obtains a value vector $B = (b_1, b_2, \ldots, b_t)$ and pID_j. The oblivious transfer is used to build a two-party private set intersection, which provides security against malicious or semi-honest adversaries. CD_i computes t values $H(b_1 \oplus \mathsf{C}(00\ldots0) \wedge a), H(b_2 \oplus \mathsf{C}(00\ldots1) \wedge a), \ldots, H(b_t \oplus \mathsf{C}(11\ldots1) \wedge a)$. CD_j obtains C_j containing t values $C_j = (c_1, c_2, \ldots, c_t)$ where $c_i = b_i \oplus \mathsf{C}(d_i) \wedge a\}$ and d_i is a choice value. The choice value is the index in oblivious transfer and it belongs to $[1, t]$. CD_i sends $H(\mathsf{D}(B, ds_i) \oplus \mathsf{C}(x) \wedge a)$ for each driving speed ds_i in DS_i to CD_j which compares them to $H(\mathsf{D}(C_j, ds_j))$ for each driving speed ds_j to determine whether there is an intersection. If so, the two drivers record the pseudo-name of each other as the proof of their pairing. We note that the attractive factors that lead the drivers to connect to nearby drivers are the benefits from enjoying a traffic service free from attacks and some monetary incentives provided by the RHSP, e.g., coupons.

Next, CD_i generates a blinded signature as follows. CD_i chooses $u_1, u_2 \leftarrow \mathbb{Z}_q^*$ and computes a blinded signature $\hat{S}_i = (\hat{S}_i^1, \hat{S}_i^2, \hat{S}_i^3, \hat{S}_i^4)$ where $\hat{S}_i^1 = (S_t^1)^{u_1}$, $\hat{S}_i^2 = (S_t^2)^{u_1}$, $\hat{S}_i^3 = (S_t^3)^{u_3}$, $\hat{S}_i^4 = (S_t^4)^{u_1 u_2}$.

CD_i generates a login token $Tk_i = g_T^{1/(r_1+H_i)}$ where $H_i = H(ETR_i)$. CD_i computes $v_0 = e(g, \hat{S}_i^4)$, $v_1 = e(X_1, \hat{S}_i^1)$, $v_2 = e(X_1, \hat{S}_i^2)$, $v_3 = e(X_1, \hat{S}_i^3)$. CD_i generates a ZKPoK $\pi_i : \{(r_1, r_2, r_2') | v_0^{r_2'} = v_1 v_2^{r_1} v_3^{r_2} \wedge Tk_i = g_T^{1/(r_1+H_i)}\}$, where $r_2' = 1/r_2$.

Last, CD_i uploads a final report to RU_o:

$$FR_i = (RU_o, ETR_i, \hat{S}_i, Tk_i, \pi_i, \{pID\}). \tag{1}$$

E. Traffic Processing

Upon receiving the FR_i from a contributing driver, the RSU RU_o first checks whether $Tk_i \in \lambda$. If so, RU_o terminates processing FR_i. RU_o verifies the $\hat{S}_i$ by checking whether

$$\hat{S}_i^1 \neq 1, \quad e(g, \hat{S}_i^2) = e(X_2, \hat{S}_i^1), \quad e(g, \hat{S}_i^3) = e(X_3, \hat{S}_i^2).$$

If not, RU_o terminates processing. Otherwise, FR_i computes the same (v_0, v_1, v_2, v_3) and verifies π_i. If the proof fails, RU_o terminates processing. RU_o sends Tk_i to the TMSP which updates $\lambda = \lambda \cup \{Tk_i\}$.

After verifying L final reports from L contributing drivers individually, RU_o constructs a dynamic weighted proximity graph by using $\{pID\}$ and screens drivers after mass propagation in the graph. Intuitively, malicious nodes are connected to each other in a fully connected subgraph and their edge weights are higher than honest nodes'. Therefore, the malicious subgraph will absorb more mass values than the rest of the graph. We assume that there is only one malicious subgraph and show defense results in Section 7.3.

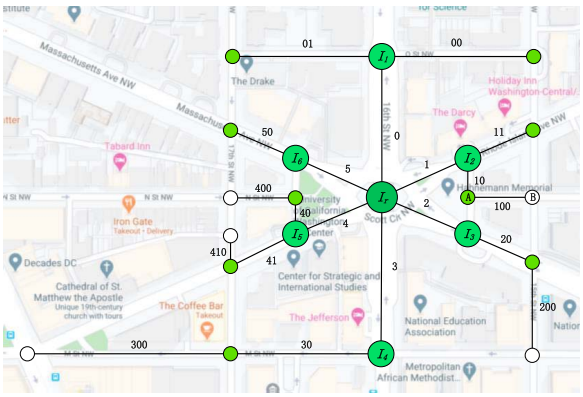


Fig. 4. Representing a traffic region as a tree.

For each index i in Arr_o, RU_o picks the ith item in each ETR, i.e., an encoded and partial traffic report, from L contributing drivers in order to form a vector for $1 \leq i \leq m : \bar{V}_i = (ETR_1^i, ETR_2^i, \ldots, ETR_L^i)$.

RU_o chooses a uniformly random permutation π and obtains an aggregated traffic report on road i for $1 \leq i \leq m : ATR_o^i = \pi(\bar{V}_i)$. The permutation for the encrypted values is used to mix the inputs to be sent to the service provider. It aims to prevent the service provider from linking the contributing drivers to their reports. The RSU can compute the aggregate results locally, but this will bring extra computational cost to the RSU. We focus on detecting the n-by-1 attack on RSUs, therefore outsourcing this local aggregate computation to the service provider.

Last, RU_o sends $\{ATR_o\}$ with a signature to the TMSP. The TSMP computes the traffic condition TC_i^o on road i from $RU_o : TC_o^i = \mathsf{A}(ATR_o^i) = \sum_{j=1}^{L} TR_o^{ij}$, where TR_o^{ij} is the value of traffic report from CD_j on road i from RU_o.

F. Traffic Querying

A requesting driver RD_j wants to know the traffic condition on road i in RU_o's coverage region. RD_j first converts i into a road code rc_j. For example, as shown in Fig. 4, if i connects I_2 and the green node below I_2, then rc_j could be a customized value, such as 1 and 10. Here, 1 stands for three roads: $I_r I_1$, $I_1 A$, and AB. 10 refers to two roads: $I_1 A$, and AB. In this way, we have achieved user-defined location privacy.

RD_j computes a similar blinded signature $\hat{S}_j = (\hat{S}_j^1, \hat{S}_j^2, \hat{S}_j^3, \hat{S}_j^4)$, a login token $Tk_j = g_T^{1/(r_1+H_j)}$ where $H_j = H(o||i)$, and a ZKPoK $\pi_j : \{(r_1', r_2', r_2'') | v_0^{r_2''} = v_1 v_2^{r_1'} v_3^{r_2'} \wedge Tk_i = g_T^{1/(r_1+H_j)}\}$. Then, RD_j submits a traffic query to RU_o:

$$TQ_j = (o, \mathsf{E}(rc_j), \hat{S}_j, Tk_j, \pi_j). \tag{2}$$

RU_o verifies $\hat{S}_j$, Tk_j, and π_j as it did in traffic processing. If they are valid, RU_o sends $(o, \mathsf{E}(rc_j))$ with a signature to the TMSP. The TMSP locates the value in corresponding index,

and returns it to RD_j via RU_o. RU_o comprehends the value as a traffic condition.

VI. Privacy and Security Analysis

In this section, we prove that TraJ achieves all the privacy and security goals, namely, identity privacy, location privacy, unlinkability, one-time and anonymous authentication, integrity, and security against the three attacks.

A. Privacy

Identity privacy. The possibility of leaking the identity of a driver lies in three steps, namely signature generation, token generation, handshaking with nearby drivers. Even if the RSU colludes with the TMSP, they cannot reveal the user identity from the received and shared user information. In signature generation, the user identity is the signature S received from the TTP. The signature is blinded such that RSUs and the TMSP cannot disclose the true signature based on the blinded signature. In token generation, the user identity is the random number r_1 generated in entity registration. A contributing driver computes a hash value of r_1 and the encrypted traffic report $Tk_i = g_T^{1/(r_1+H(ETR_i))}$. The real identity is protected by the onewayness of the hash function H. In handshaking with nearby drivers, the pseudo-name generated during handshaking with nearby drivers is a random number that does not convey anything useful about the real identity. Therefore, their real identities are hidden when users participate in the system. Identity privacy is preserved.

Data privacy. It faces two possibilities where leakage may happen, namely after encryption with E, and after permutation with π. In the former, CD_i encrypts TR_i by using E on TR_i and random numbers. As a result, E is random based on the randomness of CD_i. In the latter, an RSU permutes the L encrypted traffic reports $\{ETR\}$ to obtain an aggregated traffic report ATR. Given that the underlying split-and-mix protocol (E, A) in the anonymized model is σ-secure for computing the sum function $\mathsf{F} : \mathcal{X}^n \to \mathcal{Y}$ for any two inputs $\mathbf{a}, \mathbf{a}' \in \mathcal{X}^n$, the statistical distance SD between $\mathcal{S}_{\mathbf{a}}^{\mathsf{E}}$ and $\mathcal{S}_{\mathbf{a}'}^{\mathsf{E}}$ is no more than $2^{-\sigma}$ where $\mathcal{S}_{\mathbf{a}}^{\mathsf{E}}$ is the result of applying E to $\mathbf{x}$ and permutating the nm-tuple [19]. Therefore, contributing drivers' traffic report must be hidden from other entities. Data privacy is preserved.

Location privacy. For the contributing driver, we did not collect their detailed true locations in traffic uploading, but map each road to an index for RSUs and the TMSP. Contributing drivers' traffic reports are encrypted and permuted by the RSU before sending to the TMSP. For the requesting drivers, we use a coding method in traffic querying to allow them to control the granularity of how their locations are preserved, thus achieving user-defined location privacy. The strength of user-defined privacy preserving is that it will enable users to choose the roads to be queried while preserving their true location.

The location privacy we provide is different from k-anonymity. Our location privacy is based on location generalization and k-anonymity provides indistinguishability from other $k-1$ users. The requesting user in our scheme chooses a sequence of roads that cover the road she/he stands in. The choice of roads gives the requesting users more flexibility and location anonymity. Furthermore, a sequence of roads is a generalized area that covers more users than one road covering k users.

Unlinkability. The unlinkability relies on the an underlying pseudorandom function (PRF) [37] which show that if the Decisional Diffie-Hellman inversion (DDHI) assumption [38] holds in $\mathbb{G}$, then for any polynomially bounded O and any efficient algorithm $\mathcal{A}$, $\Pr[\mathcal{A}(g, g^{1/r}, \ldots, g^{1/(r+O)}) = 1]$-$\Pr[\mathcal{A}(g, E_0, \ldots, E_O) = 1]$ is negligible where $r \leftarrow \mathbb{Z}_q^*$ and $E_0, \ldots, E_O \leftarrow \mathbb{G} \backslash \{1\}$.

Theorem 1: If the DDHI assumption holds in $\mathbb{G}$, the identity privacy is preserved in token generation.

Proof: In order to prove the theorem above, we build an anonymity experiment Exp first. Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) adversary which is given sequential access to oracles. The anonymity experiment is as follows.

(1) We choose a random bit b and set $h = 0$.

(2) $\mathcal{A}$ generates a PK.

(3) $\mathcal{A}$ registers twice with PK. If there exists one registration failure, then Exp returns $b' = 0$. Otherwise, $\mathcal{A}$ obtains sk_0 and sk_1.

(4) $\mathcal{A}$ executes any of the operations below: change h; query $\mathsf{Login}(\cdot)$, i.e., login with sk_c, PK, and h.

(5) $\mathcal{A}$ executes any of the operations below: change h; query oracle executes any of the operations below: $\mathsf{ChallengeLogin}(\cdot)$ which replies as $\mathsf{Login}(c \oplus b)$ does.

(6) $\mathcal{A}$ performs as in (4).

(7) $\mathcal{A}$ outputs a bit b' and it succeeds if $b' = b$.

We design three new experiments based on Exp. In Exp_1, we replace all ZKPoK with simulated proofs which only affects the success probability of $\mathcal{A}$ negligibly. In Exp_2, we replace M in entity registration with a uniform group element. Since r_1 and r_2 are random, the new M does not affect the success probability of $\mathcal{A}$. In Exp_3, let S_i be the valid signature of CD_i in entity registration. In subsequent logins if CD_i or CD_j, we choose a random $r_4 \leftarrow \mathbb{Z}_q^*$ and compute $S' = ((S_i^1)^{r_4}, (S_i^2)^{r_4}, (S_i^3)^{r_4}, (S_i^4)' \leftarrow \mathbb{G})$. It is obvious that S' is distributed identically to the values in Exp_2 despite the login is from CD_i or CD_j, indicating that the success probability of $\mathcal{A}$ is not affected. Let r_1^i and r_1^j be the two respective random values of r_1 for CD_i and CD_j. Let U be the upper bound on the login times $\mathcal{A}$ can have. In Exp_4, we select uniform group elements $E_i(0), \ldots, E_i(U)$, $E_j(0), \ldots, E_j(U)$ and replace $Tk_i^{r_1^b}$ with $E_b(U)$ where $b = i$ or $b = j$. This only affects the success probability of $\mathcal{A}$ negligibly. In Exp_4, logins of CD_i are distributed identically to logins of CD_j. Conclusively, the success probability of $\mathcal{A}$ in Exp_4 is $1/2$, completing the proof.□

Thus, given two encrypted traffic reports/queries, any adversaries cannot decide whether they are produced from the same contributing/requesting driver. Unlinkability is guaranteed.

B. Security

Theorem 2: If the Lysyanskaya, Rivest, Sahai, and Wolf (LRSW) assumption [39], [40] holds, TraJ provides secure authentication.

Proof: Let Π' be the signature scheme in LRSW assumption, and Π be the proposed TraJ. Let $\mathcal{A}$ be a PPT adversary attacking Π. A makes $\hat{q}$ queries $(r_1^1, r_2^1), \ldots, (r_1^{\hat{q}}, r_2^{\hat{q}})$ and

TABLE II
COMPARISON OF PRIVACY AND SECURITY PROPERTIES

Property	CLASC	PAM	RCoM	TraJ
Identity Privacy	√	√	×	√
Data Privacy	×	√	√	√
Location Privacy	√	×	√	√
Unlinkability	√	√	√	√
Authentication	√	√	√	√
Data Integrity	√	√	√	√
Security against Attack 1[1]	×	√	×	√
Security against Attack 2[2]	×	×	×	√
1: multiple requesting/querying attack 2: n-by-1 jamming attack				

receives $\hat{q}$ signatures $(S_1^1, S_1^2, S_1^3, S_1^4), \ldots, (S_{\hat{q}}^1, S_{\hat{q}}^2, S_{\hat{q}}^3, S_{\hat{q}}^4)$ with $M_i = g^{r_1^i} X_3^{r_2^i}$. Let $(\bar{r}_1, \bar{r}_2, \bar{S})$ be $\mathcal{A}$'s forgery and $\bar{M} = g^{\bar{r}_1} X_3^{\bar{r}_2}$. Let For be the event that $\mathcal{A}$ makes a valid forgery and Eve be the event that $\bar{M} \notin \{M_1, \ldots, M_{\hat{q}}\}$.

Now we assume that $\mathcal{A}$ has non-negligible advantage in attacking Π and show how to construct an adversary $\mathcal{A}'$ attacking Π':

1. When $\mathcal{A}$ submits a signature query (r_1^i, r_2^i), $\mathcal{A}'$ chooses $x_3 \leftarrow \mathbb{Z}_q$.

2. $\mathcal{A}'$ queries $r_1^i x_3 + r_2^i$ to the LRSW oracle and receives a response (A_1, B_1, C_1).

3. $\mathcal{A}'$ returns $(A_1, A_2, A_2^{x_3}, A_3)$ to $\mathcal{A}$ as a signature.

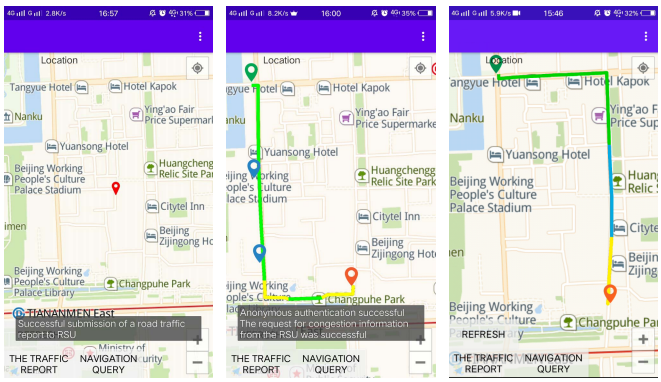
4. When $\mathcal{A}$ outputs a forgery $(\bar{r}_1, \bar{r}_2, \bar{S} = (\bar{S}_1, \bar{S}_2, \bar{S}_3, \bar{S}_4))$, $\mathcal{A}'$ outputs $(\bar{r}_1 x_3 + \bar{r}_2, \bar{S}_1, \bar{S}_2, \bar{S}_4)$.

We assume that $A_1 = g^a$ for an nonzero a, then $A_2 = (A_1)^{x_2} = g^{ax_2}$, $A_2^{x_3} = g^{ax_2x_3}$ and $A_3 = g^{a(x_1+(r_1^i+x_3r_2^i)x_1x_2)} = g^{ax_1}(g^{r_1^i+x_3r_2^i})^{ax_1x_2} = g^{ax_1}M^{ax_1x_2}$. If For happens, then Eve happens. This is because when $(\bar{r}_1, \bar{r}_2, \bar{S} = (\bar{S}_1, \bar{S}_2, \bar{S}_3, \bar{S}_4))$ is a valid forgery, $\bar{S}_2 = \bar{S}_1^{x_2}$ and $\bar{S}_3 = \bar{S}_1^{x_1+(\bar{r}_1+\bar{r}_2x_3)x_1x_2}$, completing the proof.□

The login token in traffic uploading/querying has guaranteed that only one traffic report/query is allowed under one anonymous credential at one time. Thus, TraJ resists to the multiple requesting/querying attack.

To detect the n-by-1 jamming attack in traffic processing, we analyze the specific character of the attack, i.e., malicious devices act as normal users to handshake with nearby devices and submit traffic reports. As a result, the edge weight between malicious devices, the edge weight between malicious device and normal device look no different than the edge weight between normal devices. Next, we combine WiFi challenge handshaking, proximity graph construction, and mass propagation to transfer mass values between nodes. Under the attack, the total mass value will transfer from the bootstrap nodes to the rest of the graph. Since the malicious devices are tightly connected, the mass value of the malicious nodes will show a special pattern that we present in Section 7. Thus, TraJ can detect the anomaly and resist the attack. The RSUs can cut off nodes with lower values when the mass propagation completes. Thus, TraJ resists to the new attack by making the output of normal traffic uploading indistinguishable from the one under attack, i.e., $|\Pr[F(\mathcal{CD}) \rightarrow \mathcal{TC}] - \Pr[F'(\mathcal{A}) \rightarrow \mathcal{TC}]| \leq \text{negl}(n)$.

We compare TraJ with existing works, namely CLASC [4], PAM [5], and RCoM [6] regarding privacy and security in Table II. CLASC has used plaintext to collect and process road surface message. PAM achieved most privacy and security goals but the location privacy for using a range to stand for a speed, and it could not defend against the n-by-1 jamming attack. RCoM has asked a vehicle to send an identifier V and secret key vsk to an RSU and a cloud server which violates identity privacy and unlinkability. Although it allows the root authority to see the road condition report in plaintext, the root authority is assumed to be trusted, which does not leak data privacy or location privacy.



(a) Report Uploading (b) Querying Case 1 (c) Querying Case 2

Fig. 5. Implementation of TraJ.

VII. PERFORMANCE ANALYSIS

A. Experiment Settings

We instantiated the TraJ using a laptop and eleven android smartphones. The server uses the Springboot framework and the driver-side is programmed as an Android application. We used the JPBC library to implement the encryption primitives, and the elliptic curve being defined as $y^2 = x^3 + x$ over F_p [41]. We created the driver-side application project in Android Studio 3.6 and used Gradle to introduce the JPBC library and bcprov-jdk.jar from Bouncy Castle [43]. Meanwhile, we created server-side project at IDEA and used Maven to invoke the downloaded JPBC library. The database on the server-side is MySQL 8.0 connected using the JDBCtemplate. Http is used for communication between smartphones and the server-side, WiFi-Direct [42] is used for handshakes between smartphones, and the System.nanoTime() function is used to record the return time of operation. The screenshots of the application are shown in Fig. 5: a contributing driver successfully submits four locations with their real-time traffic; a requesting driver queries the traffic of four locations and mark the corresponding roads after receiving query results. Although the requesting driver only selects the traffic of two locations, he has selected a code that covers six intersections, thus receiving the traffic of six five roads. We used the real dataset of traffic scenario: The Bologna Ringway dataset [44]. It models the traffic in Bologna, Italy, on a typical day between $08:00$ and $09:00$ with more than 22000 vehicles. There are multiple data items in the sumo-processed version, including vehicle id, location, and timestamp. Experimental parameters and hardware settings are given in Table III and Table IV.

B. Computational Costs and Communication Overhead

We now analyze computational costs, and communication overhead for CD, RD, RSU, TMSP, and the TTP in four main phases. The results are recorded in Table V.

TABLE III
KEY EXPERIMENTAL PARAMETERS

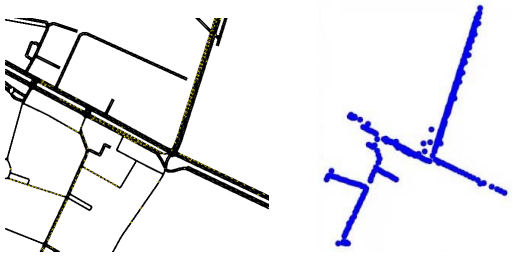
Parameter	Value
$n, m, w, t, \|q\|, \|p\|$	$[2, 30]$, 20, 4, 5, 1024, 512
H, l, E, L	SHA256, 256, ElGamal, 1000
N, n	$252, \{0, 50, 75, 100, 125, 150\}$

TABLE IV
HARDWARE SETTINGS

Smartphone Xiaomi 11	
CPU	snapdragon 888
Memory	12 GB
Operating System	Android 11
Smartphone Vivo Xplay6	
CPU	snapdragon 820
Memory	6 GB
Operating System	Android 7
Laptop (Hasee)	
CPU	Intel Core i7-7700HQ @2.6GHZ
Memory	8 GB
Operating System	Microsoft Windows 10 64-bit

TABLE V
COMPUTATIONAL COSTS (S) AND COMMUNICATION OVERHEAD (KB)

Entity	CD	RD	RSU	TMSP	TTP
Computational cost	0.27	0.27	0.2	0.026	0.1
Communication overhead	2.376	2.376	0.21	0.125	0.5



(a) Original Roads and Vehicles (Yellow Triangles) (b) Selected Vehicles (Blue Dots) After Data Processing

Fig. 6. Dataset processing.

In entity registration, both CD and RD interact with the TTP by computing and sending a ZKPoK to the TTP. The TTP verifies the proof, computes, and returns a signature S to the driver. The driver verifies S via bilinear pairings. It costs CD and TTP approximately 0.11 ms and 0.1 ms, respectively. The communication overhead of a driver is $|\text{ZKPoK}| = 1024 * 4 = 4096 = 0.5$ (KB). The communication overhead of the TTP is $|S| = |S^1| + |S^2| + |S^3| + |S^4| = 1024 * 4 = 0.5$ (KB). In traffic uploading, each RSU broadcasts an instruction message. Each CD encrypts a traffic report TR_i, handshakes with a nearby driver, computes a blinded signature $\hat{S}$, generates a login token Tk and a ZKPoK, and uploads a final report FR to the RSU. In traffic processing, the RSU verifies the blinded signature, login token, and ZKPoK, permutes m vectors $\bar{V}\}_{i=1}^{m}$, and sends a set of m aggregated traffic reports $\{ATR_o^i\}_{i=1}^{m}$ with a signature to the TMSP. The TSMP computes m traffic condition *TC* from one RSU. In traffic querying, RD computes a similar blinded signature, a login token, and a ZKPoK, and sends a traffic query to the RSU. The RSU verifies the query and sends two identifiers $(o, \mathsf{E}(rc))$ with a signature to the TMSP. The TMSP retrieves the value in the corresponding index, i.e., storing and searching in a hashmap, and returns it to RD via the RSU. The user-defined query approach does not weaken the traffic monitoring performance. This is because we have computed the traffic condition for each road for the RSU and the TMSP to generate the traffic condition result for each encoded road. No matter which road the requesting driver queries, the query processing will only be once, i.e., the query cost is constant. For example, as shown in Fig. 4, say the traffic conditions for two roads I_2A and AB are uncrowded (00) and slightly congested (10), respectively. The query on the encoded road 10, which is I_2A and AB, will be $\{10 - 00, 10\}$.

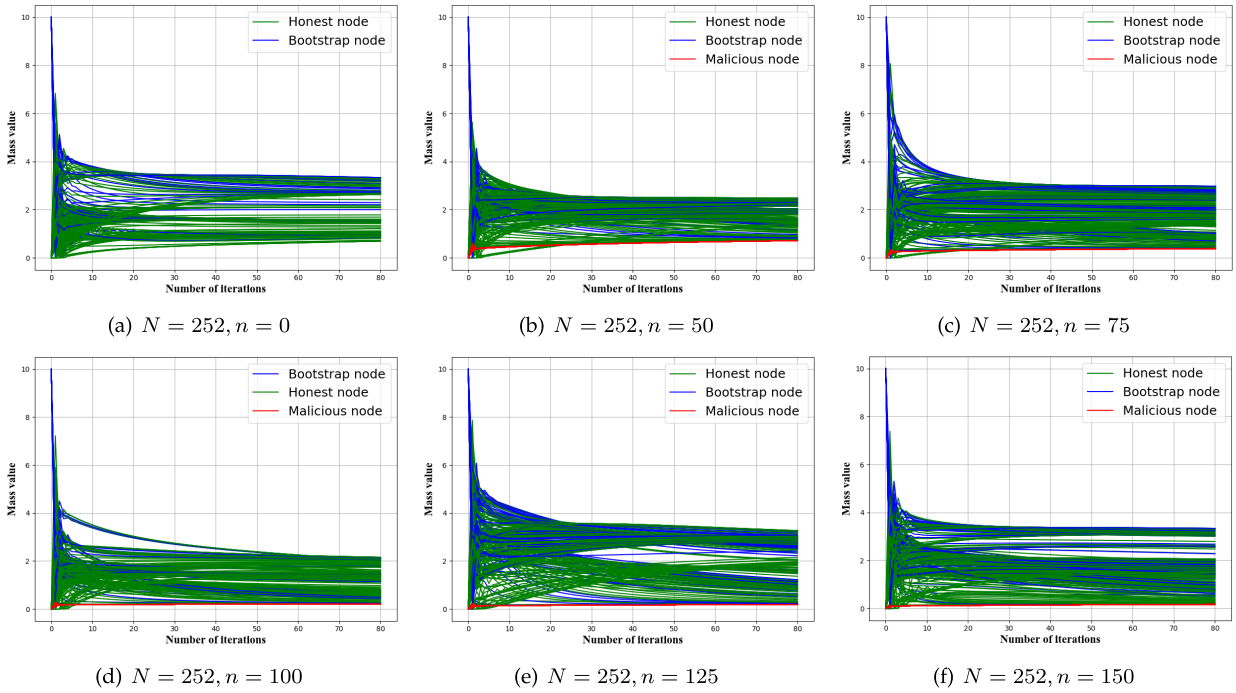
C. Defense Efficacy

We construct a weighted proximity graph of N honest nodes and num malicious nodes at an RSU. We randomly choose 20% of honest nodes (e.g., bus and taxi) as bootstrap nodes and set their mass value to 10. The mass propagation proceeded by distributing nodes' mass value to their neighbor according to their weight, i.e., $tv_i^{t+1} = \sum_j (tr_j^t * w_{ij} / \sum w_j)$, where tv_i^t is node j's mass value at time t, w_{ij} is the weight of edge e_ij, and $\sum w_j$ is the sum of all weights of edges connecting node j. When the RSU is under attack, i.e., the node degree of malicious nodes is $n - 1$.

After processing the dataset, we can select an area covering $N = 252$ honest vehicles that are close enough to bridge nearby vehicles to form a connected graph, as shown in Fig. 6. We consider one honest scenario with $n = 0$ and five attack scenarios where $n = 50, 75, 100, 125, 150$, respectively. The malicious nodes are placed in the center among the vehicles. In the honest scenario, i.e., Fig. 7(a), the mass value of each node tends to be stable and within a close range. This is because random handshakes among honest drivers will lead to an almost uniform mass distribution. In the remaining five scenarios, the edge weights between honest nodes, the edge weights between malicious nodes, and the attack edge weight between a malicious node and an honest node are drawn from $[2, 5]$ because the malicious nodes act as honest nodes. During the mass propagation, the malicious nodes will absorb and propagate mass values as the honest nodes do. When the propagation is complete, the overall mass values are stable. We have three observations: (1) the red line is basically horizontal, i.e., the mass values of malicious nodes are almost the same, because they are a tightly connected subgraph to divide mass values; (2) the mass values of malicious nodes are lower than honest nodes because n is large, while the overall mass value for the malicious zone is limited; (3) when n increases, i.e., the ratio of malicious nodes to honest nodes becomes larger but the handshakes stay the same, the mass values of malicious nodes decrease. Finally, we can screen the malicious nodes by observing these phenomena.

D. Comparison With Existing Work

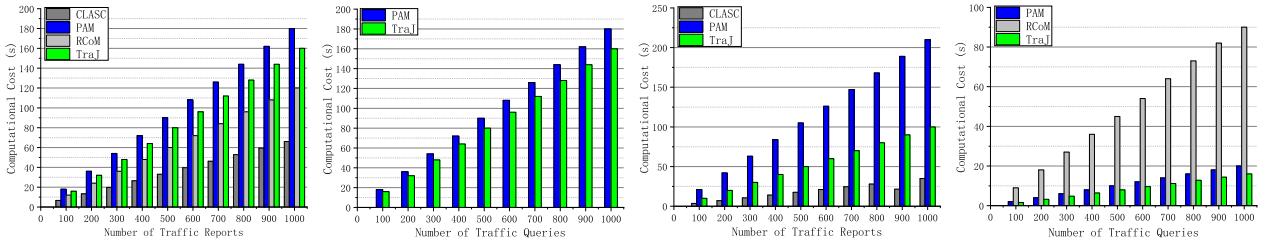
CLASC utilized a certificateless aggregate signcryption scheme to collect traffic reports and protect driver privacy,



(a) $N = 252, n = 0$ (b) $N = 252, n = 50$ (c) $N = 252, n = 75$

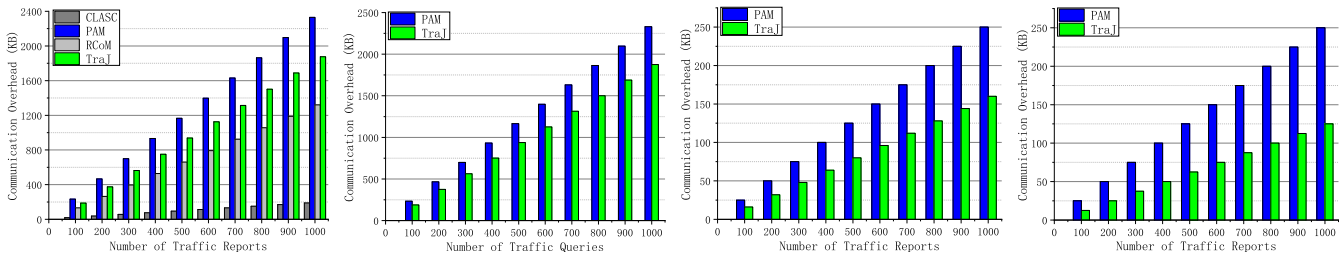
(d) $N = 252, n = 100$ (e) $N = 252, n = 125$ (f) $N = 252, n = 150$

Fig. 7. Mass propagation in a weighted proximity graph under n-by-1 jamming attack and a real-world dataset.



(a) CD (b) RD (c) RSU (d) TMSP

Fig. 8. Comparison of computational costs.



(a) CD (b) RD (c) RSU (d) TMSP

Fig. 9. Comparison of communication overhead.

which leads to a relatively low computation cost. However, it sacrifices data privacy for using traffic reports in plaintext, and it does not include traffic querying. PAM leveraged the BBS group signatures and searchable encryption to collect traffic reports and respond to traffic queries. It incurs too many computational costs and communication overhead. RCoM realized traffic collecting by authentication, token generation, and equality test on ciphertexts. We use four equivalence classes in the experiments which is the same as our setting $w = 4$. Specifically, in Fig. 8(c), we record the consumed time for an RSU in processing traffic reports. In Fig. 8(d), it excludes RCoM since it did not include the traffic querying phase. As shown in Fig. 8 and Fig. 9, TraJ has a moderate computational cost and communication overhead amid all schemes. TraJ's computational cost and communication overhead are acceptable for two reasons. First, current on-board units have already possessed enough computation and communication capabilities for vehicular services. For example, the CP400.85

from Hyperion Technologies contains an ARMv7-A system with a processing power of 500 MHz, and it can store 7.5 GB of data in reliable storage [45]. TS3290/00A from Kapsch TrafficCom supports DSRC communication [46]. Second, TraJ performs the best or moderate in the comparison experiments.

VIII. Discussions

A. Vehicle Mobility

The mobility of the vehicles affects privacy performance from two aspects. First, the speed of vehicle will impact the communication between the vehicle and an RSU. The higher the speed, the fewer interactions will be, given a fixed reporting/requesting time epoch. In this case, less sensitive information is leaked. It also impacts the handshakes between vehicles. If one honest vehicle moves faster than the vehicles nearby, the handshakes connecting this vehicle with others will decrease. In this case, it will help construct a more dynamic proximity, leading to an ideal mix-zone for users. Second, if the number of honest vehicles increases, it will help build a more compact proximity graph, leading to a high detection probability and privacy-preserving zone. But still, we cannot look at this issue from a single view of point. We will evaluate the effects of vehicle mobility under real vehicular environments in future work.

B. Differential Privacy

We did release the raw aggregated traffic report to the service provider, but we only ask the drivers to use a cloaked location, i.e., a road, to replace her/his current location. In this way, we protect their specific locations. Considered that users always submit similar locations to the service provider, e.g., when they drive from home to work, releasing locations during this part is analogous to trajectory publishing in location-based services. Therefore, we could utilize ϵ-spatiotemporal event privacy [12] as an add-on to protect users' location privacy while considering their spatiotemporal events.

C. Possible Attacks

One potential attack is that an adversary opens the service app and just puts the cellphones on a road for a long time trying to fake the traffic jam. This will create a temporary jam but still raises an alarm and even draws attention from the traffic department who checks whether there is an accident report or checks at the scene.

Another attack is that attackers report the lowest congested traffic condition to create a traffic jam. In this attack, a group of colluding drivers upload the lowest congested traffic condition from a road, e.g., reporting no traffic for a congested road, to the service provider. The service provider may be misled by the drivers to believe that this road is clear and notify other honest drivers of the false congestion. When the drivers come to this road, the road condition will become even more congested. We discuss several possible methods to defend against this attack. Similar to TraJ, we can ask reporting drivers to handshake with nearby drivers to authenticate each others location and speed. If they are not close enough or do not share a similar speed, they cannot be bridged. Next, the RSU constructs a weighted proximity graph and propagate trust values in the graph. If the proportion of the malicious drivers is not high, the RSU screens them with a high probability. This has been verified in our previous work [5] where we considered an opposing situation, where a group of colluding drivers upload the highest congested traffic condition. Furthermore, we can build a reputation mechanism for the traffic monitoring system. If some drivers misbehave, we will reduce their reputation value and add them to a blacklist if necessary. In doing so, we have to guarantee that updating a drivers reputation value does not lead to identity linkability.

D. Utilization of Sensors and Cameras

The underlying problem that is addressed is not just simple car counting. It may be solved by installing sensors or cameras to count the vehicles in a road segment. Such an approach has three disadvantages that make them not suitable for this scenario. (1) Cost. It requires a large number of devices for installation that raises a high cost. For example, a MB8450 car detection sensor costs 98.95 dollars [47], and the price of Lnd laser radar traffic sensor transport management for vehicle detection is up to 800 dollars [48]. (2) Efficiency. It is difficult to determine how many devices are needed or where to install them. The utilization of multiple devices requires time-consuming and careful experiments to test the efficiency of this approach, considering different device number and different distribution. The "intelligence" of the traffic camera is also questionable as a woman was mistaken for car [49]. (3) Privacy. The installation of extra cameras on road will infringe on privacy issues [50], [51]. This is because the collected data from cameras taking a picture or recording a video of the vehicles driving by will be transmitted to a server and such a server may leak the data (e.g., vehicle brand and location, license plate, and even a picture of the driver) due to inappropriate protection mechanism, a malicious employee, or being hacked. As reported, at least 363 cameras originally designed to monitor traffic flow are switched to spy upon pedestrians [52].

IX. Conclusion

In this paper, we have proposed a novel privacy-preserving traffic monitoring scheme TraJ for edge-computing assisted vehicular networks. TraJ preservers identity privacy and location privacy based on anonymous authentication and private aggregation. Besides the multiple uploading attack, TraJ also defends against the newly emerged n-by-1 jamming attack. This is achieved by constructing a dynamic weighted proximity graph from the private set intersection and profiling the special character of the attack. With TraJ, drivers can securely participate in the traffic monitoring system without privacy concerns.

References

[1] H. Cui, R. H. Deng, and G. Wang, "An attribute-based framework for secure communications in vehicular ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 721–733, Apr. 2019.

[2] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 703–715, Jul. 2020, doi: 10.1109/TDSC.2018.2850780.

[3] B. Hoh *et al.*, "Virtual trip lines for distributed privacy-preserving traffic monitoring," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, Breckenridge, CO, USA, 2008, pp. 15–28.

[4] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[5] M. Li, L. Zhu, and X. Lin, "Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing," *IEEE Trans. Services Comput.*, vol. 14, no. 6, pp. 1902–1913, Nov. 2021, doi: 10.1109/TSC.2019.2903060.

[6] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1779–1790, Jul. 2019.

[7] *Google Maps*. [Online]. Available: https://www.google.com/maps

[8] *Waze*. [Online]. Available: https://www.pentagram.com/work/waze

[9] G. A. Fowler. (2019). *What Does Your Car Know About You? We Hacked a Chevy To Find Out*. [Online]. Available: https://www.washingtonpost.com/technology/2019/12/17/what-does-your-car-know-about-you-we-hacked-chevy-find-out

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st, Ed., MCC Workshop Mobile Cloud Comput. (MCC)*, Helsinki, Finland, 2012, pp. 13–16.

[11] C. Luo *et al.*, "Predictable privacy-preserving mobile crowd sensing: A tale of two roles," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 361–374, Feb. 2019.

[12] Y. Cao, Y. Xiao, L. Xiong, L. Bai, and M. Yoshikawa, "Protecting spatiotemporal event privacy in continuous location-based services," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3141–3154, Aug. 2021.

[13] (2020). *An Artist Used 99 Phones to Fake a Google Maps Traffic Jam*. [Online]. Available: https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam

[14] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An online social network with user-defined privacy," in *Proc. ACM Conf. Appl., Technol., Architectures, Protocols Comput. Commun. (SIGCOMM)*, Barcelona, Spain, Aug. 2009, pp. 135–146.

[15] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "User-defined privacy grid system for continuous location-based services," *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2158–2172, Oct. 2015.

[16] G. Sun, Y. Xie, D. Liao, H. Yu, and V. Chang, "User-defined privacy location-sharing system in mobile online social networks," *J. Netw. Comput. Appl.*, vol. 86, pp. 34–45, May 2007.

[17] E. Boyle, Y. Ishai, R. Pass, and M. Wootters, "Can we access a database both locally and privately?" in *Proc. 15th Theory Cryptogr. Conf. (TCC)*, Baltimore, MD, USA, Nov. 2017, pp. 662–693.

[18] S. Patel, G. Persiano, and K. Yeo, "Private stateful information retrieval," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 1002–1019.

[19] H. Corrigan-Gibbs and D. Kogan, "Private information retrieval with sublinear online time," in *Proc. 39th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (Eurocrypt)*, May 2020, pp. 44–75.

[20] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "PSI from PaXoS: Fast, malicious private set intersection," in *Proc. 39th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (Eurocrypt)*, May 2020, pp. 739–767.

[21] B. Wang, "Defending against sybil devices in crowdsourced mapping services," in *Proc. MobiSys PhD Forum*, Jun. 2016, pp. 179–191.

[22] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. 24th Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 2004, pp. 41–55.

[23] *Phantom Traffic Jams Explained*, p. 1. [Online]. Available: https://motorlease.com/article/traffic-jams-explained

[24] *M6 Traffic Live: Six Miles of Queues and 60 Minute Delays Northbound in Lancashire After Serious Crash*, p. 1. [Online]. Available: https://www.lancs.live/news/lancashire-news/live-m6-crash-closed-lancashire-22187679

[25] P. Zhao *et al.*, "Synthesizing privacy preserving traces: Enhancing plausibility with social networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2391–2404, Dec. 2019.

[26] Y. Luo, X. Jia, S. Fu, and M. Xu, "PRide: Privacy-preserving ride matching over road networks for online ride-hailing service," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1791–1802, Jul. 2019.

[27] M. Nabil, A. Sherif, M. Mahmoud, A. Alsharif, and M. Abdallah, "Efficient and privacy-preserving ridesharing organization for transferable and non-transferable services," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1291–1306, May 2021.

[28] H. Yu, J. Shu, X. Jia, H. Zhang, and X. Yu, "LpRide: Lightweight and privacy-preserving ride matching over road networks in online ride hailing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10418–10428, Nov. 2019.

[29] M. Li, Y. Chen, C. Lal, M. Conti, M. Alazab, and D. Hu, "Eunomia: Anonymous and secure vehicular digital forensics based on blockchain," *IEEE Trans. Dependable Secure Comput.*, early access, Nov. 25, 2021, doi: 10.1109/TDSC.2021.3130583.

[30] M. Li, F. Wu, G. Chen, L. Zhu, and Z. Zhang, "How to protect query and report privacy without sacrificing service quality in participatory sensing," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–7.

[31] G. Maganis, E. Shi, H. Chen, and D. Song, "Opaak: Using mobile phones to limit anonymous identities online," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, Low Wood Bay, U.K., Jun. 2012, pp. 295–308.

[32] M. Z. Lee, A. M. Dunn, B. Waters, E. Witchel, and J. Katz, "Anon-pass: Practical anonymous subscriptions," in *Proc. 34th IEEE Symp. Secur. Privacy*, May 2013, pp. 319–333.

[33] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *J. Cryptol.*, vol. 1, no. 2, pp. 77–94, 1988.

[34] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *Proc. 11th Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA, Aug. 1991, pp. 433–444.

[35] M. Orrù, E. Orsini, and P. Scholl, "Actively secure 1-out-of-N OT extension with application to private set intersection," in *Proc. Cryptographers' Track RSA Conf. (CT-RSA)*, Feb. 2017, pp. 381–396.

[36] A. Bittau *et al.*, "Prochlo: Strong privacy for analytics in the crowd," in *Proc. 26th Symp. Operating Syst. Princ. (SOPS)*, Oct. 2017, pp. 441–459.

[37] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Proc. 8th Int. Workshop Public Key Cryptogr. (PKC)*, Les Diablerets, Switzerland, Jan. 2005, pp. 416–431.

[38] S. Mitsunari, R. Sakai, and M. Kasahara, "A new traitor tracing," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 85, no. 2, pp. 481–484, 2002.

[39] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Proc. 6th Int. Workshop Sel. Areas Cryptogr. (SAC)*, Kingston, AB, Canada, Aug. 1999, pp. 184–199.

[40] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proc. 24th Annu. Int. Cryptol. Conf. (CRYPTO)*, Aug. 2004, pp. 56–72.

[41] D. M. Kar and I. Ray, "Systematization of knowledge and implementation: Short identity-based signatures," 2019, *arXiv:1908.05366*.

[42] *Wi-Fi Direct*, p. 1. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-direct

[43] *The Legion of the Bouncy Castle*, p. 1. [Online]. Available: https://www.bouncycastle.org/java.html

[44] *The Bologna Ringway Dataset*, p. 1. [Online]. Available: http://academic.lucabedogni.it/the-bologna-ringway-dataset

[45] *CP400.85*, p. 1. [Online]. Available: https://satsearch.co/products/hyperion-technologies-cp400-85.

[46] *TS3290/00A. On-Board Unit*, p. 1. [Online]. Available: https://www.kapsch.net/ktc/downloads/datasheets/in-vehicle/5-8/Kapsch-KTC-DS-OBU-TS3290_00A.pdf

[47] *MB8450 Car Detection Sensor*, p. 1. [Online]. Available: https://www.maxbotix.com/ultrasonic_sensors/car-detection-sensor.htm

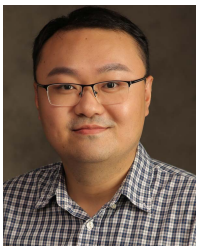
[48] *LND Laser Radar Traffic Sensor Transport Management for Vehicle Detection*, p. 1. [Online]. Available: https://g-teksensor.en.made-in-china.com/product/iZLTVecPhRWr/China-Lnd-Laser-Radar-Traffic-Sensor-Transport-Management-for-Vehicle-Detection.html

[49] M. Zhang. *Traffic Camera Mistakes Woman for Car, Issues Ticket to Car Owner*, p. 1. [Online]. Available: https://petapixel.com/2021/10/19/traffic-camera-mistakes-woman-for-car-issues-ticket-to-car-owner

[50] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1168–1174, Aug. 2008.

[51] S. Tang *et al.*, "TelosCAM: Identifying burglar through networked sensor-camera mates with privacy protection," in *Proc. IEEE 32nd Real-Time Syst. Symp.*, Vienna, Austria, Nov. 2011, pp. 327–336.

[52] H. Kelly and G. Keogh. *Big Brother is Watching Your Social Distancing*, p. 1. [Online]. Available: https://www.dailymail.co.uk/news/article-8874125/Traffic-flow-cameras-secretly-switched-monitor-millions-pedestrians.html



Meng Li (Member, IEEE) received the B.E. degree in information security from the Hefei University of Technology in 2010, the M.S. degree in computer science and technology from the Department of Computer Science and Technology, Beijing Institute of Technology, in 2013, and the Ph.D. degree in computer science and technology from the Beijing Institute of Technology in 2019. He is currently an Associate Researcher and the Dean Assistant with the School of Computer Science and Information Engineering, Hefei University of Technology. He is also a Post-Doctoral Fellow with the Department of Mathematics and HIT Center, University of Padua, Italy, where he is with the Security and PRIvacy Through Zeal (SPRITZ) research group led by Prof. M. Conti. He was sponsored by the ERCIM "Alain Bensoussan" Fellowship Programme in October 2019, to conduct post-doctoral research at CNR, Italy. He was sponsored by the China Scholarship Council (CSC) to study in the Broadband Communications Research (BBCR) Lab at the University of Waterloo and Wilfrid Laurier University from September 2017 to August 2018. In this area, he has published more than 40 papers in international peer-reviewed transactions, journals, magazines, and conferences, including IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE INTERNET OF THINGS JOURNAL, *Information Sciences*, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATION, MobiCom, ICICS, SecureComm, TrustCom, and IPCCC. His research interests include security, privacy, vehicular networks, applied cryptography, and blockchain.

Liehuang Zhu (Senior Member, IEEE) is currently a Full Professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology. He is selected into the Program for New Century Excellent Talents in University from the Ministry of Education, China. His research interests include the Internet of Things, cloud computing security, internet, and mobile security. He has published over 100 SCI-indexed research papers in these areas, as well as a book published by Springer. He serves on the editorial boards of three international journals, including IEEE INTERNET OF THINGS JOURNAL, *IEEE Network*, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He won the Best Paper Award at IEEE/ACM IWQoS 2017 and IEEE TrustCom 2018.

Zijian Zhang (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Technology, Beijing Institute of Technology. He is currently a Research Fellow with the School of Computer Science, The University of Auckland. He was a Visiting Scholar with the Department of Computer Science and Engineering, State University of New York at Buffalo, in 2015. His research interests include the design of authentication and key agreement protocol and the analysis of entity behavior and preference.

Chhagan Lal (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Malaviya National Institute of Technology, India, in 2014. During his Ph.D., he has been awarded with the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in the University of Saskatchewan, Saskatoon, SK, Canada. He is currently a Senior Researcher in cybersecurity with the Department of Intelligent Systems, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology (TU Delft), The Netherlands. Earlier, he was a Research Scientist at Simula Research Labs, Oslo, Norway. Before joining Simula, he held a post-doctoral position with the Department of Mathematics, University of Padova (UNIPD), Italy. In UNIPD, he is an active member of the Security and PRIvacy Through Zeal (SPRITZ) research group, which is led by Prof. Mauro Conti. His current research areas include the applications of blockchain technologies, security in software-defined networking, and the Internet of Things networks.

Mauro Conti (Fellow, IEEE) received the Ph.D. degree from the Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Post-Doctoral Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011, he joined the University of Padua, as an Assistant Professor, where he became an Associate Professor in 2015 and a Full Professor in 2018. He has been a Visiting Researcher at GMU, UCLA, UCI, TU Darmstadt, UF, and FIU. He is currently a Full Professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of security and privacy. In this area, he published more than 400 papers in topmost international peer-reviewed journals and conferences. He is a Senior Member of the ACM and a Fellow of the Young Academy of Europe. He was the Program Chair of TRUST 2015, ICISS 2016, WiSec 2017, ACNS 2020, and CANS 2021; and the General Chair of SecureComm 2012, SACMAT 2013, NSS 2021, and ACNS 2022. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the Area Editor-in-Chief of IEEE COMMUNICATIONS SURVEYS & TUTORIALS, and has been an Associate Editor for several journals, including IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT.

Mamoun Alazab (Senior Member, IEEE) received the Ph.D. degree in computer science from the School of Science, Information Technology and Engineering, Federation University of Australia. He is currently an Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is a cybersecurity researcher and practitioner with industry and academic experience. His research is multidisciplinary that focuses on cybersecurity and digital forensics of computer systems with a focus on cybercrime detection and prevention, including cyber terrorism and cyber warfare. He has more than 150 research articles. He delivered many invited and keynote speeches, 24 events in 2019 alone. He convened and chaired more than 50 conferences and workshops. He works closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), the Australian Communications and Media Authority (ACMA), Westpac, United Nations Office on Drugs and Crime (UNODC), and the Attorney Generals Department. He is the Founding Chair of the IEEE Northern Territory (NT) Subsection.