Ad Hoc Networks 000 (2019) 101876

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Privacy-aware query processing in vehicular ad-hoc networks☆

Yongxuan Lai[a,c,*], Yifan Xu[a,c], Fan Yang[d,c], Wei Lu[e], Quan Yu[b,*]

[a] Software School of Xiamen University, Xiamen 361005, China
[b] Key Laboratory of Complex Systems and Intelligent Computing, School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, China
[c] Shenzhen Research Institute of Xiamen University, Shenzhen 518057, China
[d] Department of Automation, Xiamen University, Xiamen 361005, China
[e] Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, Renmin University of China, Beijing 100872, China

ARTICLE INFO

Article history:
Received 11 December 2018
Revised 26 April 2019
Accepted 29 April 2019
Available online xxx

Keywords:
Privacy preserving
Query result forwarding
Query mapping
Query processing
VANETs

ABSTRACT

Recently there is a research trend to integrate the cloud and the vehicular ad-hoc networks to enhance each other on the cooperative urban sensing applications. However, it remains a challenging problem on how to forward queries to search the results and then to route back the query results, as the vehicular nodes dynamically move on the road. Also, the dark side of vehicular cloud is often ignored. Getting the owner's identity or locations of a given vehicle, or getting the queries submitted by vehicles would put the data privacy at risk. In this paper we propose a privacy preserving query processing scheme for vehicular sensor networks. The scheme consists of two phases: 1) at the index mapping phase, sensed data are stored locally at road side units (RSU), and a global index is maintained at the cloud; 2) at the query processing phase, queries are transformed based on a mapping function, and then are routed to the cloud. The cloud computes the set of RSUs that have the query result based on the mapping index, and then directs the query to the RSUs. Then the result forwarding is transformed into a query result forwarding problem at the RSU, where the data of query results are routed back at their best paths either through the 4G channel or through the DSRC (Dedicated Short Range Communication) channel. Also, the query results are encrypted and diffused to a set of RSUs that the query requester would travel along, so the requester could fetch the results before the query is outdated. Extensive experiments demonstrate the effectiveness of the proposed algorithm in vehicular sensor networks. The ratio of successful query delivery is higher than existing query schemes, while at the same time preserving the privacy of query requesters and the data owner in vehicular sensor networks.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Vehicular ad-hoc sensor networks (VANETs) have emerged as a promising technology with several applications that may have a deep impact on urban scenarios [1–3]. Vehicles traveling on roads exchange information with nearby peers, and data can be disseminated and reach a far distance by using moving vehicles as intermediates [4,5] and following multi-hop routing protocols. IEEE 802 committee define wireless communication standard IEEE 802.11p [6], which serves specifically for vehicle-to-infrastructure (V2I) communication and also realizes a fast vehicle to vehicle (V2V) wireless communication connection in the urban road environment that ensures transportation safety and communication reliability for moving vehicles. The Federal Communications Commission has allocated 75 MHz of bandwidth, which operates on 5.9 GHz channel for short range communications. Vehicles communicate with other vehicles directly forming V2V communications, or communicate with fixed equipment next to the road, referred to as road side unit (RSU), forming V2I communications. VANETs enable the concept of Smart Car and Intelligent Transportation Systems (ITS), in which information, sensing and communication technologies are applied in the fields of trip services, road transportation, and traffic management [4].

However, VANETs also bring about some challenging issues for the vehicular applications such as data gathering and query processing within the network. First, vehicular nodes move on the road on high speed. The network topology changes rapidly, and this leads to fragmented and intermittent-connected communications. Second, the data are sensed periodically and a large amount

☆ This research is in part supported by the Natural Science Foundation of China (61672441, 61872154, 61862051), the Shenzhen Basic Research Program (JCYJ20170818141325209), the Beijing Municipal Science and Technology Project (Z171100005117002), the Foundation of Qiannan Normal University for Nationalities (QNSY2018JS010).

* Corresponding authors.
E-mail addresses: laiyx@xmu.edu.cn (Y. Lai), yuquan1704@163.com (Q. Yu).

https://doi.org/10.1016/j.adhoc.2019.101876
1570-8705/© 2019 Elsevier B.V. All rights reserved.

of data are generated continuously. So it needs efficient filtering and pruning mechanisms to handle these data and extract the knowledge that could be further used for the upper decision making layers [7,8]. Last but not least, privacy and safety has been a key issue in the environment of VANETs. Users / vehicles might agree to provide part of their on-board sensing data for the benefit of a more efficient and intelligent transportation system, but they also like to have their identify not be identified and have their data stored locally on the vehicles. Actually, there have been growing concerns on the potential security risks of self-driving vehicles, as they might be used as a kind of weapons to carry out atrocities, especially on the roads [9]. Through anonymizing the properties of vehicles and users, e.g., locations and types of vehicles, the VANETs applications reduce their risk on public security. Although there are some works directly addressed the privacy issues [10–12], it remains unclear and unspecified on how to integrate the privacy preserving techniques with the query processing procedure, especially in the environment of VANETs.

In VANETs queries could be diffused far away to retrieve remote data while most of existing query schemes [13,14] assume no fixed data server available in VANETs. There are 3 steps in the query processing: 1) query requester diffuses the query to different data sources, either directly or by using multi-hop relaying techniques; 2) nodes receive the query and compute part of the query results locally; 3) nodes forward the results to the query requester. As the query requester is moving along the road, the query processing inevitably incurs relatively large query delays and routing back the query results to the query requester becomes a key issue in VANETs [15–18].

Differencing from the above-mentioned in-network query processing in VANETs, recently there is research work that integrates the cloud and vehicular networks [19–22], where a group of largely autonomous vehicles coordinate and dynamically allocate their computing, sensing, communication, and physical resources with the cloud to authorized users. This concept of VANET Cloud is also highly related to “edge computing” [23,24] that highly suits the scenario of VANETs, where the road site units (RSU) are abundant in computation, storage and communication resources. There are two main reasons that we do need store data on the RSUs or edge nodes: 1) the sensing data on VANETs are usually huge and continuously generated. It would be too expensive to use 4G to upload all the data to the web cloud [14]; 2) there are applications that need near real-time response, and storing data on the RSUs would have great advantage when accessing the data on the edge, especially for applications that has some locality characteristics [19].

However, the dark side of VANET Cloud is often ignored: the cloud might accumulate enormous amounts of private user data such as detailed location information, queries, etc. The privacy issue of VANETs arises [25–31]. Attacks on privacy over VANETs are mainly related to illegally getting sensitive information about vehicles. As there are linkages between the vehicles and drivers, getting the data from vehicles’ circumstances inevitably affects the drivers’ privacy. For example, getting the owner’s identity or locations of a given vehicle, or getting the queries that a vehicle submits could put the driver’s privacy at risk. The locations of vehicles, or the paths followed within a period of time, are usually considered as personal and private data.

In this paper we propose a privacy preserving scheme called PPQ (*Privacy Preserving Query processing*). The main challenges lie in two aspects:1) while the query is encrypted, how to direct the query to the RSUs that have the query results? 2) the node that submits the request might move out of the coverage of RSUs, or there is not enough time to receive the result through the V2I or V2V communications. How to transfer and deliver the query results to the query requester efficiently and properly? The ordinary nodes, RSUs and the cloud should cooperate with each other to process the query, and the forwarding mechanism of queries and results should be well designed. The proposed scheme consists of two phases: the index mapping phase, and the query processing phase. At the index mapping phase, sensed data are stored locally at RSUs, yet partition-based mapping indexes are maintained at the cloud for storage. The main contributions of this paper are as follows:

1. We propose a privacy preserving query processing scheme called PPQ for vehicular sensor networks, which consists of index mapping phase and query processing phase. Differencing from to existing centralized or fully in-network query processing schemes, the proposed scheme integrates the cloud, the road side units (RSU), and the vehicular networks to process the query and preserve the query-related privacy.
2. We formulate the result forwarding as a QRF problem (*query result forwarding*), where the data of query results choose their best paths either through the 4G channel or through the DSRC (Dedicated Short Range Communication). Query results are encrypted and diffused to a set of RSUs which the query requester would travel along, so the requester could fetch the results before the query is outdated. So it achieves better query ratio and less latency compared with other schemes.
3. We conduct extensive experiments to demonstrate the effectiveness of the proposed algorithm in vehicular sensing applications. The rate of successful query delivery is much higher than existing query schemes, while at the same time preserving the privacy of query requesters and the data owner in VANETs.

To the best of our knowledge, this research is one of the first steps towards the integration of the cloud and the vehicular networks, as well as the 4G channel, to preserve the privacy of queries and improve the effectiveness of query processing in VANETs. The rest of the paper is structured as follows: Section 2 describes the related work; Section 3 introduces some preliminaries and defines the network model; Section 4 presents the detailed description of the PPQ framework, including data storage, indexing, and the privacy-aware query processing; Section 5 describes the environmental setup and presents the experimental results; finally, Section 6 concludes the paper.

2. Related work

In this section we review three categories of related works to position our work in the research community.

2.1. Queries and information gathering in VANETs

There has been some recent research that views vehicles as powerful mobile sensors, and propose some data gathering or information integration schemes in vehicular networks. Lee et al. [32] proposed a system called MobEyes for urban monitoring. The system opportunistically diffuses concise summaries of the sensed data, and builds a low-cost distributed index based on these summaries to facilitate various applications. Palazzi et al. [33] proposed a delay-bounded data gathering algorithm in VANETs. It gathers data from the region that meets some specified time constraints and adaptively switches the data muling and multi-hop forwarding strategies. One of the drawbacks is that it should integrate with a geo-cast protocol for the query propagation. Lai et al. [4] proposed a fog-based data gathering scheme called TPEG in VANETs. It consists of two phases and adopts a two-level threshold strategy. Nodes sense the environment at low cost sensing mode at the monitoring phase; yet they would transfer to the event-checking

and deep sensing mode when there is high probability that an event occurs. In the deep sensing mode, nodes generate more accurate sensed data.

Also, there are several works that address the query processing in vehicular networks. Motani et al. proposed the PeopleNet [34] scheme for information exchange in the mobile environment. But when it sends queries to an area that may contain relevant information, it should depend on the existing fixed network infrastructure. Lee et al. [15] proposed a mobility assisted query dissemination scheme called FleaNet. Nodes with queries would periodically advertise the queries to their one-hop neighbors to check if they could provide query results based on the local storage. Zhang et al. [16] proposed a content sharing scheme similar to FleaNet. A node queries other encountered vehicular nodes on the way. The queries are keyword-based, and a node retrieves the most popular content relevant to the query. Differed with the proposed PPQ scheme, the above-mentioned schemes only query from the one-hop neighbors, so they don't consider query and result routing problems.

Also, there are some works that utilize the concept of Distributed Hash Table (DHT) to index the data and process the queries. Delot et al. [17] proposed a scheme called GeoVanet. It uses a DHT-based model to identify a geographical location, where sensed data are forwarded for storage and queries are routed to retrieve the query results in a bounded time. Paczek et al. [35] introduced a selective data collection scheme for traffic control applications. Based on the uncertainty determination of the traffic control decisions, it decides whether to transfer the sensed data at selected time moments. One drawback of the above-mentioned DHT-alike schemes is that only the resources of in-network nodes are considered. However, the vehicular nodes are dynamic in nature, and communications among them usually incur relatively large delays, especially when intermediate hashed locations are needed for the processing. Instead, the proposed PPQ scheme integrates the RSUs, the cloud and the dynamic vehicular nodes. It takes advantage of resources at the cloud and RSUs to direct queries and preserve the privacy.

2.2. Privacy in VANETs

Privacy awareness in VANETs has been extensively studied in recent years. Hwang et al. [36] proposed a novel time-obfuscated technique to obtain a set of similar trajectories by breaking the sequence of the query issuing time. The mechanism finds (r-1) trajectories to be distributed with a distance variance and maintains a relatively low indexing cost. Ying et al. [10] proposed a dynamic mix zone for vehicular users that can support different privacy levels and ignore a user's location at the time of request. The proposed scheme can be used for safety applications on certain types of roads. Tang et al. [11] focused on long-term location privacy protection, in which symmetry, decongestion, practicability, and consistency were proposed for dummy trajectory generation. In the scheme, a user's real trajectory is used as a generator to obtain multiple dummy trajectories to confuse attackers. However, the burden on the cloud server will be higher because multiple dummy trajectories along with a real trajectory are uploaded to the server simultaneously. To achieve anonymity, it is usually acceptable to generate multiple trajectories. Lin et al. [12] systematically discussed how to implement group signature protocol in VANETs, where one group public key is associated with multiple group private keys. Under the group signature scheme, although an eavesdropper can know that a message is sent by the group, it cannot identify the sender of the message. Li et al. [31] proposed a privacy-preserving route-sharing scheme, where fog nodes are utilized to process user's group formation and participation requests. It utilizes improved anonymous authentication, rate limiting pseudonyms, modied privacy-preserving equality test, and location geo-indistinguishability.

Most of the research directly addressed the location privacy by proposing various schemes. However, little research has addressed integrating the privacy preserving techniques with the query processing procedure, especially in the environment of VANETs. Kong et al. [30] proposed a scheme that exploits structured scalars to denote the locations of data requesters and vehicles, and achieves the privacy-preserving location matching with the homomorphic cryptosystem technique. However, it could only address the range queries, and does not consider the query result forwarding problem in VANETs.

2.3. Integration of cloud and VANETs

The integration of the cloud and vehicular networks has been a research trend recently.

Eltoweissy et al. [20] proposed the concept of Autonomous Vehicular Clouds (AVC) for the first time. Within a group of semi-autonomous vehicles, the computing, sensing, communication, and physical resources can be coordinated and dynamically allocated to authorized users. The concept of AVC or VANET Cloud also highly relates to "fog computing " or "edge computing" [23] which extends traditional cloud computing paradigm to the edge of the network. Bonomi et al. [19] defined the characteristics of fog computing and emphasized that the fog/edge would bring new elements to the realm of Internet of Things through the reduction of service latency and improvement of QoS. More recently, Kai et al. [37] gave a survey on some opportunities and challenges related to the context of fog computing in VANETs. Wang et al. [24] proposed a fog structure composed of multiple mobile sinks. Mobile sinks act as fog nodes to bridge the gap between WSNs and the cloud. They cooperate with each other to set up a multi-input multi-output (MIMO) network, aiming to maximize the throughput and minimize the transmission latency.

Our scheme takes full advantage of resources at the cloud and the RSUs for the query processing at VANETs. Data are cooperatively stored and indexed, and queries are processed and forwarded along different paths according to the cost and time bound of the query. Through the integration and mutual interaction among the clouds and the edge nodes (RSUs), queries in VANETs could be processed effectively and efficiently while at the same time preserving the privacy of vehicular nodes or users. To the best of our knowledge, this research is one of the first steps towards the integration of the cloud and the vehicular networks, as well as the 4G channel, to preserve the privacy of queries and improve the effectiveness of query processing in VANETs.

3. Preliminaries

This section introduces some preliminaries of the paper, including the vehicular network model, the query model and the attack model requests.

3.1. Network model

We assume a three layered network model in VANETs: the cloud, the RSUs, and the vehicular nodes (Fig. 1). The cloud is assumed to be untrusted in the model, yet the privacy of queries and data should be preserved during the data storage and query processing. Each vehicle, e.g., v_i, periodically senses the data and monitors its environments on board and on the road. It generates pieces of data and sends them to the roadside units (RSUs) through DSRC [38], which consist of one hop vehicle to infrastructure (V2I) or multi-hop vehicle to vehicle (V2V) communications. RSUs provide storage and networking services between the vehicular nodes

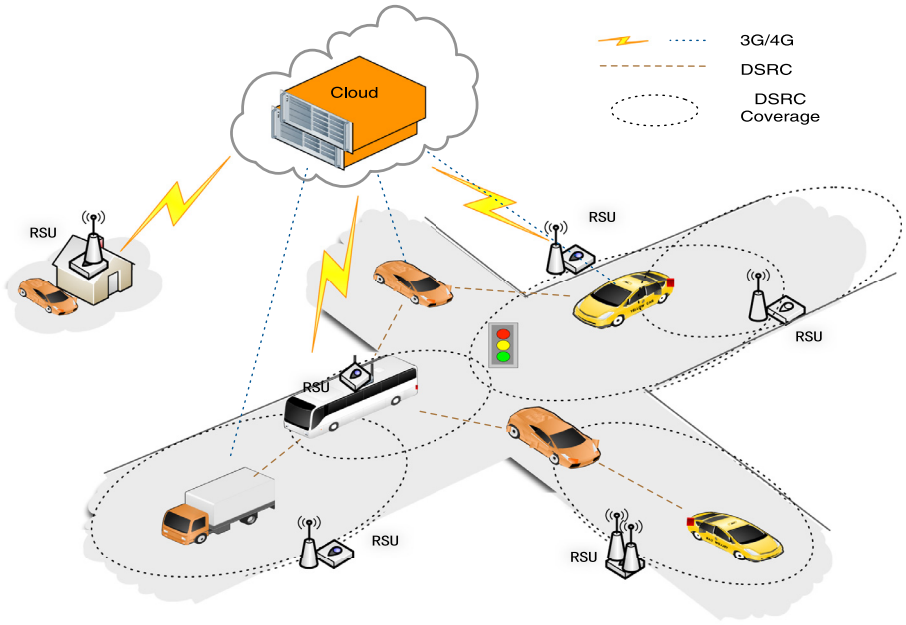


Fig. 1. Illustration of Vehicular Ad-hoc Networks with clouds. Vehicular nodes exchange data with nearby peers or RSUs through DSRC communications. Nodes could also pull/push data with the cloud through the 3G/4G channel. .

and the clouds. Each RSU has a pair of public key and private key, where public key could be diffused to other RSUs and the cloud to encrypt queries and data. Specially, when sending back data to other peers, RSUs could use its private key to decrypt the data. Previous research has proved that collision functions under some cryptographic assumption are provably hard for an enemy to find collisions [39]. Also, vehicular nodes would route the meta-data such as location, speed to RSUs when they are in contact, and upload these metadata to the RSUs, so RSUs are assumed to have the full knowledge including the speed and traffic conditions near the roads within their covered areas.

3.2. Query model

Vehicular nodes issue queries to get their desired data or query results. A query is denoted by *query* (*s, f, t, a*), where *s* is the source node, *f* is a filter, *t* is the bounded time within which results should be returned, and *a* is the attached meta-data to facilitate the query. A vehicular node keeps moving after issuing a query, and it would fetch the query result along its way to the destination before the query is expired. Note that the path and destination are assumed to be known, e.g., through the GPS navigation system, and the cloud and RSUs are assumed to work cooperatively and cache data on the intermediate RSUs on the path. During all these procedures the cloud should not be able to access the queries and the data, so that the privacy could be preserved.

3.3. Attack model

Given the role of an attacker, attacks can be classified as either insider threats and external attacks.

With insider threats, a system manager may leak information on purpose. Moreover, a user may collect queries and query results on the cloud server when packets of queries or results pass through the cloud server. In this way, the data privacy is exposed to attackers. With external attacks, there are active attacks and passive attacks. In active attacks, attackers send malicious information to mislead users, such as in luring attacks [40]. In passive attacks, attackers collect geographic and social information to estimate a user's intent queries and their privacy related information. A passive attacker has no capability to disturb the typical function of the network and it makes no alterations in the messages travelling among the nodes. The following functions may be performed by the passive attacker [41]: 1) a passive attacker is like a normal node which collects information from wireless sensor networks; 2) eavesdropping and monitoring of data from communication channel by authorized attackers or adversaries. The objectives and impacts of such type of attacker contain eavesdropping on data, information stealing and gathering, and the confidentiality and privacy necessities will be compromised.

As the data is not compromised but the confidentiality of the data is compromised, it becomes extremely difficult to detect the passive attack. We consider the passive attack model and assume that attacks might be the owner (or operator) of the cloud servers, because they could obtain all or part of the data or query. Hence the privacy is easily exposed to the owner of the cloud. Also, the RSUs deployed at the edge of networks have similar problems. When query results are forwarded or relayed through RSUs, the RSUs might make copies of the data and infer some privacy-related information. So the query and the requested data should be secured. However, we assume that the RSUs are trusted to conduct functionality properly, e.g., diffusing messages, unless users lose physical control over them.

4. PPQ framework

In this section we present an overview of the PPQ and describe the detailed components and algorithms of the scheme.

4.1. Overview

Fig. 2 and Fig. 3 depict the overall steps of the PPQ scheme, which consists of two phases, i.e. the data storage and the query processing. At the data storage phase, the sensed data are upload to RSUs through DSRC communications [38] (step 1). The RSU, e.g., $u1$, then receives the data, and it would extract the sketch of data

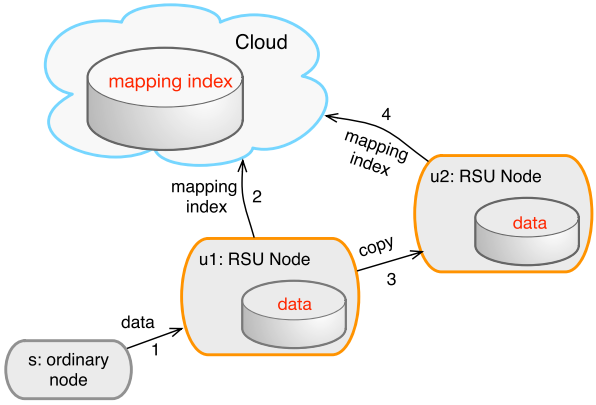


Fig. 2. Illustration of data storage and index update.

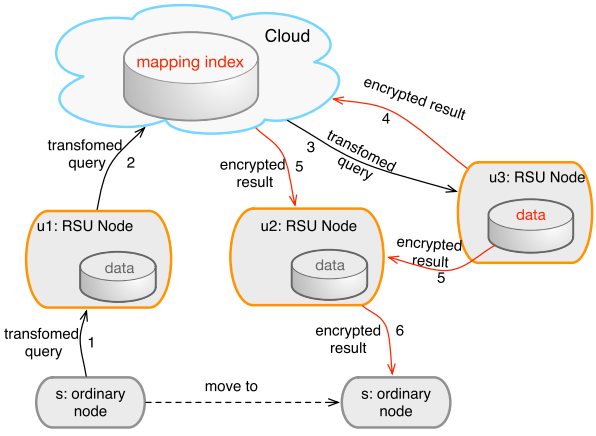


Fig. 3. Illustration of privacy preserving query processing. The arcs in black color denote the forwarding of queries, and the arcs in red color denote the forwarding of query results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and send an update message to the cloud which stores a mapping of the sensed data, denoted as *mapping index* (step 2). Data might also be forwarded and copied to other RSUs, e.g., $u2$, for storage (step 3), and its index is also updated accordingly at the cloud (step 4). At the query processing phase, a query is encrypted and transformed into an *encrypted* query before being submitted to the RSU (step 1). The query then arrives at the cloud, and the cloud processes the encrypted query by searching its indexing database (step 2) and forwards the query to RSUs (step 3), e.g., $u4$, where the desired query results are stored. The query is decrypted and processed at $u3$ and the query results are extracted. Finally, the query result is forwarded to a set of intermediate RSUs through various channels and paths (step 4 and 5), e.g., $u2$, where s moves from one location to another to fetch the query results (step 6).

The privacy of the query requesters and users in VANETs is preserved during these procedures. The cloud only stores the mapping indexes, so it would not acquire the data or their distributions that are stored at RSUs; and the cloud would not access the query and query results because they are encrypted during the procedures. The main challenges lie in two aspects: 1) while the query is encrypted, how to direct the query to the RSUs that have the query results? 2) the node that submits the request might move out of the coverage of RSUs, or there is not enough time to receive the result through the V2I or V2V communications. How to transfer and deliver the query results to the query requester efficiently and properly? PPQ integrates the cloud, and the RSUs, the vehicular nodes to cooperatively store and index the data. In this way, the queries could be processed efficiently and the query requesters' privacy could be preserved.

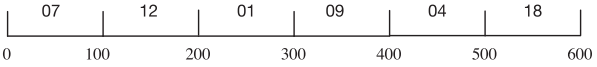


Fig. 4. Partition and identification of a value domain. The value ranges of a dimension are mapped into identifiers.

4.2. Data storage and indexing

In the scenarios of VANETs, sensed data come from various sources and formats, e.g., GPS locations, streaming videos, or monitoring data of engines. These data are injected into the network, stored at the RSUs and then indexed at the cloud. To be more specific, in PPQ a piece of data is uploaded to nearby RSUs, the RSUs then store the data, and the cloud would keep an index entry of the data. Indexes maintained at the cloud represent the global knowledge. They are used to direct queries to potential RSUs to extract the query results.

We denote a piece of data as $table\_name(a_1, a_2, \ldots, a_k, data)$, where $table\_name$ denotes the set or table that the data belongs to, a_i is the i^{th} attribute of the metadata that has k dimensions, and $data$ is an extra data object linked to the reading, e.g., video file. Then for each attribute a_i, its domain is partitioned into segments, and the value on that attribute is mapped to an identifier, denoted by $I(a_i)$:

$$E(table\_name) \leftarrow encrypt(table\_name), \quad I(a_i) \leftarrow map(a_i) \qquad (1)$$

where $encrypt$ is an encryption function that encrypts the table name $table\_name$, map is a collision free hash function that hashes a value to an identifier. Here we assume RSUs are trusted to each other and use the same encryption function to encrypt the table name. Fig. 4 illustrates an example that hashes the value range into identifiers. The value range is partitioned into 6 partitions: (0,100], (100,200], (200,300], (300,400], (400,600], (500,600], and each partition is mapped to an identifier. So a data reading would be transformed into a *mapping index*, which is in the form of $E(t\_name)(id, I(a_1), I(a_2), ..., I(a_k))$, where id is the identification of the RSU that generates the mapping index. The mapping indexes indicate the distribution of data at specific RSUs. For example, $E(Location)(0021, 02, 04)$ denotes a mapping index of a data reading that belongs to table *Location*. The data is stored at RSU with identification 0021 and has the first and second dimension mapped to 02 and 04 respectively.

Mapping indexes are sent to the cloud periodically and are stored at the cloud, so the cloud could track data and their copies within the network. However, because the hash function map is irreversible, the cloud is not able to figure out the data ranges, and let alone the detailed data that are stored at the RSUs. In this way, the privacy of the data at RSUs is preserved. Data readings might be copied to other trusted RSUs for data storage. Also, the cloud could track data and their copies within the network because RSUs would send the update of data mapping index to the cloud whenever they receive segments of data.

4.3. Query processing

A query is submitted by a vehicular node in the form of a SQL string, denoted by *query*:

```
select A1,A2 from SpeedNet
where A1>23 and A2 <87
```

Yet to preserve the privacy of querier, the query is mapped to a new query, denoted by *query′*:

```
select id from MappingIndex
where TableName = encrypt(SpeedNet)
```

```
and I(A1) in {11,06,04,19}
and I(A2) in {05,06,43}
```

where *id* is the identification of the RSUs, MappingIndex is the mapping index storage on the cloud, {11, 06, 04, 19} are the identifiers on $A1$ that satisfy the predicate $A1 > 23$, and {05, 06, 43} are the identifiers on $A2$ that satisfy the predicate $A2 < 87$.

The mapped query *query′* is also called "transformed query" that would then be forwarded to the cloud for further processing. Different methods could be adopted to process the query, depending on different surrounding environments of vehicular nodes. The public key of the query requester *s.PublicKey* is also embedded in the query message, which could be sent to the cloud and distributed within the vehicular network.

Note that each RSU has a pair of public key and private key, where its public key could be diffused to other RSUs and the cloud to encrypt queries and data. Specially, when sending back data to other peers, RSUs could use its private key to decrypt the data. If one RSU is compromised or corrupted, queries not targeted to the RSU are still safe because the encrypted data could only be accessed by those who have the private keys. Moreover, previous research [39] has proved that collision functions under some cryptographic assumption are provably hard for an enemy to find collisions.

4.3.1. Search and forward query results

When the cloud receives the mapped query *query′*, it would execute the query on the mapping index storage. So *id′s* that indicate the RSUs that store the results of the query are retrieved. We denote $\mathbb{M} = \{m_1, m_2, \ldots, m_k\}$ as the set of RSUs where the query results could be retrieved. Then the encrypted query is forwarded to these RSUs to extract the query results.

The RSU that receives *query′* would execute the query, and retrieve the query result. The query result would be encrypted using the public key of the query requester, and be sent back to the query requester. Here we assume the public-key cryptograph (also called asymmetric cryptography) schemes, e.g., RSA [42] and DSS [43], are used in the scheme.

Query results are to be forwarded back to RSUs that the query requester would pass through. The set of RSUs are denoted as $\mathbb{D} = \{d_1, d_2, \ldots, d_l\}$, which is a sequence of l RSUs that the query requester s would pass through to fetch the results. The set of RSUs are assumed to be predicted. So the query result forwarding problem is to calculate the amount of query results that are to be forwarded to the RSUs in $\mathbb{D}$. One main constraint of the query is the time bound, which is denoted by t. A query is successfully processed only when its result is returned to the requester within the time constraint t. So when an RSU, e.g., u_s, acquires the query results, it would forward back the results to the query requester as soon as possible.

In VANETs, there are three channels of communications, i.e. the DSRC-based V2V or V2I, the wired I2I, and the 4G. These channels differ in the bandwidth, transmission speed, and economic cost, and query results could be forwarded along different channels and paths. Table 1 presents the notations when defining the paths that are described below.

1. *Path*1: Query results are uploaded and downloaded both through the 4G channel. The cloud is denoted by d_0, and the amount of data forwarded from m to the cloud by x_0. The time needed for the transmission is:

$$\varphi(x_0) = \eta_{4g} + \gamma_{x_0} + max\left(\frac{x_0}{BW_{u,4g}}, \frac{x_0}{BW_{d,4g}}\right) \tag{2}$$

where η_{4g} is the time needed to establish the connection, γ_{x_0} is the time needed for the data processing at the cloud and the RSUs, BW is the bandwidth of the channel, and the subscript u and d are the up and down link of the channel. Here the *max* function implies the uploading of query results to the cloud is in parallel with the downloading to the source node s.

2. *Path*2: Query results are forwarded to RSUs in $\mathbb{D}$ through I2I or V2V communications, and data are fetched when s moves through these RSUs. The time needed could be estimated by the travel time of s to visit RSUs in $\mathbb{D}$ one by one:

$$\rho = \frac{dis(u_s, d_1)}{v_{u_s,d_1}} + \sum_{j=1}^{l-1} \frac{dis(d_j, d_{j+1}) * \psi(j)}{v_{d_j,d_{j+1}}} \tag{3}$$

$$\psi(j) = \begin{cases} 1, & if \quad \sum_{z=j+1}^{l} x_z > 0 \\ 0, & if \quad \sum_{z=j+1}^{l} x_z = 0 \end{cases} \tag{4}$$

where $dis(a, b)$ is the distance between a and b, $\psi(j)$ is a function denoting whether d_j is the last RSU participating the forwarding, and $v_{a,b}$ is the average speed of node s moving from a to b. The speeds of vehicles are assumed to be estimated through periodic data exchange between vehicles and RSUs.

Note that the query results should be returned before a bounded time. So the *Path*2 channel, although is more expensive, acts as a supplementary path to forward the results when the query may be outdated if this channel is not used.

4.3.2. Solving the query result forwarding problem

*Path*1 is faster on data transmissions, yet suffers higher economic cost as it solely uses the 4G channel. *Path*2 is cheaper as it depends on the DSRC channels, but incurs a larger delay because the "store-carry-forward" transmission strategies in VANETs. PPQ would use both the paths to gather the query results and strike a balance between the cost and the time delay. The query result forwarding (QRF) is modeled as a linear programming problem, which is well studied and could be solved efficiently in the worst case [44,45]. The notations are presented at Table 1.

$$\text{Minimize: } (x_0 * (c_{u,4g} + c_{d,4g}) \tag{5}$$

$$+ \sum_{j=l}^{l} x_i * (c_i + c_{j,s})) \tag{6}$$

Subject to:

$$x_i \geq 0, \quad i = 0, 1, ., l, \tag{7}$$

$$\left(x_0 + \sum_{i=1}^{l} x_i\right) = |data| \tag{8}$$

$$\frac{x_i}{BW_{i2i}} \leq \frac{dis(s, d_i) - R}{v_{s,d_i}}, \quad i = 1, 2., l \tag{9}$$

$$\frac{x_i}{BW_{i2v}} \leq \frac{2R}{v_{d_i}}, \quad i = 1, 2., l \tag{10}$$

$$\varphi(x_0) \leq t - \Delta \tag{11}$$

$$\rho \leq t - \Delta \tag{12}$$

The objective of the model is to minimize the total cost of query result forwarding. Term 5 and 6 are the cost of forwarding results through *Path*1 and *Path*2 respectively. x_i is the total amount of data sent from RSU m to d_i, $c_{u,4g}$, $c_{d,4g}$ are the unit cost of the up and down channel of 4G, c_i is the unit cost of transmissions from m to

Table 1
Notations and their default settings at experiments.

Notations	Meanings
s	the query requester (source node)
m	the RSU where the query results are retrieved
d_0	used to denote the cloud
$\mathbb{D}$	$= \{d_1, d_2, \ldots, d_l\}$, a sequence of l RSUs that s would pass through to fetch the results
$\mathbb{M}$	$= \{m_1, m_2, \ldots, m_k\}$, a set of k RSUs where the query results are retrieved
x_i	the total amount of data sent from m to d_i
$c_{u,4g}$, $c_{d,4g}$	the unit cost of the up and down channel of 4G, default 10^{-2} \$/MB
c_i	the unit cost of transmissions from RSU m to d_i, default 10^{-4} \$/MB
$c_{j,s}$	the unit cost of transmissions from d_j to source node s, default 10^{-4} \$/MB
R	radius of the coverage area of RSU, default 60 m
BW_{i2i}, BW_{i2v}	bandwidth of the RSU-to-RSU (I2I) and RSU-to-Vehicle (I2V) communication channel, default 50Mbps for I2I, 500 Kbps/250 Kbps for the down/up links of I2V
v_{s,d_j}, v_{d_j}	average speed between node s and d_j, and within the coverage area of d_j, computed according to trajectories
γ_{x_0}	time for data processing at the cloud or RSUs, default 1–5 s
η_{4g}	waiting time to establish a connection for the 4G channel, default 0.5 s

d_i, and $c_{j,s}$ is the unit cost of transmissions from d_j to source node s. In $path2$, results are forwarded from l RSUs in $\mathbb{M}$ to intermediate RSUs in $\mathbb{D}$, and then to the query requester s. As the 4G channel is expensive, to prevent the channel being overloaded, the unit cost $c_{u,4g}$ is set much larger than c_i so as to make $Path2$ feasible for the data transmission. Constraint 7 implies the range of x_i. Only RSUs that have x_i greater than 0 are selected as relaying RSUs where the requester could fetch the query result. Constraint 8 implies all data are routed back to the query source, either through the cloud or through intermediate RSUs. Here $|data|$ is the amount of data in the query results. Constraint 9 implies the data are prepared and received at d_i before query requester s arrives the coverage area of RSU, i.e. d_i, to fetch the data. Here $dis(s, d_i)$ is the distance between s and RSU d_i, and R is the radius of coverage of an RSU. v_{s,d_j} is the average speed between node s and d_j. Constraint 10 implies there should be enough time for the node to fetch the data when it passes through the coverage area of d_i. BW_{i2v} is the bandwidth of the RSU-to-Vehicle (I2V) communication channel, and v_{d_j} is the average speed within the coverage area of d_j. Symbol $\varphi(x_0)$ and ρ are defined at Eqs. 2 and 3, where t is the time bound for the query, Δ is the time that has elapsed before forwarding the query results. Constraint 11 and 12 ensure the forwarding time should be within the time bound of the query. Here the query results are assumed to be forwarded in parallel through different communication channels.

Note that the average speeds v_{s,d_j} and v_{d_j} are the inputs for this model, so the time needed for s to visit the RSU in $\mathbb{D}$ one by one could be estimated. We assume the current location of query requester and its traveling path are known, e.g., through the GPS navigation system, and RSUs would learn and monitor the speeds of vehicles moving along the roads. For example, whenever a vehicle s pass through an RSU u_i, it uploads its average speed from u_{i-1} to u_i, where u_{i-1} is the latest RSU that s contacts with. So each RSU is able to estimate the speeds between its neighboring RSUs. The speed information is also uploaded to the cloud so the cloud has the knowledge to predict the set of RSUs $\mathbb{D}$. RSUs are with computing capability and have the solver installed. In the concept of "edge computing" [23], RSUs are viewed as edge nodes that would calculate optimal solutions for QRF problem.

4.4. Algorithm description

As illustrated in Fig. 3, a query is forwarded to RSUs, and up to the cloud. The cloud processes the encrypted query by searching its indexing database and forwards the query to RSUs, where the desired results are stored. The query is then decrypted and processed and the query results are extracted. Finally, the query result is forwarded to a set of intermediate RSUs where the query requester moves from one location to another to fetch the query results. Algorithm 1 presents the pseudocode of the PPQ scheme, which consists of procedures at the ordinary nodes, the RSU's, and the cloud.

Algorithm 1: PPQ: Privacy Preserving Query Processing Algorithm.

```
1  ———— at ordinary nodes ————————
2  if receive data from self then
3      send(⟨data⟩, self.RSU, DSRC);
4  if receive query from self then
5      query' =transform(query);
6      send(⟨query', s.PublicKey⟩, cloud, 4G);
7  if receive ⟨eData⟩ from RSU then
8      result = decrypt(eData, s.PrivateKey);
9      return_answer(result, query);
10 ———————— at RSU ————————
11 if receive ⟨data⟩ from node then
12     store(data, localStorage);
13     MappingIndex index = map(data);
14     send(⟨index⟩, cloud);
15 if receive ⟨query', s.PublicKey, 𝔻⟩ from cloud then
16     result = get_data(query', localStorage);
17     problem = search(query', data', 𝔻);
18     commendSet CS= LPSolver.solve(problem);
19     for command C in CS do
20         data=getPartialData(result,C.x_i);
21         eData= encrypt(data, s.PublicKey);
22         send(⟨eData⟩, C.dest, C.channel);
23 if receive ⟨eData⟩ from RSU then
24     send_or_broadcast eData to s;
25 ———————— at cloud————————
26 if receive ⟨index⟩ from RSU then
27     update(index, indexStorage);
28 if receive ⟨query', s.PublicKey⟩ from node then
29     IDs = search(query', indexStorage);
30     𝔻= predictVehicle (s);
31     for each id in IDs do
32         send(⟨query', s.PublicKey, 𝔻⟩, id);
```

When a vehicular node generates a piece of data, it uploads the data to nearby RSUs through the V2I communications (line 2–3).

The RSU that receives the data would store the data (line 11–12), generate a mapping index and upload the index entry of the data to the cloud (line 14). When the cloud receives the mapping index, it stores the index in its *indexStorage* (line 27).

When a query is generated at a vehicular node, the query is transformed to *query′*. The transformed query and the public key of query requester *s* are sent to the cloud (line 5–6). Here any cryptographic system that uses pairs of keys could be used. Public keys are disseminated widely to RSUs through the help of the cloud, and private keys are known only to the owner. When the cloud receives the mapped query, it exacts *query′* and execute *query′* on the mapping index storage. So *id's* that indicate the RSUs that store the results of the query are retrieved (line 29). Then *query′* and *s.PublicKey* are forwarded to these RSUs to extract the query results (line 32). Also, the set of RSUs that the query requester would pass by, denoted by $\mathbb{D}$, would also be extracted and sent to the RSU (line 30).

When the mapped query is received by an RSU, the RSU would search its local storage to get the results of *query′* (line 16). The forwarding commands are calculated by solving a linear programming problem (line 17–18). Variables $\{x_i | i = 0, ., l\}$ are calculated by the LP solver and wrapped to the command. A command, e.g., *C*, contains the destination (*C.dest*) and channel of the forwarding (*C.channel*), and the amount of data to be forwarded x_i. For each command, its partial results are extracted and encrypted using the public key *s.PublicKey*(line 19–21). Then the encrypted data are sent according to the command.

When the encrypted query result is received by the RSU, e.g., d_i, the RSU would send or broadcast the result to the query requester *s* (line 24). Finally, when the query requester receives the result data from RSUs that it passes by, it would decrypt the data using its private key and return the query results (line 9).

5. Experimental study

5.1. Environment setup

Simulated experiments are conducted on the ONE platform [46] with real-world road network to verify the performance of the proposed scheme. 120 RSUs are evenly deployed along the roads, as depicted in Fig. 5. The Xiamen Taxi Dataset[1] that contains taxi trajectories of Xiamen city, China during July 2014 is used for the simulation. The datasets consist of one-month trajectory data of about 5000 taxis in Xiamen. The frequency of trajectory reporting is 1–2 times per minute, so in total there are about 220 million GPS position records and 8 million live trips in the dataset. For this simulation we extract 506 taxies and about 1/10 of the trajectories for the performance evaluation.

A speed network of roads is generated according to the trajectories after mapping the GPS points to the road segments. The speed of the vehicles differs according to road segments and time periods. Fig. 6 depicts the snapshots of speed network of Xiamen City, China, limited to a rectangle area of [118.0660*E*, 118.0990*E*] × [24.4300*N*, 24.5300*N*].

5.2. Experimental setting

There is no limit on the communication range of the 4*G* channel, yet the range of I2I or I2V communication is set 60 m. The time for data processing at the cloud or RSUs, denoted as γ_{x_0}, is set 1 to 5 sec, and the waiting time to establish a connection for the 4G channel, denoted as η_{4g}, is set 0.5 sec (defined at equation (2)). The bandwidth of the 4*G* channel is set 20 Mbps/5 Mbps

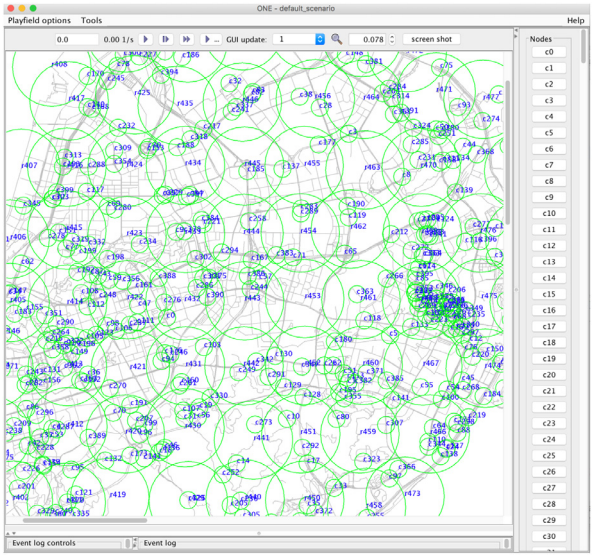


Fig. 5. Snapshot of the simulation field in Xiamen Island. The blue texts denote vehicular nodes, and green circles denote the coverage areas of RSUs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

for the down/up links, the bandwidth of the V2V or V2I channel is 500 Kbps/250 Kbps for the down/up links, and the bandwidth of the I2I channel is 124 Mbps. The cost of the channel $c_{u,4g}$ is set 10^{-2} \$/MB and $c_{i,j}$ is set 10^{-4} \$/MB. Ideal links are assumed when two nodes meet and establish a connection, and the query request, query result and metadata could be wrapped into one message respectively.

The total simulation time is 16 h, from 7:00 to 23:00 within a day. A data generator is installed to inject data to the vehicular sensor networks, where every node senses a data sample every 60 sec. The sensing data have three attributes A_1, A_2, A_3, *Data*, where attributes A_1, A_2, A_3 are the metadata attributes that range from 0 to 1000, and *Data* is the detailed sensing data. The value ranges for A_1, A_2, A_3 are partitioned into 20 per segment by default. The size of sample ranges from 0.4 to 1.0 megabytes. A query is defined as retrieving the *Data* attribute given ranges of the attribute. The time bound for the query follows a normal distribution: t ∼ N(600s, 40), and every node generates a query at an average rate of 0.1 query/minute. GLPK for Java[2] is used as a solver for the QRF problem.

5.3. Metric and compared algorithms

The ratio of successful queries, the delay of query and the total cost of query processing are the main metrics for the performance analysis. A query is successful only when results are returned to the requester before the bounded time. The query ratio is a main performance metric for the query processing scheme in VANETs, as lots of queries would be failed if they are outdated, due to their susceptibility to external interference that leads to disconnected and portioned network. The cost of queries is the sum of cost along different paths, where the cost of each path is calculated by multiplying the amount of transmitted data and the unit cost of transmission of that channel.

For the performance comparison, we implement other three query processing schemes as follows:

[1] http://mocom.xmu.edu.cn/project/show/xmdataset.

[2] http://glpk-java.sourceforge.net/.

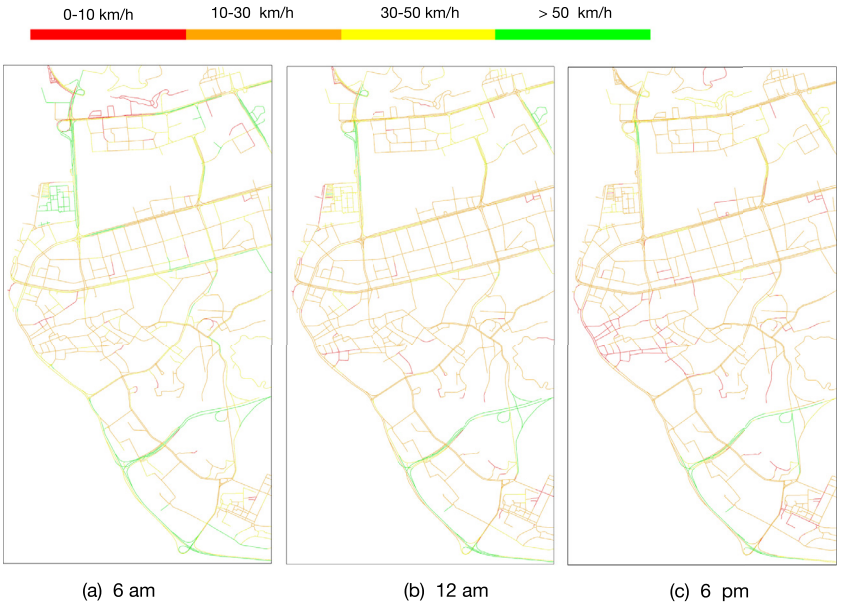


Fig. 6. Snapshots of speed network of Xiamen City, China, limited to an area of [118.0660*E*, 118.0990*E*] × [24.4300*N*, 24.5300*N*].

Table 2
Overall Performance of the Schemes.

Schemes	Query Ratio(%)	Query Delay(s)	Overall Cost($)
Centralized	100.00	3.2	2356.73
Flooding	38.63	525	56.20
FleaNet	10.24	602	13.86
GeoVanet	54.18	610	34.47
PPQ	**92.12**	**588**	**302.98**

- 1) Centralized: sensed data are uploaded to a centralized cloud server through the 4G channel, and queries are also directed to and processed at the cloud;
- 2) Flooding: vehicular nodes would relay all the received queries and the query results, so as to route back the query results to the query requester;
- 3) FleaNet [15]: vehicular nodes with queries periodically advertise the query only to the one-hop neighbors. Neighbors check and search their local storage to answer the queries;
- 4) GeoVanet[17]: vehicular nodes follow a rule to route their sensing data to a fixed geographical location based on a DHT-based (distributed hash table) model; and queries are mapped to the dedicated point to retrieve the results within a bounded time.

Besides Centralized, all the compared schemes avoid using a centralized server, and do not integrate with the cloud. To the best of our knowledge, there are no related querying schemes that consider the privacy issues in VANETs. So we mainly compare the success ratio, the delay and the cost of queries in the following studies.

5.4. Experimental analysis

5.4.1. Overall performance

Vehicular nodes follow the trajectories of the dataset. About 485×10^5 samples are sensed and 4.8×10^4 queries are processed. The overall performance of the schemes are presented in Table 2. The Centralized scheme has a query ratio of 100% and a very small query delay (3.2s). This is mainly because it uploads all the data through the 4G channel, and stores all the data at the cloud so as all the queries could be successfully processed. However, it has the largest work load for the telephony network. As the unit cost of the 4G channel is expensive, e.g., 10^{-2} \$/MB, its total cost is about 2356.73 dollars for the data upload and query result download, which is about 8 times that of the PPQ scheme or about 41 times that of the Flooding scheme. Moreover, because all the data are stored at the cloud, it incurs a high data privacy risk. On the contrary, the Flooding, FleaNet and GeoVanet store and process the data on local nodes and depend on the in-network DSRC communications for the query processing, which has a lower data privacy risk given that the RSUs are trusted. The total cost is less than 60 dollars, due to the small unit cost of V2V or I2V communications (10^{-4} \$/MB). However, the query ratio is much lower, which is less than 55% for all the three schemes. The query ratio of FleaNet is about 10.24%, Flooding is about 38.63%, and GeoVant is about 54.18%. This is due to their different relaying strategies of the queries. In FleaNet, queries are forwarded only to one-hop neighbors. In Flooding, queries and results are flooded within the network, so it has more probability to answer the query and return the result as more nodes could receive the query. However, the query result might not be able to be forwarded to the query requester because the query requester is moving to other regions while the query processing is being processed. So the query ratio is relatively lower compared to other schemes. In GeoVant, data are routed to fixed RSUs for storage, and the query results could be fetched by the query requester. So it has relatively higher query ratio and the cost is as low as about 34.47 dollars. The delay of the query is low at the Centralized scheme, yet for other schemes the query delay is close to the time bound of the queries, which is about 600 sec. Here only the queries that are successfully processed are accounted for the delay calculation.

The proposed PPQ scheme has a query ratio as high as 92.12%. The cloud plays a vital role on the indexing of data, and helps to forward the queries to RSUs that have the query answers, which speeds up the query processing and avoids query expiration. Moreover, the 4G channel is adopted to forward the result to the query requester, which makes a large number of queries processed on time. As showed in Fig. 7, more than 30% of the queries in PPQ are assisted by the 4G channel, which means part of the query results are forwarded through the *Path*1 to the query requester. Without the assistance of the cloud and the 4G channel, these

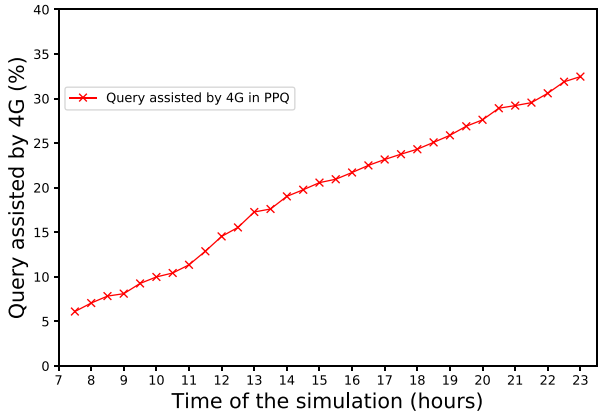


Fig. 7. Accumulated percentage of queries assisted by 4G.

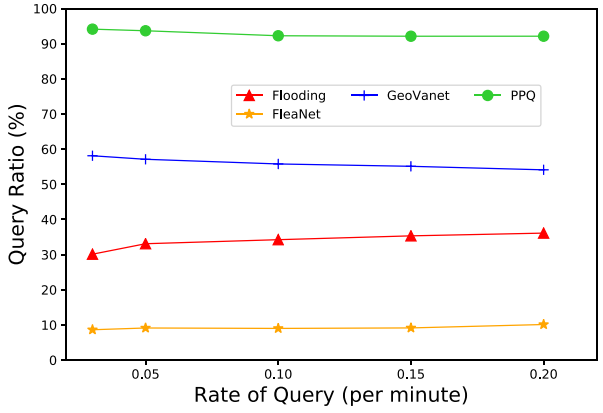


Fig. 8. Impact of query rate to the query ratio.

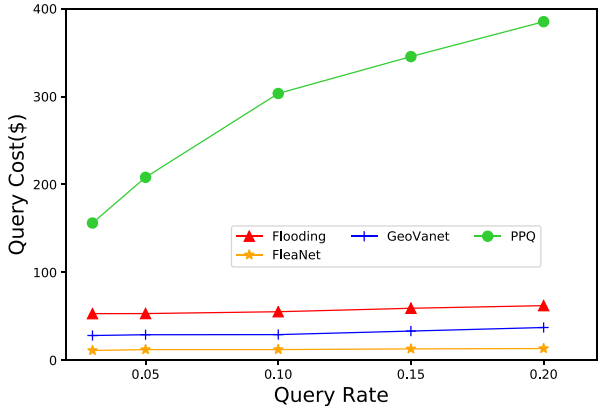


Fig. 9. Impact of query rate to the cost of query.

queries would otherwise break the bounded time and be dropped. This, on the other hand, increases the cost of the query processing. It costs about 302.98 dollars, about 5.39 times of the Flooding scheme. The economic cost of PPQ is about 12% of the Centralized scheme, which is still much cheaper.

5.4.2. Impact of query rate

Queries are generated according to a *query rate*, where larger query rate generates a larger number of queries. We first vary the rate of queries and study its impact on the performance. As depicted in Figs. 8 and 9, the impact on the Flooding, FleeNet and GeoVanet schemes is relatively small. The cost increases about 20–30 percent as more queries are to be forwarded to the encountered nodes, but the ratios of successful queries do not increase accordingly. For the PPQ scheme, the cost increases from 157 to 385 dollars when the query rate grows from 0.05 to 0.20 per minute, but the ratio of successful queries seems not be affected by the increased number of queries. When there are more queries, a larger number of query results are to be forwarded to the requester. But as the bandwidth of the I2V channel is limited and the query time is bounded, according to the QFP calculation in PPQ more data would choose the *Path*1 and be routed through the 4G channel to the requester. This leads to sharp increase for the overall cost, yet does not increase the overall query ratio.

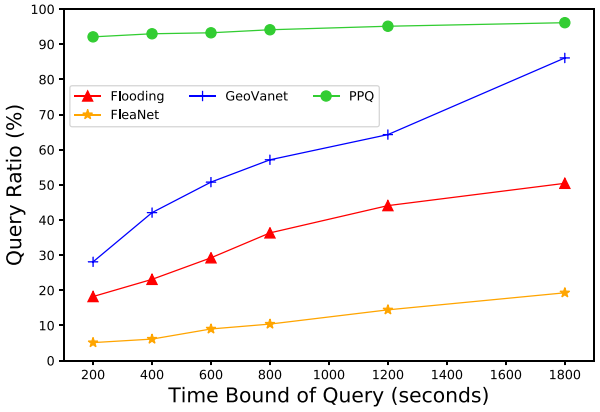


Fig. 10. Impact of bounded time to the query ratio.

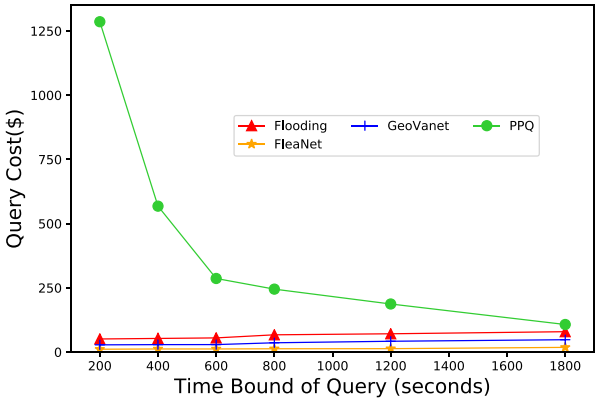


Fig. 11. Impact of bounded time to the cost of query.

5.4.3. Impact of bounded time

As mentioned before, the bounded time of query plays a key role in the success of queries, so we also vary it to study the performance. As depicted in Figs. 10 and 11, the query ratios increase with the bound or query time. It increases from about 5%, 18%, 30% to 86%, 47%, 20% for the FleaNet, Flooding, GeoVanet schemes when the time bound increases from 200s to 1800s. All these schemes are the in-network query processing schemes, and they adopt a “carry-store-forwarded” strategy to forward the queries and results. They need more time to route a piece of data to its destination, which leads to the expiration of queries as there is not enough time. Yet when the bounded time grows, there is more time to process the queries, which leads to the increase of query ratio. For the GeoVanet scheme, the data are stored at a hashed location and queries are forwarded to that location to extract the query results through multi-hop communications. So usually GeoVanet needs longer time to process the queries. When the time bound increases from 200s to 1800s, its query ratio grows from 27% to 86.2%.

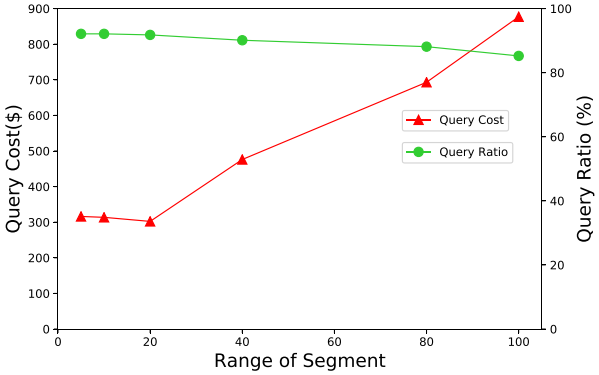


Fig. 12. Impact of range of segments in PPQ.

For the PPQ scheme, the query ratio grows from 91% to 97% as the bounded time grows from 200s to 1800s. But the cost decreases from 1286 dollars to 107 dollars. This is because PPQ would adaptively choose its path to route back the query results according to the budget of query time. When the bounded time is small, i.e. approaching to the expiration time, PPQ would choose *Path*1 and adopt the 4*G* channel to route back the query result. As the delay of the 4G channel is much smaller, more queries could be successfully processed. Also, it incurs larger cost as the 4G channel is assumed more expensive. On the other way, if the bounded time of queries are larger, i.e. there are enough time for data to be forwarded to the requester, PPQ would choose *Path*2 through the cheaper channel of DSRC to transmit the data. Hence, the economic cost of query processing decreases to about 92$.

5.4.4. Impact of segment ranges

PPQ preserves the privacy of queries through mapping the queries and data into partitioned value ranges. The partitions are mapped to mapping indexes, which are then routed to the cloud to direct the queries. So the range of segments plays an important role for the query processing. Fig. 12 shows the impact of segment ranges to the query ratio and query cost. From the figure we could see that the cost of query increases as the range of segments becomes larger. The cost is about 300 dollars when the range is less than 20, yet it grows to near 900 dollars when the range is 100. This is explicable because the transformed queries are based on the segments, and the results of the transformed query are the superset of those of the original query. Extra results lead to the anonymity of the queries and results. For the transformed query, larger range would have larger size of query results, so the cost of result forwarding would also increase.

The ratio of successful queries is about 91%, and it decreases a little bit as the range becomes larger. This is because the communication channel is limited, and the delay would increase if there are a larger number of messages to be forwarded back to the query requesters. So some query results would become expired and be dropped, which leads to the decrease of the query ratio.

6. Conclusions

We propose an efficient privacy preserving query processing scheme called PPQ in VANETs, based on the integration of the RSUs and the cloud. Sensed data are stored locally at RSUs, and partition-based mapping indexes are maintained and stored at the cloud. Queries are firstly transformed to a mapped encrypted query using a mapping function, and then routed to the cloud. The cloud computes the set of RSUs where the query results are stored based on the mapping index, and then directs the query to each of the RSU to extract the query results. The result forwarding is then transformed to a *query result forwarding problem* and query results are encrypted and diffused to a set of RSUs which the query requester would later travel along and fetch the results before the query is outdated. Experiments based on simulations on real-world trajectories demonstrate the effectiveness of the proposed algorithm in vehicular sensing applications. The ratio of successful queries is much higher than the existing schemes, while at the same time preserving the privacy of query requesters and the data owner in VANETs.

For the future work, we are going to investigate the impact of traffic patterns to optimize the privacy preserving query processing procedures in VANETs. Also, we are going to study other privacy preserving techniques in the scenarios of vehicular networks, e.g., vehicular privacy preservation of trajectories and anonymity of query requesters.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

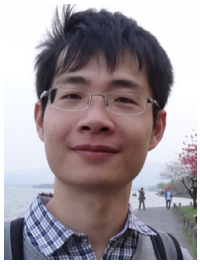
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.adhoc.2019.101876.

References

[1] S. Al-Sultan, M.M. Al-Doori, A.H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network, J. Netw. Comput. Applica. 37 (2014) 380–392.
[2] A. Dua, N. Kumar, S. Bawa, A Systematic Review on Routing Protocols for Vehicular ad hoc Networks, Vol. 1, Elsevier, 2014. Pp: 33–52.
[3] Y. Lai, L. Zhang, F. Yang, L. Zheng, T. Wang, K. Li, Casq: adaptive and cloud-assisted query processing in vehicular sensor networks, Future Genera. Comput. Syst. 94 (2019) 237–249.
[4] Y. Lai, F. Yang, J. Su, Q. Zhou, T. Wang, L. Zhang, Y. Xu, Fog-based two-phase event monitoring and data gathering in vehicular sensor networks, Sensors 18 (1) (2017) 82.
[5] Y. Lai, L. Zhang, T. Wang, F. Yang, Y. Xu, Data gathering framework based on fog computing paradigm in vanets, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data, Springer, 2017, pp. 227–236.
[6] D. Jiang, L. Delgrossi, Ieee 802.11 P: towards an International Standard for Wireless Access in Vehicular Environments, in: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, IEEE, 2008, pp. 2036–2040.
[7] Y. Lai, H. Lin, F. Yang, T. Wang, Efficient data request answering in vehicular ad-hoc networks based on fog nodes and filters, Future Genera. Comput. Syst. 93 (2019) 130–142.
[8] C. Zhang, C. Liu, X. Zhang, G. Almpanidis, An up-to-date comparison of state-of-the-art classification algorithms, Expert Syst. Appl. 82 (2017) 128–150, doi:10.1016/j.eswa.2017.04.003.
[9] F.S. De Sio, Killing by autonomous vehicles and the legal doctrine of necessity, Ethical Theory Moral Pract. 20 (2) (2017) 411–429.
[10] B. Ying, D. Makrakis, H.T. Mouftah, Dynamic mix-zone for location privacy in vehicular networks, IEEE Commun. Lett. 17 (8) (2013) 1524–1527.
[11] F. Tang, J. Li, I. You, M. Guo, Long-term location privacy protection for location-based services in mobile cloud computing, Soft Comput. 20 (5) (2016) 1735–1747.
[12] X. Lin, X. Sun, P.H. Ho, X. Shen, Gsis: a secure and privacy-preserving protocol for vehicular communications, IEEE Trans. Veh. Technol. 56 (6) (2007) 3442–3456.
[13] M.Y. Hsieh, J.W. Ding, Dynamic scheduling with energy-efficient transmissions in hierarchical wireless sensor networks, Telecommun. Syst. 60 (1) (2015) 95–105.
[14] Y. Lai, X. Gao, M. Liao, J. Xie, Z. Lin, H. Zhang, Data gathering and offloading in delay tolerant mobile networks, Wireless Netw. 22 (3) (2016) 959–973.
[15] U. Lee, J. Lee, J.S. Park, M. Gerla, Fleanet: a virtual market place on vehicular networks, IEEE Trans. Veh. Technol. 59 (1) (2010) 344–355.
[16] Y. Zhang, J. Zhao, G. Cao, Roadcast: a popularity aware content sharing scheme in vanets, ACM SIGMOBILE Mob. Comput. Commun. Rev. 13 (4) (2010) 1–14.
[17] T. Delot, N. Mitton, S. Ilarri, T. Hien, Geovanet: a routing protocol for query processing in vehicular networks, Mob Inform. Syst. 7 (4) (2011) 329–359.
[18] X. Xing, D. Xie, G. Wang, Energy-balanced data gathering and aggregating in wsns: a compressed sensing scheme, Int. J. Distrib. Sens. Netw. 11 (10) (2015) 585191.
[19] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, 2012, 13–16.

[20] M. Eltoweissy, S. Olariu, M. Younis, Towards autonomous vehicular clouds, in: International Conference on Ad Hoc Networks, Springer, 2010, pp. 1–16.
[21] Y. Lai, F. Yang, L. Zhang, Z. Lin, Distributed public vehicle system based on fog nodes and vehicular sensing, IEEE Access 6 (2018) 22011–22024.
[22] T. Wang, J. Zhou, A. Liu, M.Z.A. Bhuiyan, G. Wang, W. Jia, Fog-based computing and storage offloading for data synchronization in iot, IEEE Internet Things J. (2019), doi:10.1109/JIOT.2018.2875915. 1–1.
[23] Y.C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile edge computing a key technology towards 5g, ETSI White Paper 11 (11) (2015) 1–16.
[24] T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian, Y. Chen, B. Wang, Data collection from wsns to the cloud based on mobile fog elements, Future Genera. Comput. Syst. (2017), doi:10.1016/j.future.2017.07.031.
[25] C.Z. Gao, Q. Cheng, P. He, W. Susilo, J. Li, Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack, Information Sciences 444 (2018) 72–88.
[26] J. Li, Y. Zhang, X. Chen, Y. Xiang, Secure attribute-based data sharing for resource-limited users in cloud computing, Comput. Secur. 72 (2018) 1–12.
[27] P. Li, J. Li, Z. Huang, C.Z. Gao, W.B. Chen, K. Chen, Privacy-preserving outsourced classification in cloud computing, Cluster Comput (2017) 1–10.
[28] T. Wang, G. Zhang, A. Liu, M.Z.A. Bhuiyan, Q. Jin, A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing, IEEE Internet Things J. (2019), doi:10.1109/JIOT.2018.2870288. 1–1.
[29] T. Wang, G. Zhang, M.Z.A. Bhuiyan, A. Liu, W. Jia, M. Xie, A novel trust mechanism based on fog computing in sensor-cloud system, Future Genera. Comput. Syst.. 10.1016/j.future.2018.05.049.
[30] Q. Kong, R. Lu, M. Ma, H. Bao, Achieve location privacy-preserving range query in vehicular sensing, Sensors 17 (8) (2017) 1829.
[31] M. Li, L. Zhu, Z. Zhang, X. Du, M. Guizani, Pros: a privacy-preserving route-sharing service via vehicular fog computing, IEEE Access 6 (2018) 66188–66197.
[32] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, A. Corradi, Mobeyes: smart mobs for urban monitoring with a vehicular sensor network, IEEE Wireless Commun. 13 (5) (2006) 52–57.
[33] C.E. Palazzi, F. Pezzoni, P.M. Ruiz, Delay-bounded data gathering in urban vehicular sensor networks, Pervasive Mob. Comput. 8 (2) (2012) 180–193.
[34] M. Motani, V. Srinivasan, P.S. Nuggehalli, Peoplenet: Engineering a Wireless Virtual Social Network, in: Proceedings of the 11th annual international conference on Mobile computing and networking, ACM, 2005, pp. 243–257.
[35] B. Placzek, Selective data collection in vehicular networks for traffic control applications, Transpor. Res. Part C-emerg. Technol. 23 (2012) 14–28.
[36] R.H. Hwang, .Y.L. Hsueh, H.W. Chung, A novel time-obfuscated algorithm for trajectory privacy protection, IEEE Trans. Serv. Comput. 7 (2) (2014) 126–139.
[37] K. Kai, W. Cong, L. Tao, Fog computing for vehicular ad-hoc networks: paradigms, scenarios, and issues, J. China Univer. Posts Telecommun. 23 (2) (2016) 56–96.
[38] J.B. Kenney, Dedicated short-range communications (dsrc) standards in the united states, Proc. IEEE 99 (7) (2011) 1162–1182.
[39] I. Damgard, Collision free hash functions and public key signature schemes 304 (1987) 203–216.
[40] T. Wang, J. Zeng, M.Z.A. Bhuiyan, H. Tian, Y. Cai, Y. Chen, B. Zhong, Trajectory privacy preservation based on a fog structure for cloud location services, IEEE Access 5 (2017) 7692–7701, doi:10.1109/ACCESS.2017.2698078.
[41] S. Mohammadi, H. Jadidoleslamy, A comparison of link layer attacks on wireless sensor networks, Int. J. Applica. Graph Theory Wireless Ad Hoc Netw. Sensor Netw. 3 (1) (2011) 35–56.
[42] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun ACM 21 (2) (1978) 120–126.
[43] B. Arazi, Integrating a key distribution procedure into the digital signature standard, Electron. Lett. 29 (11) (1993) 966–967.
[44] K.G. Murty, Linear Programming, John Wiley & Sons, 1983.
[45] S. Reveliotis, An introduction to linear programming and the simplex algorithm, School of Industrial and Systems Engineering, Georgia Institute of Technology.
[46] A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in: SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ICST, New York, NY, USA, 2009.



Yongxuan Lai received his PhD degree in computer science from Renmin University of China in 2009. He is currently an associate professor in Software College at Xiamen University, China. His research interests include network data management, vehicular ad-hoc networks, big data management and analysis. He is an academic visiting scholar at Data and Knowledge Engineering (DKE) Group, the University of Queensland, Australia.

Yifan Xu He is a postgraduate student in Software School, Xiamen University, his research areas include vehicular ad hoc networks.

Fan Yang received his PhD degree in computer science from Xiamen University of China in 2009. He is currently an associate professor at Department of Automation, Xiamen University, Xiamen 361005, China. His research interests are privacy, clustering, and pattern recognition.

Wei Lu Dr. Lu Wei is an associate professor of Computer Science Department at Renmin University. Dr. Lu got his Ph.D degree from Renmin University, Master and Bachelor Degrees from both China University of Geoscience (Beijing). His research interests include Cloud database systems, Spatial and textual data management.

Quan Yu received his Ph.D in School of Computer Science and Engineering Technology from Sun Yatsen University in 2015. He is currently a professor in the School of Mathematics and Statistics at Qiannan Normal University for Nationalities. His research interests include data mining, mobile computing and knowledge representation and reasoning.