Proactive Cache-Based Location Privacy Preserving for Vehicle Networks

Long Hu, Yongfeng Qian, Min Chen, M. Shamim Hossain, and Ghulam Muhammad

Abstract

With the diversification of location-based services in vehicle networks, users can obtain such services through submitting searching locations and points of interest. However, users may worry that their real locations and other privacy information will leak out when they get such services, so appropriate location privacy protection measures are necessary. Traditional location privacy protection, such as K-anonymous, cannot be carried out directly because of the characteristics of vehicle networks, such as high mobility. In order to solve such problems, this article proposes a strategy combining cache strategy with *K*-anonymous that can not only satisfy users' demand on obtaining required services with lowest cost, but also protect the location privacy of users. Specifically, on one hand, cache strategy that deploys part of services on roadside units in advance is used to maximize the satisfaction of users' service requests. On the other hand, each user is set with a *K* value to meet the need of *K*-anonymous. The trade-off of these two aspects guarantees users' services and the location privacy. The experiments show that the strategy proposed in this article is better than other strategies.

Introduction

The continuous upgrading of mobile devices makes the acquisition of locations more and more convenient. When the vehicle needs location-based services (LBSs) [1], such as checking the traffic status of a place or searching the surrounding parking lots, the corresponding service provider needs users to submit the real location that is required. But while such LBS provides convenience for users, it will also infringe on users' privacy because they know the real location of users, or observe the track formed by location changes in continuous time [2].

Therefore, it is necessary to protect the real location of the user and also enable the user to obtain the LBS. The location privacy protection strategies based on LBSs at present are divided into two main categories.

Encryption Mechanisms: The general problem of such methods is that the computation is too large, which is not suitable for mobile devices due to their limited storage and computing resources [3].

Location Obfuscation Methods: This kind of methods usually disturb true locations. For example, the *K*-anonymous technology in application needs to submit *K* locations to a service provider or roadside unit (RSU) [4]. However, because cars' speed is fast, a vehicle making contact with one RSU cannot realize *K* disturbance location services. When users obtain *K* location services directly from the service provider, it will lead to high cost. The quality of service for users will be decreased, resulting in *K*-anonymous techniques not being directly applied in vehicle networking.

When a user requests LBSs in vehicle networks, the implementation of traditional privacy protection methods have some limitations due to the high mobility of vehicles. Several papers [5–8] studied the related security and privacy issues in vehicle networks. In [9], Liu *et al*. raised the thought of enhancing the privacy protection of LBSs by utilizing active caching. To be specific, the RSUs broadcast and cache the contents periodically, and they can check whether or not the cached contents meet the demands before the user needs to send a request to the service provider. This method does not have any protection when the user needs to send a request to the service provider. In addition, this method does not consider the influence of cache on the privacy protection.

Apart from the two types of methods mentioned above, researchers have discussed caching-based location privacy protection methods. For example, Niu *et al*. proposed that when users request LBSs, they can use cache to reduce the number of requests submitted by users to the server, so as to achieve the purpose of privacy protection [13]]. However, this method does not take into account the mobility of the user. When the mobility of the user is high, such as vehicles, the method cannot be applied directly. This article considers location privacy protection mechanisms for vehicle networks, because all the above location privacy protection mechanisms cannot be applied to this architecture directly.

Therefore, based on the proactive cache, we propose an improved location perturbation mechanism, that is, utilizing *K*-anonymity to introduce how to set up caching strategy to meet users' LBS needs, so as to minimize users' requests to service providers. Specifically, the main contributions of this article are:

- We propose a location privacy protection mechanism based on caching for vehicles requesting LBSs in vehicle networks. Specifically, the user can, according to their own needs,

Long Hu and Min Chen are with Huazhong University of Science and Technology; Yongfeng Qian is with China University of Geosciences; M. Shamim Hossain and Ghulam Muhammad are with King Saud University.

Digital Object Identifier: 10.1109/MWC.2017.1800127

1536-1284/18/$25.00 © 2018 IEEE

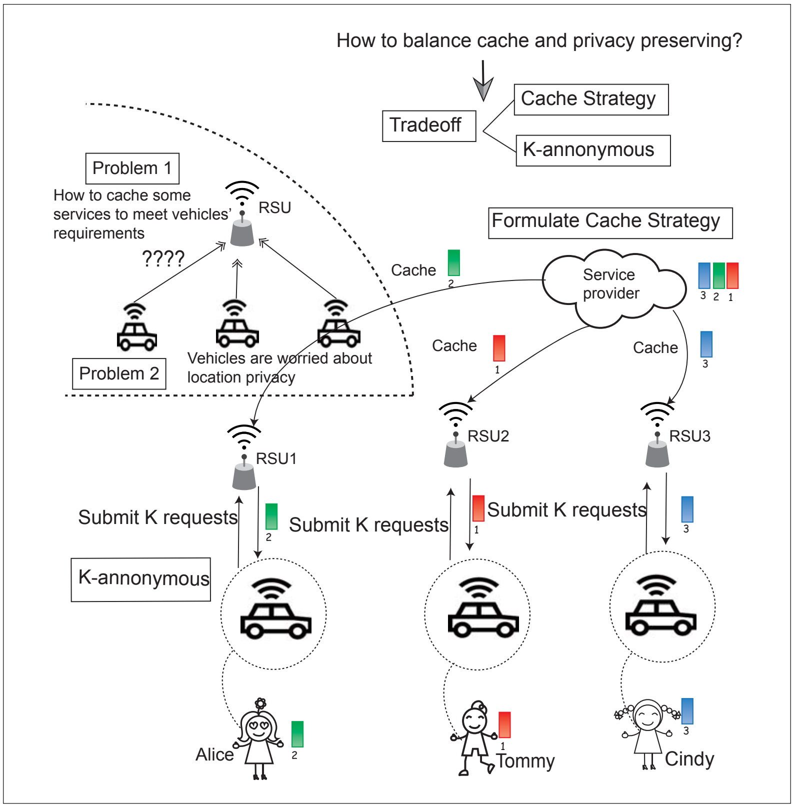

FIGURE 1. Fog-based hierarchical vehicular network architecture.

such as affordable cost and the needed degree of privacy protection, select the appropriate *K* value so that each user can submit *K* disturbance locations. This mechanism allows users to access *K* location service within the specified time, and it can satisfy requirements of vehicle networks, in which the real location of the user is protected.

- The experimental results show that the caching-based strategy proposed is better than other strategies.

The organization of this article is as follows. The following section presents the problem statement and proposed architecture. Then we describe cache-strategy-based location privacy protection. Following that, we present the experimental results and analysis. The final section presents the conclusion of the article.

Problem Statement and Proposed Architecture

This section mainly introduces the problem statement and the fog-based hierarchical vehicular network architecture.

Problem Statement

The service that users request may be repeated, so caching a part of LBSs can help to answer the service requests of users. But when the caching strategy is applied to vehicle networks, there will be the following problems. One important problem is the placement of LBSs' content. As vehicles move fast, how do we ensure that when the vehicle is in need of LBSs, the service content is cached on the RSU to meet with the vehicle? In other words, the problem is how to allocate LBSs' content to make the cache hit rate high. However, when a vehicle requests LBS content from RSUs, if it only requests it once, that is, it directly provides the real location to RSUs, although it saves time and cost, RSUs can still get the user's mobile trajectory and expose the user's privacy. These factors have led to the strategy of studying dynamically caching LBSs content to meet the user's personalized privacy protection.

Fog-Based Hierarchical Vehicular Network Architecture

As shown in Fig. 1, the architecture proposed in this article is fog-based vehicle networking, also utilizing proactive caching strategy. As we can see from this figure, we combine two problems, that is, location privacy protection of vehicles and the caching strategy to satisfy users' services. The reason for this is that when *K*-anonymous is used to protect location privacy, it is a waste to get rid of redundant location services directly. If we can cache these non-real required location services again, we can reduce the cost of users' service requests and improve the service experience of users. Specifically, it is mainly divided into three layers: different moving vehicles form the bottom layer, the middle layer is fog (mainly composed of RSUs), and the highest level is the service provider based on remote cloud. The service provider will actively cache some contents ahead in the fog, and the main functions of each layer are as follows:

•While the vehicles are moving, they may ask for LBSs. Generally speaking, vehicles first request to the RSUs; if the RSU does not have such service, the service requested will be provided by remote-cloud-based service providers.

•RSUs are to cache some services in advance, they can provide services directly to vehicles when requested. At the same time, in the caching strategy designed in this article, when the vehicle requests some kind of LBSs, it needs to submit *K* locations to the RSU. In fact, due to the high mobility of vehicles, a vehicle cannot complete such *K*-anonymous-based service requesting in the contact time with one single RSU, so multiple RSUs' interactions are needed to complete the vehicle service requests.

•Using the powerful computing and storage resources of the remote cloud, the service provider can satisfy the user's arbitrary service request. In this article, we assume users do not want to get services directly from service providers. Because users need to get services through third/fourth generation (3G/4G) in such a case, they will suffer a certain cost. Therefore, we need to design the cache strategy. Users made as few service requests as possible to the service provider, and at the same time, get more services from RSUs directly. In addition, we assume that the acquisition service from RSUs is free, because users and RSUs can connect through WiFi.

In this article, fog units and remote clouds are considered honest but curious. On one hand, fog units and remote clouds can comply with the rules and provide corresponding services according to the service requests submitted by vehicles. On the other hand, they are interested in the vehicle's real location, trying to discover the real location of the vehicle or the vehicle's point of interest (PoI).

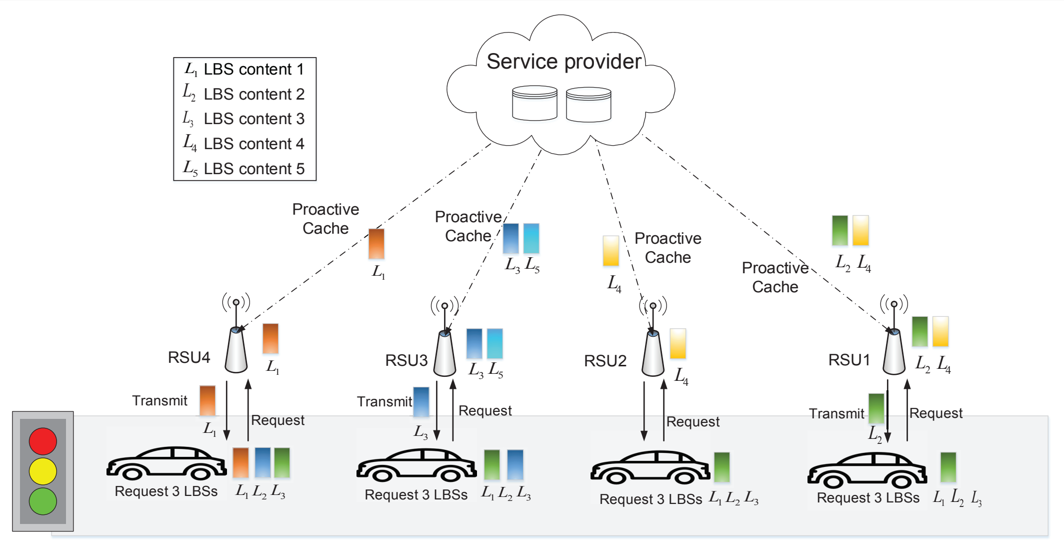

FIGURE 2. Illustration of LPP-cache strategy.

The user can set a deadline (i.e., must complete all service before a certain time) to help design the caching mechanism. It should be noted that we assume that users do not need to pay extra fees to get services from fog, but users need to spend the cost of communication when they connect to remote service providers through the base station.

Cache-Strategy-Based Location Privacy Protection

This section first introduces the cache-policy-based fog hierarchical vehicle networking architecture, and then describes it with mathematical models, and gives the solution using submodal optimization, and finally gives the security analysis.

LPP-Cache Strategy

Location privacy protection cache (LPP-cache) strategy is proposed in this article to protect the vehicles' location privacy, the main idea is, in order to improve the cache hit rate, service providers will proactively cache some contents the user may need in the fog unit. And the user, at the request of LBSs, need to submit K locations to reduce the leakage of the true location at the same time. It is worth noting that when the user selects the K locations, in addition to the location of the real requirement, the user needs to submit the $K - 1$ locations and decide how to choose the remaining K – 1 locations. In this article, we first segment the target area and obtain all the location points in the target area (assuming that there are N). Next, the ratio of the number of requests per request to the total number of requests is recorded as the degree to which the location is required (i.e., popularity). For this reason, location services that are frequently requested by users can be obtained. It may be possible to order the popularity of these locations in order from large to small, taking the location of the first M (of which $M < N$) is often requested, and each user can randomly select $K - 1$ locations from these M positions as their own disturbance locations. The advantage of this is that after each user chooses a more popular position, she can use it later and reduce the probability of repeated requests.

The user can set a deadline (i.e., all service must be completed before a certain time point) to help design the caching mechanism. It should be noted that we assume users do not need to pay extra fees to get services from fog, but users need to spend the cost of communication when they connect to remote service providers through the base station. However, the former may not satisfy users' needs in time, and the latter is able to get users' requests in time. The proposed caching strategy can satisfy users' access to services directly through fog, reducing the number of requests that users request to service providers, not only reducing cost, but also avoiding direct users' location exposure.

The system model presented in this article is shown in Fig. 2. First, fog caches a part of the LBS contents on the RSUs proactively, which means L_2 and L_4 are cached on the first RSU, L_4 is cached on the second RSU, L_3 and L_5 are cached on the third RSU, and L_1 is cached on the fourth RSU. Figure 2 takes the continuous movement of a vehicle as an example, and supposes the real location requested by such a vehicle is L_1. The vehicle submits K locations (here $K = 3$) to the RSU it encounters, namely L_1, L_2, and L_3. In this case, when the vehicle enters the coverage range of the first RSU, that RSU transmits L_2 to the vehicle. The vehicle obtains L_2 from the first RSU and then keeps moving. When it encounters the second RSU, it then submits three locations to the second RSU. But since the second RSU only caches L_4, it has not fed back any information to this vehicle. The vehicle keeps moving and enters into the coverage range of the third RSU. The third RSU feeds back L_3 to the vehicle. Similarly, the fourth RSU feeds back L_1 to the vehicle. So far, the vehicle obtains the three LBSs it requests.

This article raises the hierarchical vehicle network architecture based on fog, which means the RSU contacting vehicles are under centralized management of the service provider at a higher level. A part of LBSs' contents are cached to RSUs in advance. In this way, a vehicle can obtain the contents from the RSU directly when it is requested. To prevent the vehicle's real requesting location from being exposed when it is requesting LBSs to RSUs, in this article we propose that the vehicle submits K locations every time. Since the single time of contacting the RSU is relatively short, we suppose that the vehicle completes obtaining K LBSs through contacting with multiple RSUs multiple times. To be specific, each user submits K locations of the request to the RSU. In this way, the RSU is unable to accurately know which location is

The user submits K locations, but since the time that the RSU contacts the vehicle is relatively short (tens of seconds [10]), it is impractical to complete the service feedback on K locations by only one contact with the RSU. Therefore, our strategy is that the vehicle contacts different RSUs and submits K requests continuously.

the user's true destination, and the probability that the RSU guesses the real location is $1/k$. Every user can define its own k value, and different k values correspond to the user's sensitivity level of privacy protection. For instance, if $k = 1$, it indicates that the user only submits one location, the user's trail will be fully exposed to the RSU, and the user's exact location will be obtained if any malicious RSUs exists. If $k = 10$, it means the probability that the RSU guesses the user's real location is 10 percent. But in this situation, the user needs to submit K locations, which means the user needs to pay the service fee for the K location-based services. This is because the service provider charges some fee for each service it provides. In other words, the more locations the user submits, the higher the extent of privacy protection is, but the fee to be paid is also accordingly higher.

According to the analysis of user behaviors, every user can be analyzed for their approximate activity range. The RSUs within this range can share the contents to serve the user. Suppose that the users in the research area are denoted as u_1, u_2, ... u_n; the RSUs serving the users constitute the set R_i, which are denoted as $R_i = \{R_i^1, R_i^2, ..., R_i^{k_i}\}$. The contents requested by the user will be cached beforehand on the RSU connected to the user. In this way, when the user needs the contents, it will not require asking the service provider directly, but will satisfy the user's requirement through contents cached beforehand on the RSU. Suppose that user u_i encounters the R_i^j, $j = 1, 2, ..., k_i$ opportunistically. The encounter here means u_i is within the coverage range of R_i^j, and they can communicate directly. Since the encounter time between the vehicle and the RSU is relatively short (tens of seconds) [10], it is impractical to suppose that the whole file can be transmitted through just a one-time encounter with the RSU. It is worth noting the following two points:

- The implication of proactive cache is that the service provider caches the contents beforehand to be requested by the user onto the RSU. When the user is requesting, it can directly connect with the nearby RSU to satisfy the requirements of the user. In this way, the significance of previous caching is to reduce the response time of the service requested by the user. This is because the response time of communicating with the RSU near the user is shorter than the response time of sending the request to the content provider on the remote side.
- Since the RSU set service for user u_i is uniformly serving the user, the unified management of these RSUs requires a higher level of scheduling. However, these RSUs can be shared by R_i.

Mathematical Model

The user submits K locations, but since the time the RSU is in contact with the vehicle is relatively short (tens of seconds [10]), it is impractical to complete the service feedback on K locations by only one contact with the RSU. Therefore, our strategy is that the vehicle contacts different RSUs and submits K requests continuously. The RSU can answer part of the requests among the K locations contacted by the user according to its own caching contents. In this way, through the definition of the user's delay tolerability, the user's requests can be answered by as many as possible within the time delay that the user can tolerate.

The defined 0 – 1 symbol $x_{i,j,l}$ represents whether the service content requested by the user u_i is stored on R_j. We suppose that $\tilde{C}_j$ represents the storage capacity of RSU R_j, and $|L_l|$ represents the size of LBS L_l. Therefore, the factor needing to be considered when fog is deploying the cache strategy is that the occupied space of services deployed on each RSU shall be no greater than its storage capacity. The probability density function that the user encounters the RSU within time t is defined as $f(t)$, and its cumulative distribution function is $F(t)$. Therefore, the probability that the user can obtain the service L_1 within time T is as follows: $\Sigma_{t=1}^{T}(1 - F(t - 1))f(t)p(x_{L_1}^t = 1)$, where $p(x_{L_1}^t = 1)$ represents the probability that within time $[0, t]$ the L_1 is cached on the RSU that the user encounters. Similarly, the probability that the L_i can be obtained can be concluded. Supposing that L_i and L_j are independent, it can be concluded that within time T, the probability that the user obtains K services is as follows:

$$\prod_{l=1}^{K}\sum_{t=1}^{T}\left(1-F(t-1)\left(f(t)p\left(x_{L_l}^{t}=1\right)\right)\right), \tag{1}$$

At this time, it is quite clear that if the probability of the user obtaining K services is the maximum before the deadline, it shows that the user does not need to request the service from the service provider so as to save the fees of users. This is because the user's continuous contact with RSUs can directly and rapidly allow it to acquire the contents it demands, and this interactive cost is 0 [10]. If the user cannot obtain the contents through the RSU within the specified time, he/she will need to obtain the contents from the service provider at a higher level through a cellular network. This service request through the cellular network is normally not free of charge, and is defined as c. When a user obtains K LBSs from the RSU, the service fee to be paid is 0. When the user obtains $K - 1$ LBSs from the RSU, the fee for these $K - 1$ LBSs is 0. Therefore, to obtain the last LBS, the fee the user needs to pay is c, so as to complete the K LBSs. Similarly, when a user obtains one LBS from the RSU, the rest of the fee to be paid are $(K - 1)c$; when the user obtains no LBS from the RSU, the fee to be paid will be Kc.The measurement of location privacy protection in LBSs generally contains distortion-based metrics [11] and the entropy-based measurement [12]. This article adopts entropy to measure the privacy protection. The query probability of each LBS is available and can be obtained through Google Map statistics. The user's movement area is divided into$\tilde{M} \times \tilde{N}$, and the query probability of each cell L_i is denoted as $\tilde{p}_i$. If the user submits K locations, namely $L_1, L_2, ..., L_K$, the probability that L_i is the real location requested can be obtained through unit distribution. Similar to Niu *et al.* [13], we can utilize entropy to measure the extent of user's privacy protection, namely

$$H = -\sum_{i=1}^{K} q_i \cdot \log_2 q_i \tag{2}$$

It is observed from the definition of entropy that the greater the H value is, the more difficult the

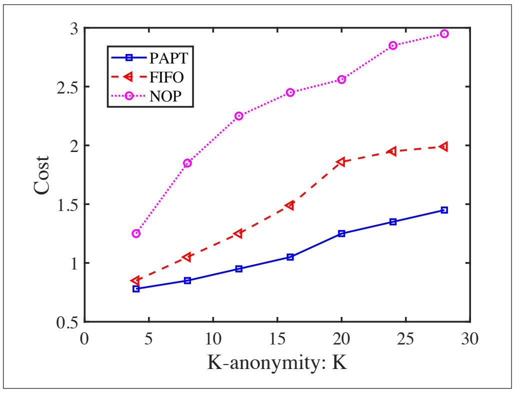

FIGURE 3. Communication cost vs. *K*.

K locations submitted by the user can be distinguished, and the real location can be determined. Therefore, the achieved privacy protection extent is greater. In other words, the other target we need to achieve is to maximize *H*.

For this purpose, we aim to find the tradeoff between the service fees of users and privacy degree to minimize the fees of users and also maximize the privacy protection. We unify the fee *c* to the interval [0, 1], which is expressed as $0 \le c \le 1$. Thus, it can be concluded that the objective function is a combination of $E(c)$ and H utilizing weight coefficient ω, wherein $0 \le \omega \le 1$. The user can define the value of different ω based on requirement. For example, the user takes a relatively greater value of ω, which means the user pays more attention on the lower service cost. On the contrary, if the value of ω is taken a smaller one, it means the user pays more attention on the extent of privacy protection.

So far, we can present our optimization problem:

$$\begin{aligned} &\underset{K, x_{i,j,L}}{\text{maximize}} && f(\cdot) \\ &\text{subject to} && \sum_{l} x_{j,L_l} |L_l| \le \tilde{C}_j \\ & && 0 \le \omega \le 1, \quad x_{j,L} = 0, 1, \qquad K \ge 1. \end{aligned} \tag{3}$$

The objective function denotes that the fee paid by the user is minimized and the entropy *H* is maximized. The constraint condition denotes that the capacity of LBSs cached on each RSU R_j cannot exceed R_j's own storage capacity. The constraint condition is the constraint range of the parameter used in the objective function.

Problem Solving

It can be observed from the expression of the optimization problem (Eq. 3) that the function $f(\cdot)$ is the function related to the variable *K* and 0 – 1 variable x_{i,j,L_k}, namely, $f(\cdot) = f(K, x_{j,L_k})$. In the course of a practical solution approach, we divide it into two steps:

- The user selects his/her own required *K*, which means the user will determine his/her own *K* value according to his/her own tolerability on the extent of privacy protection and the fees to be paid. As previously mentioned, the greater the *K* value is, the higher extent of privacy protection the user will have, but the fee needing to be paid will be higher accordingly.
- After the user selects the *K* value, the function $f(\cdot)$ is related to single variable x_{j,L_k}. At this time, with a determined *K* value, the optimization problem (Eq. 3) is turned into mixed integer nonlinear programming planning. According to the work [14] of Lan *et al.*, such a problem is the NP-hard problem. Next, we solve the remaining problem by utilizing sub-modal optimization.

Security Analysis

During the process of communication between user and RSU, and communication between RSU and service provider, the eavesdropper can be prevented by means of encryption. This article mainly considers the inference attack of RSU and service provider for the service request proposed by the users. First, the RSU can obtain the service requests submitted by the users. However, this article sets *K* service requests submitted when the user interacts with each RSU, to enable the RSU to speculate the probability of real service request of users as 1/*K*. Each RSU can gain the location request and interest point submitted by each vehicle, but we assume that RSUs cannot collaborate with each other, and the service request for each vehicle obtained by each RSU is local information. Moreover, due to the random selection of *K* services by the user, the RSU fails to guess the real request. For the service provider, there are speculation attacks in two aspects. First, the service provider guesses the real request of the user by the RSU request service. Then the information to be gained by the service provider is the service request submitted to the service provider by the RSU but failing to meet the user demands by caching. Thus, the service provider can obtain the request contents of each RSU and accordingly obtain the global service contents, but we assume that the RSU cannot collaborate with the service provider, and the RSU should only send the location and interest point to the service provider during submitting the service request, rather than submitting the corresponding user identity. Therefore, the service provider fails to directly obtain the real request of the user only based on the service request submitted by the RSU. Second, if the service request of the user fails to be fulfilled by the RSU, the user can select to directly obtain by the service provider. Then, when the user pays the service provider for the demand, the request information is also obtained by the service provider. However, except for the real request among *K* services submitted by the user, other services are selected randomly, so the service provider cannot obtain the real request. Moreover, the advanced caching strategy proposed in this article is aimed at reducing the probability of the user directly asking the service provider by caching, and accordingly lowering the risk that the user information is gained by the service provider.

Simulation Experiment

This section first introduces the experiment parameter setting, datasets, and so on, and then discusses the experimental result. Note that the proposed scheme in this article is the PAPT algorithm, and we compare it to the classical first-in first-out (FIFO) algorithm and NOP algorithm, which do not protect location privacy.

During the process of communication between user and RSU and communication between RSU and service provider, the eavesdropper can be prevented by the means of encryption. This article mainly considers the inference attack of RSU and service provider for the service request proposed by the users.

Compared with the traditional method of non use of location privacy protection and the latest FIFO mechanism, the cache based location privacy protection mechanism presented in this article has the best performance. With the continuous increase of K, the mechanism proposed in this article can effectively reduce the service cost and protect the privacy of users.

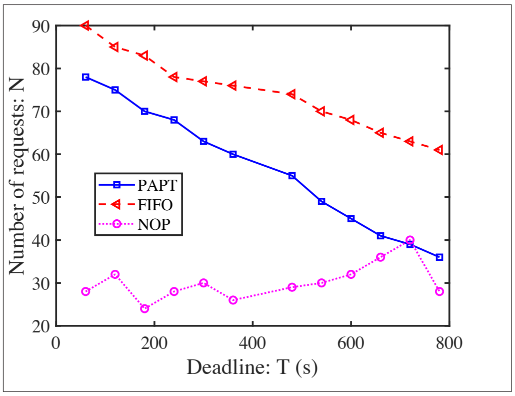

FIGURE 4. Number of requests vs. deadline T.

The datasets used in this article are two kinds. One kind is synthetic datasets, namely supposing that when a vehicle encounters RSUs it obeys the exponential distribution. The other is the real dataset, Diesel [15]. It has recorded the moving route of buses in summer 2006. From June 23, 2006 to July 21, 2006, the specific recording time every day was random, and the statistical time was about 376 hours. We recorded the data concentration throwbox as an RSU. The vehicle connects to the current RSU when it detects that it is in the communication range of the RSU. To explain the Diesel dataset better, we counted that 14 vehicles were connected with RSU and 768 records were generated during these 29 days. As we can see from Fig. 3, compared to the traditional method lacking use of location privacy protection and the latest FIFO mechanism, the cache-based location privacy protection mechanism presented in this article has the best performance. With the continuous increase of *K*, the mechanism proposed in this article can effectively reduce the service cost and protect the privacy of users.

We selected vehicle No. 3120, and recorded the relational graph between the vehicle and RSU 1 connection time. The sub-graph indicates the data of No. 3120 and RSU1 connection time in the dataset Diesel. The following sub-graph indicated the probability density function obeyed by the contact time. From the density function, it could be roughly seen that the vehicle and RSU contact time obeyed the normal distribution. It was just a rough estimate, because the data for the contact between vehicle No. 3120 and RSU1 was only 96 times according to the statistics. We also made the corresponding probability density of contact time for other vehicles such as No.3117, No. 3102, No. 3027, and No. 3116, and discovered that the trend basically obeyed the normal distribution. Thus, the dataset selected by us would be different from the exponential distribution of the contact time obedience assumed previously. From Fig. 4, we can see that when the maximum delay users can tolerate is increasing, users can use cache to get more and more services. It can be seen from the figure that the strategy proposed in this article has more stable performance and can effectively meet the needs of users.

Conclusion

In terms of vehicle networks, when a user requests LBSs, to avoid that user's location being tracked, we propose a privacy protection mechanism based on proactive caching and *K*-anonymity. Specifically, the user submits *K* locations at one time when requesting the service. In this way, the LBS the user really needs can be hidden. However, in consideration of the condition when the user needs to pay fees, we utilize RSUs to cache some services in advance. In this way, the contents can be obtained directly through communication with RSUs when the user needs them. Therefore, we formulate an optimization scheme concerning privacy protection and fees, we analyze it through sub-modular optimization, and finally do the problem solving through a greedy algorithm.

Acknowledgment

The authors are thankful to the Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia, for funding through the Research Group Project no. RG-1436-023. M. Shamim Hossain is the corresponding author.

References

[1] K. Hwang *et al.*, *Big Data Analytics for Cloud/IoT and Cognitive Computing*, Wiley. ISBN: 9781119247029, 2017.

[2] Y. Zhang *et al.*, "SOVCAN: Safety-Oriented Vehicular Controller Area Network," *IEEE Commun. Mag.*, vol. 55, no. 8, Aug. 2017, pp. 94–99.

[3] J. Chen *et al.*, "Blind Filtering at Third Parties: An Efficient Privacy-Preserving Framework for Location-Based Services," *IEEE Trans. Mobile Computing*. DOI: 10.1109/TMC.2018.2811481, 2018.

[4] Y. Gong *et al.*, "Protecting Location Privacy for Task Allocation In Ad Hoc Mobile Cloud Computing," *IEEE Trans. Emerging Topics in Computing*, vol. 6, no. 1, 2018, pp. 110–21.

[5] X. Du *et al.*, "An Effective Key Management Scheme for Heterogeneous Sensor Networks," *Ad Hoc Networks*, Elsevier, vol. 5, no. 1, Jan. 2007, pp 24–34.

[6] Z. Zhou *et al.*, "Prometheus: Privacy-Aware Data Retrieval on Hybrid Cloud," *Proc. IEEE INFOCOM 2013*, Turin, Italy, Apr. 2013.

[7] Y. Cheng *et al.*, "A Lightweight Live Memory Forensic Approach Based on Hardware Virtualization," *Elsevier Information Sciences*, vol. 379, Feb. 2017, pp. 23–41.

[8] M. Chen *et al.*, "Cognitive Internet of Vehicles," *Computer Commun.*, vol. 120, May 2018, pp. 58–70.

[9] B. Liu *et al.*, "Silence Is Golden: Enhancing Privacy of Location-Based Services by Content Broadcasting and Active Caching in Wireless Vehicular Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 12, 2016, pp. 9942–53.

[10] M. Chen and Y. Hao, "Task Offloading for Mobile Edge Computing in Software Defined Ultra-dense Network," *IEEE JSAC*, vol. 36, no. 3, Mar. 2018, pp. 587–97.

[11] R. Shokri *et al.*, "A Distortion-Based Metric for Location Privacy," *Proc. 8th ACM Wksp. Privacy in the Electronic Society*, 2009, pp. 21–30.

[12] A. R. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Pervasive Computing*, vol. 2, no. 1, 2003, pp. 46–55.

[13] B. Niu *et al.*, "Enhancing Privacy Through Caching in Location-Based Services," *Proc. IEEE INFOCOM 2015*, 2015, pp. 1017–25.

[14] R. Lan *et al.*, "Device-to-Device Offloading with Proactive Caching in Mobile Cellular Networks," *Proc. IEEE GLOBECOM 2015*, 2015, pp. 1–6.

[15] J. Burgess *et al.*,"CRAWDAD Dataset Umass/diesel (v. 2008-09-14)"; http://crawdad.org/umass/diesel/20080914, Sept. 2008.

Biographies

Long Hu (hulong@hust.edu.cn) has been a lecturer in the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), China, since 2017. He was a visiting student at the Department of Electrical and Computer Engineering, University of British Columbia from August 2015 to April 2017. His research includes the Internet of Things, software defined networking, caching, 5G, body area networks, body sensor networks, and mobile cloud computing.

Yongfeng Qian (yongfengqian@ieee.org) is a specially assigned associate professor in the School of Computer Science at China University of Geosciences. She obtained her Ph.D. degree from the School of Computer Science and Technology at Huazhong University of Science in June 2018. She got her M.S. degree from the School of Mathematics and Statistics, HUST, in 2015. Her research interests include software defined networking, security and privacy, cloud computing, and the Internet of Things.

Min Chen [SM'09] (minchen2012@hust.edu.cn) has been a full professor in the School of Computer Science and Technology at HUST since February 2012. He is Chair of the IEEE Computer Society STC on Big Data. His Google Scholars Citations have reached 14,200+ with an h-index of 58. He received the IEEE Communications Society Fred W. Ellersick Prize in 2017. His research focuses on cyber-physical systems, IoT sensing, 5G networks, SDN, healthcare big data, and so on.

M. Shamim Hossain [SM'09] (mshossain@ksu.edu.sa) is a professor with the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He has authored or co-authored more than 200 publications. He is the recipient of an ACM TOMM Nicolas D. Georganas Best Paper Award. He currently serves on the Editorial Board of *IEEE Multimedia*, *IEEE Network*, and *IEEE Wireless Communications.*. His research focuses on social media, the Internet of Things, cloud and multimedia for healthcare, wireless cloud networking, and smart health.

Ghulam Muhammad (ghulam@ksu.edu.sa) is a professor in the Department of Computer Engineering, College of Computer and Information Sciences at King Saud University. He received his Ph.D. in electrical and computer engineering from Toyohashi University and Technology, Japan, in 2006. His research interests include deep learning, and image and speech processing. He has authored and co- authored more than 200 publications including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters.