Privacy-preserving Task Scheduling for Time-sharing Services of Autonomous Vehicles

Mohammad Hadian[1], Thamer Altuwaiyan[1], Xiaohui Liang[1], and Haojin Zhu[2]
[1]Department of Computer Science, University of Massachusetts Boston
[2]Computer Science & Engineering Department, Shanghai Jiao Tong University
Email: {mshadian,thamerfa}@cs.umb.edu, xiaohui.liang@umb.edu, zhu-hj@cs.sjtu.edu.cn

Abstract—**Sharing of autonomous vehicles between multiple users can potentially be the ultimate solution for increasing the efficiency of the transportation system. In a time-sharing scenario, AV owners share their vehicles to others at their unwanted times without incurring any human efforts. However, such sharing service requires the disclosure of users' locations and route information, raising severe privacy concerns and issues. In this paper, we propose a privacy-preserving task scheduling scheme for time-sharing services of autonomous vehicles. First, we design a matching scheme that finds the feasible requesters for each available AV. Then we propose a scheduling scheme using different approaches for assigning requesters to the AVs based on different system parameters. Both schemes work efficiently without requiring users to share their exact locations and route details while maximizing the AV owner profit and minimizing the requester cost. Specifically, our schemes enable an untrusted matching server to match owners and requesters using a set of Intermediate Destination (ID) locations in the travel paths. Only if the service can be given to the requester efficiently, the owner and the requester share the details of the routes. All the calculations for verification of the feasibility of the service are done on the untrusted server. We show the accuracy of our proposed scheme through extensive simulations on real data. The results confirm that our traffic-based ID selection scheme, with consideration of the traffic information and patterns in the area, outperforms the baseline scheme where the IDs are chosen randomly. Furthermore, the effectiveness of different scheduling schemes including greedy-based and first come first served are evaluated.**

I. Introduction

Technological advancements are brought to the automobile industries recently, to bring automation and computerization to car driving [1]–[5]. Autonomous Vehicles (AV) can dramatically reduce crashes, assist and manage traffic flows, reduce travel durations and energy consumption, provide critical mobility to the elderly and disabled. It can also make vehicle sharing convenient, popular, and necessary to fundamentally change the transportation systems. The technological hardware and software needed for AVs include the addition of new sensors, communication and guidance technology, and software for each automobile. These equipment are costly and significantly affect the price of these cars which hamper large-scale production and availability of the AVs. For example, Light Detection and Ranging (LIDAR) systems used in many of today's AVs cost $30,000 to $85,000 [6]. Sharing of AVs [7], [8] is an interesting and promising solution to increase vehicle usage efficiency and alleviate the cost problem by making the AV profitable for the original owner and dividing the cost over multiple users.

Availability, time and effort of a human driver are necessary for human-driven cars being transferred from one location to another. This results in families often requiring multiple cars, which are not effectively used all the times by their users and are often parked for long time periods in garages and parking lots. A study indicates cars are parked 95% of the time [9]. This situation will be significantly changed by autonomous driving becoming popular and practical among the users. In a time-sharing scenario, after an AV finishes one user's task, it drives away to accomplish other tasks without the need of any extra human effort. A requester task is defined as the pickup and drop-off locations and their times. Similarly, an owner's task is defined as the time and location of availability of the AV and the time and location the AV is needed back after finishing the requester's tasks. With this service, the high cost can be shared by a group of AV users. However, to coordinate sharing services, users need to disclose their origin and destinations and times of their commute [10]. Certainly, the coordination of AV usage will acquire very sensitive detailed information not only about people's current locations but also future locations. Locations can include residences, places of employment, places of amusement, common routes of travel [11]–[14].

Leaking the location information has direct and clear negative impacts for users [15]–[18], e.g., information about the time users leave and return to their places of residence can be useful for thieves and can cause harm to the users. Furthermore, information about visits and frequency of visits of users to healthcare facilities such as hospitals, clinics, and drug stores can be used to effectively infer the user's state of health, and places of dining and entertainment reveal user's personal preferences. The most common location privacy preservation techniques such as location obfuscation and location generalization are not effective enough in the scenario of sharing the pickup and drop-off locations for routing, because a small change in the location coordinates can result in highly different routes to be selected as the travel path due to the traffic and route conditions around the locations. As another problem with these techniques, in many cases, the privacy preservation achieved is not adequate because by intersecting the generalized or randomized locations with the location with the highest probability of user presence, i.e., the points of interest, the user locations are inferrable.

In time-sharing services, the scheduling scheme needs to efficiently calculate the travel durations and distances from the AV location to the requester's location, then to the requester's destination and finally back to the owner's current location. In this paper, we investigate the time-sharing service of autonomous vehicles from the usability and location privacy preservation perspectives and propose a novel privacy-preserving task matching and task assignment scheme over encrypted data. Usability in this work is referred to as the ability of the scheme to accurately find and match the feasible requesters and AV owners. While focus of our work is on the technical necessities of implementing a secure sharing service, legal agreements and policies of services between the owners and requesters required for a safe and trustworthy service are beyond the scope of this work, and apply to all similar sharing services such as Uber and Lyft. The task matching scheme matches the AV owners and requesters without knowing their exact routes and times of travel. Thus, their exact locations cannot be inferred by the server. We propose an Intermediate Destination (ID) assisted task matching technique [19], which instead of using the users' real location information, use the travel times and distances from these locations to a set of pre-defined hotspot locations for matching [20].

• First, we design and implement a privacy-preserving task matching scheme over encrypted travel information, which matches the AV owner and the requester. The scheme checks complying owner/requester pairs and estimates a cost of service value for each, without knowing and disclosing any of the user's sensitive location information. To do so, we consider two different cases of random and traffic-based selected IDs.

• Second, we investigate different approaches for scheduling the tasks and assigning them to the AVs, including cost and profit based greedy approaches and first come first served. These different approaches chosen based on the conditions and requirements of the system are necessary for the efficient and optimized performance of the scheme, which cannot be done using existing optimization techniques due to highly dynamic parameters of the environment, such as vehicular traffic, road blocks and road conditions.

• Third, we obtain the accuracy and efficiency analysis results through simulation on real traffic data collected from Google Maps API. We evaluate the effect of different system settings and parameters on the performance of the system, such as average travel time, distance and profit. The results confirm that our traffic-aware ID selection scheme significantly outperforms the baseline scheme (with ID locations chosen randomly) from the matching accuracy perspective, i.e., provides results closer to the ground truth. The results also confirm the detection of conflicting tasks with a high accuracy resulting in a 0.05% to 0.12% false negative rate. Furthermore, a trade-off between the accuracy and efficiency of the scheme is observable from the results. Evaluation of the scheduling scheme confirms the effect of different scheduling schemes on different cost metrics.

II. Related works

A. Location privacy

Location Privacy-Preserving Mechanisms (LPPMs) have been studied in many recent books and surveys [21], [22]. Obfuscation is generally used in these mechanisms, e.g., spatial cloaking, cell merging, location precision reduction or dummy cells to achieve anonymity and uncertainty based privacy preservation. Based on Sweeneys concept of k-anonymity for data privacy [23], [24] introduced k-anonymity for location privacy. In this approach, instead of pseudonymously reporting their exact locations, users report a region containing k-1 other people. [25] examines the technique of adding multiple false locations to a true location report, and speculate on how to make the false locations realistic and to reduce the cost of the inevitable extra communication. [26] introduces the idea of obfuscation for location privacy, formalizing the concepts of inaccuracy and imprecision as examples. Also, Beyond the obfuscation of spatial information, [27] considers spatiotemporal obfuscation to protect movement trajectories of users. The recent work [28] proposed a notion of geo-indistinguishability which extends differential privacy. Several works use Markov models for modeling users' mobility and inferring user locations or trajectories [13], [29]. [30] proposed an insightful technique with a provable privacy but it used suppression instead of perturbation. Furthermore, in case of using these methods for a routing scheme, similar to ours, due to road and traffic conditions in the area, the accuracy will suffer significantly. However, our proposed scheme provides exceptionally high accuracy by using a smart selection of IDs.

B. Privacy-preserving profile matching

Privacy-preserving profile matching protocols [31]–[34] are used to check the similarity of two profiles without disclosing them. [35], [36] have developed profile matching with full anonymity. However, location privacy-preserving systems and AV sharing schemes have never used these schemes previously.

Homomorphic encryption enables addition/subtraction operations in the cipher-text domain. This feature makes them widely used in many applications [37], [38]. Additionally, to enable searching encrypted data, keyword search techniques have been recently developed as a fundamental approach [39]. Improvement of search functions, e.g., multi-keywords, range-query, ranking, relevance scores, and top-k retrieval are the main focus of these approaches. [40] has developed a practical and efficient multi-keyword search scheme that can support complicated logic search using the mixed "AND, OR and NO" operations of keywords. [41] has introduced a new cryptographic primitive, called proxy re-encryption with keyword search: if a pre-defined keyword "urgent" is found in the encrypted data, the proxy re-encryption will be executed on the encrypted data such that others may access the data in an emergency case. However, none of these techniques are used in task matching in the time sharing scenarios.

III. System Model

We consider a scenario in which the AV owners use the scheme to advertise to share their AVs with others during the times they do not use them. The requesters on the other side, use the scheme to request the AV. Neither of the two sides needs to share their location information and details of their travel. If the AV can be available for the requester, the matching scheme will consider whether or not these two tasks are matching. The scheme uses different scheduling approaches to maximize its performance and efficiency, based on different parameters.

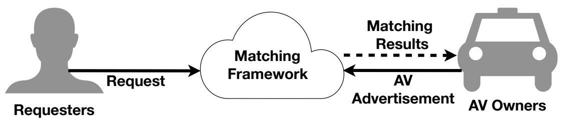

Fig. 1: System Model

A. System components

Our proposed matching and scheduling scheme consists of five components:

• AV Owners (O): Owners of the AVs who want to share them. The owners are responsible for advertising the AVs and indicating the time and location of where the AVs would be available for giving service, as well as time and location where the AVs are needed to be back after finishing their tasks.

• Requesters (R): Passengers who request for this service to use the available AVs. The requesters also indicate the time, origin and destination of their trips using their local devices.

• Matching server (S): The semi-trusted server which finds and matches the suitable and complying owners and requesters based on the encrypted details of their availability and transfer needs, considering the public traffic information. The matching server uses different approaches for performing the matching between the requester and the owner tasks.

• Scheduling server (S_S): After matching between the requester and AV tasks is done by the matching server, and feasible requesters are detected for each AV, now its time for the scheme to assign them to each other. The scheduling server, considering different given parameters, is in charge of performing the scheduling with respect to maximizing the profit given to it as arguments.

• Public traffic server (S_T): The traffic server providing information regarding routes and paths and the vehicular traffic information, such as Google maps and other navigation applications and software. Due to the privacy requirements of the owner and requester locations, this server is used by the owner and requester devices directly and is not controlled and managed by the matching scheme.

B. Design goals

• Privacy-preserving matching of requesters and owners: The server in our scheme is trusted to perform the operations required for matching AVs and requesters correctly and completely. However, it is untrusted with the users' location information. Based on this setting, location and time information of the neither of the users can be shared, and this cannot be done simply by de-identification of the users. Because locations themselves which may be places of residence or business can highly identify the users. Besides the server, location privacy of the users must also be protected from each other. Thus, the requester information will only be shared with the owner if they both have confirmed the service.

• Accuracy: Process of detecting the feasible requesters by the matching server should be done accurately, resulting in a low False Positive rate and False Negative rates. This can particularly be challenging since the actual and exact locations of the users are not provided to the server.

• Customized scheduling: The scheme should be able to accept different parameters, provided by the system admin, based on environment and admin preferences, and perform the scheduling of the matching requester and AV tasks, based on these parameters. This allows the scheme to remain efficient for different environments and different settings, with different definitions of cost and profit.

• Computation and communications efficiency: Matching of the tasks needs to be done efficiently. This means highly computational operations are expected to be mostly done on the servers, with high computational powers. On the other hand, the communications and transfer of the information between the users and the server should remain minimum.

C. Trust model

• Matching and scheduling server: The server semi-trusted in our scheme, i.e., the location and time information of the AVs and requesters are not shared with the server, but the task matching and scheduling operations are considered to be performed correctly by the server.

• Traffic server: Similar to the matching server, the traffic server is trusted with providing accurate information about the travel routes and times between the given locations. These routing operations are done locally on the users' devices and thus do not incur additional privacy concerns for the service.

• Owners and requesters: The users, i.e., requesters and owners are not considered trusted, and do not share any information with each other. Only when the service is finalized, requester's location and travel information are sent to the AV for completing the service.

IV. Proposed scheme

In this section, we introduce our proposed privacy-preserving time-sharing task matching and scheduling schemes, including security and privacy-preservation techniques and their details for maximizing profit or minimizing costs of the system.

A. Overview of the scheme

The overview of our proposed framework is shown in Fig. 2.

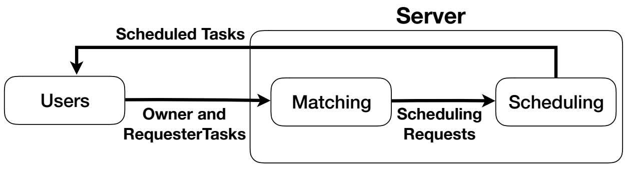

Fig. 2: Overview of the Scheme

The matching and scheduling services are done entirely on the Cloud where the AV owners and requesters use their Internet-connected devices to use them. The key privacy point in this scheme is related to the fact that the matching server does not receive the location information from any of the users, and the only information sent to the server is the encrypted travel time and distances, from users' locations to a set of pre-defined intermediate destinations (ID). These time and distances are privately calculated by the users devices, using a routing and traffic service such as Google Maps, Waze, etc. The matching server then using these information checks a series of matching criteria to determine if two owner and requester tasks are matching. Finally, based on different scheduling schemes, the server chooses the most efficient requester task to be assigned to an AV owner, and vice-versa. While the parameters of the scheduling are set based on the preferences of the owners, the scheduling scheme simplifies the role of the owners by automatically and accurately selecting the most profitable requester for each of the owners.

B. Task matching scheme

Consider a scenario where the AV owners advertise their AVs to share with others at the times they don't need them. To do so, instead of revealing the actual times and locations of the AVs, which are considered private, the scheme automatically uses the location of some "Intermediate Destinations", e.g. public places, restaurants, etc. and calculates the travel durations and distances from AV locations to these points using publicly available traffic information, and uses these times for the matching. Each AV task is created based on two locations and one time value; the current location of the AV (L), the location which the owner needs it back (L'), and the time at which the owner needs the AV back at that location (T'). We use a triple (L, L', T') to represent each AV. This triple is then processed locally in the traffic and routing server on owner's device, and a set of non-privacy-sensitive values are generated using the set of IDs to be sent to the matching server. These values consist of four main categories:

• Times of arrival of AV to each of the ID locations, calculated based on the current location of the AV, and travel time from that location to the ID location. We use the notation $AV_i.T_j$ to represent the time of arrival of AV i to ID j.

• Times of required availability of the AV at the ID location, in order for the owners to have them back at the required time and location. This time is calculated using the ID location and the time and location which the AV is needed to be back. These times are denoted as $AV_i.T'_j$ for AV i and ID j.

• Travel distance from the current location of the AV to all of the ID locations; denoted by $AV_i.D_j$ for AV i and ID j.

• Finally, the travel distance from all of the ID locations to the AV's final location, where the owner needs it back; denoted by $AV_i.D'_j$ as the distance from ID j to AV i's final location.

This set is called the "AV task" and denoted by

$$av_i = \bigcup_{j=1}^{d} \{AV_i.T_j, AV_i.T'_j, AV_i.D_j, AV_i.D'_j\}$$

for all ID locations j in the system.

All these times and distances are then encrypted using the AV's public key (k_i for AV i) and are sent to the matching server. We denote the encrypted data m using k_i by $E_i(m)$, for example $E_i(AV_i.D_j)$ is the encrypted distance from AV i's current location to the ID j.

On the other side, the requesters, who are willing to rent a shared AV also use details of their request, including pickup time ($\bar{T}$) and pickup and drop-off locations ($\bar{L}$ and $\bar{L}'$) of their trip. First these privacy-sensitive information are locally converted into non-sensitive information using the traffic and routing information server, using the ID locations as intermediate destinations. Similar to the AV owner case, output of this step, which will be sent to the matching server, consists of a set of travel times and distances as follows:

• Travel time from requester's current location, i.e. origin of his/her trip to each of the ID locations. We use the notation $R_i.T_j$ to represent the travel time for requester i to ID j.

• Times at which the AV would be available at the ID locations after finishing the requester tasks. This time is calculated using the ID location and the location of requester's destination. These times are denoted as $R_i.T'_j$ for requester i and ID j.

• Travel distance from the current location of the requester. i.e. origin, to all of the ID location. For requester i and ID j, these distances are denoted by $R_i.D_j$.

• Finally, the travel distance from requester's drop-off location to the all of the ID locations; denoted by $R_i.D'_j$ as the travel distance from requester i's destination to ID j.

We refer to this set of information as the "requester task", and for all ID locations j, denote them by:

$$req_i = \bigcup_{j=1}^{d} \{R_i.T_j, R_i.T'_j, R_i.D_j, R_i.D'_j\}$$

All these information are encrypted before being sent to the server. Requester first gets a set of public encryption keys of all of the AVs in the system, i.e $k_1, k_2, ...k_n$, by making a request to the matching server, then uses these keys to encrypt their data and send them to the matching server. Specifically, each requester uploads a copy of his/her task information encrypted by all of the AV keys, i.e. $E_1(req_i), E_2(req_i), ...E_n(req_i)$ are uploaded by requester i.

In order to make the encryption and matching more efficient, our scheme uses an estimating technique to remove the less likely IDs from the list of the available IDs. Thus a fewer number of encryptions and calculations is required by the

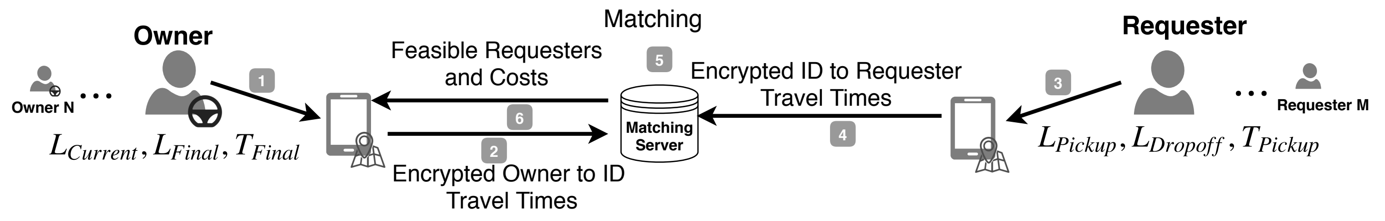
Owner
Owner N
$L_{Current}, L_{Final}, T_{Final}$
1
Matching
Feasible Requesters and Costs
5
6
2
Encrypted Owner to ID Travel Times
Matching Server
Encrypted ID to Requester Travel Times
4
3
Requester
Requester M
$L_{Pickup}, L_{Dropoff}, T_{Pickup}$

Fig. 3: Overview of the Task Matching Scheme

matching server to be performed. Details of the efficiency improvement techniques are presented later in section VI.

The server matches the encrypted AV and requester tasks and returns to the owner a set of encrypted cost values associated with each requester based on the AV and requester tasks. The owner then decrypts and first finds the set of feasible tasks for itself, then sends back all the decrypted costs associated to each requester to the server. Server uses these decrypted cost values for scheduling the owner and requester tasks. The information sent back to the server include the path with lowest cost, i.e. through the most efficient ID, for each requester which the AV can give service to. Specifically, the AV sends to the server a scheduling request in the following format:

$$sch\_req = \bigcup(i, j, j', t, t', d, d') \text{ for all feasible requester } i,$$

where j and j' are the most efficient ID for first and second trips, incurring in costs c and c' respectively. Please note that without loss of any generality, we can use either travel time or distance or both, as the cost for each trip. Finally, the matching server uses different scheduling schemes to select the most profitable scheduling requests from all of the AVs, and will send details of the matched tasks to the respective requester and AV owner.

In this scheme, as depicted in Fig. 4, after matching a requester with an owner and finishing the requester task, the AV will return to the owner. The details of each step in this scheme are as followed:

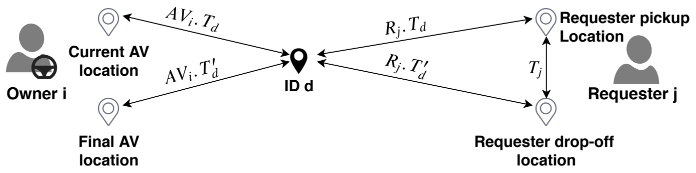

Fig. 4: Matching Criteria for Basic Scheme

1) Advertising AV: First the owner i is advertising her AV with sharing the time (T') and location (L') where she needs it back along with current location of the AV (L) using her personal device (step 1). The personal device then using a Traffic Information Service (S_T), e.g. Google Maps Server, having the owner's location information (L, L', T') along with locations of a set of Intermediate Destinations (Hotspots) $(ID_1...ID_d)$, derives the travel durations from these ID locations to the owner's locations $(AV_i.T_1...AV_i.T_d)$. It then encrypts these times with owner's public encryption key (k_i) and shares the encrypted times $(E_i(AV_i.T_1)...E_i(AV_i.T_d))$ along with the key with the matching server (step 2).

TABLE I: Notations

AV's current location	L
AV's final location	L'
Time of AV's final location	T'
AV information sent to traffic server	(L, L', T')
AV i's Task	av_i
AV i's public key	k_i
Encrypted data using k_i	E_i
Requester pickup time	$\bar{T}$
Requester pickup location	$\bar{L}$
Requester drop-off location	$\bar{L}'$
Requester information sent to traffic server	$(\bar{T}, \bar{L}, \bar{L}')$
Time of arrival of AV_i to ID_j	$AV_i.T_j$
Time of required availability of AV_i at ID_j	$AV_i.T'_j$
Distance from AV_i's current location to ID_j	$AV_i.D_j$
Distance from ID_j to AV_i's final location	$AV_i.D'_j$
Travel time of R_i to ID_j	$R_i.T_j$
Travel time from R_i's destination to ID_j	$R_i.T'_j$
Distance from R_i's current location to ID_j	$R_i.D_j$
Distance from ID_j to R_i's final location	$R_i.D'_j$
Requester i's task	req_i
Scheduling request	sch_req
Requester i's travel time	T_i
Requester i's travel distance	D_i
Current Time	T_c

2) Requesting AV: The requester j on the other side, requests an AV with sharing her needed pickup and drop-off locations $(\bar{L}, \bar{L}')$ along with her pickup time $(\bar{T})$ with her personal device (step 3). Her personal device first makes a request to the server for all available owner encryption keys and receives a set of public keys $(k_1...k_N)$ from the sharing server. Similar to advertising step, requester's personal device then uses the S_T to get travel durations to ID locations $(R_j.T_1...R_j.T_d)$ using her location information $(\bar{L}, \bar{L}')$ along with ID locations $(ID_1...ID_d)$. The device replicates these times for each of the available owners, then encrypts them using available public keys $(k_1...k_N)$ and sends the encrypted times $([E_1(R_j.T_1)...E_1(R_j.T_d)]...[E_N(R_j.T_1)...En_N(R_j.T_d)])$ to the matching server (step 4).

3) Matching server: The matching server performs the task matching after receiving the AV and requester tasks (step 5). Server performs the following addition and subtraction operations and returns the result to the owner i to help determine if requester j is able to be matched with her/him, using ID d

for first trip and ID d' for the second (return) trip. A positive results will indicate conflicting requester (check Fig. 4). These operation are done for all requester-owner pairs in the system.

$$Result_1 = E_i(T_c) + E_i(AV_i.T_d) + E_i(R_j.T_d) - E_i(\bar{T}).$$

This result being negative guarantees that AV is able to be at the requester's location on time, where T_c is current time, and;

$$Result_2 = E_i(T_c) + E_i(AV_i.T_d) + E_i(R_j.T_d) + E_i(T_j) + E_i(AV_i.T_{d'}) + E_i(R_j.T_{d'}) - E_i(T').$$

This result being negative shows that the AV is able to be back at the owner's location on time after dropping-off the requester, where T_j is requester j's trip time.

Other than these two values, which determine the travel times needed for each matched AV and requester task, we also need to find the travel distances, which will later be used in scheduling schemes to find the most efficient tasks. The server calculates the value for encrypted transitional trip distance as follows:

$$D_{Trans} = E_i(AV_i.D_d)+E_i(R_j.D_d)+E_i(AV_i.D_{d'})+E_i(R_j.D_{d'}).$$

These values along with the encrypted requester distance ($E_i(D_j)$), provided by the requester, is then sent to the AV owner for decryption. Finally, the owner after indexing the requesters randomly, sends a set scheduling requests (sch_req), each associated with one of the feasible requesters to the server for scheduling. Since the indexing of the scheduling requests are random and only known to the owner, the server cannot associate each request to the requester and can only send back to the owner an index after scheduling, as the index of the requester to be services by the owner.

C. Homomorphic encryption

Homomorphic encryption provides the addition and multiplication operations over cipher-texts, i.e. heavy operations can be performed by untrusted parties without knowing the shared secret. This method is widely used in computation on privacy-sensitive content. The Paillier Cryptosystem [42] which achieves homomorphic properties consists of three algorithms:

• Key Generation: Given the security parameter K_1, two large prime numbers p_1, q_1 are first chosen, where $|p_1| = |q_1| = K_1$. Then, the RSA modulus $n = p_1q_1$ and $\lambda = lcm(p_11, q_11)$ are computed. Define a function $L(u) = \frac{u1}{n}$, after choosing a generator $g \in Z^*_{n^2}, \mu = (L(g^\lambda \mod n^2))^{-1} \mod n$ is further calculated. Then, the public key is $pk = (n, g)$, and the corresponding private key is $sk = (\lambda, \mu)$.

• Encryption: Given a message $m \in Z_n$, choose a random number $r \in Z^*_n$, and the ciphertext can be calculated as $c = E(m) = g^m r^n \mod n^2$.

• Decryption: Given the ciphertext $c \in Z^*_{n^2}$, the corresponding message would be $m = D(c) = L(c^{\lambda \mod n^2}).\mu \mod n$.

D. Scheduling schemes

Other than finding feasible and matching tasks, assigning them to the most suitable counterpart is also important and has a key role in an efficient and usable system. With the input of set of all AVs and their matching requester tasks, the main purpose of the scheduling scheme is to assign best requester tasks to each AV based on different preferences of the system. Note that the creating and feasibility checking of the tasks are done by the privacy-preserving matching scheme. Thus, the input to the scheduling scheme has already gone through the privacy-preservation process. Tasks to be sent to this scheme include time and distances of travel for each of feasible tasks for each Av using the best and most efficient ID. Generally, scheduling can also be done on tasks created by exact travel locations and times for both requester and AVs, if the location privacy is not important to be preserved. Effect of privatization of the user information on scheduling will be different for each of the approaches and will be discussed in section V.

In this scheme, after the cost of service to each of the requesters by feasible owners is calculated by the matching server, now these cost values are used to determine the most efficient task to be performed by each owner, and to assign the requester tasks to the owner who can perform it with the lowest cost. Based on the conditions of the system and environment, and the preference of users, different scheduling methods may be user. For instance, in labor-expensive environment, a greedy approach considering the smallest travel time of the AV is preferred, where as in some environmentally-restricted locations, the shortest travel distance would be considered using the greedy approach. Alternatively, some systems may aim to minimize the wait time of the owners or requesters for being assigned a task, thus choose a "first come, first served" approach. Details of each approach is presented as follows:

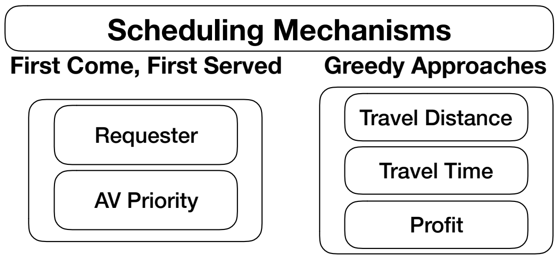

Fig. 5: Overview of Scheduling Schemes

1) Greedy: With the aim of minimizing the cost and maximizing the profit for either of the users or the collective system of the AVs, three different metrics are considerable:

• Total travel distance: in the locations where vehicular traffic is preferred to be minimal, such as environmentally-restricted cities, or fuel consumption-restricted areas, this metric is preferred. In this case, for each requester task to be addressed by an owner, the total travel distance of the AV, including transitional and profitable trips is considered as the cost and the aim of the scheduling scheme is to minimize this cost. With this approach, every requester is assigned to the AV which can perform the task with lowest total distance traveled. Amongst all the scheduling requests arrived to the matching server, it calculates the total travel distances for all of the AV and requester pairs in the system, and the pics the minimum as the best scheduling, resulting in the minimum total cost for the system. Specifically, following equation needs to be solved:

$\min\{\sum_{i=1}^{N}\sum_{j=1}^{M}(d+d')\}$ for all scheduling requests for AV i with feasible requester j.

• Total travel time: similar to the previous case, the total travel time of the AV may be important in some systems, where availability and accessibility of the AV transportation system is required and preferable. In such cases, the AVs are preferred to be done with the assigned tasks sooner, thus they can perform more tasks and handle more requesters, resulting in more availability of the system in general. Although in general the travel time and distances are highly related, in crowded and large cities traffic and the road conditions, such as accidents and road blocks can significantly change this relation between the time and distance of a trip. In the time-based greedy approach, the tasks with lowest travel time, considering current traffic and road conditions, are assigned to the AVs. Similar to previous case, the server finds the minimum total cost value for the system, considering the travel times, from the following equation:

$\min\{\sum_{i=1}^{N}\sum_{j=1}^{M}(t+t')\}$ for all scheduling requests for AV i with feasible requester j.

• Profit: to consider more advanced criteria for scheduling, not only the total travel time and distance for each task of the AV, but also the rate of profitable part of the trip to the transitional trip needs to be considered. Profitable trip is the requester's trip, which will be compensated, and transitional trip is the time or distance to be traveled by the AV to get to the requester's pickup location, and to travel back from drop off location to the next destination. Considering this method, some tasks which are considered expensive and un-efficient in previous cases may be considered profitable, based on the relative locations of the AV and requester, traffic, and other system parameters like fuel and labor costs. These additional parameters can penalize or reward the task's profit value. Although generally the travel times are used for profit calculations, using distance is also possible. To enable the system to calculate and use this profit metric, the server needs owner to perform two additional decryptions. First,

$$Prof_T = \frac{Dec_i(E_i(T_j))}{Dec_i(Result_1)+Dec_i(Result_2)-T_j-T_c+T'}$$

Note that since we have

$$\begin{aligned}&Result_1+Result_2-E_i(T_j)-E_i(T_c)+E_i(T')\\&=E_i(T_c)+E_i(AV_i.T_d)+E_i(R_j.T_d)-E_i(\bar{T})+E_i(\bar{T})+E_i(T_j)\\&+E_i(R_j.T.d')+E_i(AV_i.T_{d'})-E_i(T)-E_i(T_j)-E_i(T_c)+E_i(T)\\&=E_i(AV_i.T_d)+E_i(R_j.T_d)+E_i(R_j.T_{d'})+E_i(AV_i.T_{d'}),\end{aligned}$$

the decrypted denominator of this fraction is precisely the transitional trip duration and the decrypted numerator is the requester trip duration.

Second, we have the distance-based profit as

$$Prof_D = \frac{Dec_i(E_i(D_j))}{Dec_i(D_{Trans})}$$

These two extra values need to be sent to the server by the owner after decryption in case the profit-based method is used.

In this case, this equations needs to be solved by the server:

$\max\{\sum_{i=1}^{N}\sum_{j=1}^{M}(Prof'_T)\}$ for all scheduling requests for AV i with feasible requester j.

for the time-based profit, and

$\max\{\sum_{i=1}^{N}\sum_{j=1}^{M}(Prof'_D)\}$ for all scheduling requests for AV i with feasible requester j.

for the distance-based profit.

2) First come, first served: If a fastest service to the requesters, and/or the AV owners is preferred over maximizing the profit by the system, then a new approach may be used for scheduling the tasks. In this approach, a queue is used to store and manage the tasks, and based on the time first inserted to the queue, the task is prioritized to be extracted and fulfilled. Two different variations of the approach are considerable; to prioritize requesters, or owners:

• Requester priority: if the number of requester tasks in the system are larger than a pre-defined threshold, meaning there is a backlog of un-processed requesters and there is an increased wait time for them, then the system should adopt this approach to enable faster response to the users and improve the quality of the service, rather than the profit of the system. In this case, as soon as an AV is available in the system, it will be assigned to the requester with longest wait time in the queue. If the owner and this requester tasks don't match,based on their location and time constraints, then the next requester in the queue will be considered. Specifically, server uses the following algorithm:

Algorithm 1: Requester-priority Scheduling

```
1: for each available AV do
2:     i = 0;
3:     while !(req=$req\_queue$.peak(i) Match AV) do
4:         i++;
5:         req=$req\_queue$.peak(i);
6:     end while
7:     Assign req to AV;
8:     $req\_queue$.remove(i)
9: end for
```

• Owner priority: similarly, if there are a large number of AV owners idle in the system with no requester task to be assigned to them, they will be sorted and be assigned requesters based on their wait times in the system. This approach will be adopted if the number of currently idle AVs in the system exceeds a pre-defined threshold. In case the matching criteria of the requester and owner do not match, then the next owner in line will be assigned the task. Following algorithm is used by the server:

A comparison between all of the different approaches is presented in Table II.

Algorithm 2: Owner-priority Scheduling

```
1: for each available req do
2:     i = 0;
3:     while !(AV=$av\_queue$.peak(i) Match req) do
4:         i++;
5:         AV=$av\_queue$.peak(i);
6:     end while
7:     Assign req to AV;
8:     $av\_queue$.remove(i)
9: end for
```

TABLE II: Scheduling schemes Comparison

Approach	Parameter	Criteria	Preferred Env.
Greedy	Travel Time	min. total time	Labor-intensive
	Travel Distance	min. total distance	Traffic-restricted
	Task Profit	max. profit	Both
First come, First Served	Requester Priority	req. wait time	Requester Backlog
	AV Priority	AV idle time	Idle AV Backlog

V. EVALUATION

In this section, we evaluate the performance of our proposed schemes. We have conducted real experiments using "Google Maps Directions API" on randomly generated locations within two residential and one commercial districts of a limited area, i.e. the city of Boston and its vicinities. We have used Python-3 on an Ubuntu machine with an Intel Core i7 processor @3.4 GHz and 16 GB of memory for coding and testing.

A. Privacy and security evaluation

In our scheme, the requester and AV owners only share their request and AV tasks with the server, which does not include any sensitive and private location information of either AV or the requester. On top of removing the sensitive location information using the local traffic servers and the ID locations, the requester and AV tasks are also encrypted, which adds another level of security and privacy preservation to the scheme. Also, the locations of the AV is never shared with the requester, and only when scheduling is done and the index of the scheduled requester is sent to the AV, owner and the requester will be directly in contact, and requester will share his/her exact locations with the owner.

Also on the scheduling phase, the server does not have any information about the actual and exact locations of the users. Further, the task costs used by the server for scheduling are only associated with a set of randomized indexes produced by the owner, which is not linkable to any requester by the server.

These settings make the matching and scheduling privacy-preserving and secure, and prevents the server form knowing any details about the location of the users. Also it removes any unnecessary sharing of location information between the users.

B. Efficiency evaluation

In this section we evaluate the performance and efficiency of our proposed scheme in terms of communication and computation overheads of the system. Specifically, we evaluate the effect of system parameters on the communication and computation

TABLE III: Computation and Communication Overheads

Computations Overhead Summary	
Travel duration calculation	$(m+n)(4\times d+1)$
Travel duration encryptions	$(m+n)(4\times d+1)$
Cost calculations	$11\times m\times n\times d$
Cost value decryptions	$3\times m\times n\times d$
$\min$ operations of travel cost	$2\times m\times n$
Communications Overhead Summary	
Locations sent to server	$2\times(m+n)(4\times m+1)$
Travel times sent to server	$(m+n)(4\times m+1)$
Travel distances sent to server	$(m+n)(4\times m+1)$
Encrypted cost values sent to AV	$m\times(2\times n\times d+1)$
Cost and requester index values	$m\times n$

overheads for the server and users. We also evaluate scheduling overheads for different scheduling mechanisms.

1) Matching overheads: Considering m owners, n requesters and d IDs, for each owner-requester pair, there are two transitional trips, as depicted in Fig. 4. Thus, there are $(m+n)(4\times d+1)$ travel duration calculations performed by the owner's local traffic server and the same number of encryptions, i.e. $(m+n)(4\times d+1)$ encryptions, performed by interface devices. Matching server performs 11 addition and subtractions on encrypted values to calculate $Result_1, Result_2$, and D_{Trans} for each owner-requester-ID triple. This results in $11\times m\times n\times d$ operations. Then each owner interface will decrypt $3\times n\times d$ values received from the server, corresponding to $Result_1, Result_2$, and D_{Trans}. This adds $3\times m\times n\times d$ operations in total on the owner device. After decryption of these values, the requester needs to perform a two $\min$ operations for each requester to find the most efficient ID for the first and second transitional trips. This will result in a total number of $2\times m\times n$ minimum operations on d values.

As for the communication overhead, a total number of $(m+n)(4\times m+1)$ pairs of locations are sent to the traffic server and travel durations are returned. This results in $2\times(m+n)(4\times m+1)$ locations, and $(m+n)(4\times m+1)$ travel time and distances. Matching server sends a maximum of $2\times n\times d+1$ encrypted values to the owner. In total, system would have a maximum of $m\times(2\times n\times d+1)$ communication overhead for this part. A maximum of n decrypted total cost, associated with each of the requesters is then sent to the server by each owner, resulting in $m\times n$ cost value and requester index transfers. Finally, after scheduling is done by the server, an index, corresponding to the matched requester, is sent to each AV, incurring a total of m message transfers communication cost to the system. A summary of all computation and communication costs is presented in Table III.

C. Accuracy evaluation

In this section we evaluate the added travel time to the ground truth due to use of the privacy-preservation matching technique. We also evaluate the rate of False Negatives in the matching of a requester and AV task, as an indication of accuracy of the scheme. Note that due to added travel time as a result of

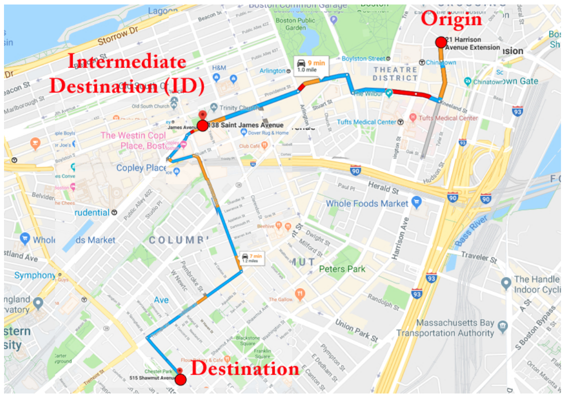

Fig. 6: Example of Routing Using the Intermediate Destination

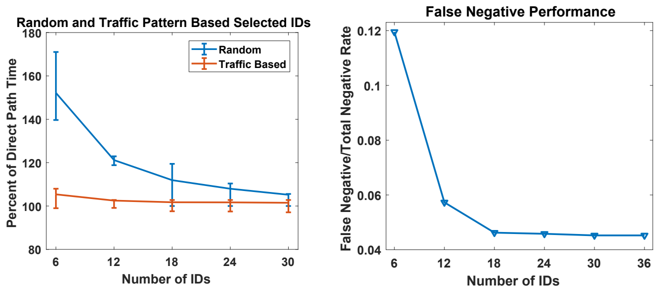

Fig. 7: (a) Accuracy of random and traffic-based cases. (b) False negative ratios performance of the traffic-based case

using the intermediate destinations in the matching scheme, some of matching users are detected non-matching, resulting in an increased rate of FN. Since we do not expect to reduce the travel times by the use of privacy-preservation matching technique, the False Positive rate is expected to be zero.

Specifically, we evaluate the effect of ID locations in accuracy, as well as effect of the used scheduling approach.

1) Random and Traffic-based Scheme: We evaluate the accuracy of our basic scheme by comparing the results where the IDs are chosen randomly, against the traffic-based case where the ID locations are chosen based on the vehicular traffic patterns in the area. We compare the added travel duration and distance when using ID locations as intermediate points, compared to the direct path between origin and destination. In each case we compute the accuracy of the scheme based on the number of IDs. Intuitively, there exists a trade-off between the efficiency and the accuracy achieved. We have tested the algorithm against different numbers of IDs (i.e. 6, 12, 18, 24 and 30) and 30 locations as origins and destinations; 10 locations in each of the three districts, all drawn randomly distributed within the district limits. One example is presented in Fig. 6, where origin, destination and the ID are chosen from same district. We have calculated the travel durations and distances for all origin-destination pairs using all IDs and then selected the minimum as the best option.

In random case, at the worst case, i.e. having 6 IDs, on average among 300 pairs of origin-destination, the travel duration is increase by 52% compared to the direct path, and gradually decrease to reach 5% for the case with 30 IDs. Our traffic-based case on the other hand, performs significantly better and reaches to 5% and 1% for worst and best case respectively. As show in the Fig. 7, the accuracy increases by increasing the number of IDs, even for the random case, because the possibility of having a more suitable ID will increase. It is observable that with increased number of IDs, both cases' performance would get close to the ground truth. On the other hand, the efficiency will suffer from increased number of IDs due to added number of operations. It is very important to mention, using our traffic-based case, which takes advantage of the traffic patterns for choosing the IDs, we achieve improved accuracy with the same number of IDs, thus we improve accuracy without compromising efficiency.

We also consider the number of false negatives of our scheme, i.e. cases where a requester is feasible but the scheme flags it conflicting due to the effect of the IDs. Taking 75 minutes as the required time window for the trip, our traffic-based case with 6 and 18 IDs reaches the false negative ratios of 0.119 and 0.046 on average among 4000 tests. In addition, the false positive ratios are always zero, because our scheme outputs a required time window that is longer than it should be. As such, our scheme using traffic-based IDs can well identify the conflicting trips from the encrypted information only.

2) Scheduling mechanisms: To evaluate the performance and accuracy of our scheduling scheme, we use different approaches for scheduling and evaluate their effect on different parameters of the system, such as total travel time and distance cost of the system, and profit, as the rate of requester trip to the transitional trips for each task, in terms of both time and distance. We use a set of scheduling requests, created from 30 origin and destination locations, chosen from the three districts, and different number of randomly selected ID locations. We randomly take 6, 12, 18, 24, and 30 ID locations in the area and report the total cost of the system for each case. We test all three of greedy-based approaches, and for FCFS approach. Since the cost of the system is irrelevant to the different cases where the priority is either given to the AV or the requester, we only evaluate the requester-prioritized case.

Results, as depicted in Fig. 8, confirm the effectiveness and performance of our scheduling mechanism. We consider the FCFS case as the baseline, since the cost is not considered as a parameter for this approach. As shown by the Fig. 8(a), the travel time-based greedy approach, results in an average travel time decrease of 15.2 minutes for ten tasks among all the cases, with a minimum and maximum improvement of 5 and 47 minutes respectively, compared to the baseline. Similarly, results confirm an average travel-distance reduction of 6.36 miles, as a result of the travel distance-based greedy scheduling approach among all of the test cases. As depicted in Fig. 8(b), increasing number of IDs used for matching, positively effects the total profit of the system, from both travel time and distance point of view. Also, it is shown that the distance-based profit

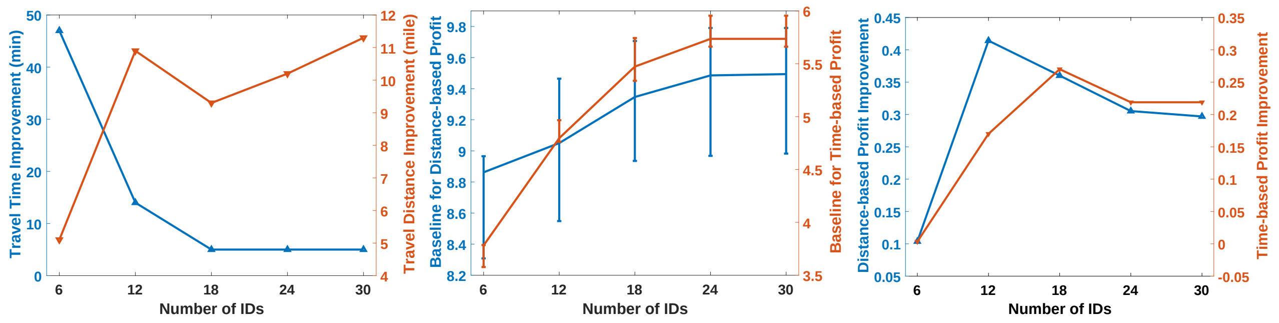

Fig. 8: (a) Travel Time and Distance Improvement by the Greedy Scheduling Approach Compared to the Baseline (b) Time and Distance-based Profit Baseline and Minimum/Maximum (c) Time and Distance-based Profit Improvement by Greedy Approach Compared to the Baseline

rate shows more variation (average 0.79) while using different scheduling mechanism, compared to the time-based profit rate with average variation of 0.28. Finally, as shown in Fig. 8(c) the result confirm an average profit rate improvement of 0.176 and 0.49, compared to the baseline, for travel time-based, and travel distance-based approaches respectively.

VI. Discussion

Our experiments validate effectiveness of our scheme under some conditions, but to make the scheme more practical we discuss following parameters for future works.

• **ID Selection.** In our experiments, we have considered a small and limited number of areas as districts with the locations of the owners, requesters and IDs. Although it is fair to assume the majority of the requests will happen in most crowded residential and commercial districts of the area, there still exists a chance of presence of an owner or requester in less probable areas which may never been selected as a district. This may result in dramatic increase of the travel durations calculated by our scheme compared to the ground truth because the IDs are all chosen from the pre-defined districts and there is a high probability of a long travel duration from the pickup and drop of locations of the requester to any chosen ID.

• **Number of IDs in the system.** An increased number of ID locations in the system will result in an overhead for the scheme during the matching, because each AV-requester-ID triple needs to be individually considered and evaluated form the cost-of-service point of view.However, there are three important point to make here: 1) Since most of the heavy operations related to the matching and cost-of-service evaluation are done on the cloud servers, which are computationally powerful, this increased number of operations does not have a considerable effect on the over performance of the scheme. 2) For many users, the privacy protection offered in our system is worth the short delay in the matching process, both AV owners and requesters. 3) The aim of our scheme is to reduce the number of active ID locations using various static and dynamic methods, in order to minimize the overhead of the system. Thus, although there exist an enormous number of IDs, to ensure usability and accurate matching of the tasks, the number of active IDs for each pair of AV-requester are always limited.

• **Matching efficiency improvement.** To increase the general efficiency of our ID-based matching scheme, we can consider methods that can be used to reduce the communication and computation overheads of the matching scheme by reducing the number of candidate intermediate destinations.

1) Future Travel Estimation: The travel durations between two locations are highly dependent to the traffic and conditions of the routs in the area, such as roadblocks, roadworks and accidents. Although in general the route conditions do not follow a pattern, the vehicular traffics in the area usually have visible patterns based on the days of the week and times of the day . Also, the accidents and congestions in the routes are partially related to the vehicular traffic patterns and are most likely to happen in heavier traffics. Based on this observation, for each day of the week and each time of the day, we define a "traffic coefficient" and use this as a baseline for comparing current traffic of the area to take out the IDs in areas with higher than usuall traffic.

2) Dynamic ID Selection: Other than the techniques discussed in previous section, we also repeatedly monitor the route conditions and traffic flow of the ID areas. This helps to remove the IDs where the traffic around them is higher than usual or there are any accidents or traffic jams. Removing such IDs from the list of available IDs results in more realistic and accurate travel durations used for matching of the tasks which results in lower rate of false negatives in task matching.

VII. Conclusion

In this paper, we proposed a novel privacy-preserving task scheduling scheme for the time-sharing service of autonomous vehicles. To achieve the privacy preservation of location and time, we choose Intermediate Destination (ID) based on traffic patterns and use these IDs to enable the server to perform a privacy-preserving matching for AV owners and AV requesters. Then we propose a set of different scheduling schemes to assign the matched requester tasks to the AVs. We analyzed the security of our proposed scheme and concluded that the semi-trusted scheduling server is unable to infer user's travel times and locations. We conducted a set of extensive experiments on real data sets to show that our enhanced scheme with traffic pattern consideration significantly outperforms the baseline scheme, and its results are close to the ground truth. For future works, we will consider methods for dynamically evaluating and selecting the ID locations for better accuracy and efficiency.

References

[1] C. K. Brownell, "Shared autonomous taxi networks: An analysis of transportation demand in nj and a 21 st century solution for congestion," Ph.D. dissertation, Citeseer, 2013.

[2] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.

[3] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "Ac-poca: Anticoordination game based partially overlapping channels assignment in combined uav and d2d-based networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1672–1683, 2018.

[4] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, 2017.

[5] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Transactions on Computers*, vol. 67, no. 9, pp. 1287–1300, 2018.

[6] N. Shchetko, "Laser eyes pose price hurdle for driverless cars," *The Wall Street Journal*, vol. 21, 2014.

[7] Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. S. Shen, "Privacy-preserving partner selection for ride-sharing services," *IEEE Transactions on Vehicular Technology*, 2018.

[8] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, G. Lin, and X. Shen, "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Transactions on Vehicular Technology*, 2018.

[9] D. Shoup, "Pay as you park," August 2005, [Online; posted 2005]. [Online]. Available: http://shoup.bol.ucla.edu/PayAsYouPark.htm

[10] M. Ghaffari, N. Ghadiri, M. H. Manshaei, and M. S. Lahijani, "p^4qs: A peer-to-peer privacy preserving query service for location-based mobile applications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9458–9469, Oct 2017.

[11] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," *Pervasive computing*, pp. 390–397, 2009.

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

[13] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, vol. 171, no. 5-6, pp. 311–331, 2007.

[14] Y. Qiao, Y. Cheng, J. Yang, J. Liu, and N. Kato, "A mobility analytical framework for big mobile data in densely populated area," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1443–1455, 2017.

[15] I. Bilogrevic, M. Jadliwala, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2011, pp. 77–96.

[16] X. Liang, X. Li, T. H. Luan, R. Lu, X. Lin, and X. Shen, "Morality-driven data forwarding with privacy preservation in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3209–3222, 2012.

[17] B. Ying, D. Makrakis, and Z. Hou, "Motivation for protecting selfish vehicles' location privacy in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5631–5641, Dec 2015.

[18] Z. M. Fadlullah, C. Wei, Z. Shi, and N. Kato, "Gt-qosec: A game-theoretic joint optimization of qos and security for differentiated services in next generation heterogeneous networks." *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1037–1050, 2017.

[19] M. Hadian, T. Altuwaiyan, and X. Liang, "Privacy-preserving time-sharing services for autonomous vehicles," in *Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–5.

[20] W. Zhao, H. Nishiyama, Z. Fadlullah, N. Kato, and K. Hamaguchi, "Dapa: Capacity optimization in wireless networks through a combined design of density of access points and partially overlapped channel allocation," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3715–3722, 2016.

[21] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[22] G. Ghinita, "Privacy for location-based services," *Synthesis Lectures on Information Security, Privacy, & Trust*, vol. 4, no. 1, pp. 1–85, 2013.

[23] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[24] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *MobiSys*, 2003.

[25] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *International Conference on Pervasive Services*. IEEE, 2005, pp. 88–97.

[26] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 152–170.

[27] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 31–42.

[28] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 901–914.

[29] S. Qiao, C. Tang, H. Jin, T. Long, S. Dai, Y. Ku, and M. Chau, "Putmode: prediction of uncertain trajectories in moving objects databases," *Applied Intelligence*, vol. 33, no. 3, pp. 370–386, 2010.

[30] M. Götz, S. Nath, and J. Gehrke, "Maskit: Privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 289–300.

[31] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mhealthcare social network," *Mobile Networks and Applications*, vol. 16, no. 6, pp. 683–694, 2011.

[32] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 646–660, 2018.

[33] W. Tang, J. Ren, and Y. Zhang, "Enabling trusted and privacy-preserving healthcare services in social media health networks," *IEEE Transactions on Multimedia*, 2018.

[34] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A blockchain-based non-repudiation network computing service scheme for industrial iot," *IEEE Transactions on Industrial Informatics*, 2019.

[35] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. S. Shen, "Fully anonymous profile matching in mobile social networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 641–655, 2013.

[36] K. Rabieh, M. Mahmoud, A. Siraj, and J. Misic, "Efficient privacy-preserving chatting scheme with degree of interest verification for vehicular social networks," in *Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[37] W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar, "Exploring the feasibility of fully homomorphic encryption," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 698–706, 2015.

[38] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, 2015.

[39] R. Li, Z. Xu, W. Kang, K. C. Yow, and C.-Z. Xu, "Efficient multi-keyword ranked query over encrypted data in cloud computing," *Future Generation Computer Systems*, vol. 30, pp. 179–190, 2014.

[40] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.

[41] J. Shao, Z. Cao, X. Liang, and H. Lin, "Proxy re-encryption with keyword search," *Information Sciences*, vol. 180, no. 13, pp. 2576–2587, 2010.

[42] P. Paillier, "Composite-residuosity based cryptography-an overview," *Cryptobytes*, vol. 5, no. 1, pp. 20–26, 2002.