Privacy Preservation in Location-Based Services: A Novel Metric and Attack Model

Sina Shaham, Ming Ding, Bo Liu, Shuping Dang, Zihuai Lin, and Jun Li

Abstract—Recent years have seen rising needs for location-based services in our everyday life. Aside from the many advantages provided by these services, they have caused serious concerns regarding the location privacy of users. Adversaries can monitor the queried locations by users to infer sensitive information, such as home addresses and shopping habits. To address this issue, dummy-based algorithms have been developed to increase the anonymity of users, and thus, protecting their privacy. Unfortunately, the existing algorithms only assume a limited amount of side information known by adversaries, which may face more severe challenges in practice. In this paper, we develop an attack model termed as Viterbi attack, which represents a realistic privacy threat on user trajectories. Moreover, we propose a metric called transition entropy that enables the evaluation of dummy-based algorithms, followed by developing a robust algorithm that can defend users against the Viterbi attack while maintaining significantly high performance in terms of the traditional metrics. We compare and evaluate our proposed algorithm and metric on a publicly available dataset published by Microsoft, i.e., Geolife dataset.

Index Terms—k-anonymity, spatio-temporal trajectories, location-based services, privacy preservation.

1 Introduction

With the ubiquitous use of smartphones and social networks, location-based services (LBSs) have become an essential part of contemporary society. The users of smart devices can download LBS applications from Google Play or Apple Store, and 'query' for LBSs they desire. Query refers to a request for a service by providing a location. For example, users can query their locations from an LBS provider to find restaurants nearby [1], refine route planning [2], and receive location-based advertisements [3]. The annual market for LBSs is expected to reach 77.84 billion US dollars by 2021, with an annual growth rate of 38.9% [4].

Unfortunately, the privacy issues associated with the LBSs have raised many concerns. Notably, after the recent Facebook data privacy scandal occupying the headlines of major media [5]. Different from the security of data, which is mainly concerned with secure encryption and integrity, privacy indicates how in control users are to prevent the leakage of their data; Can LBS providers analyze users' locations to find out their home address? Can LBS providers take advantage of users' data to figure out their shopping habits? Can LBS providers share user data with third-parties? And these are just some of the issues that may compromise the location privacy of users.

This work was submitted in part and accepted to present in IEEE INFOCOM, 2019.

S. Shaham and Z. Lin are with the Department of Engineering, The University of Sydney, Sydney, NSW, 2006 Australia (Email: sina.shaham, zihuai.lin}@sydney.edu.au).

M. Ding is with Data61, CSIRO, Sydney, NSW, 2015 Australia (email: ming.ding@data61.csiro.au)

B. Liu is with University of Technology Sydney, NSW 2007, Australia (Email: bo.liu@uts.edu.au)

S. Dang is with Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia (Email: shuping.dang@kaust.edu.sa).

Jun Li is with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210094, CHINA. He is also with the School of Computer Science and Robotics, National Research Tomsk Polytechnic University, Tomsk, 634050, RUSSIA. (Email: jun.li@njust.edu.cn.)

Krumm et al. [6] warn about the current location privacy threats. The authors show that just by having the last location of a day, it is possible to estimate the home location within 60 meters of the actual site. The authors in [7] demonstrate that even when locations are queried from LBS providers as members of a community, sensitive locations associated with users can still be identified based on the distribution of queries. Beresford and Stajano [8] also warn that a system collecting users' locations may invade their location privacy. Therefore, it is crucial to devise new ways to preserve the location privacy of users formally defined as "the ability to prevent other parties from learning one's current or past locations" [8].

Researchers have proposed several approaches to preserve the location privacy of users, among which dummy-based algorithms have drawn a great deal of attention [8]–[12]. For a given user location, the dummy generation algorithms aim at generating $k-1$ dummy locations aside from the actual location of the user and submitting them all together to the LBS server. Thus, making it difficult for untrusted servers, or so-called adversaries, to identify the actual user location. All algorithms are executed in the application layer of mobile phones before sending queries to LBS providers. The groundwork in this field was laid by the authors of [13]. They generated dummies randomly throughout the map and evolved them as users move. Followed by this work, the authors in [14] and [15] proposed to choose the candidate dummies from a virtual circle or grid constructed around the current location of the user.

More recently, an enhanced algorithm was proposed in [16], termed as the dummy-location selection (DLS) algorithm. This algorithm considers the likelihood of locations being real or fake predicated on the history of queries on the map. The basic idea of the DLS algorithm can be explained intuitively in Fig. 1. Assume that a user is at location A and a dummy generation algorithm is required to generate one dummy to preserve the location privacy of user shown by A'. The DLS algorithm argues that A' cannot just be any point on the map but a location that has a similar

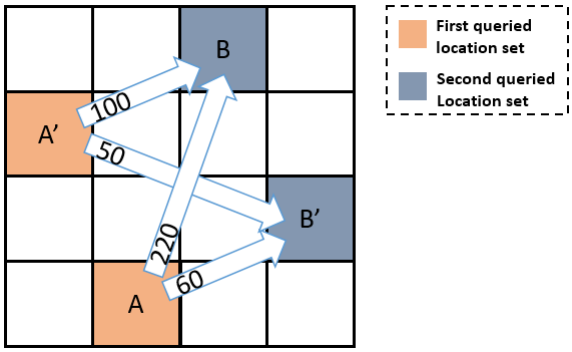


Fig. 1: An example of location privacy of the user being compromised by considering the introduced side information.

likelihood of being queried as to the location A. Such a likelihood can be calculated from the history of queries on the map. For instance, if the location A has been queried 1000 times, and A' has been queried only 5 times, the LBS provider can infer with a high likelihood that the location A is the real location of the user. Based on this logic, the DLS algorithm attempts to select dummies with the same likelihood as the actual locations. Information such as the query likelihood, and extra details that adversaries may know are usually referred to as 'side information'. Unfortunately, the DLS algorithm overlooks a significant piece of side information, which can severely compromise the location privacy, as explained in the following.

Suppose that the user A moves to location B and the DLS algorithm generates another dummy B' associated with the location B. Based on the history of trajectories traveled on the map, the adversary may know the likelihood of paths which have been traveled by users. For instance, the location B has been queried sequentially after the location A for 220 times. This is shown by a directed arrow connecting A to B in the map. Let us now look at the four directed edges connecting the two sets of locations and consider the number of times that each path has been traveled. It can be seen from Fig. 1 that in total, location B has been called 320 times after locations A and A', whereas location B' has been queried only 110 times. Therefore, the adversary can infer with a high likelihood that the real location is possibly location B, and thus, compromise the location privacy of user.

In this work, we study the impact of such side information on the location privacy of users. Compared with the existing literature, the main contributions of this paper are presented as follows:

- We propose an attack model based on the Viterbi algorithm and term it as Viterbi attack. This attack shows how susceptible currently existing algorithms can be, as they mostly see locations as independent snapshots.
- We propose a novel metric called 'transition entropy', which considers the privacy of users in trajectories and not just the static snapshots of the queried locations. We explain the calculation of the metric for two consecutive locations and then expand it to paths with higher lengths. Moreover, we develop an exhaustive algorithm that can improve the transition entropy for a given dummy-generation algorithm.
- We develop an algorithm called robust dummy generation (RDG) that has high resilience against the Viterbi attack while maintaining the high performance in terms of the traditional cell entropy metric, in addition to having a robust performance in terms of transition entropy.
- We compare and evaluate the performance of the proposed metrics and algorithms on a publicly available dataset published by Microsoft, i.e., Geolife dataset.

The rest of the paper is organized as follows. We start by explaining the existing literature in Section 2. Section 3 describes the system model used throughout the paper including the system architecture, the adversary model, and the side information that may be exploited by adversaries. In Section 4, we introduce our proposed metrics followed by explaining the proposed attack model in Section 5. Next, the proposed algorithms are illustrated in Section 6. Finally, we compare and evaluate the proposed metrics and algorithms in Section 7, and we conclude our work in Section 8.

2 Related Works

Anonymity is defined as "the state of being not identifiable within a set of subjects, the anonymity set" [17]. Also, the location of a user is said to be k-anonymous if it is not distinguishable from at least $k-1$ other user locations [18]. To obtain k-anonymity for users, several approaches have been proposed, from which we have identified four broad categories: location cloaking, mixed-zones, pseudonyms, and dummy aided algorithms. The location cloaking technique is based on requesting LBSs for an area consisting of k locations via a trusted party, mixed-zones are predicated on anonymous regions for users, the pseudonyms approach takes advantage of fake IDs for users, and finally, the dummy generation algorithms query fake locations to confuse adversaries.

Gruteser and Grunwald [19] initiated the research on location cloaking. The key idea is to employ a trusted server in order to aid users become k-anonymous. Upon receiving a query from a user, the location anonymizer server computes a cloaking box including the location of the user and $k-1$ other user locations and queries the requested service from the LBS provider for all the k locations. Therefore, making it difficult for the LBS provider to identify the user [20], [21]. Several algorithms have been proposed to implement location cloaking scheme such as ICliqueCloak [22] and MaxAccuCloak [23]. The main drawback of the location cloaking is the need for a location anonymizer, which is an additional cost overhead to the system. Also, the location anonymizer can become a data privacy threat itself.

The authors in [8] proposed the idea of mixed zones. Mixed zone is defined as the spatial zone where the identity of users is not identifiable. All users entering into a mixed zone will change their pseudonym to a new unused pseudonym making it difficult for adversaries to identify the users. The anonymization process is performed by a middle-ware mechanism before transferring the data to third-party applications. The authors further extended their work in [24] by considering irregular shapes for mixed zones. Moreover, the use of mixed zones has particularly attracted attention in vehicular communications. Applying mixed zones on road networks is considered in [25], [26], where a mixed zone construction method called MobiMix is proposed. Lu et al. [27] exploited the pseudonym changes in mixed zones at social spots, and Gao et al. [28] applied mix zones approach on trajectories for mobile crowd sensing applications. Furthermore, the use of cryptography for the generation of mixed zones in vehicular communications is considered in [29]. As it is the case for location cloaking approach, the main drawback of mixed zones is also

the need for a middle-ware mechanism or a trusted party before transferring the data to an untrusted LBS provider.

Another technique to increase the location privacy of users is based on the assignment of pseudonyms to hide the identity of users. The identity of a user can be the name of the person, a unique identifier, such as IP address, or any properties that can be related to the user. The authors in [30] proposed a scenario called the intermediary scenario, in which a trusted intermediary collects the location information of users, such as GPS data and assigns a pseudonym before sending them to an untrusted third-party LBS provider. The paper claims that the use of pseudonyms prevents the third-party LBS provider from identifying and tracking users. The work in [31] suggests that instead of delegating the generation of pseudonyms to the location intermediary, users are suggested to generate the pseudonyms themselves. The use of pseudonyms for preserving the location privacy has also been considered in vehicular communication systems, such as the work presented in [32]. There are several drawbacks associated with this approach. First of all, many of the location-based applications require users to subscribe in order to use services. Secondly, similar to the last two categories, this approach also requires a trusted intermediary, and more importantly, by analyzing the patterns in location data, an adversary can discover the identity of the users [33].

The dummy-based algorithms are considered to be a more promising approach as there is no need for a trusted anonymizer. This technique was initially proposed in [13]. The key idea is to achieve k-anonymity by sending $k-1$ dummy locations aside from the real location of the user while requesting for a service. All locations use the same identifier corresponding to the user, and therefore, it would be difficult for adversaries to identify the real locations of users. Several algorithms have been proposed to help users generate dummies. The authors in [14] proposed to use a virtual circle or a virtual grid that is based on the real location of users to generate dummies. The idea was further developed in [15]. More recently, an algorithm called dummy-location selection (DLS) was proposed in [16]. The algorithm takes the number of queries made on the map into consideration and demonstrates via simulations that the previous algorithms are susceptible to probability attacks. Although the algorithm provides an excellent framework for the generation of dummies, it does not take into account the susceptibility of users in trajectories and the privacy threats associated with that. Do et al. [34] utilized conditional probabilities to generate realistic false locations, and Hara et al. [35] proposed a method based on physical constraints of the real environment.

Among the other metrics proposed to preserve the privacy of users, the notion of differential privacy, proposed by the authors in [36], has attracted attention from both academia and the industry. The primary goal of the metric is to publish aggregated information queried by the users without compromising the privacy of any individual in the database. Most works in the literature aim to achieve differential privacy by addition of noise to the database. Differential privacy is a practical approach for preserving the privacy of population queries; however, it is not suitable for more operational scenarios, where specific queries are made to obtain specific answers. A recent metric proposed in [37], [38] offer a more promising solution to preserve the privacy of users in mobile networks. Still, unfortunately, adversaries are able to limit the surrounding area around the users with a high probability.

3 System Model and Problem Formulation

3.1 System Architecture

Following the recent standards and the current system designs used in the telecommunications industry [39]–[41], we adopt a non-cooperative system architecture as shown in Fig. 2. In this design, there are two main parties involved: LBS users and an LBS server. There is also the telecommunication infrastructure in between which works as a medium for communications between the two parties. The role of each party is explained in the following.

1) LBS users: The system model consists of multiple users equipped with mobile phones with embedded GPS modules. Users can benefit from numerous services provided by downloading and installing LBS applications on their mobile phones. The LBS applications do not necessarily require users to log in to the system, and users can request for services by providing their (I) identifiers such as IP address, username, etc., (II) location information, (III) type of services, (IV) some dummy locations to hide their exact locations. Moreover, in this paper, we focus on 'explicit' trajectory data in which queries are made in equal time intervals. GPS data is the most representative example of explicit trajectory data, which is widely adopted in the studies of trajectory analysis [42]–[44].

2) LBS server: The LBS provider is responsible for providing queried services by users. It is capable of storing the queried information and may have access to other databases and side information. This configuration enables the LBS server to infer the historical query probabilities of users, which can severely compromise the privacy of users. After each query from a user, the server stores the requested information and updates the database accordingly. The LBS server may be untrusted and aim at abusing the personal information of users. Thus, we refer to it as an 'adversary' throughout the paper.

3) Intermediary infrastructure: The queried services from the LBS server are transmitted through telecommunications infrastructure. The telecommunications infrastructure is controlled by mobile operators and regulated by government agencies [45], [46]. Therefore, such infrastructure is considered to be trusted in the system model. Admittedly, this assumption might not hold for untrusted operators and governments that violate the privacy of users in the name of national security. Such a consideration is out of the scope of our work here.

3.2 Preliminaries

Assume that the location map is divided into an $n \times n$ grid, and a user communicates with an LBS server for service. At the time t^q, the user intends to make his/her q-th query from the service provider, preserving k^q-anonymity. Here, k^q quantifies the privacy protection requirement of the user. This metric implies that the adversary is not able to identify the real location of the user with a probability higher than $1/k^q$. Hence, such a user needs to transmit $k^q - 1$ dummy locations to hide his/her true location from the observer. Note that the term 'location' refers to the cell in which the user is located. We denote the set of locations transmitted to the LBS provider at q-th query by

$$LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}. \tag{1}$$

Also, the real location is shown by r^q, where $r^q \in LS^q$. The probability of location l_x^q being the real location can be expressed as

$$\Pr(l_x^q = r^q), \ \ \forall x = 1, ..., k^q. \tag{2}$$

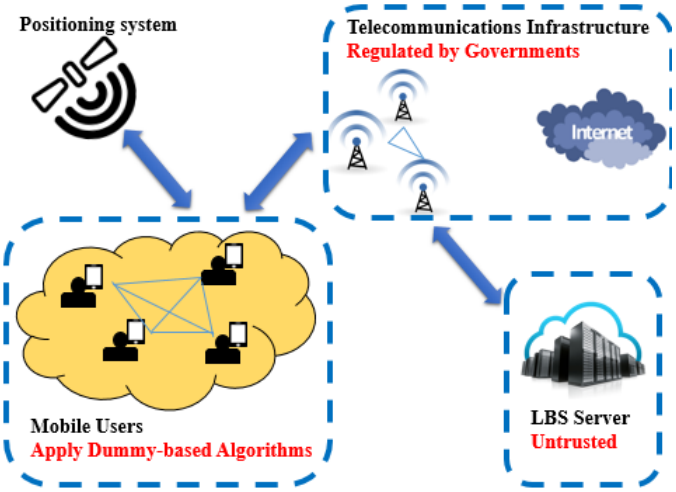


Fig. 2: System architecture of LBSs.

In the next query, the user requires k^{q+1}-anonymity and queries the location set $LS^{q+1} = \{l_1^{q+1}, l_2^{q+1}, ..., l_{k^{q+1}}^{q+1}\}$ from the LBS provider. The probability of $l_y^{q+1} \in LS^{q+1}$ being queried consecutively after $l_x^q \in LS^q$ is denoted by

$$\Pr(l_x^q \Rightarrow l_y^{q+1}). \tag{3}$$

3.3 Cell Entropy Metric

The cell entropy metric was implicitly proposed as part of the DLS algorithm in [16]. The metric is predicated on two factors: query probabilities of cells and the concept of entropy explained as follows.

For a given location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$ which includes the real location of a user and $k^q - 1$ dummies chosen to preserve k^q-anonymity, the set of query probabilities are shown by $B^q = \{b_1^q, b_2^q, ..., b_{k^q}^q\}$, where b_j^q is the query probability of location (cell) l_j^q for $j = 1, 2, ..., k^q$. The query probability of cell l_j^q is calculated by

$$b_j^q = \frac{\text{number of queries in } l_j^q}{\text{number of queries in the whole map}}. \tag{4}$$

The cell entropy borrows the concept of entropy from information theory to quantify uncertainty in query probabilities. The cell entropy metric for location set LS^q can be calculated as [16]

$$h_c = -\sum_{j=1}^{k^q} b_j^q \log_2(b_j^q). \tag{5}$$

3.4 Side Information

There are several side information that adversaries may possess to compromise the location privacy of users. Adversaries may know about the probability of a query being made in different locations of the map. For instance, if a location has been queried five times among the overall 1000 queries made on the map, its query probability can be calculated as $5/1000$. Exploiting query probabilities, adversaries can understand the likelihood of locations being genuine or fake. For instance, if a user queries two locations at the same time, one with a comparably higher probability, it is more likely that the real location is the one with the higher probability. Following the literature [13], [14], [47], we assume that the current side information is accurate without any false information and adversaries may possess the partial or complete side information. In the short-term, the impact of generated dummies are considered to be negligible on the calculation of entropies.

Query probability has always been a critical consideration in the generation of dummy locations. In this work, apart from the possession of traditional side information by adversaries, we consider another prominent side information that can severely compromise the privacy of users. That is, the trajectories users have traveled, which reveals how many times a location has been queried after its neighbor locations. Authorities do not specify any time limit for storing the location information of the users, as it is the case in the US [48]. This lack of legislation enables adversaries to monitor users and get access to trajectories they travel.

4 Transition Entropy

The Cell Entropy metric is a well-known and formidable approach for preserving the privacy of users in telecommunication networks. However, it is proven that the algorithms predicated on Cell Entropy are susceptible to inference attacks if adversaries withhold background information about users. In this section, expanding the novel idea of Cell Entropy, we propose a metric termed as Transition Entropy to quantify privacy preservation in LBSs. We start by explaining the metric for two consecutive queries, then expanding it to trajectories with higher lengths. This metric quantifies the privacy of users in trajectories and can be used as a benchmark to compare and evaluate the performance of dummy-based algorithms in trajectories. Therefore, Transition Entropy no longer has the drawbacks of Cell Entropy as it is based on the traveled paths of the users.

4.1 Transition entropy metric for two consecutive queries

Consider q-th and $(q+1)$-th query of a user from the LBS provider. In the q-th query, the user requests service for the location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$ including $k^q - 1$ dummies and the real location of the user to achieve k^q-anonymity. This follows by moving to a new location with the anonymity constraint of k^{q+1} and making its $(q+1)$-th query. Note that dummies can be generated using any of the existing algorithms in the literature. To start with, based on the sets LS^q and LS^{q+1}, we generate a bipartite graph shown in Fig. 3, where each set forms vertices at one side of the bipartite graph. Looking at the history of queries on the map, we denote the number of times location $l_y^{q+1} \in LS^{q+1}$ has been queried after location $l_x^q \in LS^q$ by n_{xy}, and assign it to the directed edge connecting l_x^q to l_y^{q+1}. Also, as explained in the system model section, for every location $l_x^q \in LS^q$, we denote the query probability of location l_x^q by b_x^q. Query probabilities are also calculated from the historical data stored at the LBS provider.

Our goal is to find out how probable it is for each member of the location set LS^{q+1} to be the real location, given the location set LS^q. In other words, the aim is to calculate the posterior probability of members in LS^{q+1} with respect to LS^q. This probability for each member of LS^{q+1} can be written as

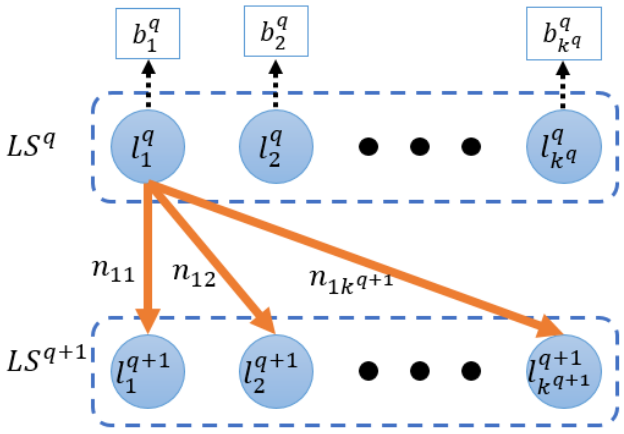


Fig. 3: Bipartite graph generated by two consecutive queries of a user.

$$\forall\, l_y^{q+1} \in LS^{q+1}:$$

$$\Pr(l_y^{q+1} = r^{q+1} | LS^q) = \tag{6}$$

$$\sum_{s=1}^{k^q} \Pr((l_s^q \Rightarrow l_y^{q+1}), (l_s^q = r^q)) = \tag{7}$$

$$\sum_{s=1}^{k^q} \Pr(l_s^q \Rightarrow l_y^{q+1} | l_s^q = r^q) \Pr(l_s^q = r^q), \tag{8}$$

where (7) is the joint probability of l_s^q being the real location of LS^q, and moving to the location l_y^{q+1} after l_s^q. The former probability in (8) can be calculated as

$$\forall\, l_y^{q+1} \in LS^{q+1},\ \forall\, l_x^q \in LS^q:$$

$$\Pr(l_x^q \Rightarrow l_y^{q+1} | l_x^q = r^q) = \frac{n_{xy}}{\sum_{y=1}^{k^{q+1}} n_{xy}}, \tag{9}$$

and the latter probability indicates the normalized query probability and is given by

$$\forall\, l_x^q \in LS^q:\ \Pr(l_x^q = r^q) = \frac{b_x^q}{\sum_{j=1}^{k^q} b_j^q}. \tag{10}$$

Note that (10) indicates that the posterior probabilities of cells in LS^q are set to the normalized query probability of the locations in LS^q. By calculating (8) for every member of LS^{q+1}, the posterior probabilities of locations in LS^{q+1} are determined based on LS^q. Having these probabilities, we exploit the concept of entropy to infer the uncertainty in identifying dummies. The entropy can be derived by

$$h_t = -\sum_{y=1}^{k^{q+1}} \Pr(l_y^{q+1} = r^{q+1} | LS^q) \log_2(\Pr(l_y^{q+1} = r^{q+1} | LS^q)). \tag{11}$$

We define h_t as the transition entropy of the location set LS^{q+1} with respect to LS^q. The transition entropy metric reveals the uncertainty in identifying the real location by adversaries. Having a higher transition entropy indicates that for each member of LS^{q+1}, the probability of paths originating from LS^q to the destination of that member is similar to the other members of LS^{q+1}. Hence, it would be more difficult for the adversary to compromise k^{q+1}-anonymity of the user. The formal algorithm for computing the transition entropy in two consecutive queries is presented in Algorithm 1. The main advantages of the transition entropy metric are:

- Considering the performance of the dummy based algorithms in trajectories and not just for a stationary set of locations.
- Being able to investigate the performance of the dummy based algorithms for users with varying k-anonymity requirements in their trajectories.

Algorithm 1: Transition entropy for two consecutive queries.

Input: LS^q and LS^{q+1}
Output: h_t
Initialization: $CellSum = 0$, $h = 0$.
for $1 \le x \le k^q$ **do**
  $EdgeSum = 0$
  for $1 \le y \le k^{q+1}$ **do**
    $EdgeSum = EdgeSum + n_{xy}$
  end
  for $1 \le y \le k^{q+1}$ **do**
    $\Pr(l_x^q \Rightarrow l_y^{q+1} | l_x^q = r^q) = n_{xy}/EdgeSum$
  end
end
for $1 \le x \le k^q$ **do**
  $CellSum = CellSum + b_x^q$
end
for $1 \le x \le k^q$ **do**
  $\Pr(l_x^q = r^q) = b_x^q/CellSum$
end
for $1 \le y \le k^{q+1}$ **do**
  $\Pr(l_y^{q+1} = r^{q+1} | LS^q) = 0$
  for $1 \le x \le k^q$ **do**
    $\Pr(l_y^{q+1} = r^{q+1} | LS^q) = \Pr(l_y^{q+1} = r^{q+1} | LS^q)$
      $+\Pr(l_y^{q+1} = r^{q+1} | l_x^q = r^q) \Pr(l_x^q = r^q)$
  end
  $h_t = h_t -$
    $\Pr(l_y^{q+1} = r^{q+1} | LS^q) \log_2(\Pr(l_y^{q+1} = r^{q+1} | LS^q))$
end
return h_t

Algorithm 2: Transition entropy for trajectories of length $c+1$.

Input: $LS^q, LS^{q+1}, ..., LS^{q+c}$
Output: h_t
Start:
Run Algo. 1 for LS^q and LS^{q+1}
for $q+1 \le query \le q+c-1$ **do**
  Normalize posterior probabilities of LS^{query}
  Query probabilities of $LS^{query} \leftarrow$ posterior probabilities of LS^{query}
  Run Algo. 1 for LS^{query} and $LS^{query+1}$
end
$h_t \leftarrow$ Normalize posterior probabilities of LS^{q+c} and calculate their entropy
return h_t

$$\sum_{s^c=1}^{k^{q+c-1}} \sum_{s^{c-1}=1}^{k^{q+c-2}} \cdots \sum_{s^1=1}^{k^q} \Big(\Pr(l_{s_1}^q = r^q) \Pr(l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c} | l_{s^c}^{q+c-1} = r^{q+c-1}) \prod_{i=1}^{c-1} \Pr(l_{s^i}^{q+i-1} \Rightarrow l_{q+i}^{q+i+1} | l_{s^i}^{q+i-1} = r^{q+i-1}) \Big) \quad (15)$$

- Elimination of the need for many other previously considered factors, such as time reachability and direction similarity.

4.2 Transition entropy metric for trajectories

Here, we generalize the transition entropy metric for trajectories with different lengths. Consider a user requesting for its $(c+1)$-th query at time t^{q+c}. Hence, providing the LBS provider with the location set $LS^{q+c} = \{l_1^{q+c}, l_2^{q+c}, ..., l_{k^{q+c}}^{q+c}\}$ in order to preserve k^{q+c}-anonymity. The previous queried location sets of the user are shown by LS^{q+i} for $i = 0, ..., c-1$, each with the privacy requirement shown by k^{q+i}. Initially, we aim to calculate the posterior probability of each location in LS^{q+c}. The posterior probabilities indicate the likelihood of any location in LS^{q+c} being the real location based on the previous queries of the user. The posterior probability for each location in LS^{q+c} can be written as

$$\forall\, l_y^{q+c} \in LS^{q+c} : \Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1}) = \quad (12)$$

$$\sum_{s^c=1}^{k^{q+c-1}} \Pr((l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c}), (l_{s^c}^{q+c-1} = r^{q+c-1}) | LS^q, ..., LS^{q+c-2})) = \quad (13)$$

$$\sum_{s^c=1}^{k^{q+c-1}} \Pr(l_{s^c}^{q+c-1} \Rightarrow l_y^{q+c} | l_{s^c}^{q+c-1} = r^{q+c-1}) \times \Pr(l_{s^c}^{q+c-1} = r^{q+c-1} | LS^q, ..., LS^{q+c-2}). \quad (14)$$

Following the same process of moving from (12) to (14), the probability of $\Pr(l_{s^{c-1}}^{q+c-1} = r^{q+c-1} | LS^q, ..., LS^{q+c-2})$ can be solved recursively to reach (15). Also, the transition probabilities in (12) can be calculated as (9). Therefore, evaluating (15) for each node in LS^{q+c}, we can determine the likelihood of a location being the real location of the queried set LS^{q+c}. Finally, we borrow the concept of entropy to characterize the uncertainty in probabilities of LS^{q+c}. So that:

$$h_t = -\sum_{y=1}^{k^{q+c}} \Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1}) \log_2(\Pr(l_y^{q+c} = r^{q+c} | LS^q, ..., LS^{q+c-1})). \quad (16)$$

We call h_t, the transition entropy of the set LS^{q+c} with respect to location sets $LS^q, ..., LS^{q+c-1}$. Our experiments demonstrate that the proposed transition entropy metric shows the high possibility of revealing the real location of users from their previous queries made on the map. The algorithm to calculate the transition entropy metric is presented formally in Algorithm 2.

It is important to note that in the derivation of transition entropy, the first queried location set is the only place in which query probabilities of locations play a role. The transitions between the queried locations determine the remaining factors. It is essential to understand why the query probabilities of the other locations on the path are not used in the calculation of transition entropy. We explain the concept using an example. Fig. 4 demonstrates a case where a user requests an LBS in two consecutive queries. The numbers written on the nodes indicate the normalized query probability of locations, and the numbers printed on the edges indicate the normalized probability of that transition. Now, consider the calculation of LS^{q+1} based on the previous queried location set LS^q. The purpose of the example is to illustrate why the posterior probabilities calculated by previous queries for LS^{q+1} is more reliable than the query probabilities of locations in LS^{q+1}. First, let us calculate the posterior probabilities of LS^{q+1} and its entropy. According to (15), the posterior probabilities can be written as

$$\text{Posterior probability of A being the true location} = \frac{3}{5} \times \frac{1}{3} + \frac{1}{5} \times \frac{1}{4} + \frac{1}{5} \times \frac{1}{4} = \frac{6}{20} \quad (17)$$

$$\text{Posterior probability of B being the true location} = \frac{3}{5} \times \frac{1}{3} + \frac{1}{5} \times \frac{2}{4} + \frac{1}{5} \times \frac{3}{4} = \frac{9}{20} \quad (18)$$

$$\text{Posterior probability of C being the true location} = \frac{3}{5} \times \frac{1}{3} + \frac{1}{5} \times \frac{1}{4} = \frac{5}{20} \quad (19)$$

According to the query probabilities of LS^{q+1}, the location A is more likely to be the real location as it has a significantly higher query probability. However, looking at the posterior probabilities calculated for the location set, we can see that based on LS^q, location B is more probable to be the real location of the user. This discrepancy can be explained by looking at what the actual meaning of query probability is. The query probability indicates the number of times a location has been called but does not specify if it has been called after any particular location. Therefore, although the location A has been called more times than the other locations in LS^{q+1}, most of these queries have been made after locations E and D, which are not a member of the location set LS^q. Hence, it can be seen that the posterior probabilities are more credible, as they are considering the number of times queries made after the previous location set LS^q.

5 Viterbi Attack

The Viterbi algorithm is a well-known dynamic programming algorithm proposed in 1967 [49]. Initially, it was designed specifically for convolutional codes, but then it found numerous applications, such as exploring the most likely sequence of hidden states in Hidden Markov Models (HMMs). For a given graph, the aim of the algorithm is to find the shortest path or the so-called Viterbi path. The Viterbi algorithm provides several features that distinguish this algorithm from the other existing algorithms. The most essential feature of the algorithm is the low computational complexity. Here, we design an attack model based on the Viterbi algorithm and name it Viterbi attack, since the principal idea behind the attack is inspired by the Viterbi algorithm. The proposed

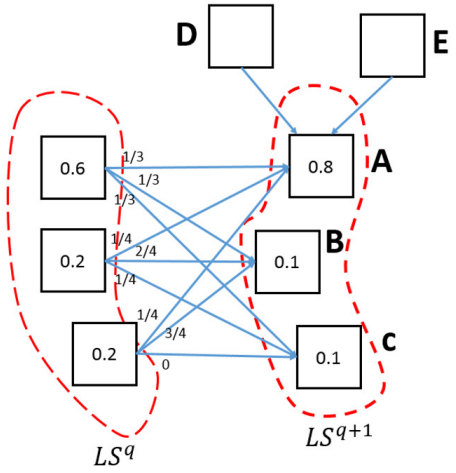


Fig. 4: An example of two consecutive queried location sets.

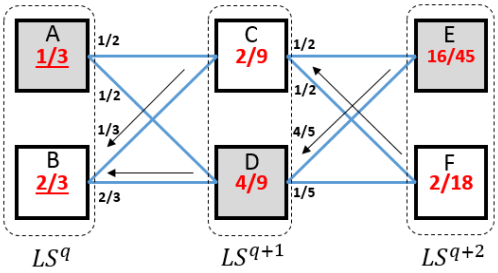


Fig. 5: An example of a trajectory with the length of three. The numbers on the cells indicate the posterior probabilities, and the numbers on the edges indicate the probability of travelling the edge.

Viterbi attack can significantly compromise the location privacy of users if it is not considered in the design of the dummy generation algorithms. As it will be demonstrated in simulations, even for short trajectories, the Viterbi attack can reveal a considerable number of user locations.

Given the location sets LS^q, LS^{q+1},...,LS^{q+c}, corresponding to a trajectory of length $c+1$ of a user, an adversary seeks to find the most probable location sequence or, which is also known as the state sequence. Hence, the attacker aims to identify locations which are most likely to be the actual locations of the user and not the dummies. The desired state sequence of the adversary includes all the real locations of the user shown by $\{r^q, r^{q+1}, ..., r^{q+c}\}$.

We define $\mu(c+1, u)$ to be the maximum probability of a state sequence with the length of $c+1$, given $z^q, z^{q+1}, ..., z^{q+c}$, where $z^j \in LS^j$ and $z^{q+c} = u \in LS^{q+c}$. This function can be mathematically expressed as

$$\mu(c+1, u) = \max_{z^{q:q+m} | z^{q+m} = u} \Pr(z^{q+m} = r^{q+m}), \qquad (20)$$

where for each $u \in LS^q$, and the initial value of the μ function is set to be

$$\mu(0, u) = \Pr(u = r^q). \qquad (21)$$

As the most credible information for the first queried location set is the query probability, $\Pr(u = r^q)$ is calculated via equation (10). Starting from the second queried location set the most probable path can be calculated recursively as

$$\mu(m+1, u) = \max_{u^0 \in LS^{q+c}} \mu(c, u') \Pr(u' \to u). \qquad (22)$$

Algorithm 3: Viterbi attack.

1 **Input:** Location sets LS^q, LS^{q+1},...,LS^{q+c} and the normalized query probability for the location set LS^q
2 **Output:** $EstState$ which is the most likely path
3 **Start:** .
4 **for** $1 \le u \le k^q$ **do**
5 $\quad \mu(q, u) = \Pr(l_u^q = r^q)$
6 $\quad pointer(q, u) = 0$
7 **end**
8 **for** $1 \le j \le c$ **do**
9 $\quad$ **for** $1 \le u \le k^{q+j}$ **do**
10 $\quad\quad \mu(q+j, u) = \max_{u^0 \in LS^{q+j}} \mu(q+j-1, u') \Pr(u' \to u)$
11 $\quad\quad pointer(q+j, u) \leftarrow$ state of $\max_{u^0 \in LS^{q+j}} \mu(q+j-1, u')$
12 $\quad$ **end**
13 **end**
14 $EstState[c] =$ state of $max(\mu(q+c, :))$
15 **for** $c-1 \ge j \ge 0$ **do**
16 $\quad EstState[j] = pointer(q+j+1, EstState[j+1])$
17 **end**
18 **Output:** $EstState$.

The formal presentation of Viterbi attack is given in Algorithm 3. The algorithm starts by setting the initial values of the μ array to their normalized query probability in lines $4-7$. An array called $pointer$ is used to keep track of the most likely state of the previous queried location set, and the most probable path is calculated in lines $8-13$. Finally, the most probable path is chosen, and the corresponding states are returned as outputs.

The example shown in Fig. 5 explains the Viterbi attack on a trajectory with the length of three, where the value of k is set to two for all queried location sets. The grey cells indicate the actual user locations and the cells with no filling are dummies associated with the user locations. Also, as before, the values on the edges indicate the probability of travelling through the edge, and the values on the cells indicate the posterior probabilities of the paths so far. The purpose of the example is to show how the Viterbi algorithm can identify and distinguish most of real user locations from dummies. The Viterbi attack successively traverses through the location sets and ultimately backtracks the most likely path. Such a path is shown in the simulation section of this paper to severely compromise the location privacy of users severely.

The cells A and B are the members of the first location set queried from the LBS provider, i.e., LS^q. As LS^q is the initial set, the posterior probabilities of the cells are set to their query probabilities. In other words, these probabilities indicate how likely these cells are to be the real locations of the user. Next, the Viterbi attack considers the second location set. For each cell, the algorithm calculates the most likely path that can reach the cell by (22). Once the most likely path so far is calculated and identified, a pointer from the cell to its previous neighbour is stored, which will be used for backtracking the most likely path. For example, there are two paths ending at C, which are originated from A and B. Based on (22), the probabilities are:

Algorithm 4: Exhaustive search algorithm

Input: $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, $\{l_1^{q+1}\}$, k^{q+1}
Output: LS^{q+1}
Start:
$D \leftarrow$ generate a pool of $4k^{q+1}$ dummies using the X algorithm
$\{S_1, S_2, ...S_m\} \leftarrow$ choose m distinct $(k^{q+1}-1)$-subsets of D
for $1 \leq y \leq m$ **do**
  $S_y \leftarrow S_y \cup \{l_1^{q+1}\}$
  $h_y \leftarrow$ calculate transition entropy of S_y
  $H \leftarrow H \cup \{h_y\}$
end
$LS^{q+1} \leftarrow S$ corresponding to the maximum h
return LS^{q+1}

$$\text{Probability of path from } A \text{ to } C = \frac{1}{3} \times \frac{1}{2} = \frac{1}{6} \quad (23)$$

$$\text{Probability of path from } B \text{ to } C = \frac{2}{3} \times \frac{1}{3} = \frac{2}{9} \quad (24)$$

As the path from B to C has a higher probability, and this value is shown on the cell. A pointer is used to store B as the previous neighbour of C. The pointer is shown by an arrow on the edge. A similar approach is continued in sequence for the cells D, E, and F. After the calculation of probabilities, as shown in the figure, the most likely path is identified based on the probabilities shown on the last location set. In this example, the path ending at the cell E has the highest probability. Therefore, based on the stored pointers, the most likely path corresponds to the cells B, D and E. In this example, the Viterbi attack is able to distinguish two out of three of the user locations correctly.

6 Proposed Algorithms to Improve Location Privacy of Users

In this section, we start by developing an exhaustive search algorithm to improve the transition entropy metric for a given dummy-generation algorithm. We denote this hypothetical algorithm by X and aim at increasing its transition entropy in trajectories. Next, we propose an algorithm called RDG that significantly increase the privacy of users against the Viterbi attack, while maintaining the high performance in terms of transition entropy and cell entropy.

6.1 Exhaustive Search Algorithm

Suppose that a user has made its q-th query shown by $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, which includes the real location and its associated dummies. The dummies in LS^q are generated using a given algorithm X. In the next query, the user moves to a new location (l_1^{q+1}) and seeks to generate $k^{q+1}-1$ dummy locations. The following approach will help the user increase its transition entropy while generating LS^{q+1}.

The idea is to generate a pool of dummies based on the algorithm X instead of only $k^{q+1}-1$ dummy locations. Having the dummy pool, the exhaustive search algorithm goes through $k^{q+1}-1$ subsets of the pool to find the one that maximizes the transition entropy. The formal description of the proposed exhaustive approach is presented in Algorithm 4. The inputs of the algorithm are the location set $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, the real location of the user in $(q+1)$-th query, and the privacy requirement of the user in $(q+1)$-th query. The exhaustive search algorithm starts by generating a pool of $4k^{q+1}$ dummies using the X algorithm and assigns them to an empty set D. Then, m distinct subsets of D with $(k^{q+1}-1)$ members are chosen and assigned to $\mathcal{S} = \{S_1, S_2, ...S_m\}$. Any of the members in $\mathcal{S}$, once attached to l_1^{q+1} will form a complete k^{q+1} set, preserving k^{q+1}-anonymity. Note that the constraint m is chosen to limit the number of subsets computed in case of large pool size. Next, the transition entropies resulted from the members of $\mathcal{S}$ are calculated and stored in H. Finally, the member of S that results in the maximum transition entropy is returned as the output.

6.2 RDG Algorithm

We propose a robust algorithm termed the RDG algorithm to preserve the location privacy of LBS users in mobile networks. The RDG algorithm has three primary advantages compared with the existing algorithms: (I) Achieves near-optimal cell entropy; (II) results in a significantly high transition entropy compared with the current approaches; (III) provides a robust performance against Viterbi attack. The algorithm is based on the idea of posterior probabilities, and it is formally presented in Algorithm 5.

Following the same setup as the proposed exhaustive search algorithm, a user has made its q-th query shown by $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, which includes the real location and its associated dummies. The dummies in LS^q are generated using the DLS algorithm. In the next query, the user moves to a new location (l_1^{q+1}) and seeks to generate $k^{q+1}-1$ dummy locations. If LS^q is the initial query of the user from the LBS provider, then the initial posterior probabilities are set to the normalized query probabilities of the locations in LS^q; otherwise, the posterior probabilities are calculated by (12). In the algorithm, posterior probabilities are assigned to an array called $weight$.

The algorithm starts with the generation of a pool of dummies using the DLS algorithm based on the real location of LS^{q+1}. Using the DLS algorithm to generate the pool of dummies will ensure high performance in terms of cell entropy. From our experiments, setting the pool size to four times of the k^{q+1} maintains the cell entropy sufficiently high, while resulting in a robust performance in terms of the transition entropy and Viterbi attack resilience. Next, the algorithm continues by employing a greedy approach to add the most suitable dummies for the location set LS^{q+1}. For choosing the i-th member of the set LS^{q+1}, each of the remaining dummies in the pool is checked one by one. A criterion chosen here is based on maximizing the entropy for the array $weight$. For each member $u \in LS^{q+1}$, the $weight$ array is calculated as

$$weight(q+1, u) = \max_{u' \in LS^q} weight(q, u') \Pr(u' \rightarrow u). \quad (25)$$

The first index of the $weight$ array is used to distinguish between weights corresponding to different location sets. For each member of the dummy pool, its weight is calculated, followed by the entropy of the weight array. After calculation of the entropy for all possible members, the member having the maximum entropy is chosen as the next member of LS^{q+1}. The process continues until

Algorithm 5: RDG algorithm.

Input: $LS^q = \{l_1^q, l_2^q, ..., l_{k^q}^q\}$, $\{l_1^{q+1}\}$, k^{q+1}
Output: LS^{q+1}
Start:
for $1 \leq u \leq k^q$ **do**
  $weight(q,u) \leftarrow$ Posterior probability of l_u^q
end
$D \leftarrow$ generate a pool of $4k^{q+1}$ dummies using the DLS algorithm
for $1 \leq member \leq k^{q+1} - 1$ **do**
  $entropy = zeros(1 \times |D|)$
  for $1 \leq d \leq |D|$ **do**
    $LS^{q+1} = LS^{q+1} \cup \{D[d]\}$
    for $1 \leq u \leq k^{q+1}$ **do**
      $weight(q+1,u) = \max_{u^0 \in LS^q} weight(q,u')\Pr(u' \rightarrow u)$
    end
    normalize $weight(2,:)$
    $entropy[d] \leftarrow$ entropy of $weight(q+1,:)$
    $LS^{q+1} = LS^{q+1} - \{D[d]\}$
  end
  $NewMember \leftarrow$ {member of D which maximize $entropy$}
  $LS^{q+1} = LS^{q+1} \cup \{NewMember\}$
  $D = D - \{NewMember\}$
end
return LS^{q+1}

all $k^{q+1} - 1$ dummies of LS^{q+1} are chosen. Note that before the calculation of entropy, the weights are normalized to make the accumulation of probabilities add up to one. The algorithm is designed to provide a high cell entropy and transition entropy for users' of the LBS applications while protecting them from the Viterbi attack on trajectories.

7 Performance Evaluation

7.1 Experimental Setup

In our experiments, we use the data collected by Geolife project [50]–[52]. The Geolife dataset includes the GPS trajectories of 182 users from April 2007 to August 2012 in Beijing, China. The dataset contains $13,561$ trajectories with a total distance of $1,292,951$ km. Two main advantages are distinguishing Geolife dataset for our work: Firstly, the recorded data aside from monitoring the daily routines of users, such as going to work or home, includes trajectories involving sports activities such as hiking and cycling. Secondly, many of the recorded trajectories are tagged with transportation modes, which indicate the use of various means of traveling from bus and car to airplane and train.

We conducted our experiments on 1km× 1km central part on the Beijing map with the resolution of 0.01km×0.01km for each grid cell. The location privacy requirements of users are investigated for values 2 to 30. For each value of k, the trial is repeated 3000 times to ensure the reliability of results. The experiments were performed on a PC with a 3.40 GHz Core-i7 Intel processor, 64-bit Windows 7 operating system, and an 8.00 GB of RAM. Python programming is used to implement algorithms.

TABLE 1: Statistics of Geolife dataset.

Dataset	Geolife
Total number of samples	47581
Number of trajectories	13561
Number of users	182
Total distance	$1,292,951$ km

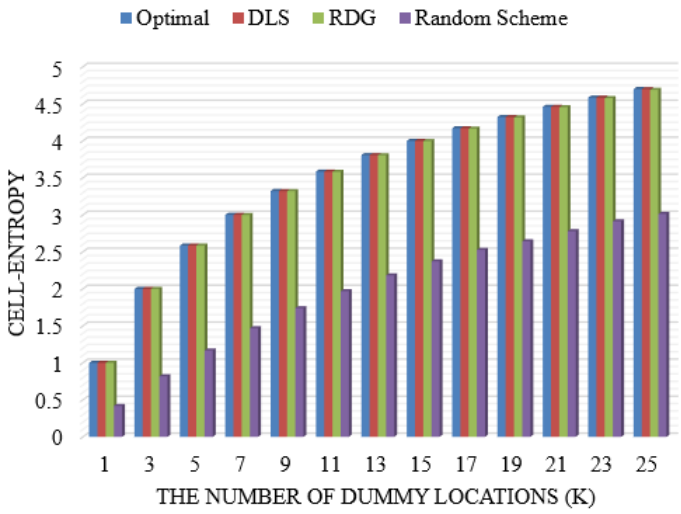


Fig. 6: Comparison of algorithms in terms of cell entropy for different k.

7.2 Performance Analysis

We evaluate the performance of the proposed algorithms and metrics through extensive experiments. We intend to show that the proposed RDG algorithm can achieve:

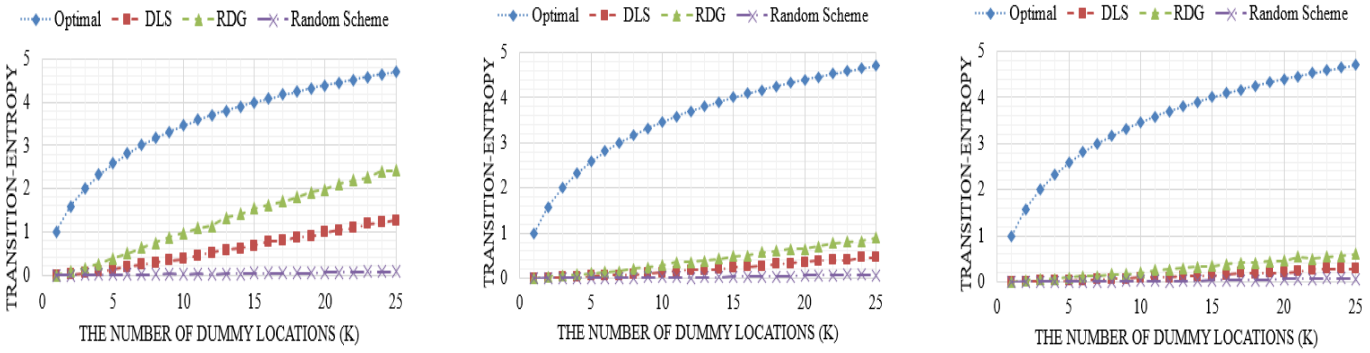
- Near-optimal cell entropy;
- Robust transition entropy performance compared to prior works;
- Improved resilience against Viterbi attack.

Therefore, in the following subsections, we start by evaluating the performance of algorithms in terms of cell entropy, followed by transition entropy analysis and the investigation of the resilience to the Viterbi attack.

7.2.1 *Cell entropy performance evaluation*

Cell entropy indicates how different is the query probability of the actual user location from its associated dummies. A higher cell entropy is desirable, as it results in higher uncertainty of finding the real location. Fig. 6 presents the comparison among different algorithms in terms of cell entropy. The optimal value is achieved when all k locations queried from the LBS provider have the same probability of $1/k$, or equivalently, the location set has the cell entropy of $h = \log_2(k)$. The optimal value is the upper bound for all algorithms since it is the maximum entropy that a location set can achieve.

In Fig 6, three algorithms are compared, including the DLS algorithm which is the conventional method for generation of dummy locations, our proposed RDG algorithm, and the random scheme by which dummies are chosen randomly. Moreover, optimal cell entropy values are shown as a benchmark. As expected, the random scheme proposed in [13] results in a lower cell entropy



(a) Trajectories of length 2. (b) Trajectories of length 3. (c) Trajectories of length 4.

Fig. 7: Comparison of algorithms in terms of transition entropy for different k.

compared to the other algorithms due to the random generation of dummies. On the other hand, the RDG and DLS algorithms both consider the query probability of cells in the generation of dummies. Therefore, the cell entropy of these two algorithms is higher than the random scheme and almost achieves near-optimal performance. Having such a high cell entropy ensures that the adversary is not able to compromise the location privacy of users from a stationery set of locations submitted to the server. Unfortunately, although the DLS algorithm has a robust performance for a single collection of queried locations, no consideration has been given to locations queried as part of trajectories.

7.2.2 *Transition entropy performance evaluation*

The currently established cell entropy metric only considers the location privacy for a stationary set of queried locations submitted to the LBS server but overlooks the fact that users may ask for services successively. If users query location sets in consecutive attempts, they reveal the trajectory they are traveling. Therefore, adversaries can use the likelihood of traveling different paths between consecutive location sets to calculate the posterior probabilities and compromise the location privacy of users.

Fig. 7 compares the performance of different algorithms in terms of the transition entropy metric for various k. Having a lower transition entropy suggests a lower privacy level for users of the LBS applications and a higher likelihood for adversaries to find out the actual coordinates of the users. We start our evaluation by trajectories of length 2 in Fig. 7a, and then focus on the transition entropy for longer paths in Figs. 7b and 7c. In all the three graphs, the comparison is conducted among the optimal transition entropy values, the widely adopted algorithm DLS, the proposed RDG algorithm, and the random scheme.

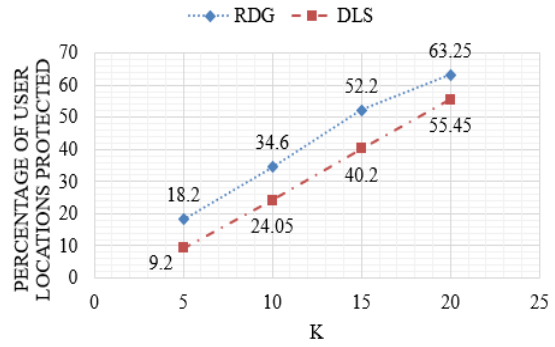
In Fig. 7a, two consecutive location sets are generated based on the specified value of k. Each of the locations sets includes the real location of the user and its associated dummies. To make experiments as realistic as possible, the movement pattern is chosen randomly from the recorded trajectories in the dataset. The optimal value corresponds to a scenario, in which all members of the second location set are equally likely to be called consecutively after the members of the first location set. This outcome is desirable, as it results in achieving k-anonymity for users and protecting their location privacy. The optimal values can be calculated in a similar way as the optimal number for the cell entropy. Considering Fig. 7a, the random scheme in which the dummy locations are chosen randomly achieves the lowest transition entropy, indicating that the adversary can easily recognize most of the dummies from the transition entropy even for the two consecutive location sets queried by the user.

Furthermore, it can be seen from the figure that although the DLS algorithm achieves near-optimal performance in terms of cell entropy, it results in significantly low privacy protection in trajectories. Even for two consecutive queries from the LBS provider, the DLS algorithm indicates a significantly low transition entropy. Such performance shows that adversaries can compromise the location privacy of users by calculating the posterior probabilities. Fortunately, the proposed RDG algorithm can significantly improve the transition entropy, achieving almost twice as high transition entropy as the DLS algorithm. In other words, the likelihood of compromising the k-anonymity requirement is decreased by the proposed algorithm, which leads to a higher location privacy level for users of the LBSs.

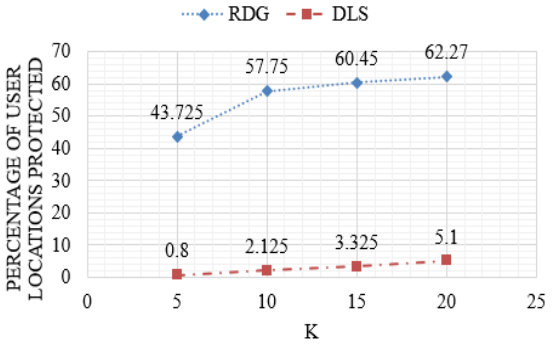
Figs. 7b and 7c extend our analysis of transition entropy to trajectories with higher lengths. Both graphs indicate that as more locations are queried from the LBS provider, the transition entropy decreases. These experimental outcomes match well with the theory because having more information results in a more accurate calculation of posterior probabilities by adversaries; Hence, we expect to see less uncertainty and transition entropy. Further investigating the figures, the DLS algorithm can be seen to have a very low transition entropy compared to the proposed RDG algorithm. Therefore, the RDG algorithm is viable in increasing the transition entropy of users while maintaining the cell entropy to a near-optimal level. However, as the adversary acquires more location points, the threat to location privacy of users gets more serious. This fact can be seen in Fig. 7. As the length of trajectory increases, the transition entropy decreases, which refers to having a higher chance for adversaries to get access to the location data of users.

7.3 Performance of Algorithms against Viterbi Attack

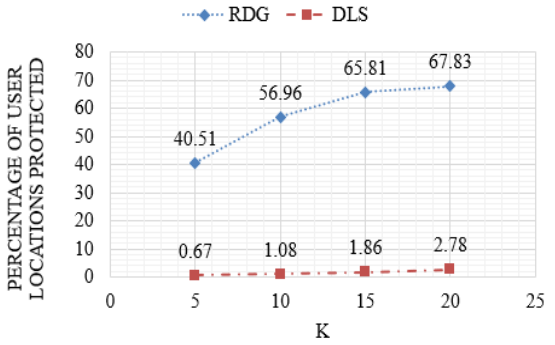
In this section, we compare and evaluate the performance of our proposed RDG algorithm and the widely accepted DLS method against the attack model proposed in Section 5. The Viterbi attack considers users in trajectories instead of just taking into account snapshots of the real locations and dummies. The Viterbi attack is based on the calculation of posterior probabilities of user locations revealed to the LBS provider. It will be shown in this section that applying such an attack can significantly compromise the location



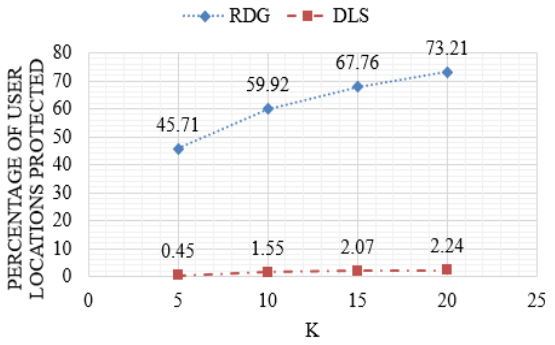
(a) Percentage of user locations protected for the trajectory length of 2.



(b) Percentage of user locations protected for the trajectory length of 4.



(c) Percentage of user locations protected for the trajectory length of 6.



(d) Percentage of user locations protected for the trajectory length of 8.

Fig. 8: The performance evaluation and comparison of algorithms against the Viterbi attack considering various path lengths and privacy requirement k.

privacy of users. Therefore, having a robust algorithm such as RDG is crucial to protect the location privacy.

Fig. 8 illustrates the performance of the RDG and DLS algorithms once the Viterbi attack is applied to the dataset. The figure consists of four subfigures to show the performance with various length of trajectories. In each subfigure, the percentage of real locations of users which have been protected are exhibited for different privacy requirements k. For instance, in Fig. 8b, when $k = 5$, the graph indicates that the DLS algorithm can only protect 0.8 percent of the queried locations, and therefore, adversaries can almost distinguish all true locations of users from their associated dummies. This indicates how dangerous and powerful the Viterbi attack can be in compromising the privacy of users.

Considering the performance of the DLS algorithm, it can be seen in the figure that for path lengths greater than 2, the Viterbi attack can almost find out all real locations of the users despite the existence of dummy locations. Therefore, although in a single query user locations are protected using the existing dummy generation algorithms, when users are considered in trajectories, due to the extra side information that adversaries may hold, they are able to identify user locations. Furthermore, another mainstream observation is that increasing the number of dummies can improve location privacy. Such an effect is expected as having a larger k indicates the generation of more dummies to protect user privacy. Unfortunately, the boost in privacy by increasing the value of k is not sufficient even when the trajectory length is two.

From Fig 8, our proposed RDG algorithm can help users to protect their privacy significantly better. The RDG algorithm takes into account the posterior probabilities that adversaries may hold and aims at making the likelihood of different paths equal. Doing so, the algorithm confuses adversaries in identifying exact locations of users. In contrast to the DLS algorithm, the performance of RDG algorithm improves as the path length increases. It means that for longer trajectories, the adversary has a less chance of compromising user privacy. Also, expectedly, increasing the value of k improves the privacy of users for the RDG algorithm as well.

8 Conclusions and Future Directions

In this paper, we investigated the location privacy of users in trajectories and considered the threats that their previous queries could pose on their location privacy. We developed an attack model based on the Viterbi algorithm that demonstrates how susceptible the location privacy of users is. Therefore, to overcome the difficulties current research face in the evaluation of user privacy in trajectories. We have extended the well-known metric cell entropy to consider users throughout the entire travelled path. The new metric transition entropy can be used to identify how susceptible the users are to contextual location privacy threats.

Furthermore, to improve the transition entropy metric, an exhaustive search approach was proposed, which can increase the transition entropy for a given dummy generation algorithm. We also proposed a powerful algorithm called RDG that results in a robust performance in terms of both transition entropy and cell entropy, while protecting users against the Viterbi attack, particularly in short trajectories.

Based on the results and achievements in this paper, there are several potential research directions worth further investigations:

- Improve the RDG algorithm to achieve higher transition entropy levels. Although the RDG algorithm can significantly enhance user privacy against the Viterbi attack,

as the trajectory length increases, adversaries learn more information about users, and consequently, the transition entropy is decreased. Therefore, improvements to the algorithm are required to ensure that transition entropy stays above an acceptable level.

- Extend our approach to 'implicit' datasets, in which the time intervals between queries are not equal.
- Improve the comprehensiveness of posterior probabilities in the calculation of transition entropy to incorporate the temporal information of users.

REFERENCES

[1] D. Arribas-Bel and J. Bakens, "Use and validation of location-based services in urban research: An example with dutch restaurants," *Urban Studies*, vol. 56, no. 5, pp. 868–884, 2019.

[2] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie, "Tour recommendation and trip planning using location-based social media: a survey," *Knowledge and Information Systems*, pp. 1–29, 2018.

[3] X. Chen, F. Xu, W. Wang, Y. Du, and M. Li, "Geographic big data's applications in retailing business market," in *Big data support of urban planning and management*. Springer, 2018, pp. 157–176.

[4] MARKETSANDMARKETS, "Location-based services (lbs) and real-time location systems (rtls) market by component (hardware, software and services), location type (indoor and outdoor), vertical, region - global forecast to 2024," Aug 2019. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/location-based-service-market-96994431.html

[5] Financial Times, "Facebook privacy breach," Mar 2018. [Online]. Available: https://www.ft.com/content/87184c40-2cfe-11e8-9b4b-bc4b9f08f381

[6] J. Krumm, "Realistic driving trips for location privacy," in *International Conference on Pervasive Computing*. Springer, 2009, pp. 25–41.

[7] B. Liu, W. Zhou, S. Yu, K. Wang, Y. Wang, Y. Xiang, and J. Li, "Home location protection in mobile social networks: a community based method (short paper)," in *International Conference on Information Security Practice and Experience*. Springer, 2017, pp. 694–704.

[8] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive computing*, no. 1, pp. 46–55, 2003.

[9] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, 2007, pp. 246–257.

[10] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 171–178.

[11] M. L. Yiu, C. S. Jensen, J. Møller, and H. Lu, "Design and analysis of a ranking approach to private location-based services," *ACM Transactions on Database Systems (TODS)*, vol. 36, no. 2, p. 10, 2011.

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

[13] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Pervasive Services, 2005. ICPS'05. Proceedings. International Conference on*. IEEE, 2005, pp. 88–97.

[14] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 957–962.

[15] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*. ACM, 2008, pp. 16–23.

[16] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 754–762.

[17] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity a proposal for terminology," in *Designing privacy enhancing technologies*. Springer, 2001, pp. 1–9.

[18] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," Technical report, SRI International, Tech. Rep., 1998.

[19] 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.

[20] C.-Y. Chow and M. F. Mokbel, "Enabling private continuous queries for revealed user locations," in *International Symposium on Spatial and Temporal Databases*. Springer, 2007, pp. 258–275.

[21] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008, pp. 547–555.

[22] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2012.

[23] J. Xu, X. Tang, H. Hu, and J. Du, "Privacy-conscious location-based queries in mobile environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 3, pp. 313–326, 2010.

[24] A. R. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pp. 127–131.

[25] B. Palanisamy and L. Liu, "Mobimix: Protecting location privacy with mix-zones over road networks," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 2011, pp. 494–505.

[26] P. Balaji and L. Liu, "Attack-resilient mix-zones over road networks: architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 14, no. 3, pp. 495–508, 2014.

[27] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[28] S. Gao, J. Ma, W. Shi, G. Zhan, and C. Sun, "Trpf: A trajectory privacy-preserving framework for participatory sensing," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 874–887, 2013.

[29] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux, "Mix-zones for location privacy in vehicular networks," in *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, no. LCA-CONF-2007-016, 2007.

[30] T. Kölsch, L. Fritsch, M. Kohlweiss, and D. Kesdogan, "Privacy for profitable location based services," in *International Conference on Security in Pervasive Computing*. Springer, 2005, pp. 164–178.

[31] T. Rodden, A. Friday, H. Muller, A. Dix *et al.*, "A lightweight approach to managing privacy in location-based services," *Eprint Lancaster University*, 2002. [Online]. Available: https://eprints.lancs.ac.uk/id/eprint/12967

[32] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[33] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and ubiquitous computing*, vol. 18, no. 1, pp. 163–175, 2014.

[34] H. J. Do, Y.-S. Jeong, H.-J. Choi, and K. Kim, "Another dummy generation technique in location-based services," in *Big Data and Smart Computing (BigComp), 2016 International Conference on*. IEEE, 2016, pp. 532–538.

[35] T. Hara, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, "Dummy-based user location anonymization under real-world constraints," *IEEE Access*, vol. 4, pp. 673–687, 2016.

[36] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[37] K. Chatzikokolakis, E. Elsalamouny, and C. Palamidessi, "Efficient utility improvement for location privacy," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 308–328, 2017.

[38] S. Oya, C. Troncoso, and F. Pérez-González, "Is geo-indistinguishability what you are looking for?" in *ACM WPES*, 2017.

[39] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," in *International Workshop on Privacy Enhancing Technologies*. Springer, 2006, pp. 393–412.

[40] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preserving location data publishing: A machine learning approach," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[41] R. Angmo, V. Mangat, and N. Aggarwal, "Preserving user location privacy in era of location-based services: Challenges, techniques and framework," in *Proceedings of 2nd International Conference on Communication, Computing and Networking*. Springer, 2019, pp. 43–52.

[42] X. Kong, F. Xia, Z. Ning, A. Rahim, Y. Cai, Z. Gao, and J. Ma, "Mobility dataset generation for vehicular social networks based on floating car data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 3874–3886, 2018.

[43] X. Kong, M. Li, K. Ma, K. Tian, M. Wang, Z. Ning, and F. Xia, "Big trajectory data: A survey of applications and services," *IEEE Access*, vol. 6, pp. 58 295–58 306, 2018.

[44] J. Fan, C. Fu, K. Stewart, and L. Zhang, "Using big gps trajectory data analytics for vehicle miles traveled estimation," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 298–307, 2019.

[45] H. Talat, T. Nomani, M. Mohsin, and S. Sattar, "A survey on location privacy techniques deployed in vehicular networks," in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. IEEE, 2019, pp. 604–613.

[46] B. Liu, W. Zhou, T. Zhu, L. Gao, and Y. Xiang, "Location privacy and its applications: a systematic study," *IEEE access*, vol. 6, pp. 17 606–17 624, 2018.

[47] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1017–1025.

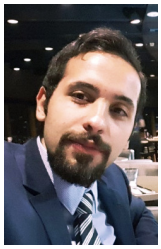
[48] US Gov., "Text - s.1223 - 112th congress (2011-2012): Location privacy protection act of 2012," Dec 2012. [Online]. Available: https://www.congress.gov/bill/112th-congress/senate-bill/1223/text

[49] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[50] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

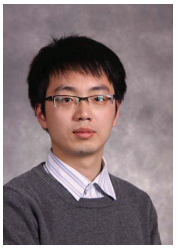
[51] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 312–321.

[52] Y. Zheng, X. Xie, and W.-Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, 2010.

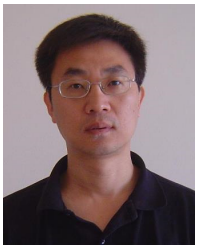


Sina Shaham received B.Eng (Hons) in Electrical and Electronic Engineering from the University of Manchester (with first class honors). He is currently an MPhil student at the University of Sydney. He has years of experience as a Data Scientist and Software Engineer in companies such as InDebted. His current research interests include applications of artificial intelligence in big data and privacy.

Ming Ding (M'12-SM'17) received the B.S. and M.S. degrees (with first class Hons.) in electronics engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the Doctor of Philosophy (Ph.D.) degree in signal and information processing from SJTU, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked at Sharp Laboratories of China in Shanghai, China as a Researcher/Senior Researcher/Principal Researcher. He also served as the Algorithm Design Director and Programming Director for a system-level simulator of future telecommunication networks in Sharp Laboratories of China for more than 7 years. Currently, he is a senior research scientist at Data61, CSIRO, in Sydney, NSW, Australia. He has authored over 100 papers in IEEE journals and conferences, all in recognized venues, and about 20 3GPP standardization contributions, as well as a Springer book "Multi-point Cooperative Communication Systems: Theory and Applications". Also, he holds 21 US patents and co-invented another 100+ patents on 4G/5G technologies in CN, JP, EU, etc. Currently, he is an editor of IEEE Transactions on Wireless Communications. Besides, he is or has been Guest Editor/Co-Chair/Co-Tutor/TPC member of several IEEE top-tier journals/conferences, e.g., the IEEE Journal on Selected Areas in Communications, the IEEE Communications Magazine, and the IEEE Globecom Workshops, etc. He was the lead speaker of the industrial presentation on unmanned aerial vehicles in IEEE Globecom 2017, which was awarded as the Most Attended Industry Program in the conference. Also, he was awarded in 2017 as the Exemplary Reviewer for IEEE Transactions on Wireless Communications.



Bo Liu (M'10) received the B.Sc. degree from the Department of Computer Science and Technology from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and then he received the and MEng. and PhD degrees from the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2007, and 2010 respectively. He is currently a Senior Lecturer with the University of Technology Sydney, Australia. His research interests include cybersecurity and privacy, wireless communications and networks.



Zihuai Lin received the Ph.D. degree in Electrical Engineering from Chalmers University of Technology, Sweden, in 2006. Prior to this he has held positions at Ericsson Research, Stockholm, Sweden. Following Ph.D. graduation, he worked as a Research Associate Professor at Aalborg University, Denmark and currently at the School of Electrical and Information Engineering, the University of Sydney, Australia. His research interests include source/channel/network coding, coded modulation, MIMO, OFDMA, SC-FDMA, radio resource management, cooperative communications, small-cell networks, 5G cellular systems, etc.

Shuping Dang (S'13–M'18) received B.Eng (Hons) in Electrical and Electronic Engineering from the University of Manchester (with first class honors) and B.Eng in Electrical Engineering and Automation from Beijing Jiaotong University in 2014 via a joint '2+2' dual-degree program. He also received D.Phil in Engineering Science from University of Oxford in 2018. Dr. Dang joined in the R&D Center, Huanan Communication Co., Ltd. after graduating from University of Oxford and is currently working as a Postdoctoral Fellow with the Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology (KAUST). He is a co-recipient of the 'best paper' award for work presented at 2019 19th IEEE International Conference on Communication Technology. He serves as a reviewer for a number of key journals in communications and information science, including IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE COMMUNICATIONS LETTERS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. Dr. Dang is recognized as the Exemplary Reviewer of IEEE Communications Letters in 2019. His current research interests include novel modulation schemes, cooperative communications, terahertz communications, and 6G wireless network design.

Jun Li (M'09-SM'16) received Ph. D degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, P. R. China in 2009. From January 2009 to June 2009, he worked in the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he is a Research Fellow at the School of Electrical Engineering, the University of Sydney, Australia. From June 2015 to now, he is a Professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a visiting professor at Princeton University from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and industrial Internet of things. He has co-authored more than 200 papers in IEEE journals and conferences, and holds 1 US patents and more than 10 Chinese patents in these areas. He was serving as an editor of IEEE Communication Letters and TPC member for several flagship IEEE conferences. He received Exemplary Reviewer of IEEE Transactions on Communications in 2018, and best paper award from IEEE International Conference on 5G for Future Wireless Networks in 2017.