M.Sc. Engg. Thesis

Real-Time Sharing of Privacy Protected Location Data in Road Networks

by
Nazmun Naher

Submitted to

Department of Computer Science and Engineering

in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka 1000

February 2016

Dedicated to my loving parents

Author's Contact

Nazmun Naher

Software Engineer

Email: nazmun.cse@gmail.com

The thesis titled "Real-Time Sharing of Privacy Protected Location Data in Road Networks", submitted by Nazmun Naher, Roll No. **0411052019P**, Session April 2011, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on February 7, 2016.

Board of Examiners

1. Tanzima

Dr. Tanzima Hashem — Chairman

Associate Professor — (Supervisor)

Department of CSE, BUET, Dhaka 1000.

2. _____

Dr. Mohammad Mahfuzul Islam — Member

Head and Professor — (Ex-Officio)

Department of CSE, BUET, Dhaka 1000.

3. _____

Dr. Md. Mostofa Akbar — Member

Professor

Department of CSE, BUET, Dhaka 1000.

4. Rifat

Dr. Rifat Shahriyar — Member

Assistant Professor

Department of CSE, BUET, Dhaka 1000.

5. Nova A.

Dr. Nova Ahmed — Member

Assistant Professor — (External)

Department of ECE

North South University, Bashundhara, Dhaka-1229.

Candidate's Declaration

This is hereby declared that the work titled "Real-Time Sharing of Privacy Protected Location Data in Road Networks" is the outcome of research carried out by me under the supervision of Dr. Tanzima Hashem, in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Nazmun Naher

Nazmun Naher

Candidate

Acknowledgment

I express my heart-felt gratitude to my supervisor, Dr. Tanzima Hashem for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. She guided me with proper directions whenever I sought one. I could not have imagined having a better supervisor and mentor for my M.Sc. study and research. Her patient hearing of my ideas, critical analysis of my observations and detecting flaws (and amending thereby) in my thinking and writing have made this thesis a success.

Besides my supervisor, I would like to thank the rest of my thesis committee for their valuable suggestions. I thank Dr. Md. Mostofa Akbar, Dr. Rifat Shahriyar and specially the external member Dr. Nova Ahmed for their encouragement, insightful comments, and hard questions.

Last but not the least, I am always grateful to my beloved family, who have been the source of inspiration behind my every success I have ever made.

Abstract

Location-based applications like nearby information services (e.g., finding the nearest bus station or a hospital), traffic monitoring, urban planning and transport management facilitate the development of smart cities and improve the quality of life for citizens. A user's location is a sensitive data and can reveal private information about the user's health, habit and preferences. Due to privacy concerns, people may hesitate to share their locations and prohibit the growth of location-based services and analysis. In this thesis, we develop a novel approach to share privacy protected location data with others in real time. Our approach does not need to trust any party including a centralized server or peers in a distributed setting for protecting location privacy.

Researchers have developed techniques for sharing a user's locations in a privacy preserving manner in the Euclidean space and road networks. However, none of these approaches can ensure both user privacy and data utility in real time. We identify the possible privacy threats in the literature and develop solutions to overcome the privacy attacks. In our approach, a user reveals a cloaked region (i.e., a region that includes her current location) instead of an actual location if the disclosure of the user's actual location enables others to infer a user's visit to a sensitive place. Existing approaches fail to protect a user's privacy for not considering upcoming sensitive locations in advance. We develop a technique to precompute the *warning zone*, i.e., the refined area where the disclosure of a user's actual location may enable adversaries to identify the user's sensitive locations. Warning zones also enable users to reduce the frequency of not sharing locations for privacy reasons and thereby improve the accuracy and utility of shared locations. In addition, we develop an algorithm to compute a user's cloaked regions using the pre computed warning zones with reduced processing time. We evaluate our proposed approach using a real dataset and compare our algorithm with the most recent state-of-the-art technique in road-networks, in terms of privacy, data utility and computational overhead.

Table Of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The proliferation of location aware mobile devices has enabled users to continuously share their locations and enjoy a range of location based services (LBSs) like nearby information services, location-based advertisements, social networking, and navigation systems. Furthermore, urban planners, traffic monitoring authorities and researchers can analyze shared location data to make effective decisions to facilitate urban computing. However, a user's location is a sensitive data and can reveal private information about the user's health, habit and preferences [2–6]. Due to privacy concerns, users may hesitate to share locations with others and prohibit the growth of location-based analysis and services for the development of a smart cities [7–9]. In this thesis, we develop an efficient approach for real time sharing of location data without violating a user's privacy in road networks.

A user needs to periodically report her locations in real-time to enjoy a continuous LBS (e.g., searching for a nearest gas station with respect to a moving user) or facilitate a location based analysis (e.g., live traffic monitoring). Assume that a user named Alice reports her visited locations as shown in Figure 1.1. Alice visits a hospital on the way to her office and after work she visits a park and then returns to home. The hospital may be a sensitive place for Alice and if an adversary can convince the hospital authority, she can have detail information about Alice's health condition. Thus, Alice reveals her daily activities and sensitive information while sharing the location data.

In this thesis, we present an approach, where users ensure privacy of location data before sharing with others. Our approach does not need to trust any party including a centralized server or peers in a distributed setting for protecting location privacy. In parallel to protecting privacy of location data, it is important to preserve the utility of location data. For example, if a user reveals location

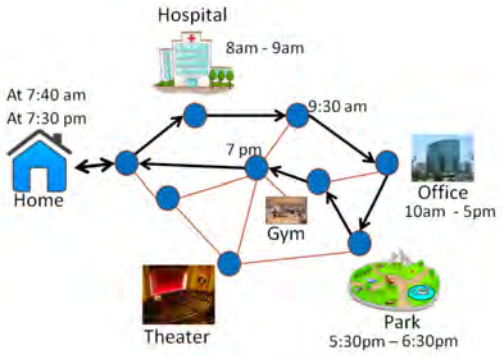


Figure 1.1: Privacy violation while sharing location data

data in city level, the revealed information can contribute little for the development of applications and analysis in a smart city. Our proposed approach ensure both utility and privacy of location data. Furthermore, if an approach requires high computational overhead to protect privacy of location data before sharing with others, it cannot support real-life applications (e.g., real time traffic monitoring). We propose an efficient algorithm that can share privacy protected location data in real-time.

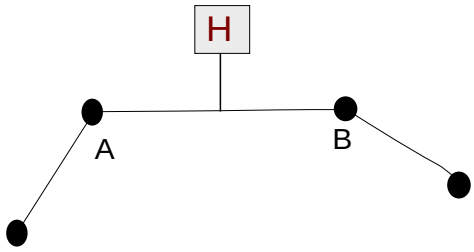


Figure 1.2: A privacy threat when a user stops sharing a sensitive location

1.1 Privacy Threats and Solutions

A naive approach to protect a user's privacy can be to stop sharing sensitive locations. For example, if Alice does not want to let others know that she visits a hospital, Alice can stop sharing location when she is nearby a hospital. However, protecting privacy of users who periodically shares location data in real-time is not straightforward as an adversary can exploit spatio-temporal constraints in road networks and infer sensitive information.

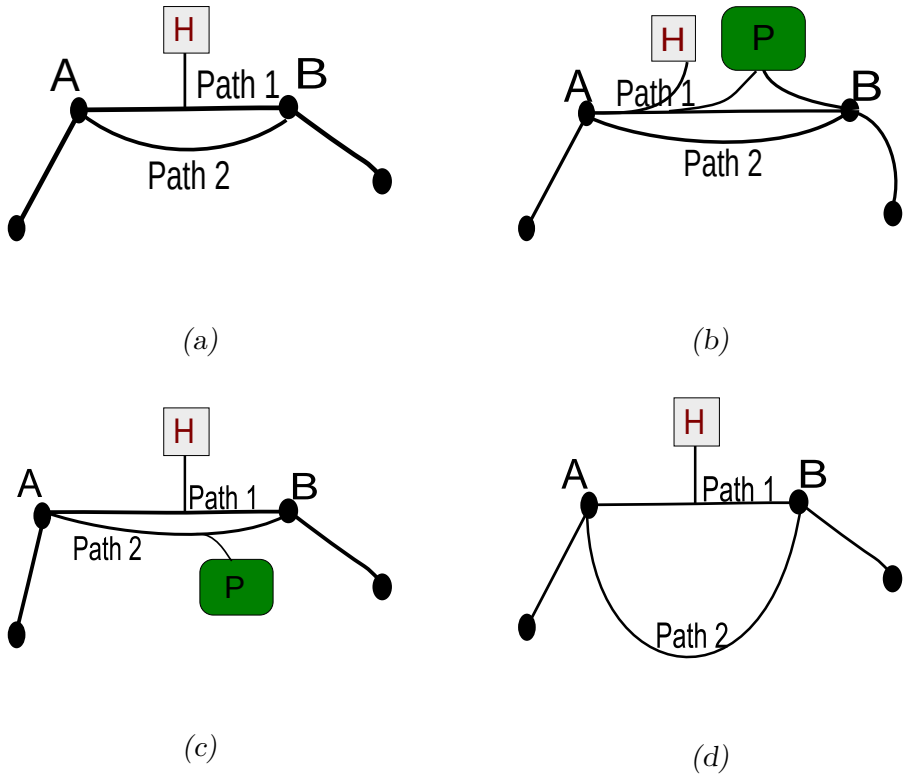


Figure 1.3: (a): No stay point on Path 2, (b)-(c): A park on the way from A to B, (d): Path-2 is long

Let Alice reveal location A at time t_A and then stop. After a medical visit, Alice starts sharing location again when she reaches a non-sensitive place B at time t_B. Considering the traffic speed, an adversary can figure out that Alice needs maximum time t to travel from A to B, where t is less than $(t_B - t_A)$. From this difference of time, the adversary can infer that Alice stays somewhere between A and B and if there is a single stay point, say a hospital, between A and B, then the adversary becomes confident that Alice stays in the hospital while traveling from A to B.

One may argue that there can be more than one path to travel from A to B and there is no guarantee that Alice takes the path through the hospital. However, the fact is that in some scenarios (Figure 1.3), even if there are more than one path, an adversary can identify that Alice visits the hospital.

Figure 1.3(a): There are two paths to travel from A to B. In this case, the probability of Alice to take Path 1 is 50%. Since there is no stay point in Path 2 and $t << (t_B - t_A)$, an adversary still can assume that Alice takes Path 1 and the probability of Alice to be in the hospital is 100%.

Figure 1.3(b)-(c): There is a park in addition to the hospital on the way from A to B and Alice

can spend time in the park other than hospital. Hence an adversary can infer that the probability of Alice to be in the hospital is 50%. On the other hand, if Alice travels from A to B on Sunday evening, when the park remains closed, then the adversary is 100% confident that Alice visits the hospital.

Figure 1.3(d): The minimum time to travel from A to B via Path-2 is greater than $(t_B - t_A)$. Thus, it is not possible for Alice to take Path 2 and an adversary can infer with 100% probability that Alice visits the hospital.

Hiding a user's identity while sharing location data cannot always protect a user's privacy [10–15] as the location data such as home or office address can act as an identifier. Once the identity of a user is determined, the privacy of sensitive location data revealed by the user is violated. On the other hand, identifies are often required for personalized services (e.g., friend finder applications) and authorization purposes. Thus we focus on hiding sensitive location data instead of identities.

A large number of approaches exist to protect location privacy in the Euclidean space [16–22]. However, users move through road networks and an adversary can exploit the constraints of road networks to infer a user's sensitive locations. In recent years, researchers have also proposed a few techniques [1, 23] for sharing location data in road networks. However, both of these techniques are vulnerable to privacy attacks. In [1], when a user is on a sensitive location, a false(safe) location nearby to the actual location is published. Since safe (false) locations for sensitive locations are precomputed irrespective of time, the safe location may remain closed when it is published. Thus, the solution proposed in [1] fails to protect location privacy in real-time. Furthermore, this approach does not take *velocity based linkage attack* [16] into consideration. A user's movement is restricted by the maximum allowed speed in road networks and thus, if a user's shared safe (false) location falls outside the maximum movement bound with respect to the user's last revealed location, an adversary can easily identify the revealed location as false using the velocity based linkage attack.

In [23], the authors use popular cloaking techniques to protect location privacy under road network constraints. Using the cloaking technique, instead of sharing a sensitive location, the user reveals an area that includes the user's sensitive location. The privacy requirement of a user is expressed in terms of the popularity, i.e., the number of people located at places (e.g., a bank, school, hospital) included in a cloaked region. The popularity of places may vary with time and the area of the cloaked region location can change to satisfy the privacy requirement of users in real-time. In [23], the authors propose that to reduce the computational overhead in real-time, the cloaked regions for sensitive locations can be precomputed. However, like [1], precomputed cloaked regions may not always satisfy

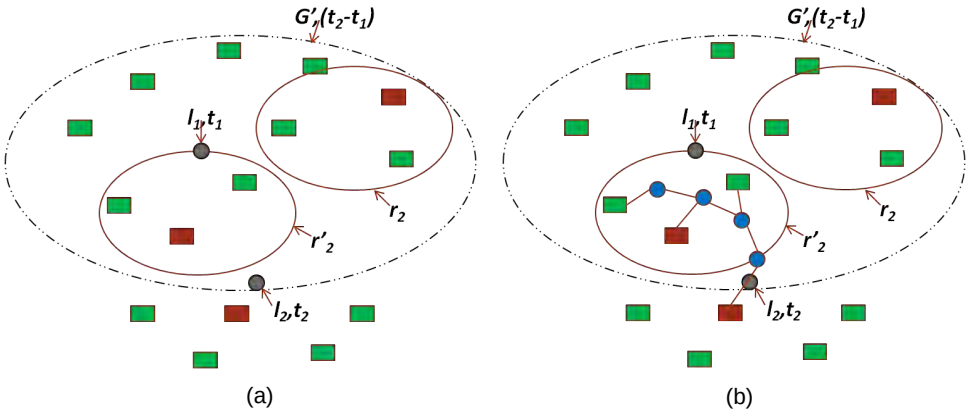


Figure 1.4: An example of the upcoming linkage attack, where red and green places are considered as sensitive and safe, respectively

the privacy requirement of users due to change in popularity of places (e.g., the popularity of a place may drop to 0 if the place remains close at a certain time). We call this attack on a user's location data as *temporal linkage attack*.

To address the temporal linkage attack, the authors in [23] also propose an computationally expensive online algorithm to compute cloaked regions in real-time. However, online cloaking also fails to protect privacy due to the lack of knowledge about upcoming sensitive locations. Assume that a user shares l_1 at time t_1 (Figure 1.4(a)). While revealing the location at time t_2, the algorithm only considers the subgraph G' which consists all the vertex and edges reachable from l_1 within $t_2 - t_1$ time and computes cloaked regions r_2 and r'_2 for red sensitive places. If the user's location at time t_2 falls inside r_2 or r'_2, the user reveals corresponding cloaked region. Otherwise, the user reveals her actual location. Let the user's location is at l_2 as shown in Figure 1.4(a), and thus the user reveals l_2 at time t_2. If we consider the city network in Figure 1.4(b), we see that the user is actually heading to a sensitive location, which has not been recognized by the solution at time t_2. This implies that the user shares her sensitive movement with others because of not knowing about upcoming sensitive locations. We call this attack as the *upcoming linkage attack*. Another limitation of this approach is that it frequently fails to satisfy privacy requirements of users, stops sharing location data and reduces the utility of the revealed information.

To overcome the limitations of existing approaches, we introduce a new concept called *warning zone*. Warning zones enable us to overcome the upcoming linkage attack, increase the utility of location data and reduces the computational overhead to protect location privacy in real-time. Each

warning zone consists of a sensitive location and places and road networks whose disclosure may enable an adversary to apply the upcoming linkage attack. If a user's location falls inside the warning zone, our approach checks whether the user can reveal her actual location or not based on the popularity of places in real-time. If not then our approach compute the cloaked region in real-time. On the other hand, if the user's location falls outside the warning zone, our approach reveals the user's actual location. Existing approaches stop sharing locations when it is not possible to protect a user's sensitive location using a cloaked region because of previously revealed locations. Computation of warning zones enable us to prethink about upcoming sensitive locations and thereby reduce the frequency of not sharing locations and increase the utility of location information. Furthermore, warning zones refine the search space in road networks, where a user needs to consider cloaked regions and thus, reduce the computational overhead of our real-time privacy preserving algorithms.

1.2 Contributions

In summary, the contributions of this thesis are as follows:

- We identify the possible privacy attacks in existing privacy preserving approaches and develop solutions to overcome the identified attack.
- We develop an approach that ensures both privacy and utility of location data of users. Our approach does not need to trust anyone to protect location privacy of users.
- We present efficient algorithms that can protect privacy of a user's location with reduced computational overhead and share location data in real-time.
- We validate the effectiveness and efficiency of our proposed approach using a real dataset.

1.3 Organization

The remainder of the thesis is organized as follows. In Chapter 2, we formulate the problem and present the privacy model considered for our system. Related works are discussed in Chapter 3. The detail of our approach is proposed in Chapter 4. In Chapter 5, we present a comprehensive security analysis and in Chapter 6, we elaborate our experimental evaluation of the proposed approach. Finally, Chapter 7 concludes the thesis.

Chapter 2

Problem Formulation

In this chapter, in Section 2.1, we first discuss the terms and concepts we use throughout the thesis and then in Section 2.2, we present the privacy model we consider for our approach.

2.1 Preliminaries

City network: City network is a connected weighted graph $G = (V, E, W)$, where

- V denotes a set of vertices and each vertex $v \in V$ represents a place or a road junction.
- E denotes a set of edges and each $e \in E$ represents a road segment connecting two vertices.
- W denotes a set of weights and each $w_e \in W$ represents the minimum time required to travel the edge $e \in E$.

Place type: PT denotes a set of place types in the city network G. The type of a place v in denoted by $v.pt$, where $v.pt \in PT$ can be a school, hospital, park etc. Based on a user's privacy profile, place type is divided into sensitive PT_S and non-sensitive PT_{NS} sets. For example, university and park are non-sensitive and a hospital is sensitive for a specific user (Figure 2.1).

Popularity: POP denotes a set of popularity of places in the city network G. The popularity of place is denoted as $v.pop$, where $0 \geq v.pop \leq 1$. The popularity of a place means the probability of a random user to be located in v at a specific time t. The popularity varies with time, for example, a school has almost 0 popularity at night if it does not have night shift. We also consider that the

popularity of a road junction and edges are 0 for our computation. In Figure 2.1, the popularity of a university and a hospital are 0.4 and 0.5, respectively.

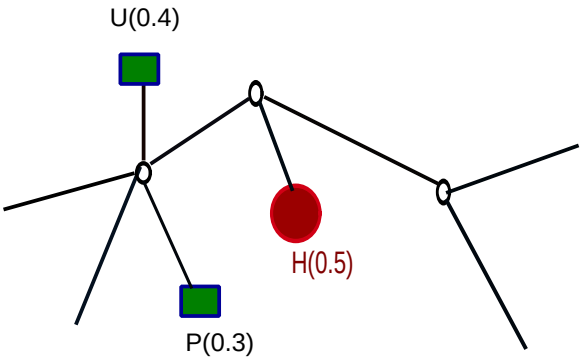


Figure 2.1: Example of city network

Cloaked region r: Cloaked region r is connected sub-graph of city used for revealing location instead of a user's actual location. r should maintain the following properties:

- should contain one sensitive place PT_S and atleast one non-sensitive place PT_{NS}.
- should satisfy minimal disclosure requirement. Let a user's defined disclosure threshold value is β_{pt_s} indicating the maximum allowed probability of a user being in a place of such type. Hence probability of such place in a specific region r should not exceed β_{pt_s} for that specific user. Hence privacy requirement for single r is:

$$\frac{(v.pop)_{pt_s}}{r.pop} \leq \beta_{pt_s}$$

where $r.pop$ is popularity of the specific cloaked region r.

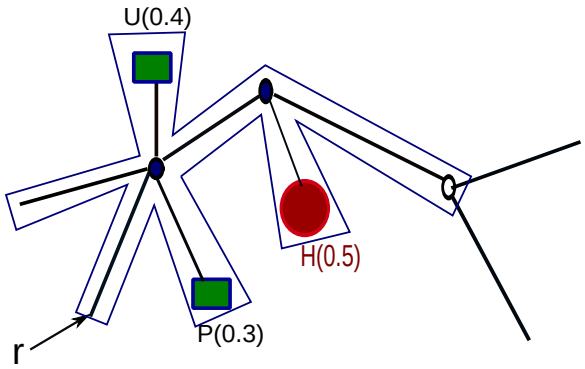


Figure 2.2: Cloaked region r for hospital(H)

In Figure 2.2, hospital(H) is sensitive and university(U) and Park(P) are non-sensitive. Popularity for hospital is 0.5, for university is 0.4 and for park is 0.3 . Assume that disclosure threshold for hospital is 0.5. Hence cloaked region r in the Figure 2.2 satisfies the user's privacy requirement because in the region r the probability of the user being in a hospital $(0.5/0.5 + 0.4 + 0.3 = 0.42)$ is less than or equal to 0.5.

When revealing sequence of locations there is a risk of velocity based linkage attack[16]. Analyzing the velocity and previously reported location, adversary can limit the area of a user's position within reported r. Therefore, to prevent this attack, we define another property for r. As we said before, each edge is weighted a minimum time to traverse that edge. If previous position is a cloaked region r_1 at time t_1 then next revealed cloaked region r_2 at time t_2 should maintain the following property:

- time-distance from r_1 to r_2 is defined as maximum required weighted time of shortest path between any vertex in r_1 to any vertex in r_2. Since time-distance is measured by r to r, we can say that if a user reveals exact location then the cloaked region r contains one vertex.
- Hence the privacy requirement for these sequenced r is : time-distance from r_1 to r_2 (denoted by $d(r_1, r_2)$) is lower than the time t spent between previous request t_1 and current request t_2. i.e.

$$d(r_1, r_2) \leq t$$

Warning zone of a sensitive place: Warning zone consists of one sensitive location with all the vertices and edges related to that sensitive location maintaining a user's required l diversity. Warning zone means we not only consider the sensitive stay point to take action against privacy breach but also consider the sensitive movements that can be used for upcoming linkage attack. We calculate a warning zone wz_v with $l-$diversity (Figure-2.3) for each sensitive place v of a user. Bamba et al.[24] first defines $l-$diversity for a cloaked region i.e. a region with $l-$ diversity contains l places. We adopt the concept and in our solution it means warning zone wz_v has l different outgoing road directions towards l non-sensitive locations. If a user is in any warning zone then we have to take necessary steps for privacy protection.

User Profile PP_{user}: User profile denoted by PP_{user}, consists of list of sensitive place types(PT_S) with their minimal population requirement for disclosure(β).

$$PT_S = (pt_{s_0}, \beta_{s_0}), (pt_{s_2}, \beta_{s_2}), ..., (pt_{s_n}, \beta_{s_n})$$

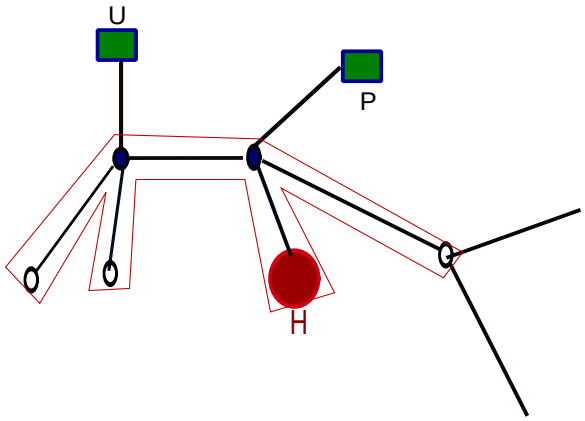


Figure 2.3: Warning zone for hospital(H)

For example, Hospital(pt_{s_0}) is sensitive for a user and its minimal disclosure requirement is .25(β_{s_0}). That means the probability of a random user to be located in that type of place at a specific time t should be less than or equal to .25 when sharing location. A user also gives input about the number of diversity l she needs to create warning zone wz and maximum allowed time delay θ to process her request of sharing location.

$$PP_{user} = (l, \theta, PT_S)$$

Notations, those are used in this thesis are summarized in Table 2.1

2.2 Privacy Model

An adversary can have information about the city network that includes the popularity of places, a user's published locations and the privacy algorithm used to protect location privacy. We assume that an adversary does not know about a user's privacy profile. However, an adversary can predict which type of place is sensitive for a user by reverse engineering with previously published cloaked regions r. Hence we consider that an adversary has the information about a user's sensitive places. The adversary can apply the velocity based linkage attack, the temporal linkage attack and the upcoming linkage attack.

The user's privacy is the maximum and the utility of location data is 0, if a user does not reveal locations in any form (actual or cloaked). On the other hand, the utility is the maximum and privacy is 0, if a user reveals actual locations of her movements. To balance between privacy and utility, we assume that a user specifies the required privacy level in terms of disclosure threshold, i.e., the

Table 2.1: Notations and their meanings

Symbol	Meaning of the symbols
G	A graph representing a city network
V	A set of vertices in G and each vertex $v \in V$ represents a place or a road junction.
E	A set of edges in G and each $e \in E$ represents a road segment connecting two vertices.
W	A set of weights in G and each $w_e \in W$ represents the minimum time required to travel the edge $e \in E$
PT	A set of place types, where $v.pt$ represents the place type of v and $v.pt \in PT$
POP	A set of popularity of places, where $v.pop$ represents the popularity of a place v and $v.pop \in POP$
WZ	A set of warning zones, where $wz_v \in WZ$ represents a warning zone for a place v
r	A cloaked region
$d(r_1, r_2)$	the required maximum time distance between two vertices of cloaked regions r_1 and r_2
$PT_S = (pt_{s_0}, \beta_{s_0}), (pt_{s_2}, \beta_{s_2}), ..., (pt_{s_n}, \beta_{s_n})$	$PT_S \leftarrow$ Set of sensitive place types $(pt_{s_0}, \beta_{s_0}) \leftarrow$ Disclosure threshold β_{s_0} for sensitive place type pt_{s_0}
$PP_{user} = (l, \theta, PT_S)$	$PP_{user} \leftarrow$ A user's privacy profile $l \leftarrow$ Required diversity $\theta \leftarrow$ Maximum allowed time delay to process request $PT_S \leftarrow$ Set of sensitive place types with their minimal disclosure threshold

user does not want to be identified at a sensitive place with a probability higher than the disclosure threshold.

We propose an approach that ensures the required privacy level of a user while maximizing the utility of location data. We develop techniques to overcome the velocity based linkage attack, the temporal linkage attack and the upcoming linkage attack and thus, adversaries cannot refine a user's sensitive locations with a probability higher than the disclosure threshold using linkage attacks.

Chapter 3

Related Works

Privacy-preserving location publishing has been extensively studied in recent years. There exist many different models like False locations/path, Cloaking methods, Position sharing approaches etc to protect from revealing sensitive data when a user uses different types of LBSs. Some of the approaches work in Euclidean space and some consider the road network constraints. Section 3.1 describes different types of privacy techniques to protect a user's privacy and Section 3.2 elaborates the approaches based on Euclidean space and road network. Finally in Section 3.3, we discuss the approaches whether they process a user's request with offline or online data.

3.1 Different Types of Privacy Techniques

3.1.1 False locations

The basic idea is to send either one or more fake locations that are related to the user's actual location [22, 25]. Without the help of any trusted third party, a mobile user can generate fake location trajectories, called dummies to protect trajectory privacy. A trajectory is the path that a moving object follows through space. [22] proposed an anonymous communication technique in which a user sends position data with some dummies different from the actual position generated using Dummy generation algorithm based on realistic user movements. However in long run an adversary can distinguish a user's true trajectory from dummies by exploring data mining technique based on moving pattern of users [26]. The authors in [25] proposed two scheme to derive dummy trajectories that exhibit long-term user movement patterns. Random pattern scheme randomly generates dummies with

consistent movement patterns and the key idea of the rotation pattern is to have some intersections between trajectories of dummies and the user. However these dummy locations could still fall within sensitive areas since there is no distinction between sensitive or non-sensitive locations.

3.1.2 Cloaking techniques

A typical approach to protect location privacy is to generate a cloaked region (r) [16–18, 23, 24, 27–36] that encloses a user's position. A broad range of existing cloaking techniques rely on k anonymity [27–30, 32, 33] where the region contains $k-1$ other LBS users along with the requestor or k anonymity with l diversity [24, 35] where region consists of l places or l different types of places.

Initially k anonymous is introduced to protect shared medical data such that an adversary can not distinguish an individual record from other $k-1$ records [37] which is then extended by l diversity [38]. The authors in [39] develop a privacy notion t closeness which is a further refinement of l-diversity approach where each sensitive attribute has l well represented values. The distance between two distributions of a sensitive attribute should be no more than the disclosure threshold t.

A region is considered location k anonymous if the location information inside that region shared by a user to LBSs is indistinguishable from the locations shared by other $k-1$ users. Location k-anonymity is first studied by Gruteser et al. [27]. However location privacy should be based on personalized preferences unlike [27] where users do not have the ability to define their own privacy requirements. [31] presented a framework called personalized anonymity technique for the protection of sensitive attributes while providing k anonymity. However the approach aims to protect data privacy. The authors of [32] proposed a solution to provide a personalized k at per user level instead of a uniform k for all users. A user can specify the minimum value k with specific privacy requirement like maximum level for spatial and temporal error a user can accept while preserving k anonymity of a region. The techniques used in [29, 40] also support customization the value of K. However these solutions need to rely on a anonymity server which works as a middleware server between a user and service providers. The author in [41] developed an cloaking algorithm which works in mobile peer to peer environment. The k anonymity process has been extended to support road networks where a users location is cloaked into a region satisfying the privacy requirements of k-anonymity and containing at least l different road segments [33].

However, location k anonymity approaches fail to protect privacy since all k users may belong

to a single location. PrivacyGrid, a framework for supporting anonymous location-based queries is developed by Bamba et al. which satisfies required location $k-$anonymity and each region with l diversity where l diversity means l places [24]. However they did not distinguish between sensitive and non-sensitive place types for a user when considering l diversion. All l places can be of same sensitive place type. To overcome this problem, Xue et al. [35] defined l diversity as the number of different place types.

Problem of these solutions is that they consider only the number of users and types of places in the cloaked region. However number of users can differ with different types of places and the impact of a huge populated place in a cloaked region is omitted. For example, a cloaked region can be built with k anonymity and l diversity where all k users belong to one sensitive location. Gruteser et al. [34] classified a area as sensitive or non-sensitive. When a user's location is in a region surrounded by sensitive areas, the algorithm reports an area containing $k1$ other sensitive areas. Hence an adversary can not distinguish one sensitive area where the user belongs. However still the adversary has the information that the user has visited her sensitive place.

The Probe framework (Privacy Preserving Obfuscation Environment) [18] overcomes this limitations. It provides a set of alternative cloaking heuristics which blur sensitive semantic positions based on a user's preferences. All locations on the map are represented as features, and each feature has a type, sensitive (e.g., hospitals, bars) or innocuous (e.g., shopping centers, parks). Each user defines her own privacy profile which specifies sensitivity thresholds with respect to each feature type. PROBE generates cloaked regions that cover a mix of sensitive and innocuous regions such that the association probability between the user and sensitive features is bounded below the specified threshold.

Ghinita et al. [16] protects against velocity based linkage attacks that infer exact locations based on previously reported locations. They propose two techniques to preserve the privacy of user requests: temporal cloaking and spatial cloaking. Temporal cloaking is suitable when the partition of the map into cloaked regions is fixed in advance i.e., the map is partitioned into a set of tiles. In spatial cloaking, cloaked regions can be dynamically computed at the time of the request. Regions must be constructed taking into consideration the sets of sensitive features and associated sensitivity thresholds.

[17] proposed an extension of the Probe framework [18] to include criteria for the evaluation of the cloaking methods accuracy with respect to spatial accuracy bounds. They have presented the key features of a privacy-preserving approach for the personalized protection of sensitive locations.

Yigitoglu et al. [23] presents an extension of the semantic location cloaking model [17] originally

developed for the cloaking in an unconstrained space. Unlike some other cloaking approach, it considers road networks and places. For example, a user remains in the cloaked region which contains only the sensitive place for a long time, more than the time that is needed to traverse all the road inside the region. Hence an adversary can predict that the user walked into that sensitive place since there is no alternative place in the cloaked regions in which the user can spend much time. The user's sensitive stay point is thus disclosed. The authors in [23] defines cloaked region which contains one or more non-sensitive places. It also protects user from the velocity-based linkage attack. They developed two cloaking method: offline cloaking and online cloaking. In offline cloaking method, cloaked regions are pre-computed and when a user requests to share her location the solution transforms the user's actual location into a cloaked region. On the other hand, cloaked regions are built at runtime in online cloaking method.

3.1.3 Position sharing approaches

False path or cloaking technique none of them provide different precisions to different clients with different quality of service demands and trust levels. Position sharing approaches [19–21] protect a user's location privacy and also provide different precisions to different clients based on different quality of service demand and trust levels. It splits the precise user position into a set of imprecise position shares and distributes these shares among LSs of different providers. Location-based applications (LBA) can query these shares from the LSs and fuse them to a position of well-defined precision depending on the number of shares they got access rights for from the tracked user.

3.1.4 Mix-zones and path confusion

A mix zone exists whenever two or more users occupy the same place at the same time and their paths become indistinguishable. In Mix zone based approaches [10, 11, 42] when users enter a mix-zone, they change to a new and unused pseudonym such that the mapping between their old pseudonyms and new pseudonyms are not revealed. Hence when users exit the mix zone, an adversary can not predict which user takes which trail. Path Confusion approaches [43, 44] extended the method developed by mix zones. They incorporate a delay during sharing users' locations until the paths intersect such a way that they create anonymity. However, these techniques actually aim to publish trajectory data by protecting a user's identity not location privacy.

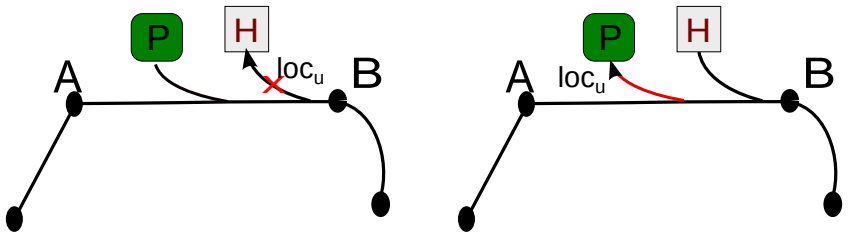
3.2 Approaches in Euclidean Space and Road Networks

3.2.1 Approaches in Euclidean space

Most of the researchers developed their solution working in Euclidean space [16–22] where a user can go through anywhere without any bindings. However in real world, users only move through the road network. If a user's locations are continuously reported, an attacker can correlate the regions through road networks from multiple timestamps to accurately predict the user's position inside a region.

3.2.2 Approaches in road networks

There are few works which consider road network constraints while sharing a user's location continuously. Claudio et al. [1] have developed a model to protect location privacy considering sensitive and non-sensitive locations in road network. Safe path/poi for a user is pre-computed using a user's historical movements based on Markov chains. Any unusual path/poi is also considered as sensitive. Whenever a user is on a sensitive/unusual path/poi, a false safe path/poi is published nearby to that sensitive path/poi.



(a) (b)

Figure 3.1: (a) Real path (b) Released path in [1]

For example in Figure 3.1(a) a user's location loc_u falls into a sensitive path which is heading to a hospital, a sensitive location for the user. The solution [1] detects the risk and instead of publishing the user's actual location, it publishes the pre-computed safe path indicated by red arrow in Figure 3.1(b).

Static cloaking techniques which aim to protect location privacy for snapshot LBSs under road network constraints [33, 45–47] are not applicable. Snapshot LBS means the location based service

where a user shares her location once to get the service. Hence an adversary can limit a user's actual area inside the cloaked region using temporal correlation with spatial information (e.g., time needed to traverse the roads of the city) while sharing location continuously. The authors of [16] identified velocity based linkage attack while a user shares her location data continuously. However they did not consider the road network constraints. The authors in [23] developed the solution in which the cloaked region are built under road network constraints during sharing location data continuously. A cloaked region for a sensitive place contains non-sensitive places required to meet a user's privacy requirement along with one sensitive place and the places are connected through road networks. (Figure 3.2).

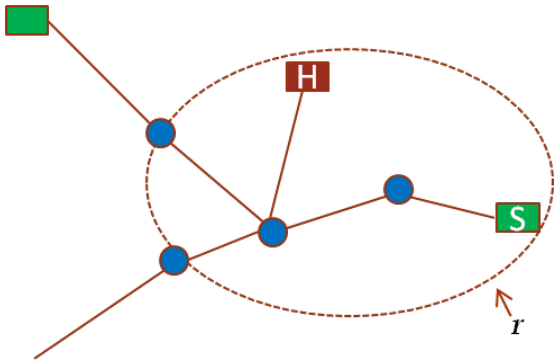


Figure 3.2: Cloaked region in road network

Before publishing a cloaked region instead of a user's actual location while a user requests to share her location, solution in [23] checks whether all the places and roads inside the region are reachable from the previously shared location through road networks within the time gap. There also exist some approaches under road network constraints which preserve a user's privacy by protecting her identity [12, 13, 48, 49]. However all the personalized and authorization based services that need a user's identity can not be available. Hence our aim is to protect a user's privacy while publishing her location continuously with her identity.

3.3 Offline and Online Approaches

3.3.1 Offline approaches

Offline solution means safe regions or paths which are needed to publish instead of a user's actual location, are either pre-computed or generated based on historical data. The solution in [50] ensures k

anonymity using $k-1$ other individuals' historical footprints instead of using their real-time locations. However an adversary might find only the requesting user or less than k users into that region considering real-time. Another solution called CliqueCloak [51] also compromises real-time operation since it waits until it gets k different queries requested from a particular region.

Claudio et al. [1] pre-computes a user's safe poi/path using a user's historical movements based on Markov chains model. Markov model is a type of predictors where the mobility behavior of an individual is represented as a Markov model and the next location is predicted based on the previously visited locations [52–55]. However, the solution in [1] does not consider the velocity based linkage attack while publishing a user's location. On the other hand, population of poi varies time to time. Hence it fails to protect privacy due to ignoring time constraint as safe poi which is pre-computed based on a user's historical movement may not have the validity for the requesting time period. The offline cloaking method developed in [23] also does not consider the population of poi under time constraints since cloaked regions are pre-computed. For example, a cloaked region for hospital is pre-computed (Figure 3.2) with a hospital and a school which remains closed at night. Hence at night an adversary can predict that a user belongs to the hospital inside the cloaked region if she spends more time than needed to traverse all the roads of that region.

3.3.2 Online approaches

Some anonymization cloaking processes like [56–58] ensure k anonymity by blurring a user's actual location or path into a region containing $k-1$ other users at real time. However, previously we described that k anonymity process can not provide location privacy if all k users belong to one sensitive location or path. Online cloaking method developed in [23] overcomes it along with all the problems described for offline techniques. Since cloaked regions are built at runtime, instead of considering whole city network the solution only considers the sub graph containing all the places and roads reachable from previously shared location within the time difference. It then publishes a cloaked region instead of a current location when it finds that a user's position falls into a cloaked region(r) of a sensitive location inside the sub graph. However they are not aware of upcoming sensitive locations and can not provide privacy against upcoming linkage attack. On the other hand, sub-graph and cloaking regions are generated at run time based on time and popularity of the places at that time. Hence size of the sub-graph and number of cloaked regions within it can affect the effectiveness and

efficiency of location-dependent services.

In our approach, we propose a solution which protects a user from revealing her sensitive information while sharing real-time location continuously. The solution checks whether a user needs to reveal a cloaked region even if she is not in a sensitive location and if needed, it publishes a cloaked region instead of the user's actual location. We also improve the rate of service drop and the risk of privacy violation because of the service drop.

Table 3.1 summarizes the limitations in different techniques which we overcome in our solution.

Table 3.1: Privacy preserving techniques with their limitations

Techniques	Temporal linkage attack	Velocity based linkage attack	Upcoming linkage attack	Utility	Efficiency
False location based on Markov chains [1]	x	x	x	high	high
Online spatial cloaking[23]	✓	✓	x	low	low
Position sharing approach[21]	x	x	x	high	high
Our approach	✓	✓	✓	high	high

Chapter 4

Our Approach

In this chapter we elaborate our proposed approach to ensure a user's privacy requirements while sharing location data with others. A user provides her required l diversity, disclosure threshold β and maximum allowed time delay θ with city network G into her privacy profile. Our aim is to protect the user's privacy from all types of linkage attacks based on the privacy profile. Our approach is a combination of offline and online process. In the offline part, we calculate the warning zone wz_v with the concept of l diversity for each sensitive location v in the map when a user first defines her privacy profile. Warning zones allow the system to pre-think whether the disclosure of a user's actual location enables others to infer that the user has visited a sensitive place and take privacy actions based on that. Transformation process (i.e. generating a cloaked region and revealing a user's location) is online such that we can calculate the real-time probability to associate a user with her sensitive location inside a cloaked region. During service request when a user wants to share her location, if the user's actual location falls into any of her warning zones then we create the cloaked region for that warning zone. After creating the cloaked region, we check whether the user's actual location falls into that cloaked region or not. If the user lies into that region then system reveals the cloaked region instead of the user's actual location and if the user is outside of the region then system reveals the actual location. It may happen that a user's location falls into multiple warning zones. In that case, we create cloaked regions for all the warning zones and select a random cloaked region to publish amongst all cloaked regions which contain the user's actual location. On the other hand if the user is outside of her warning zone then her actual location is considered as safe to publish.

The remainder of this chapter is organized as follows. Process of generating list of warning zones

is stated in the next section. Section 4.2 provides the algorithm for creating a cloaked region for a warning zone. Finally, Section 4.3 states the process to publish a user's actual location or a cloaked region based on the user's privacy profile.

4.1 Generate List of Warning Zones (WZ)

Warning zones are computed offline. Every warning zone wz_v contains one sensitive place v and has l outgoing road directions towards l non-sensitive places. At first we create warning zones for each sensitive place. After that we also consider the non-sensitive places which are connected directly with the pre-computed warning zones. The process repeatedly uses the function $GenerateZone(G, v, PP_{user})$ to create a zone z_v with l diversity for every sensitive or non-sensitive place $v \in V$ in graph G which is used later to compute the list of warning zones. The function is invoked from Algorithm 1. Hence we first elaborate the procedure of generating a zone z_v with l diversity for a place v before going into the detail process to compute list of warning zones using Algorithm 1.

4.1.1 Generate a zone z_v with $l-$diversity

Pseudo-code of generating a zone z_v for a place v with $l-$diversity is given in Algorithm 1. The algorithm takes city network G, a place v and a user profile PP_{user} as input and returns the zone z_v with l diversity. We continuously add road junctions to extend a subgraph for the place v until we find the required diversity. Breadth-first-search (BFS) is used to generate the zone and the place v will be the starting vertex. The algorithm maintains a first-in, first-out queue Q to manage the set of vertices discovered during the searching algorithm. At first we initiate the zone z_v and the queue Q empty, mark the vertex v as visited and then initialize the queue Q containing just the vertex v (Line 1-4). The **while** loop of lines 5-21 iterates as long as required $l-$diversity is not found. For first iteration, Q only contains the root vertex v. Line 6 determines the vertex u at the head of Q and removes it from Q.

Algorithm 1: GenerateZone(G, v, PP_{user})

Input: City network $G = (V, E, W)$, sensitive or non-sensitive place v, a user's privacy profile $PP_{user} = (l, \theta, PT_S)$

Output: zone z_v

```
1:  z_v ← φ
2:  queue Q ← φ
3:  mark v visited
4:  ENQUEUE(Q, v)
5:  while Q ≠ φ do
6:    u ← DEQUEUE(Q)
7:    for all unmarked u' ∈ AdjacentList(u) do
8:      if u'.pt ∉ PT_S then
9:        mark u' visited
10:       if u'.pt ∈ PT_NS then
11:         decrease l by 1
12:       else
13:         z_v.addEdge(u, u')
14:         ENQUEUE(Q, u')
15:       end if
16:     end if
17:   end for
18:   if l ≤ 0 then
19:     break
20:   end if
21: end while
22: return z_v
```

The **for** loop of lines 7-17 considers each non-visited vertex u' in the adjacency list of u. If place type of u' is non-sensitive for the user then the algorithm marks the vertex u' as visited (Line 9). If u' falls into the user's non-sensitive place type, it means one diversity is found for the vertex v. Hence the algorithm decrease the value of l by 1 (Line 10-11).

Otherwise u' is a road-junction and we place u' at the tail of Q and add the edge(u, u') to the z (Line 12-14). When all the vertices on u's adjacency list are marked as visited, we check the number of still required l diversity in Line 18. If required l diversity is less than or equal 0 then it means generated zone z_v has atleast l different outgoing road directions towards l non-sensitive places. Hence BFS is stopped (Line 18-19). After completing BFS, we add all the original edges between vertices in resulting zone to preserve the shortest path among them. The output of this algorithm is the zone z_v for the place v with l diversity.

Figure 4.1 illustrates the progress of the Algorithm 1. Places are denoted by their initials and road-junctions are denoted by v_i where $i = 1, 2, .., n$. In this scenario, we create a zone for hospital H where $l = 2$ i.e. generated zone have to have atleast 2 outgoing road directions towards 2 non-sensitive places. First Q is initialized with the root vertex H (Figure 4.1(a)). H has only one adjacent vertex v_1. Hence after first iteration, H is removed from the head of queue Q and after traversing H's adjacent vertex v_1, v_1 is assigned at the tail of queue (Figure 4.1(b)). In Figure 4.1(c) v_1 is removed from queue Q and then the algorithm traverse its adjacent vertices v_2 and v_3. Since the vertices are road junctions, we add these two vertices (v_2 and v_3) into queue Q. Required diversity still is 2. In Figure 4.1(d), v_2 is removed from queue Q and after traversing its adjacent vertices U, v_4 and v_5, we found a non-sensitive place U according to the user profile. Hence l is decreased by 1 and add the other two vertices into queue Q. Now required diversity is 1. The process thus continued as long as l is greater than 0. Whenever the condition is fulfilled, BFS will be stopped [Figure 4.1(e)]. Output of the algorithm is zone for hospital H with $l = 2$ diversity [Figure 4.1(f)] after adding all the original edges between vertices in resulting tree.

4.1.2 Generate list of warning zone for sensitive and non-sensitive places

Since the procedure to generate zone with l diversity is stated, we can move to the main procedure i.e. to generate list of warning zones for sensitive and non-sensitive places. Algorithm 2 takes city network G and user profile PP_{user} as input and returns list of warning zones (WZ). At first, we initialize the list of all type of warning zones WZ and list of warning zone for both sensitive (WZ_S) and non-sensitive(WZ_{NS}) places empty (Line 1-3). The first **for** loop of lines 4-7 considers each sensitive vertex $v \in V$ in graph G. Zone z_v created for a sensitive vertex v is directly added to the

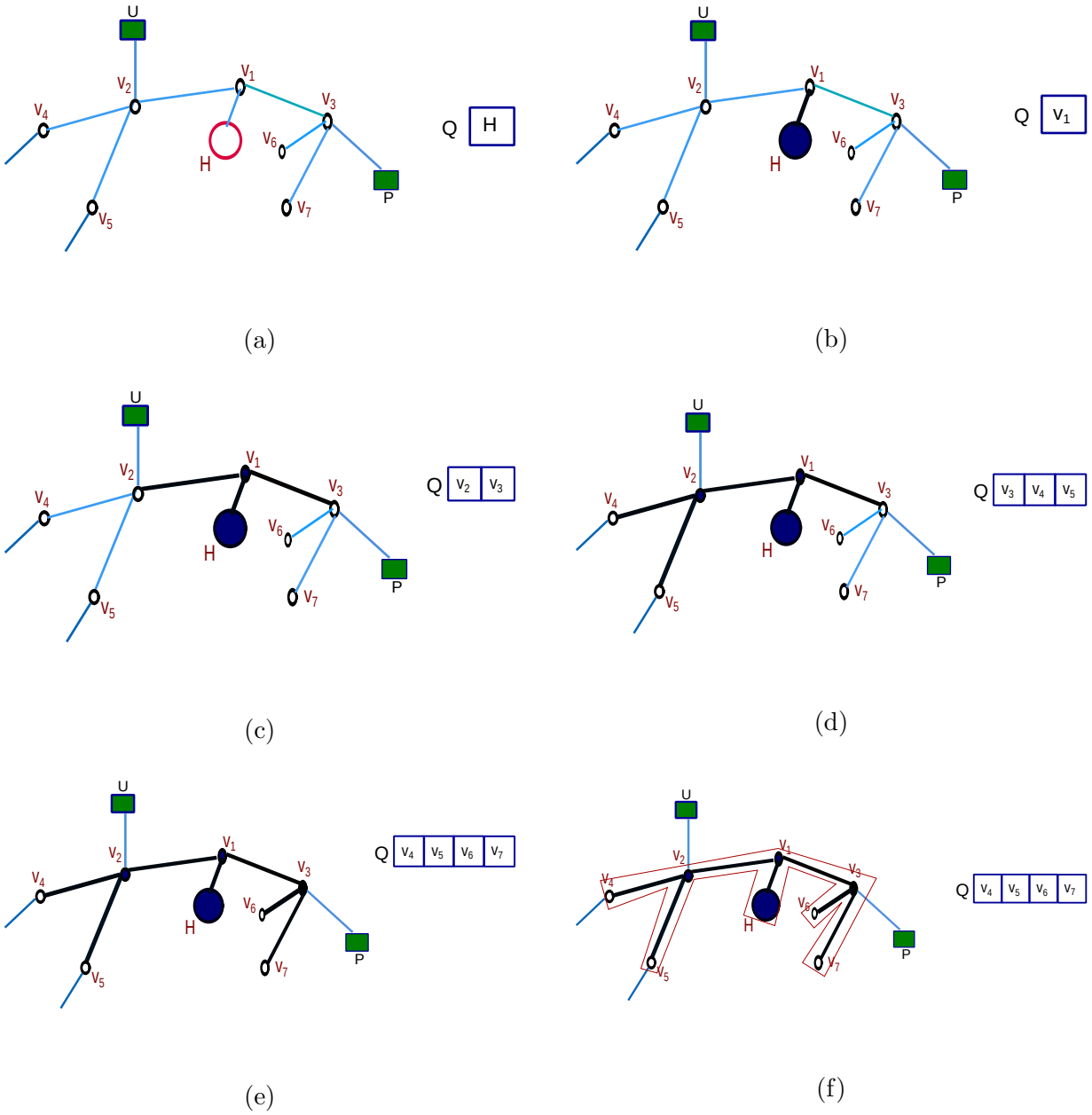


Figure 4.1: Generating zone for hospital H with l diversity where $l = 2$

list of warning zones of sensitive place WZ_S (Line 5-6) and after calculating all warning zones for sensitive places, WZ_S is added to WZ and WZ_{NS} (Line 8-9).

Algorithm 2: GenerateWZ(G, PP_{user})

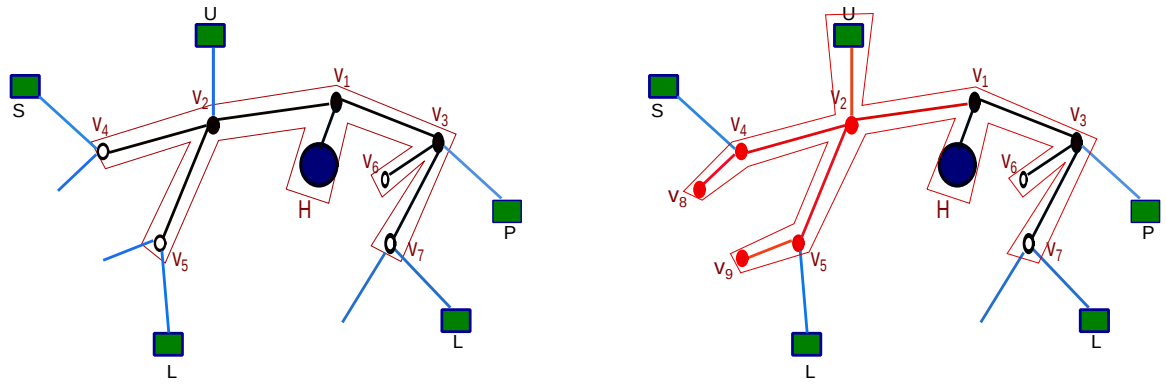
Input: City network $G = (V, E, W)$, a user's privacy profile $PP_{user} = (l, \theta, PT_S)$

Output: list of warning zone WZ

1: $WZ \leftarrow \phi$
2: set of warning zone for sensitive places, $WZ_S \leftarrow \phi$
3: set of warning zone for non-sensitive places, $WZ_{NS} \leftarrow \phi$
4: **for all** $v \in V$ s.t. $v.pop \neq 0$ and $v.pt \in PT_S$ **do**
5: $z_v \leftarrow GenerateZone(G, v, PP_{user})$
6: $WZ_S.add(z_v)$
7: **end for**
8: $WZ.add(WZ_S)$
9: $WZ_{NS}.add(WZ_S)$
10: **for all** $v \in V$ s.t. $v.pop \neq 0$ and $v.pt \in PT_{NS}$ and $AdjacentList(v) \pm wz_s$ where $wz_s \in WZ_S$ **do**
11: $z_v \leftarrow GenerateZone(G, v, PP_{user})$
12: **for all** $wz_{ns} \in WZ_{NS}$ where $z_v \pm wz_{ns}$ **do**
13: $WZ_{NS}.remove(wz_{ns})$
14: $wz_{ns} \leftarrow wz_{ns} \cup z_v$
15: $WZ_{NS}.add(wz_{ns})$
16: **end for**
17: **end for**
18: $WZ.add(WZ_{NS})$
19: **return** WZ

The algorithm also creates warning zones for non-sensitive places such that an adversary can not limit a user's actual location inside a cloaked region with the help of warning zones for sensitive places. The second **for** loop of lines 10-17 considers each non-sensitive vertex (except the road junction) $v \in V$ which is part of created warning zones of sensitive places(i.e. $AdjacentList(v) \pm wz_s \in WZ_S$) in graph G to generate zone z_v. If z_v intersects any other warning zone $wz_{ns} \in WZ_{NS}$, then our algorithm removes that wz_{ns} from WZ_{NS} and combines the warning zone wz_{ns} and z_v. The algorithm then adds the combined zone to the warning zone list of non-sensitive places WZ_{NS} (Line 13-14). After

calculating all warning zones for non-sensitive places, WZ_{NS} is added to WZ (Line 15). Output of the algorithm is list of warning zone (WZ) for both sensitive and non-sensitive places.



(a) (b)

Figure 4.2: (a) Warning zone for hospital(H) with $l = 2$, (b) Combined warning zone for hospital(H) and university(U) after generating the zone for university(U)

In Figure 4.2(a), we have warning zone for hospital using Algorithm 1. Assume, university is non-sensitive for the user and it is a part of warning zone of hospital. Hence, we create the zone of university(U) with $l = 2$. In Figure 4.2(b), a zone created for university(indicated by red edges) intersects previously created warning zone for hospital. Hence Algorithm 2 combine these two zones and the combined zone is added into the list of warning zone WZ.

4.2 Online Cloaking Method

Generating cloaked region and transformation of a user's actual location is done during service request. Population of a place may vary time to time. Hence we create cloaked regions online to preserve a user's location privacy from temporal linkage attack. Pseudo-code of the cloaking method is given in Algorithm 3.

Algorithm 3: CreateCR($G, POP, wz_v, PP_{user}, r_1, t$)

Input: City network $G = (V, E, W)$, popularity list POP, a warning zone wz_v, a user's privacy profile $PP_{user} = (l, \theta, PT_S)$, previously shared location r_1, time difference between two requests t

Output: cloaked region r_v

$r_v \leftarrow \phi$
$r_v.pop \leftarrow 0$
queue $Q \leftarrow \phi$
$v \leftarrow$ associated sensitive place of wz_v
Let the type of place v is pt_s and the required disclosure threshold is β_{pt_s}
consider the warning zone wz_v as a compound vertex $v_{compound}$
$r_v \leftarrow wz_v$
$r_v.pop \leftarrow wz_v.pop$
if $\frac{v.pop}{r_v.pop} \leq \beta_{pt_s}$ **then**
  return r_v
end if
mark $v_{compound}$ visible
$ENQUEUE(Q, v_{compound})$
while $Q \neq \phi$ **do**
  $u \leftarrow DEQUEUE(Q)$
  for all unmarked $u' \in Adj(u)$ **do**
    mark u' visible
    if $u'.pt \notin PT_S$ and $d(r_1, u') \leq (t + \theta)$ **then**
      $r_v.addEdge(u, u')$
      $ENQUEUE(Q, u')$
      if $u'.pt \in PT_{NS}$ **then**
        $r_v.pop \leftarrow r_v.pop + u'.pop$
        if $\frac{v.pop}{r_v.pop} \leq \beta_{pt_s}$ **then**
          return r_v
        end if
      end if
    end if
  end for
end while

The algorithm takes city network G, set of popularity POP, warning zone wz_v of the sensitive place v for which a cloaked region will be created, a user's privacy profile PP_{user} and previously shared location r_1 with time difference between two requests t as input and returns the cloaked region r_v which meets the user's privacy requirement. The algorithm uses BFS for creating a cloaking region for a sensitive place. Since both warning zone and cloaked region use BFS with same sensitive place, we reuse the warning zone to create a cloaked region. We continuously add non-sensitive locations to the warning zone and each time we merge a non-sensitive place, we calculate the probability of associating a user with her sensitive location. The process continues until the region meets the user's disclosure threshold.

At first the algorithm initializes the region r_v and queue Q empty and assigns associated sensitive vertex of the warning zone into vertex v (Line 1-4). Since we already used BFS algorithm for generating a warning zone, Algorithm 3 considers the warning zone of that sensitive place as a compound vertex and start BFS from that compound vertex as root to create a cloaked region instead of starting from the sensitive place v (Line 6). Computation time for creating one cloaked region is thus minimized from previous existing solution. In lines 7-8, the algorithm initializes the cloaked region r_v with warning zone wz_v and assigns the popularity of the warning zone into the cloaked region's popularity. If the cloaked region r_v meets the user's privacy requirement then the algorithm returns the resulting r_v (Line 9-11). Otherwise the algorithm inserts the compound vertex $v_{compound}$ into Q. The **while** loop of lines 14-29 iterates as long as generated cloaked region r_v does not meet the user's privacy requirement. In the iteration, it first determines the vertex u at the head of queue Q and removes it from Q. The **for** loop of lines 16-28 considers each non-visited vertex u' in the adjacency list of u. If u' is not a sensitive place for the user and the time needed to travel from r_1 to u' is less than or equal to $t + \theta$ (condition checked for velocity based linkage attack) then the algorithm adds the edge (u, u') to the r_v (Line 19) and places u' at the tail of Q (Line 20). If u' is a non-sensitive place then the algorithm adds the popularity of u' to the popularity of r_v *i.e* $r_v.pop$ (Line 22). During traversal, the algorithm checks whether the cloaked region r_v meets the user's privacy requirement or not. If yes then BFS traversal is stopped (Line 23-24) and the algorithm adds all original edges between vertices in the resulting tree. Output of the algorithm 3 is called a cloaked region r_v for that sensitive place v.

An example of creating a cloaked region from a warning zone is illustrated in Figure 4.3. The graph in Figure 4.3(a) shows one sensitive place: hospital(H), some non-sensitive places: like park(P)

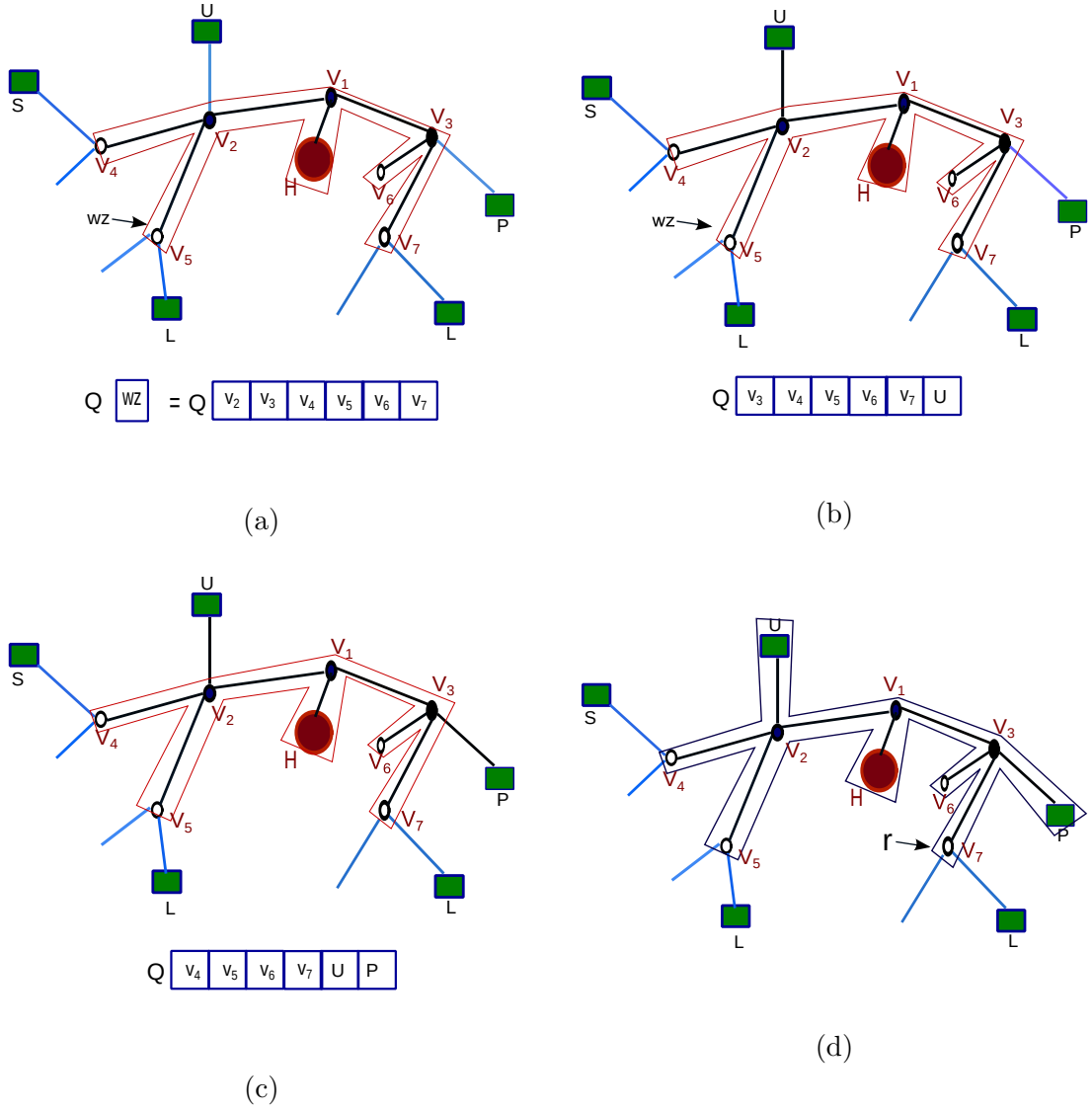


Figure 4.3: Creating cloaked region for hospital H

, university(U), school(S) etc and previously generated warning zone for hospital (H). We assume that all type of places have same popularity (0.2) and the disclosure threshold for hospital is 0.4 based on the user's privacy profile. According to this algorithm, it considers the warning zone wz as a compound vertex $v_{compound}$. The algorithm maps wz_v into a cloaked region r_v and since wz_v contains only one sensitive place(hospital), $wz_v.pop$ becomes 0.2. Hence, $r_v.pop$ is also 0.2 and the disclosure threshold is 1 which does not meet the user's privacy requirement. Then the algorithm inserts the compound vertex i.e. all the leaf nodes for warning zone wz_v into the queue Q. In Figure 4.3(b), it removes vertex v_2 from queue, traverses it's adjacent vertex i.e. a non-sensitive place: university U and

then inserts the vertex U into r_v. After adding U, $r_v.pop$ becomes 0.4 and the disclosure threshold is 0.5. Since the resulting zone still does not meet the user's privacy requirement, the iteration continues in Figure 4.3(c). In this figure, the algorithm removes vertex v_3 from the head of queue and traverses its adjacent vertex. Adjacent vertex (P) is non-sensitive and after adding the vertex into r_v, $r_v.pop$ becomes 0.6. Since $\frac{H.pop}{r_v.pop} \leq 0.4$, resulting r_v satisfies the constraint for hospital. Figure 4.3(d) then shows the created cloaked region for hospital(H).

4.3 Transformation

Pseudo-code of the transformation process is given in Algorithm 4 that takes city network G, set of popularity POP, set of warning zones WZ, a user's privacy profile PP_{user}, service requested time t_{req}, a user's location loc_u as input.

Let r_1 be the user's previously revealed location/cloaked region at time t_1 (Line 2). When a user requests to share her location, first the algorithm checks whether the user needs to reveal a cloaked region or not i.e. the algorithm checks if the user falls into any of her warning zones or not (Line 3). If the user's location is outside of her warning zone then the user can reveal her actual location (Line 4-5). If a user's actual location falls into any of her warning zones then the algorithm considers that warning zone from the set (WZ) for a cloaked region r_v and adds it to the list of cloaked regions CR (Line 7-14). After creating cloaked regions needed, the algorithm checks whether the user's actual location lies into any of those cloaked regions $r_v \in CR$ or not. If yes then it checks for velocity based linkage attack. Cloaked region r_v is considered as safe to publish if all the vertices of cloaked region r is reachable from previously shared location r_1 within $(t + \theta)$ where t is denoted for time between previous and current request and θ is denoted as threshold value for maximum allowed time delay) (Line 15-23).

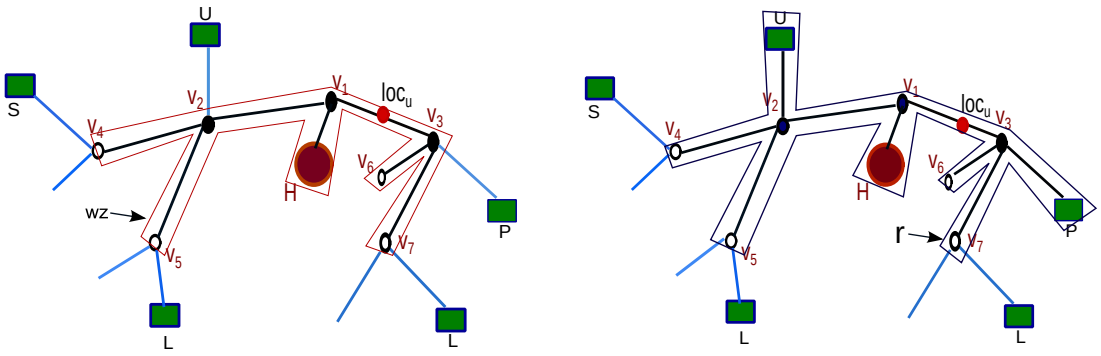
On the other hand if the user is not in any generated cloaked regions then the algorithm publishes the user's actual location. A user's actual location may fall into multiple cloaked region. In that scenerio, the algorithm randomly chooses a cloaked region to publish. If Algorithm 3 fails to create a cloaked region due to not having enough non-sensitive places then service will drop (Line 15).

Algorithm 4: Tranformation($G, POP, WZ, PP_{user}, t_{req}, loc_u$)

Input: City network $G = (V, E, W)$, Popularity list POP, list of warning zone WZ, a user's privacy profile $PP_{user} = (l, \theta, PT_S)$, service requested time t_{req}, a user's location loc_u

Output: a user's actual location or a cloaked region r_v

1: Set of cloaked region $CR \leftarrow \phi$
2: Let, previously revealed location is r_1 at time t_1
3: Determine the list of warning zones $WZ_u \in WZ$ which contains the user's actual location loc_u at time t_{req}
4: **if** WZ_u is ϕ **then**
5: **return** r_v and time t where $t \leftarrow (t_{req} - t_1)$
6: **end if**
7: **for all** $wz_v \in WZ_u$ **do**
8: $r_v \leftarrow CreateCR(G, POP, wz_v, PP_{user}, r_1, t)$
9: **if** r_v is null **then**
10: $FlagForFailedCR \leftarrow 1$
11: **else**
12: $CR.add(r_v)$
13: **end if**
14: **end for**
15: **if** $loc_u \in r_v$ where $r_v \in CR$ **then**
16: **if** $d(r_1, r_v) \leq (t + \theta)$ **then**
17: **if** $d(r_1, r_v) > t$ **then**
18: **return** r_v and time $t + \theta$
19: **else**
20: **return** r_v and time t
21: **end if**
22: **end if**
23: **end if**
24: **if** *FlagForFailedCR* is 1 **then**
25: Drop service
26: **else**
27: **return** loc_u and time t
28: **end if**



(a) (b)

Figure 4.4: (a) Warning Zone wz_v for Hospital (b) Cloaked region r_v for Hospital

An example is given in Figure 4.4. Figure 4.4(a) shows that a user's current location loc_u falls into a warning zone wz_v for hospital. According to the algorithm, it creates a cloaked region r_v for wz_v (Figure 4.4(b)). Since the user's actual location is in the cloaked region r_v, algorithm publishes r_v instead of a user's actual location.

4.4 Performance Analysis

As previously mentioned, we only generate a cloaked region for a warning zone if it contains a user's actual location at the requested time. Hence number of sharing a user's actual location is increased which ensures minimal degradation in quality of a user's shared data. However number of created cloaked region per request is also minimized which leads less online processing time than previous solutions. In our approach we never publish a user's actual location nearby her sensitive places before considering her all privacy requirements. In that way, service drop (if needed) can not cause any privacy breach which is discussed in Figure 1.3 from Section 1. On the other hand, we reuse pre calculated warning zones to create cloaked regions since both processes use same BFS algorithm to expand with same starting node. Hence computation time to create each cloaking region is also improved.

Chapter 5

Security Analysis

In this chapter, we show that our approach ensures the required privacy level of a user, i.e., an adversary can not refine a user's sensitive locations with a probability higher than the specified disclosure threshold. Furthermore, we show that in our approach, an adversary cannot apply the velocity based linkage attack, the temporal linkage attack and the upcoming linkage attack refine a user's sensitive locations with a probability higher than the disclosure threshold.

Our approach computes a cloaked region for every sensitive location of a user by considering the real time popularity of places and shares the cloaked region instead of the user's actual location, if the user is inside the cloaked region. The cloaked region generation algorithm expands the cloaked region by gradually adding non sensitive places until the probability to identify the user's sensitive place becomes lower or equal to the disclosure threshold.

However, an adversary may try to increase the probability to identify a user's sensitive location greater than disclosure threshold by reverse engineering the cloaked region. To avoid this, during warning zone generation we include warning zones of non-sensitive locations as a part of warning zone of a sensitive location. This ensures that whenever a location is cloaked, an adversary is not able to do any reverse engineering to exclude any non-sensitive place from the revealed cloaked region to increase the probability for identifying a sensitive location. For example, we assume that our algorithm does not include the warning zones of non-sensitive locations and whenever a user belongs to a warning zone of a sensitive location, a cloaked region is created to publish. Since an adversary has the knowledge about a user's sensitive place types and from the privacy algorithm, she knows that warning zones of sensitive locations does not include any non-sensitive places, she can predict

that the user actually belongs to a sensitive location. The reason behind that is if the user belongs to a non-sensitive location, she publishes her actual location because it is outside of the warning zone. Hence we construct warning zones for non-sensitive locations also such that an adversary can not identify whether a user belongs to a warning zone for a sensitive location or for non-sensitive locations.

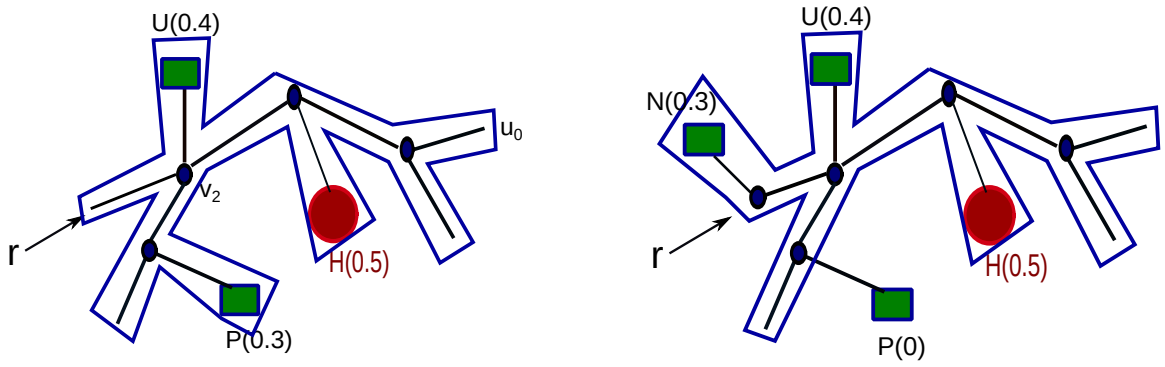
Upcoming linkage attack: The upcoming linkage attack occurs for not considering the movement towards a sensitive location in advance. To protect against upcoming linkage attack we introduce the concept of warning zone. Warning zone refines the area where the disclosure of a user's actual location may reveal a movement towards a sensitive place. In our approach, a user can reveal her actual location in two scenarios:

- A user's location is outside of warning zones.
- A user's location is inside of a warning zone but outside of a cloaked region.

However, by the technique we construct a warning zone, it is not possible that a location outside of warning zones is directly connected to any sensitive locations. In addition, a cloaked region for a sensitive location is generated by reusing the primary warning zone of that sensitive location. Thereby, it is also not possible for an adversary to link a user's actual location with a sensitive location directly if a user lies outside of a cloaked region. Hence, in both cases of sharing a user's actual location, our approach overcomes the upcoming linkage attack.

Temporal linkage attack: An adversary may increase the probability to associate a user with her sensitive location inside the cloaked region if cloaked regions are pre-computed and the popularity of places changes at the time of sharing cloaked regions. Let the cloaked region r shown in Figure 5.1(a) is pre-computed. The popularity of the park(P) becomes 0 at night and the probability of associating Alice with the hospital(H) becomes 0.56 which is greater than the specified disclosure threshold (0.5) of Alice. Thus, if Alice shares the cloaked region at night, the cloaked region fails to meet the privacy requirement of a user due to ignoring time constraints.

In our approach, Algorithm 3 computes cloaked regions at the time of sharing location data and thus considers the current popularity of places. At night, instead of considering the park for the cloaked region r, the algorithm considers another non-sensitive place, i.e., the night-club(N) with the popularity 0.3 to compute the cloaked region (Figure 5.1(b)). Hence our approach omits the possibility of applying the temporal linkage attack using correlation between popularity of a places



(a) (b)

Figure 5.1: (a) Cloaked region r for Hospital precomputed (b) Cloaked region for Hospital at night

and the time constraint.

Velocity based linkage attack: Ghinita et al.[16] shows that an adversary can use the maximum allowed velocity in a road network to refine a user's location inside a cloaked region and the attack is known as the velocity based linkage attack. To protect a user from the velocity based linkage attack, our cloaked region computation technique considers only those road junctions or non-sensitive places that are reachable from previously shared locations by the user at the time of sharing current cloaked region. If the algorithm fails to create the required cloaked region due to lack of enough non-sensitive places reachable from previously shared location then the algorithm drops the service. Hence our approach protects a user from the velocity based linkage attack while sharing location data continuously.

Chapter 6

Experiment

In this chapter, we present the experiments to evaluate the performance of our proposed approach using real datasets. Though our solution provides better privacy in terms of addressing linkage attacks, we compare our solution with the most recent online cloaking technique [23] in road-networks in terms of data utility and processing time while varying different parameters of a user's privacy profile. Since we run our experiments in different experimental environment, we also implemented the existing online algorithms with their provided pseudocode for comparison purpose. We measure the data utility in terms of the service drop and the frequency of the actual location revealed. The service drop represents the number of times a user stops sharing locations (in both precise or imprecise format). The data utility is the maximum, if a user always shares her actual locations and the data utility is 0 if the user stops sharing her locations. On the other hand, the data utility decreases if a user reveals cloked locations instead of her actual locations. Thus, the data utility depends on the precision of the shared location data, and the data utility increases with the decrease of the service drops and the increase of the number of the actual location revealed per trajectory. We run experiments for 1000 sample trajectories and determine the average processing time, service drops, and the frequency of actual location revealed.

The remainder of this chapter is organized as follows. In Section 6.1, we describe our experimental setup for evaluation and in Sections 6.2-6.4, we compare our solution with the most recent online cloaking technique in road-networks [23] in terms of data utility and efficiency while varying disclosure threshold, a user's sensitive place types, l diversity and number of sensitive buildings respectively. Finally, Section 6.5 validates our implementation by analyzing the output pattern of the existing

online technique for different privacy parameters of a user's privacy profile.

6.1 Experimental Setup

We chose a selected area of California covering approximately $32,320\ km^2$ from OpenStreetMap[1] and processed the raw datasets to prepare the city network according to our definition. The raw data consists of more than $15,000$ nodes (road junctions and places/stay points), $16,894$ edges (road networks) and different place types like hospital, shopping mall, and school. Among these nodes, there are $7,990$ places (sensitive or non-sensitive). For simplicity, we use the following fixed popularity for places of different types: hospital(0.3), school(0.4), religious place(0.09), entertainment(0.15), social/working place(0.06) and others (0.01). However, our approach can work for variable popularity of places, i.e., the popularity of a place can change with time and more than one places of the same type can have different popularity. Weight of edges represent the travel time required to traverse the edges and values of weights are collected from OpenStreetMap. Since we have to check the distance a user can cover during every request, we pre-compute all shortest paths between every pair of nodes in the graph using Dijkstra's algorithm. All algorithms are implemented in Java and experiments are executed on an Intel Core i5 3.20GHz, windows 7 machine equipped with 4GB main memory. After processing raw datasets, to simulate the movement of users, we randomly generate 1000 trajectories in the considered city network, where we assume that users periodically share their locations and the maximum number of the shared location per trajectory is 100. It is also estimated that users take around 7 hours on average to travel through each of these trajectories.

Table 6.1 summarizes parameter settings for our experiments.

[1] https://www.openstreetmap.org/

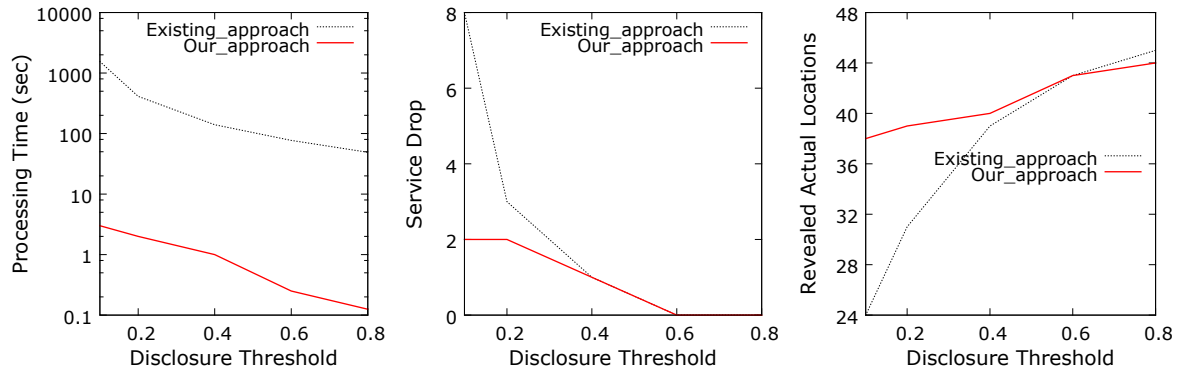
Table 6.1: Experimental Setup

Parameter	Range	Default
Disclosure threshold	0.8, 0.6, 0.4, 0.2, 0.1	0.1
Sensitive place type	Hospital, club, religious place	Hospital
No. of sensitive buildings	50 to 400	-
l diversity	2 to 8	4

6.2 Effect of Disclosure Threshold and Sensitive Place Types

Figure 6.1(a) presents the comparison in terms of efficiency i.e. the required processing time while varying disclosure threshold. Privacy profile contains only one sensitive place type (Hospital) and the required l diversity is 4. Dotted black line indicates existing online cloaking approach [23] and red solid line means our approach. For better visibility we set logarithmic scale on the $y-$axis when the output is in terms of processing time. According to the definition of disclosure threshold, we need to add more non-sensitive locations to decrease the probability to identify a user's sensitive location inside a cloaked region if the disclosure threshold is lower. Hence the processing time to create a cloaked region is increased with the decreased value of the disclosure threshold (Figure 6.1(a)). In addition, from Figure 6.1(a) we can see that our solution works faster than the previous solution. This is the benefit of warning zone. We generate cloaked region only for the warning zone which contains a user's actual location. The total number of created cloaked regions is thus reduced, which effects the processing time. We also reuse a generated warning zone to create a cloaked region. Hence the computation time to create each cloaked region is reduced.

Figures 6.1(b)-(c) show the utility comparison between both solutions. We can see that rate of service drop in previous method [23] is higher than ours while increased values of privacy, i.e., disclosure threshold (Figure 6.1(b)). The number of service drops in the figure is the average count per trajectory. Cloaked region generation precesses (our approach and existing online solution) need



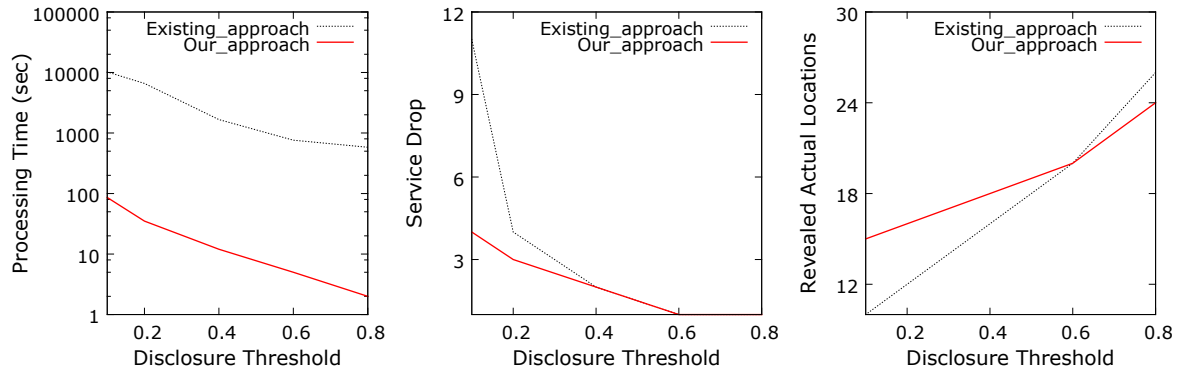
(a) (b) (c)

Figure 6.1: Comparison in term of processing time and utility while varying disclosure threshold

less non-sensitive locations to create a cloaked region for a sensitive location if disclosure threshold is higher and a user can share her actual locations all the time if her disclosure threshold becomes 1. Hence the probability of the service drop due to the failure to generate cloaked regions for not satisfying privacy requirements decreases while increasing disclosure threshold (Figure 6.1(b)) and it comes down to 0 when disclosure threshold is 1. However existing online solution creates cloaked regions in every request for all the sensitive locations which is reachable from previously shared location within the request time difference. When they fail to create, they drop the service. On the other hand, we create a cloaked region only if the warning zone contains a user's actual location. Since the number of required cloaked regions generated per request is less, the rate of service drop due to the failure to compute cloaked regions is also less in our solution.

Our solution also provides better result than previous online method (Figure 6.1(c)), if we compare the average number of revealed actual locations while varying disclosure threshold. With a lower disclosure threshold, the size of each cloaked region is increased due to adding more non-sensitive locations inside the cloaked region. Thereby, a user can share less actual locations per trajectory, which degrades data utility. Figure 6.1(c) shows that for both processes (our approach and existing online solution), the average number of actual revealed locations per trajectory decreases while decreasing the disclosure threshold. Since we only need to create a cloaked region when a user lies into a warning zone, the total number of required cloaked regions is thus reduced and we can publish a user's actual locations more without having any privacy issue that in turn increases the shared data

quality. On the other hand, both existing and our approaches use the **BFS** algorithm for computing cloaked regions. Hence the size of the computed cloaked region for same sensitive place is almost same in both approaches. Since our approach publishes higher number of actual locations than the existing online method, considering the actual location as a cloaked region containing a single location, the average cloaked region size per trajectory thus becomes less in our approach compared to the existing online method.



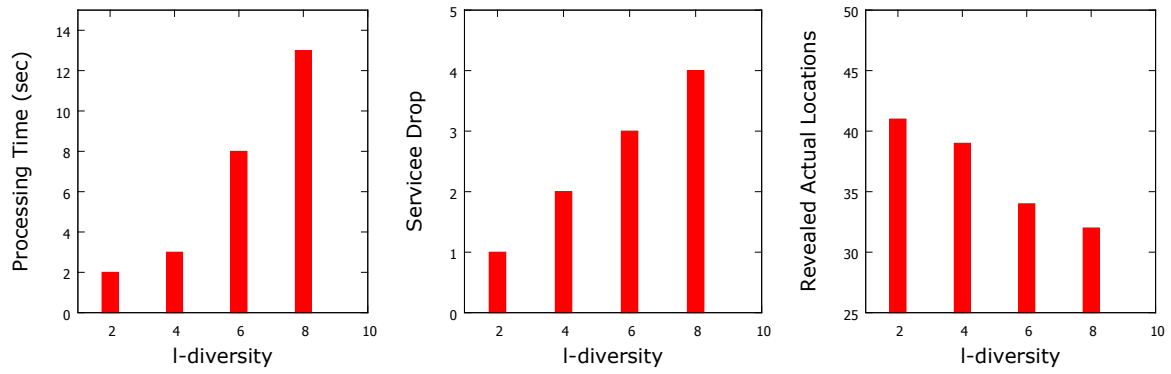
(a) (b) (c)

Figure 6.2: Comparison in term of efficiency and utility while containing more sensitive place types in one privacy profile

In reality, it is possible that a user has multiple sensitive place types (Figure 6.2). Figure 6.2(a) presents the comparison between the processing time and the disclosure threshold while having multiple sensitive place types in the privacy profile. It clearly shows that our solution is significantly better in terms of the processing time. Figures 6.2(b)-(c) show the comparison for utility, where we can see that the shared data quality of existing approach in terms of the average number of revealed actual locations and average service drop per trajectory, degrade with higher rate than ours while decreasing the disclosure threshold.

6.3 Effect of l Diversity

Figure 6.3 shows the effect of l diversity on the processing time and utility for our approach. We do not consider the existing online method for this set of experiments as the online cloaking method does not use the concept of l-diversity. In this experiment, the privacy profiles of users contain only one sensitive place type (Hospital) and the required disclosure threshold is fixed at 0.1. If we increase the diversity l, the processing time increases as well (Figure 6.3(a)). On the other hand, the average service drop count increases (Figure 6.3(b)) and the number of revealed actual locations decreases (Figure 6.3(c)) with the higher values of l. The reason behind is that with more diversity a warning zone covers more area. Hence the probability of a user being in a warning zone is increased. We thus need to generate more cloaked regions per request. The quality of shared data is thus degraded while increasing l diversity (Figure 6.3(c)).

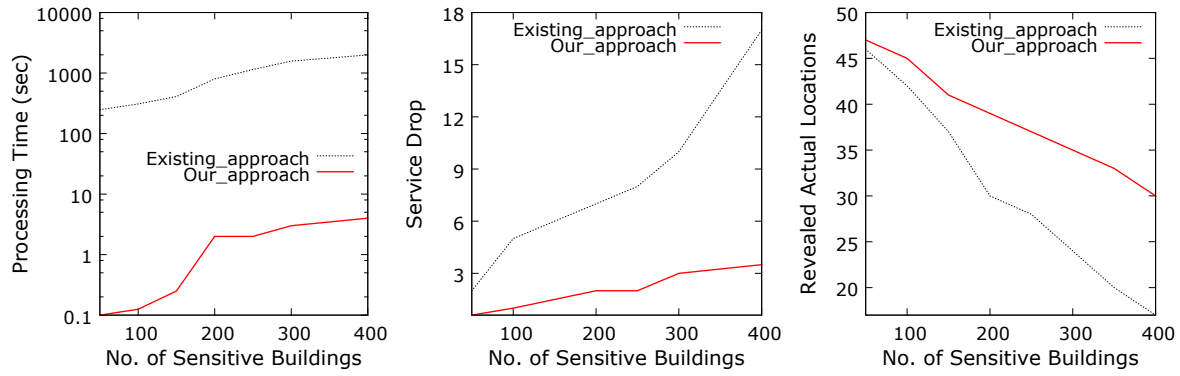


(a) (b) (c)

Figure 6.3: Effect of l diversity on processing time and utility

6.4 Effect of Number of Sensitive Buildings

Since we are working with an urban network, it is possible that the number of sensitive places is high. For scalability test, we randomly generate sensitive places inside the city network. By generating sensitive places we mean that we randomly change the type of existing places as sensitive place type and remaining places are of non-sensitive place type. In the first iteration of our experiment, we



(a) (b) (c)

Figure 6.4: Comparison in term of processing time and utility while varying number of sensitive buildings

consider 50 sensitive locations inside the city network and then for every next iteration, we add 50 more sensitive locations into it. This process continues until the number of sensitive locations reaches 400. Figure 6.4 presents the comparison on efficiency and utility with respect to the number of sensitive buildings. Other parameters are set to their default values. Figure 6.4(a) shows that solutions work faster while the number of sensitive buildings becomes lower inside the city network. The reason behind that is that we need to create less number of cloaked regions per request that in turn reduces the computational overhead. On the other hand, with less number of required cloaked regions, the service drop due to the failure to compute cloaked regions and the probability of a user being in a cloaked region are reduced. Hence we can publish more actual locations per trajectory and increase the data quality. Figures 6.4(b)-(c) show the impact in terms of service drop and revealed actual locations per trajectory. From the figures we also see that existing online solution shows poor performance compared our approach in terms of efficiency (Figure 6.4(a)) and utility (Figures 6.4(b)-(c)).

In existing online solution, the number of sensitive locations inside the subgraph (i.e., a graph containing all the vertices reachable from previously shared location within the time difference of two consecutive requests) per request is increased for increasing number of sensitive buildings overall in the city network. Hence they need to create more cloaked regions per request and the probability of the failure to compute the cloaked region while satisfying the privacy constraint is increased as

well. The number of sensitive buildings has also impact on our approach. With the increased number of sensitive buildings, the number of warning zones that contain the user's actual locations is also increased. However we only need to be concern about those warning zones, not all the sensitive buildings that are reachable from previously shared location. Hence the negative impact of the high number of sensitive buildings on the shared data quality in our approach less than that of the existing one.

6.5 Code Validation

In the existing online cloaking solution, we found that the processing time decreases with the increase of the disclosure threshold. In Figure 6.1(a), we also see that for both the online cloaking technique and our approach the experimental results show similar trends, i.e., the processing time decreases with the increase of the disclosure threshold.

If we consider the data utility in terms of the average service drop while varying disclosure threshold in existing online solution, we see that the number of times a user stops sharing her location per trajectory i.e., the average service drop increases with the decreased value of disclosure threshold. Figure 6.1(b) also shows the similar pattern for the average service drop per trajectory when the value of a user's disclosure threshold for her sensitive location is decreased.

From [23] and Figure 6.2(a)-(b), we also find that even if a user have multiple sensitive place types in her privacy profile, the impact of disclosure threshold on the processing time and data utility are same.

During scalability test, we found from existing online solution that the processing time increases with increase number of sensitive buildings. Figure 6.4(a) also shows that both our approach and existing online solution work slower while the number of sensitive buildings becomes higher inside the city network.

In addition, with the increased number of sensitive buildings, we need to compute more cloaked regions per request. Hence the service drop due to failure to compute cloaked regions increases with higher value of number of sensitive buildings. If we consider Figure 6.4(b), we can find the similar output pattern as analyzed.

Though experiments show that our approach is superior to the existing literature, the similar increasing/decreasing trends of the data utility and processing time as [23] for varying different privacy

parameters validate the correctness of the implementation of the existing and our approach.

Chapter 7

Conclusion

Protecting location privacy of users has become an important research topic in recent years. In this thesis, we have developed a novel approach to ensure a user's privacy while sharing location data without involving any third party. Researchers have developed solutions for privacy preserving sharing of location data. However, most of them focus on the Euclidean space and do not consider the the constraints of road networks. Since an adversary can infer a user's sensitive location using the maximum allowed velocity in road networks, solutions considering Euclidean space cannot overcome velocity based linkage attacks. Researchers have also proposed few techniques considering road network constraints for sharing location data continuously. However, none of them are able to ensure a user's privacy for not protecting temporal linkage attack and/or upcoming linkage attacks. Furthermore, these approaches frequently stop sharing location data for not considering upcoming sensitive locations in advance and thereby reduce the utility of location data. We develop techniques to protect a user's location privacy from all types of linkage attacks and increase data utility by reducing the frequency of not sharing locations of users.

Our solution is a combination of offline and online phases. In the offline phase, our approach pre-identifies *warning zones* for a user based on the user's privacy profile. Warning zones refine the areas where the disclosure of a user's actual location may enable adversaries to infer that the user has visited a sensitive place and thus, overcome the linkage attacks. At the time of sharing locations, in the online phase, our approach checks whether a user falls inside a warning zone. If not, our approach shares the user's actual location. Otherwise, i.e., if the user is inside the warning zone, our approach shares either cloaked or actual location depending on the the real time popularity of places. If the

disclosure of the actual location inside a warning zone can violate the user's privacy requirement, our approach computes cloaked regions. We have developed an efficient algorithm to compute a user's cloaked locations using the pre-computed warning zones.

We have performed experiments to evaluate the performance of our solution using real datasets in terms of privacy, utility and efficiency. Even though our solution provides better privacy by addressing all types of linkage attacks, we compare our algorithms with the most recent technique [23] in road-networks. Experimental results show that our approach incurs significantly less processing overhead than the existing approach because our approach considers the computation of cloaked locations only if a user is inside the precomputed warning zones. In addition, the reuse of computations performed in the warning zones for computing the cloaked locations further reduces the processing overhead. On the other hand, with our solution, a user is able to share her actual locations more without compromising her privacy than the existing approach, which increases the utility of shared data.

In the future, we aim to consider a user's habits, characteristics, background information while computing a cloaked location. For example, based on a user's financial condition an adversary can check whether the user can afford an expensive place which lies inside a cloaked location, based on the religious view an adversary can check whether the user can be in a religious place if it is located inside the cloaked location. An adversary can exploit these information and predict a user's location inside a cloaked location with higher probability. Our challenge will be to develop an efficient algorithm that can integrate a user's habits, characteristics, background information while computing a cloaked location to protect the user's privacy.

References

[1] Claudio Agostino Ardagna, Giovanni Livraga, and Pierangela Samarati. Protecting privacy of user information in continuous location-based services. In *CSE*, pages 162–169, 2012.

[2] J. Voelcker. Stalked by satellite - an alarming rise in gps-enabled harassment, July 2006.

[3] FoxNews. Man accused of stalking ex-girlfriend with gps, 2004. `http://www.foxnews.com/story/2004/09/04/man-accused-stalking-ex-girlfriend-with-gps.html`.

[4] Dateline NBC. Tracing a stalker, 2007. `http://www.msnbc.msn.com/id/19253352`.

[5] USAToday. Authorities. Gps system used to stalk woman, 2002. `http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm`.

[6] Microsoft Corporation. Microsoft survey finds that 52% of user of location based service express strong concern with sharing their location with other people or organizations, January 2011. `http://www.winrumors.com/microsoft-warns-of-privacy-concerns-with-location-based-services/`.

[7] Luca Calderoni, Dario Maio, and Paolo Palmieri. Location-aware mobile services for a smart city: Design, implementation and deployment. *JTAER*, 7(3), 2012.

[8] Antoni Martínez-Ballesté, Pablo A. Pérez-Martínez, and Agusti Solanas. The pursuit of citizens' privacy: a privacy-aware smart city is possible. *IEEE Communications Magazine*, 51(6), 2013.

[9] Constantinos Patsakis, Paul Laird, Michael Clear, Mélanie Bouroche, and Agusti Solanas. Interoperable privacy-aware e-participation within smart cities. *IEEE Computer Society*, 48(1):52–58, 2015.

[10] Xinxin Liu, Han Zhao, Miao Pan, Hao Yue, Xiaolin Li, and Yuguang Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *INFOCOM*, pages 972–980, 2012.

[11] Balaji Palanisamy and Ling Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *ICDE*, pages 494–505, 2011.

[12] Mehmet Ercan Nergiz, Maurizio Atzori, and Yücel Saygin. Towards trajectory anonymization: a generalization-based approach. In *GIS*, pages 52–61, 2008.

[13] Dan Lin, Sashi Gurung, Wei Jiang, and Ali R. Hurson. Privacy-preserving location publishing under road-network constraints. In *DASFAA*, pages 17–31, 2010.

[14] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.

[15] Kyriakos Mouratidis and Man Lung Yiu. Anonymous query processing in road networks. *IEEE*, 22(1):2–15, 2010.

[16] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. Preventing velocity-based linkage attacks in location-aware applications. In *ACM-GIS*, pages 246–255, 2009.

[17] Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. Fine-grained cloaking of sensitive positions in location-sharing applications. *IEEE Pervasive Computing*, 10(4):64–72, 2011.

[18] Maria Luisa Damiani, Elisa Bertino, and Claudio Silvestri. The PROBE framework for the personalized cloaking of private locations. *Transactions on Data Privacy*, 3(2):123–148, 2010.

[19] Marius Wernke, Frank Dürr, and Kurt Rothermel. Efficient position sharing for location privacy using binary space partitioning. In *MobiQuitous*, pages 263–275, 2012.

[20] Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. Map-aware position sharing for location privacy in non-trusted systems. In *Pervasive*, pages 388–405, 2012.

[21] Marius Wernke, Frank Dürr, and Kurt Rothermel. Pshare: Position sharing for location privacy based on multi-secret sharing. In *PerCom*, pages 153–161, 2012.

[22] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *ICPS '05*, pages 88–97, 2005.

[23] Emre Yigitoglu, Maria Luisa Damiani, Osman Abul, and Claudio Silvestri. Privacy-preserving sharing of sensitive semantic locations under road-network constraints. In *MDM*, pages 186–195, 2012.

[24] Bhuvan Bamba, Ling Liu, Péter Pesti, and Ting Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *WWW*, pages 237–246, 2008.

[25] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In *MDM*, pages 278–282, 2007.

[26] Wen-Chih Peng, Yu-Zen Ko, and Wang-Chien Lee. On mining moving patterns for object tracking sensor networks. In *MDM*, pages 41–44, 2006.

[27] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[28] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Trans. Knowl. Data Eng.*, 19(12): 1719–1733, 2007.

[29] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 763–774, 2006.

[30] Bharath Krishnamachari, Gabriel Ghinita, and Panos Kalnis. Privacy-preserving publication of user locations in the proximity of sensitive sites. In *SSDBM*, pages 95–113, 2008.

[31] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *ACM SIGMOD*, pages 229–240, 2006.

[32] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mob. Comput.*, 7(1):1–18, 2008.

[33] Ting Wang and Ling Liu. Privacy-aware mobile services over road networks. *PVLDB*, 2(1): 1042–1053, 2009.

[34] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy*, 2(2):28–34, 2004.

[35] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. Location diversity: Enhanced privacy protection in location based services. In *LoCA*, pages 70–87, 2009.

[36] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. PRIVE: anonymous location-based queries in distributed mobile systems. In *WWW*, pages 371–380, 2007.

[37] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[38] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.

[39] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, pages 106–115, 2007.

[40] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *PET*, pages 393–412, 2006.

[41] Chi yin Chow. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In *ACM GIS*, pages 171–178, 2006.

[42] Marco Gruteser and Baik Hoh. On the anonymity of periodic location samples. In *SPC*, pages 179–192, 2005.

[43] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm 2005*, pages 194–205, 2005.

[44] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *ACM*, pages 161–171, 2007.

[45] Chi-Yin Chow, Mohamed F. Mokbel, Jie Bao, and Xuan Liu. Query-aware location anonymization for road networks. *GeoInformatica*, 15(3):571–607, 2011.

[46] Wei-Shinn Ku, Roger Zimmermann, Wen-Chih Peng, and Sushama Shroff. Privacy protected query processing on spatial networks. In *ICDE*, 2007.

[47] Po-Yi Li, Wen-Chih Peng, Tsung-Wei Wang, Wei-Shinn Ku, Jianliang Xu, and John A. Hamilton Jr. A cloaking algorithm based on spatial networks for location privacy. In *SUTC*, pages 90–97, 2008.

[48] Zheng Huo, Xiaofeng Meng, Haibo Hu, and Yi Huang. You can walk alone: Trajectory privacy-preserving through significant stays protection. In *Database Systems for Advanced Applications - 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part I*, pages 351–366, 2012.

[49] Joseph T. Meyerowitz and Romit Roy Choudhury. Hiding stars with fireworks: location privacy through camouflage. In *MOBICOM*, pages 345–356, 2009.

[50] Toby Xu and Ying Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM*, pages 547–555, 2008.

[51] Bugra Gedik and Ling Liu. Location privacy in mobile systems: A personalized anonymization model. In *ICDCS*, pages 620–629, 2005.

[52] Daniel Ashbrook and Thad Starner. Learning significant locations and predicting user movement with GPS. In *ISWC*, pages 101–108, 2002.

[53] Akinori Asahara, Kishiko Maruyama, Akiko Sato, and Kouichi Seto. Pedestrian-movement prediction based on mixed markov-chain model. In *ACM-GIS*, pages 25–33, 2011.

[54] Pratap S. Prasad and Prathima Agrawal. Movement prediction in wireless networks using mobility traces. In *Consumer Communications and Networking Conference (CCNC)*, pages 1–5, 2010.

[55] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Nú nez del Prado Cortez. Next place prediction using mobility markov chains. In *MPM*, 2012.

[56] Chi-Yin Chow and Mohamed F. Mokbel. Enabling private continuous queries for revealed user locations. In *SSTD*, pages 258–275, 2007.

[57] Xiao Pan, Xiaofeng Meng, and Jianliang Xu. Distortion-based anonymity for continuous queries in location-based mobile services. In *ACM-GIS*, pages 256–265, 2009.

[58] Toby Xu and Ying Cai. Location anonymity in continuous location-based services. In *ACM-GIS*, page 39, 2007.