Improved privacy preservation approaches in mobile networks for dummy generation and spatial range queries

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Shadaab Siddiqie
201502030
mashadaab.siddiqie@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Improved privacy preservation approaches in mobile networks for dummy generation and spatial range queries" by Shadaab Siddiqie, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. P. Krishna Reddy,
Data Sciences and Analytics Center,
Kohli Center on Intelligent Systems,
IIIT Hyderabad.

To my mamma

Acknowledgments

I express my deepest gratitude to Prof. P. Krishna Reddy for introducing me to research. His kind and humble nature have taught me how to interact with my team when in a position of authority. His constant guidance has contributed significantly to my personal and professional growth. I would also like to extend my heartfelt thanks to the late Dr. Anirban Mondal, whose invaluable support and rigorous review process of research papers before submission to conferences greatly enhanced my understanding of how to articulate ideas effectively. Dr. Mondal's dedication and passion for research will forever remain an inspiration. Additionally, I would like to acknowledge the tremendous assistance of Dr. Srinivas Annappalli, who stepped in to help me with my work at a crucial time.

I would also like to thank my colleagues Akhil Ralla, Saideep Chennupati, Mamatha Alugubelly, Mittapally Kumara Swamy, Srinivas Annappalli, Narendra Babu Unnam and Revanth Parvathaneni for their constant support and for providing a positive learning environment. I especially want to thank Akhil Ralla for the long brainstorming discussions during my research. I would also like to thank my friends Bhanu Teja Pachipulusu, Ralla Akhil, Sravan Mylavarapu, Sudheer Achary, Vaibhav Gupta, Kartikey Pant, Pandramish Vinay and Rama Rohit Reddy Gangula for making my college life more cherishable.

Last but not least, I express my deepest gratitude toward my mamma, Dr Rahamath Begum and my brother, Shoaib Siddiquie, for their unconditional love and support throughout my journey.

Abstract

Location-Based Services (LBSs) have become increasingly prevalent in today's mobile technology sector, delivering tailored information relevant to the users' precise locations. These services grant users access to location-centric information like the proximity of hospitals, restaurants, or other points of interest, thereby facilitating routine tasks. However, such LBSs can pose significant concerns about user privacy. Consider a user querying, "What are the directions to the best cancer hospital from the current location?". Such queries expose the user's current location information to the LBS provider and other intermediate nodes (intruders) in the mobile network. Query location information can reveal sensitive information about the user, such as relationships, health, religion, and nightlife habits. In this thesis, we propose two improved approaches to preserve the privacy of users' query location in the mobile environment.

As the first approach, we propose an improved dummy generation approach for better privacy. In a dummy generation approach, the user sends additional dummy locations along with the user's actual location in its query, thereby confusing the LBS provider and the other nodes. The existing approaches have the issue of generating dummies in regions with more infeasible regions (inaccessible regions). Moreover, the existing approaches do not consider the presence of *time-dependent infeasible regions*. For example, consider a supermarket with opening and closing times as 9am and 9pm, respectively. From 9am to 9pm, this supermarket can be considered a feasible region; otherwise, this area can be regarded as an infeasible region. Furthermore, if the intruder estimated the centre of cloaking region (CR) using the dummy locations, it would become more accessible for the intruder to know a given user's actual location. To improve the performance, we propose an **Annulus-based Gaussian Dummy Generation** (AGDG) approach. AGDG introduces the concept of a virtual cloaking region to generate cloaking regions. In AGDG, unlike traditional methods, the user's location is not fixed at a fixed distance from the centre of the cloaking region. Additionally, AGDG considers the infeasible regions and query probability in the surrounding environment when generating dummy locations. The approach also incorporates the concept of time-dependent infeasible regions and ensures that the generated dummy locations abide by these time-dependent constraints.

As the second approach, we propose a cloaking-based approach to improve the privacy of spatial range queries. In distributed spatial cloaking-based approaches, the user's query location information is cloaked using the distributed mobile network around the user (e.g., the p2p network). Existing approaches do not preserve the user's intent privacy. For example, suppose a user queries all the cancer

hospitals near her. In that case, her location and health information (searching for intent, which is about cancer hospitals) must be preserved from both LBS providers and peers in the surrounding. Moreover, the existing approaches require a large number of peers to be employed to cloak the user query location. Maintaining such structures in a highly dynamic mobile network is challenging. We propose the notion of ***ijk*Cloak** framework to improve existing distributed spatial cloaking-based approaches. The ijkCloak framework introduces the notion of ijk-anonymity to protect both the user's query location and intent information. This method divides the user's query location information into multiple fragmented locations. This process helps keep the user's query location private from their peers and the LBS provider. Additionally, dummy intents are sent to the LBS provider along with the user's actual query to protect the user's query intent. The proposed approach ijkCloak, adopts ijk-anonymity in a mobile network environment. Because of the efficiency of ijk-anonymity, this proposed method requires fewer peers to maintain user privacy, making it more practical in a highly dynamic mobile network environment.

For each approach, the theoretical analyses and comprehensive experimental study exhibits its potential to preserve location privacy in different scenarios. We hope this research encourages further research and leads to the development of improved privacy preserving approaches in mobile networks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the advancement of mobile communications technology and the widespread use of GPS devices, location-based services (LBSs) have gained significant popularity. These services enable mobile users to obtain location-specific information by requesting LBS providers to retrieve the desired data. However, one major issue with this type of service is the possibility of malicious entities analyzing a user's requested location (query location) to track user movements illegally or leak their data [33, 17]. Consider a user querying, "What are the directions to the cancer hospital from the current location?" Such queries expose the user's current location. Moreover, malicious entities may infer the location of any given user by analyzing their query location data [19]. In addition, query location information can reveal sensitive information about the user, such as relationships, health, religion, and nightlife habits, further compromising the user's privacy [23]. Therefore, developing an approach to provide location-based services to mobile users is crucial while reducing the risk of violating user location privacy.

In this thesis, we have focused on improving the performance of existing dummy generation and distributed spatial cloaking-based privacy preservation schemes. In the following sections, we first provide background information on these approaches and the research gaps. Next, we summarize the proposed approaches. Subsequently, we list the contributions and present the thesis's organization.

1.1 Background on dummy generation approaches

Consider a real-life scenario where a user wants to request a location-based query to a given Location-Based Service (LBS) provider. In such scenarios, sending a request with the user's location to the LBS provider would expose the user's location to adversaries (including the LBS provider). In such cases, the user can choose an application that uses a dummy generation approach, which generates fake locations (dummies). The application will take the user's location and the number of dummy locations required as input. After completing on-device computations, the application will output dummy locations. The user can now use these dummy locations to send multiple requests to the LBS provider. Once the user receives the result of all the location-based queries, she can prune the results with her location as the

final result. This way, it is possible to preserve the user's location privacy using dummy generation approaches.

Overview of existing approaches and issues: In the literature, multiple dummy generation approaches [9, 53, 44, 18] have been proposed. The challenge lies in generating realistic dummies while considering environmental factors like hard-to-reach areas or population density. Incidentally, existing dummy generation approaches are inadequate for regions with more **infeasible regions** [9, 53]. A geographical region is defined as an *infeasible region* for an entity if an entity cannot possibly be physically present at that location. Examples include locations without road or transportation infrastructure, restricted government facilities or military zones and forest areas to preserve endangered species. Understandably, a higher number of infeasible regions decreases the area of CR, thereby increasing the probability of violation of user location privacy. The presence of **time-dependent infeasible regions** is not considered in past works like [18]. For example, consider a supermarket with opening and closing times as 9am and 9pm, respectively. From 9am to 9pm, this supermarket can be considered a feasible region; otherwise, this area can be regarded as an infeasible region. Additionally, if an intruder successfully determines the centre of the CR based on dummy locations, it can potentially expose the user's real location. This is because in the case of existing approaches such as [44], the user's location is always at a fixed distance from the centre of CR.

1.2 Spatial range queries and privacu issues

Generating accurate and realistic dummy data to deceive adversaries may not always be feasible. As a result, spatial cloaking-based distributed approaches utilize nearby peers to conceal the user's location information [11]. Here, users are self-organized into a distributed network as peers and collaborate to generate cloaked regions. These approaches preserve the location privacy of mobile users querying for different queries. Among other queries, mobile users issue spatial range queries (SRQs) [30], which are location-dependent queries to determine the user's desired objects within a (spatial) range restriction. For example, consider Figure 1.1. Here, u_c is a mobile user and u_c's spatial region contains 13 cancer hospitals: $\{d_1, d_2, \ldots, d_{13}\}$. A sample SRQ Q_1 of u_c is as follows: *"What are the cancer hospitals within 5km of my current location?"*.

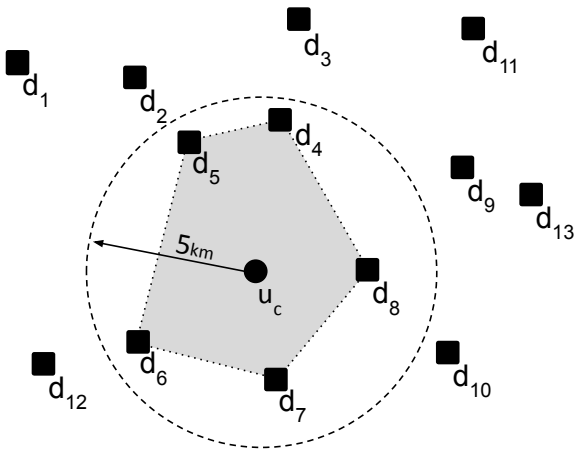

Figure 1.1: Spatial Range Query

Challenges in SRQs: For SRQs, in addition to the user's query location privacy, protecting the privacy of the user's query intent is also a major concern [54]. Understandably, the user would prefer

to maintain the confidentiality of her intent information and not disclose the intent to any unauthorized parties. In Figure 1.1, we have to preserve u_c's location, which is "my current location" and the user's intent, which is "cancer hospitals". The result for the spatial range query Q_1, requested by a user u_c, would be $\{d_4, d_5, d_6, d_7, d_8\}$, i.e., the cancer hospitals within 5km of u_c's location. Note that the location information of SRQ may not be the user's current location. For example, suppose u_c possesses the following query: *"Where is the best mental institution within 20 km of the gala area?"*. In such a case, the issue is to preserve the query location, which is "the gala area" and the intent, which is "best mental institution".

Overview of existing approaches and issues: In addition to intent and location privacy, privacy-preserving approaches for SRQs should also consider the presence of infeasible regions in the user's surroundings. Such infeasible regions can decrease the Cloaking Region (CR) area, thereby increasing the probability of violating user location privacy. The user's information should be protected from both the LBS provider and peers. Moreover, existing spatial cloaking-based approaches [15, 14, 31] use a sizeable distributed network around them to preserve users' location information. However, consistently maintaining such large clusters of peers in a dynamic mobile network is challenging.

1.3 Research gaps

Following are the research gaps in existing dummy generation and spatial cloaking-based distributed approaches for preserving privacy:

- **Dummy generation approaches:** Previous dummy generation approaches are vulnerable to an adversary who can determine the centre of the cloaking region. Additionally, these approaches need to consider time-dependent infeasible regions.

- **Distributed spatial cloaking approaches:** Existing distributed spatial cloaking approaches for preserving location privacy rely on a high number of peers to cloak the user's location information. Moreover, these approaches do not preserve the user's intent information.

Addressing these research gaps is essential because of the growing use of Location-Based Services (LBSs). This need is what motivated our research. We aim to improve the current methods, ensuring people can utilize LBSs while preserving their location information from an external entity.

1.4 Overview of the proposed approaches

In this section, we present the overview of the proposed approaches.

1.4.1 Improved dummy generation approach

The issue is to protect the user's location information even if an adversary somehow knows the centre of CR. For this, we propose the concept of **virtual cloaking region**. Our proposed approach, **Annulus-based Gaussian Dummy Generation (AGDG)**, randomizes the distance between the user and the centre of cloaking region (CR) by predetermining the user's placement using a virtual cloaking region. The virtual cloaking region is computed such that the dummies have a similar query probability to the actual users. Moreover, in the case of areas with a higher number of infeasible regions, we propose an **annulus-based cloaking region** with a Gaussian probability distribution for placing the candidates such that the distance between the candidates is increased.

Moreover, to preserve user location privacy at varying times of the day, we present the concept of **time-dependent infeasible regions**. To illustrate, consider a supermarket operating from 9am to 9pm. During its operational hours, this supermarket represents a feasible region; outside these hours, it is treated as an infeasible region. Our approach is well-equipped to consider these changes and generate appropriate dummy locations even in such dynamic scenarios.

1.4.2 Improved spatial cloaking-based approach for spatial range queries

The issue with the spatial cloaking-based distributed approaches is to preserve the user's query location and intent information from the LBS provider and the peers while using fewer peers. To address these issues for **SRQ**, we present the concept of ijk**-anonymity**. In ijk-anonymity, the user's SRQ is fragmented into j **fragmented Spatial Range Queries (fSRQs)** by handling the case of infeasible regions. Each fSRQ is sent to a peer, which further forwards it to the LBS provider through peers. Typically, an adversary (an LBS provider, peers or someone accessing LBS server data) could receive k ($k \geq j$) SRQs generated by multiple users. As a result, the j number of fSRQs obfuscates the actual location information of the user among the k SRQs received by the adversary. As a result of fragmentation, location information is also preserved from the peers. To preserve the user's query intent, $i-1$ **dummy intents** are sent along with the user's actual query intent to the LBS provider. The proposed approach, which we term as ***ijk*Cloak**, adopts ijk-anonymity in a mobile network environment. The proposed approach is more practical for a highly dynamic mobile network environment as it employs fewer peers to preserve the user location and intent privacy.

1.5 Contribution of the thesis

The major contributions of the thesis are as follows:

1. We proposed an improved dummy generation approach that considers the presence of *time-dependent infeasible regions* termed as AGDG.

2. We proposed an improved distributed cloaking-based approach for spatial range queries, termed *ijk*Cloak. This approach introduces the concept ijk-anonymity to achieve improved location and intent privacy.

3. We conduct theoretical analysis and experiments to demonstrate that our proposed approaches are more effective than existing approaches.

1.6 Organization of the thesis

The rest of the thesis is organized as follows:

- In Chapter 2, we discuss the related work.
- In Chapter 3, we present Enhanced Dummy Generation Approach.
- In Chapter 4, we present Improved Location and Intent Privacy for Spatial Range Queries.
- In Chapter 5, we conclude the thesis with a summary and discuss future research directions.

Chapter 2

Related work

This chapter discusses the literature on location privacy-preserving techniques. The work in [26] underlines scenarios where unauthorized location data disclosure can lead to privacy invasion and misuse of sensitive information. Studies have also examined the challenges LBS providers face in balancing utility and privacy, emphasizing the need for privacy-preserving mechanisms to maintain user trust and promote LBS adoption [13, 42]. Recent works [48, 1] have delved into the legal and ethical aspects of location privacy, stressing the importance of privacy-by-design and a comprehensive regulatory framework. To preserve users' location privacy, various approaches have been proposed in the literature. These methods can be broadly categorized into three types: *obfuscation*, *dummy generation*, and *spatial cloaking*. Section 2.1 provides an overview of obfuscation-based approaches. Section 2.2 covers research on dummy generation-based approaches, while Section 2.3 discusses spatial cloaking-based approaches.

2.1 Obfuscation-based approaches

Obfuscation-based approaches, such as those proposed in [3, 2, 7, 12, 24, 47, 40, 50, 22], aim to protect the privacy of location-based service (LBS) users by substituting their real locations with nearby landmarks or intersections. However, this method may not be effective if the user is in an area with a limited number of appropriate landmarks or intersections, resulting in significant degradation of the quality of LBS. Additionally, as described in [12], spatial transformation methods may be used to distort actual user locations by adding random noise. However, as shown in [25], the amount of noise required to prevent tracking attacks can be quite large, potentially diminishing the usefulness of the LBS.

The work in [32] proposes a game-theoretic model for location obfuscation in LBS, aiming to balance the trade-off between privacy protection and the quality of LBS. The works of [4] introduce a model based on application zones, representing geographic areas with similar interests, and mix zones, areas where user tracking is inhibited. Here, identities within these mixed zones become blended and unidentifiable, disrupting the linkage between users entering and exiting.

2.2 Dummy-generation approaches

Dummy generation approaches [18, 53, 9, 44] involve the generation of dummy locations, which are mixed with the user's actual location and provided to the LBS provider as a list of indistinguishable locations in a query. The user can then filter out the dummy locations and select only the information relevant to their actual location.

The work in [18] proposes a method for generating dummies that behave like real humans to improve further the effectiveness of the dummy generation approach for mobile users. A technique, designated as Circle-divided Dummy Generation (CDG) [53], generates dummies by considering an angle. Moreover, the Obstacle-based Dummy Generation (ODG) approach, which considers the surrounding environment, was proposed in [9]. Furthermore, the Efficient Dummy Generation (EDG) approach [44] created dummies not on the circumference of the circle, but rather on a thick strip of a circle (forming an annulus) to reduce the probability of exposing the users' location in areas with a high density of infeasible regions. These approaches use k-anonymity to hide the user's actual location. The k-anonymity approach [46] guarantees that an individual's location data remains indistinguishable from the information of k - 1 other users via the process of generalization. Consequently, adversaries have a probability of 1/k in accurately identifying the user's actual location.

2.3 Spatial cloaking-based approaches

Further, the existing spatial cloaking approaches based on their architecture can be categorized into *centralized* and *distributed*.

2.3.1 Centralized cloaking approaches

In centralized architecture [16, 28, 29, 36], an anonymizer is used for mixing a given user's actual location with at least $k - 1$ other users. Consequently, the LBS provider cannot identify the actual user location with a more than $1/k$ probability. However, this could pose a scalability issue because it requires all mobile users to periodically report their locations to the anonymizer. Moreover, the centralized architecture assumes a trusted third-party server to mediate interactions between the users and the LBS server.

Centralized architecture-based approaches relying on certification authorities face a similar issue of deploying a safe and practical third-party server. In [38], particular peers run as serving nodes responsible for caching all data for the other peers. This might also act as a pseudo-anonymizer and can be attacked by an adversary. An approach proposed in [54] attempts to remove the linkage between users' identity and the issued query to prevent privacy breaches on the user's intent. However, in this case, such personalization is performed on the server side.

2.3.2 Distributed cloaking approaches

For the distributed architecture model, the work in [11] first proposes a mobile-P2P model, which we refer to as *CloakP2P*. Here, users are self-organized into a P2P network as peers and collaborate to generate cloaked regions. Mobile users can work together to blur their locations without using any fixed communication infrastructure or centralized/distributed servers. When a peer wants to get its cloaked region, it must find another $k-1$ nearest peer. Then it needs to exploit the minimum region covering these k peers as its cloaked region to achieve k-anonymity. However, this approach assumes that the peers are trusted entities. This assumption may not always hold good in practice because the peers can expose the shared information to an adversary.

The works in [15, 14, 31] adopt a distributed architecture on a P2P network. Within these schemes, Privé [15], and MobiHide [14] use fixed communication infrastructure to maintain location anonymization in a P2P network. Moreover, in these schemes, each user is assigned an index based on a Hilbert-space curve [35], and the peers are organized in a structured topology such as Chord [45]. However, these complex data structures [35] make it challenging for these models to be applied in highly dynamic mobile applications. Furthermore, these models assume that the communication cost between any two users in the network is the same; this may not necessarily hold good in practice. Additionally, these schemes assume that peers are trustworthy and share information like true locations. This assumption may also not necessarily be true for real-world scenarios.

To solve the trustworthiness of peers, the authors in work [20] distributively computes the cloaking region for peers without revealing the precise location information to other peers. However, the proximity information used in cloaking is measured by peers through the received signal strength or the time difference of arrival of beacon signals among the peers themselves. This information still poses a risk of exposing the user location data since the same method is also used for positioning technologies.

The authors in work [31] propose a distributed negotiation algorithm to address the issue of user privacy. This method helps users conduct negotiations among themselves to find their cloaked regions without exposing their precise locations to peers. In [31], decentralized P2P architecture Kademlia [34] is used to achieve location privacy. However, maintaining Kademlia in a highly dynamic mobile network is challenging. Moreover, the intent of the user's query is not protected from an adversary in this approach.

In the field of location-based queries, an extensive amount of research has been done, with a particular focus on four types of queries that are primarily used by mobile users: spatial range queries, nearest neighbour (NN) queries, K nearest neighbours (KNN) queries, and multidimensional range queries.

The paper cited as [30] provides an in-depth investigation into applying and optimizing spatial range queries. It presents a novel methodology and highlights the importance of density-based techniques. The study offers insights into improving the accuracy and efficiency of results compared to existing approaches.

The study in [52] delves into these types of queries, particularly focusing on the reduction of computational and communication cost in mobile computing environments. The paper proposes innovative methods to increase the efficiency of NN queries, making it a pivotal piece of research in the field.

The K-nearest neighbours (KNN) query is extensively researched in work cited as [49]. This paper proposes an efficient algorithm that uses the power of spatial networks to perform KNN queries. This study's results underscore spatial networks' potential to enhance the performance of KNN queries.

Lastly, the multidimensional range query is the focus of the research paper [41]. This paper proposes an innovative index structure to optimize range queries in a spatial database, particularly multidimensional ones. The novel indexing structure discussed in this research significantly reduces query latency and enhances performance, significantly contributing to this field.

Although NN and KNN queries offer viable solutions to identify the nearest objects to a specific location, they cannot replace spatial range queries, as each serves a unique purpose in different scenarios. Occasionally, a multidimensional range query can replace a spatial range query, especially when the desired results are within a rectangular region rather than a circular one. However, accuracy is a crucial factor. Thus, compromising on precision is not desirable.

Given this scenario, ensuring privacy in spatial range queries becomes a significant concern, warranting the need to explore and develop privacy-preserving methodologies. Hence, this thesis proposes a robust privacy-preserving approach to protect the intent and location information associated with spatial range queries.

In the landscape of dummy generation approaches, protection of a user's location privacy against an adversary aware of the center of the cloaking region is a challenging task. Our proposed approach, AGDG, aims to enhance privacy safeguards in this context,

Meanwhile, distributed cloaking-based methods often face a balance between using fewer peers and preserving the user's query intent. Our novel solution, the ijk-*anonymity* concept for spatial range queries, is designed to address these challenges, enhancing the security of both the user's query location and intent privacy.

2.4 Differences with the existing approaches

In the existing dummy generation approaches, protection of a user's location privacy against an adversary aware of the centre of the cloaking region is a challenging task. The proposed approach (AGDG) improves the privacy performance by proposing the notion of time-dependent infeasible regions.

Moreover, none of the existing distributed spatial cloaking-based approaches protects the user's query location and intent information while using fewer peers. Our novel concept ijk-*anonymity* addresses these challenges, enhancing the user's security. The proposed approach *ijk*Cloak adopts ijk-anonymity in a mobile network environment.

2.5 Summary

In this chapter, we have provided an overview of existing research on obfuscation-based, dummy generation, and spatial cloaking-based approaches for protecting the privacy of LBS users. In the next chapter, we present the improved dummy generation approach.

Chapter 3

Enhanced Dummy Generation Approach

In this chapter, we present our proposed approach for improving the efficiency of existing dummy generation approaches. In Section 3.1, we present the problem background. In Section 3.2, we outline the basic idea of the proposed algorithm. Section 3.3 provides a detailed explanation of the AGDG approach. In Section 3.4, we incorporate time-dependent infeasible regions into the AGDG approach. Experimental results demonstrating the superiority of AGDG compared to existing approaches are presented in Section 3.5. Finally, in Section 3.6 we conclude the chapter with a summary.

3.1 Problem Background

In this section, we introduce the concept of a dummy generation. We will then define essential terms, provide the problem statement, and discuss the issues encountered with previous methods concerning this problem statement.

Consider a mobile user, who desires to obtain services from a given LBS provider by providing their current location. Any dummy generation approach aims to obtain services from LBS providers without disclosing the users' real location by sending an additional k-1 dummy locations. Observe how the user's privacy is enhanced, albeit at the cost of increased communication costs. We shall explain the following terms relevant to our problem statement.

k-anonymity: A release of data is said to have the k-anonymity property if the information for each user contained in the release cannot be distinguished from at least k-1 (dummy or real) users whose information also appears in the release. Many location-based privacy preservation algorithms (including dummy generation approaches) use the notion of k-anonymity.

Cloaking Region (CR): The region in which LBS providers cannot identify the exact location of a given user is designated as the cloaking region [53] of that user. Users can choose the area of CR based on the desired level of location privacy. Intuitively, relatively small cloaking regions make it easier for adversaries to cause user location privacy violations. However, large cloaking regions will increase the query cost. Notably, privacy preservation schemes should set the maximum and minimum bounds for CR based on this trade-off.

Infeasible Region (IR): A geographical region is defined as an infeasible region for a user if the user cannot be physically present [9] at that location, i.e., the location is essentially inaccessible to users. Examples include locations without road or transportation infrastructure, restricted government or military zones, and forest areas to preserve endangered species.

Time-dependent infeasible region: A region can be considered as an infeasible region for a while and a feasible region for the rest of the time. For example, a school's premises can be considered an infeasible region when the school is closed (i.e., during holidays and nighttime when the school is closed). We call such regions time-dependent infeasible regions. Note that this time dependency need not be consistent for different regions at different times. For example, the closing time for a school might be different from the closing time of a movie theatre. Moreover, the closing times for different schools might also be different.

Query probability: Query probability of a location is the probability that a user from that location has issued a query to the LBS provider in the past [37]. It is based on querying history. Each possible location's query probability is typically used towards constructing the entropy-based privacy metric.

Problem statement: Consider a user, who wants to hide his/her location by sending k-1 extra dummy locations $\langle l_1, l_2, \cdots, l_k - 1\rangle$ to the LBS provider along with his/her real location l_k with the purpose of achieving k-anonymity. Consider that we are given geographical information about the surrounding environment, like time-dependent and non-dependent infeasible regions and query probability at each point in the vicinity. The problem is to obtain services from the LBS provider without disclosing the users' real location to the LBS provider by placing dummies in locations similar to that of the real user. Here, the LBS provider is considered an untrusted entity with information on the locations of the infeasible regions and the respective query probabilities of the surrounding environment. Here, the LBS provider can also predict the cloaking region's center by evaluating all locations $\langle l_1, l_2, \cdots, l_k\rangle$ sent by the user.

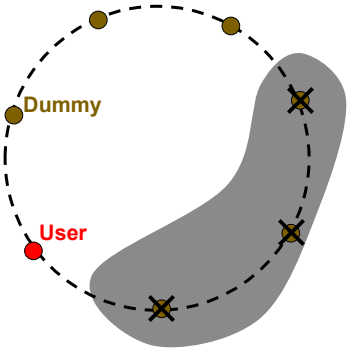

Figure 3.1: CDG

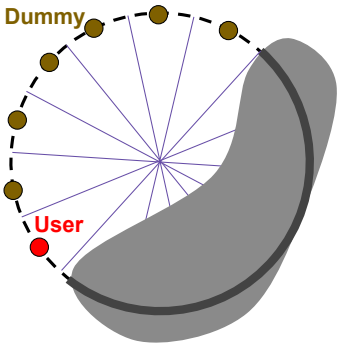

Figure 3.2: ODG

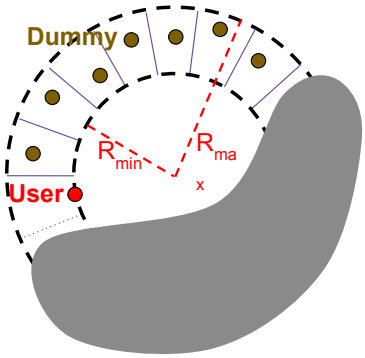

Figure 3.3: EDG

With reference to the above context of the problem, we shall now explain about the drawbacks of existing dummy generation approaches w.r.t. the preservation of user location privacy.

- The Circle-Divided Dummy Generation (CDG) approach [53] does not consider the placement of infeasible regions (grey shaded region), as depicted in the Figure 3.1. Hence, if the dummies are generated in the infeasible regions, the k-anonymity property cannot be appropriately satisfied. Observe that crosses denote the dummies in the infeasible regions in Figure 3.1. An adversary can remove the dummies placed in the infeasible regions from the list of k locations, thereby increasing the probability of finding the real users' locations.

- The Obstacle-Based Dummy Generation (ODG) approach [9] creates dummies on the arc of the circle (indicated by the dashed line) by excluding the area with the infeasible regions (indicated by the bold solid line), as depicted in Figure 3.2. However, if there are many infeasible regions, using this technique would lead to high-density cluster formation, which would make the users' location vulnerable to attack.

- The Efficient Dummy Generation (EDG) approach [44] creates dummies on a thick strip of a circle (forming annulus), as shown in Figure 3.3. It reduces the probability of exposing users' locations in areas with a higher number of infeasible regions. However, in [44] dummies are deployed randomly, thus leaving room for improvement. Moreover, the placement of the user in EDG is always at a fixed distance of $R_{max}/2$ from the center of the cloaking region. Thus, if an adversary knows the center of the cloaking region, he/she would be able to find the user's location with high probability. Additionally, the construction of the annulus in EDG is such that its thickness can only range from $R_{max}/2$ to R_{max}.

3.2 Proposed Approach: Basic Idea

Existing dummy-generation approaches fail to preserve the user's location privacy if an adversary somehow knows the centre of the cloaking region. Moreover, these approaches generate less realistic dummies in areas with more significant infeasible regions. Furthermore, these approaches need to consider the presence of infeasible regions that can change with time. For example, consider a supermarket with opening and closing times as 9 am and 9 pm, respectively. From 9 am to 9 pm, this supermarket can be considered a feasible region; otherwise, this area can be regarded as an infeasible region.

Given these issues with the existing dummy generation approaches, we present an improved dummy generation approach which aims to address these issues. We designate this approach as Annulus-based Gaussian Dummy Generation (**AGDG**). To randomize the distance between the user and the centre of CR we introduce the concept of *virtual cloaking region* to predetermine the user's placement in the CR. Hence, even if an adversary somehow knows the centre of CR, it would not be able to know the user's actual location. The virtual CR is computed such that the dummies have a similar query probability to that of real user. Moreover, under our proposed AGDG approach, dummies are placed such that users' location privacy is maintained even in locations with a higher number of infeasible regions. For this purpose, we propose a notion of an *annulus-based cloaking region* with a Gaussian probability distribu-

tion for placing the candidates such that the distance between candidates is increased. Furthermore, our proposed approach is more flexible with constructing its CR w.r.t. existing approaches. Moreover, to incorporate the presence of a *time-dependent infeasible region*, we use a multi-layered structure to obtain an infeasible region's layout at any particular time. The workflow of AGDG is depicted in Figure 3.4.

3.3 AGDG approach

The AGDG comprises the following steps:

1. Constructing the virtual cloaking region (VCR)
2. Determining the placement of the user in VCR
3. Computing the real cloaking region
4. Determining the placement of dummies

Now we shall discuss each of these steps in detail.

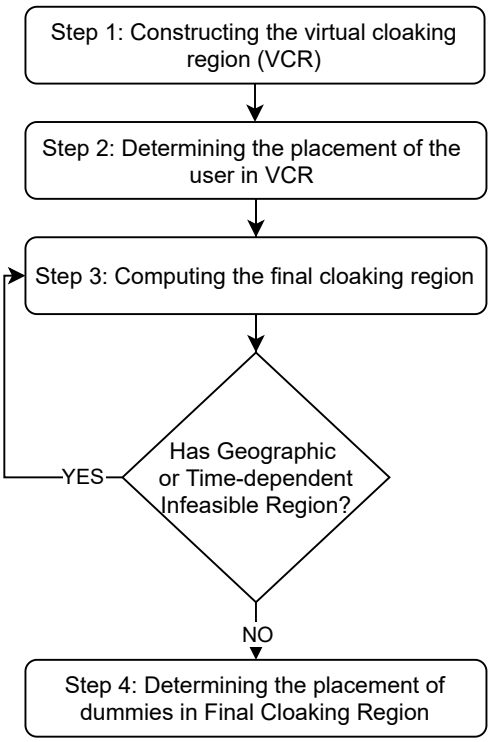

Figure 3.4: Workflow of AGDG

3.3.1 Step 1: Constructing the virtual cloaking region:

We construct a virtual cloaking region (VCR) to make AGDG independent of the users' placement. Using a virtual cloaking region, we can predetermine the user placement w.r.t. the virtual cloaking region by randomizing the distance between the center of the VCR and the user.

Construction of the virtual cloaking region proceeds as follows. A virtual circle with center C is constructed using a user-defined cloaking area A_{min}. The radius R_{max} of the virtual circle should satisfy $\pi R_{max}^2 \geq A_{min}$. In our case, we simply choose $R_{max} = \sqrt{\frac{A_{min}}{\pi}}$.

Another circle with radius R_{min} at the same center C is constructed, thereby forming an annulus (ring shape) with R_{max} and R_{min} as the outer radius and the inner radius, respectively. Here, R_{min} is a user-specified constant. Since the virtual circle is used to create a virtual cloaking region, we have more control over the range of R_{min}, which can range from 0 to R_{max}.

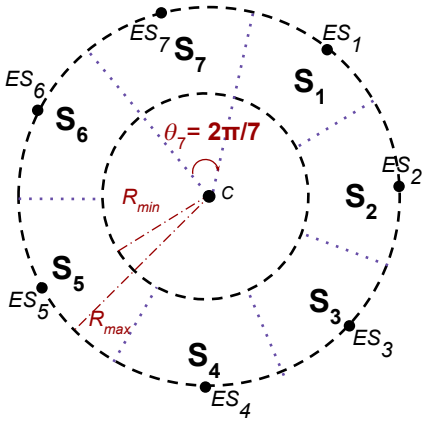

Figure 3.5: Sectors of annulus

To achieve k-anonymity, this annulus is divided into k equal sectors. These k sectors are denoted as $\langle S_1, S_2, \cdots, S_k \rangle$. Either the real user or one of the dummy users

would later be placed in each sector. In the illustrative example in Figure 3.5, k = 7 and $\langle S_1, S_2, \cdots, S_7 \rangle$ represent the sectors of the annulus. In this case, the angle projected by any sector is $2\pi/7$.

3.3.2 Step 2: Determining the placement of the user in VCR:

Using the virtual cloaking region, we are free to predetermine the users' placement w.r.t. the center of VCR. The users' placement must be independent of the distance from the center of the cloaking region. To this effect, a probability distribution at each point in the cloaking region is formed using a Gaussian distribution. This ensures that the user in S_i is placed closer to ES_i as depicted in Figure 3.5. Here S_i is a sector of the VCR, and ES_i is at the edge of S_i. This randomizes the distance between the centre of CR and the user while maximizing the distance between any two candidates. Thereby maximizing the cloaking region. Let (x, y) be a point in sector S_i. The probability distribution $\Phi_i(x, y)$ is given as follows:

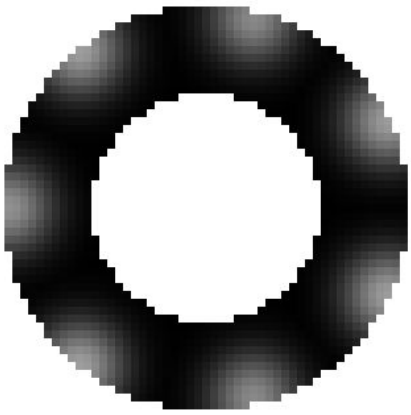

Figure 3.6: Gaussian distr. Φ

$$\Phi_i(x, y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{dist((x,y),ES_i)}{\sigma}\right)^2} \tag{3.1}$$

Here, $dist((x, y), ES_i)$ is the distance between the point (x,y) and ES_i. Figure 3.6 shows the probability distribution Φ, where white represents a higher probability of placing the user, while black indicates a lower probability of placing the user. Then a random sector is selected, and the users' placement is determined in the virtual cloaking region using the Φ distribution. Then we superimpose our virtual cloaking region onto the real-world map with the determined users' placement coinciding with the actual geographical location of the real user on the map.

3.3.3 Step 3: Computing the final cloaking region:

Since the virtual cloaking region is placed in a geographical layout, we now have to consider all the infeasible regions in its neighborhood and select the real final cloaking region.

To place the annulus in the best possible region, we now rotate this annulus n times with an angle of θ= $2\pi/n$ with the real users' position as the pivot, as shown in Figure 3.7. In Figure 3.7, n = 4 and $\theta = 90°$. Here, the shaded regions are considered as the infeasible regions, and n is constant. Then we deal with all the infeasible regions using Equation

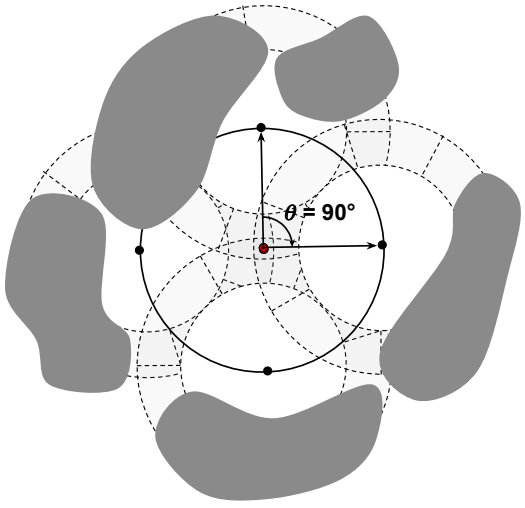

Figure 3.7: Multiple cloaking regions after rotation

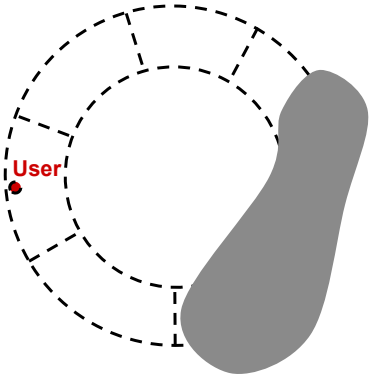

Figure 3.8: Annulus with least θ_{ir}

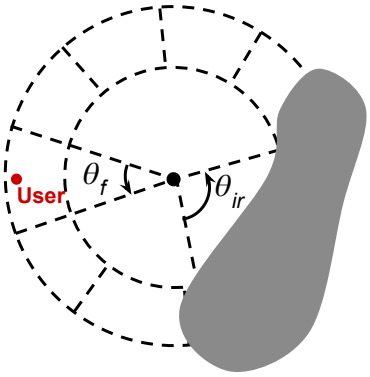

Figure 3.9: Redefined sector size

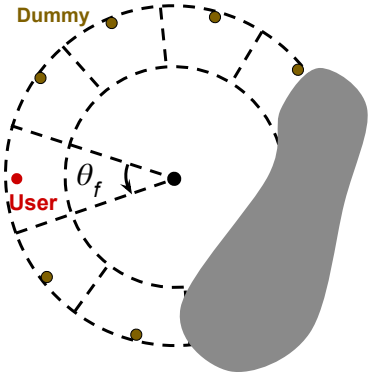

Figure 3.10: Final dummies

3.2 below and select the final cloaking region with the least amount of infeasible regions, as depicted in Figure 3.8.

$$CR_f = (CR_{R_{max}} - CR_{R_{min}}) - CR_{ir} \tag{3.2}$$

Here, CR_f is the area of the cloaking region after removing all the infeasible regions. $CR_{R_{max}}$ is the area covered by the larger circle and is equal to πR_{max}^2. $CR_{R_{min}}$ is the area covered by the smaller circle and is equal to πR_{min}^2. CR_{ir} is the region occupied by infeasible regions. After selecting the cloaking region with the least amount of infeasible regions as the final cloaking region, we now re-adjust the sizes of the sectors in the final cloaking region by using the following equation:

$$\theta_f \times (i-1) \leq S_i < \theta_f \times i \tag{3.3}$$

where θ_f is given as follows:

$$\theta_f = \frac{2\pi - \theta_{ir}}{k} \tag{3.4}$$

where θ_{ir} is the total angle projected by the infeasible region at the center of annulus C, as shown in Figure 3.9.

3.3.4 Step 4: Placing dummies in the final cloaking region:

Once the final cloaking region has been selected, we compute the appropriate placement of dummies in the final cloaking region. We normalize Φ such that the sum of all probabilities of valid cells (locations with no infeasible regions) in a sector S_i equals 1.

$$\bar{\Phi}_i(x,y) = \frac{\Phi(x,y)}{\sum_{(x_j,y_j)\subset S_i} \Phi(x_j,y_j)} \tag{3.5}$$

Any privacy preservation scheme is supposed to maximize the similarity between the dummy and the real users' location. To this accord, a new Gaussian distribution is required to increase the chance of dummies being placed at locations with query probability closer to that of the real user. Let (x, y) be a point in sector S_i and $P_{x,y}$ be the query probability at (x, y). Let (x_r, y_r) be the location of the real user and P_{x_r,y_r} be the query probability at (x_r, y_r). Then the probability distribution $\Psi_i(x, y)$ is given as follows:

$$\Psi_i(x,y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(-\frac{|P_{x_r,y_r} - P_{x,y}|}{\sigma}\right)^2} \tag{3.6}$$

To consider all the infeasible regions and also to ensure that the sum of the values of Ψ in a given sector is equal to 1, we normalize as follows:

$$\bar{\Psi}_i(x,y) = \frac{\Psi(x,y)}{\sum_{(x_j,y_j)\subset S_i} \Psi(x_j,y_j)} \tag{3.7}$$

We then combine the two probability distributions as follows:

$$\Omega = \frac{a\bar{\Psi} + b\bar{\Phi}}{a+b} \tag{3.8}$$

Here, a and b are constants, which are the weight coefficients for each distribution.

Using probability distribution Ω, we now deploy dummies at each sector $\langle S_1, S_2, \cdots, S_k - 1\rangle$ at locations $\langle l_1, l_2, \cdots, l_k - 1\rangle$, as shown in Figure 3.10. Since we use virtual cloaking regions to determine the real users' placement, AGDG would still be relatively safe from attackers with knowledge of the location of the cloaking region's center. Additionally, since we have used the annulus-based cloaking regions, AGDG has better privacy-preserving performance even in locations with more infeasible regions. Moreover, since we have made an effort to maximize both CR and similarity between the user and the dummies' query probability, AGDG can be reasonably expected to perform better than existing approaches.

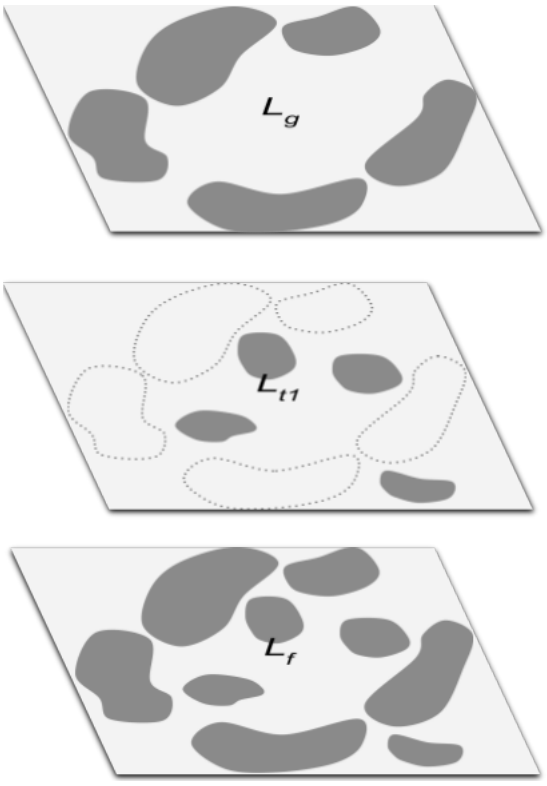

Figure 3.11: Multiple infeasible layers

3.4 Incorporating Time-dependent Infeasible Regions

In the previous approaches, infeasible regions are fixed and do not change with time. However, this is not true in

real-life situations. For example, a school region can be considered an infeasible region from 8 pm to 5 am since school will be closed at that time. Thus if a dummy is placed at this location at that particular time, it can be pruned out with a higher probability. To this end, we propose an improvement to the above approach to incorporate these time-dependent infeasible regions. The main idea is to incorporate a multi-layered structure [6] to obtain an infeasible region's layout at any particular time. Let L_g be the geography-based infeasible region layer as shown in Figure 3.11. Let L_{t_1} be the time-dependent infeasible region layer at time t_1 as shown in the Figure 3.11. The final infeasible region layer, as shown in Figure 3.11 can be calculated as:

$$L_f = L_g \wedge L_{t_1} \tag{3.9}$$

This multi-layered approach can be used to find the final infeasible region layer even if there are more than two layers. For example, areas like the movie theatre will be closed on a particular day. Even those areas can be considered infeasible regions for that day L_d.

$$L_f = L_g \wedge L_d \wedge L_{t_1} \tag{3.10}$$

Thus, this approach can be used for both periodic and non-periodic time-dependent infeasible regions.

Algorithm 1 depicts the steps for determining the location of k-1 dummies. We construct the virtual annulus-based cloaking region and divide it into sectors (see Lines 1-7). Then we determine the placement of the real user in the virtual cloaking region (see Lines 8-11). The virtual cloaking region is placed on a real map, and multiple annuli are constructed to determine the final real annulus with the least infeasible regions (see Lines 12-18). Sizes of the sectors in CR are re-adjusted to consider all the infeasible regions in the surrounding regions (see Lines 19-25). Probability distribution Φ is constructed to consider entropy and then we normalize both Φ and Ψ after removing all the infeasible regions (see Lines 26-28). We then construct probability distribution Ω using Φ and Ψ and then determine the location of $k-1$ dummies (see Lines 29-36).

3.5 Performance Evaluation

This section reports our performance evaluation. Our experiments are performed using a computer having a fifth-generation Intel Core-i5 2.7 GHz processor with 8 GB RAM using *Python 3.0*.

Algorithm 1 AGDG Approach

Input : A_{min}: User defined cloaking region; k: total number of candidates; L_g: geo-based infeasible regions; L_{t1}: time-dependent infeasible regions;

Output : list of $k-1$ dummy locations;

Construct virtual annulus A with inner radius $R_{min} \leftarrow \sqrt{\frac{A_{min}}{\pi}}$;
$\langle S_1, S_2, \cdots, S_k \rangle \leftarrow \{\}$; {sectors of the virtual annulus}
for $S_i \in \langle S_1, S_2, \cdots, S_k \rangle$ **do**
    for $(x, y) \in S_i$ with 'angle at center' in range $(\frac{2\pi(i-1)}{k}, \frac{2\pi(i)}{k})$ **do**
        S_i.append((x, y));
    end for
end for
Select a sector S_u randomly from $\langle S_1, S_2, \cdots, S_k \rangle$;
for (x, y) in S_u **do**
    $(x, y) \leftarrow \Phi_i(x, y)$ {using Equation 3.1}
end for
Select (x_u, y_u) from S_u using Φ distribution;
Place virtual annulus such that (x_u, y_u) co-inside with users' real location on the map;
$\langle A_1, A_2, \cdots, A_n \rangle \leftarrow \{\}$; { multiple annuli with user's location as pivot}
for $A_i \in \langle A_1, A_2, \cdots, A_n \rangle$ **do**
    $A_i \leftarrow$ rotate A by $2\pi/k$ with (x_u, y_u) as an anchor point
end for
$L_f = L_g \wedge L_{t1}$; {final infeasible region layer}
Select annulus A with the least amount of infeasible region in L_f; {using Equation 3.2}
Readjust size of sectors {using Equation 3.3}
for $S_i \in \langle S_1, S_2, \cdots, S_k \rangle$ **do**
    for (x, y) with 'angle at center' in range $(\theta_{eff} \times (i-1), \theta_{eff} \times (i))$ **do**
        S_i.append((x, y)) {readjust size of sectors}
    end for
end for
Create probability distribution Ψ to consider Entropy {using Equation 3.6};
Normalize Ψ {using Equation 3.7};
Normalize Φ {using Equation 3.5};
Combine two distribution $\Omega = \frac{a\bar{\Psi} + b\bar{\Phi}}{a+b}$;
$\langle l_1, l_2, \cdots, l_{k-1} \rangle \leftarrow \{\}$; {location of dummies}
for $S_i \in \langle S_1, S_2, \cdots, S_k \rangle$ **do**
    for (x, y) in S_i **do**
        $l_i \leftarrow \Phi_i(x, y)$ {using Equation 3.1}
    end for
end for
return $\langle l_1, l_2, \cdots, l_k - 1 \rangle$;

Our experiments consider a two-dimensional layout with 1000 × 1000 cells. Each cell has a dimension of 10 × 10 square meters. We assume R_{max} = 25 meters hence, the basic CR requested by the user was assumed to be 1963.4 as ($\pi \times 25 \times 25$) cells. Infeasible regions were randomly arranged in the layout, depending on the infeasible region ratio (IRR), which ranged from 0 to 0.9. Time-dependent infeasible regions (TIR) were randomly arranged in the layout. The ratio of TIR in the layout ranged from 0 to 0.3 of the entire layout. Our performance study parameters were adopted from the existing work in

[44]. Table 3.1 summarizes all the performance study parameters. Each experiment was conducted 100 times; hence the results presented here represent the average values over 100 runs of each experiment.

Table 3.1: Parameters used in our experiments

S.N.	Parameter	Default Values	Variations
1	k	10	[2, 3, $\cdots$, 29, 30]
2	IRR	0.3	[0,0.1, $\cdots$, 0.8, 0.9]
3	TIR	0.1	[0,0.1, $\cdots$, 0.3]
4	R_{min}	15	-
5	R_{max}	25	-
6	n	4	-
7	a	1	-
8	b	1	-
9	σ_{Φ}	0.001	-
10	σ_{Ψ}	2	-

3.5.1 Comparative Theoretical Analysis

The user's location privacy must hold against an adversary, which can detect the center of the cloaking region using the k locations sent by the user to the LBS provider. It is crucial to evaluate the robustness of AGDG in such scenarios. Figure 3.12 shows our experimental results w.r.t. such an adversary. In Figure 3.12, the x-axis represents the number of candidates (k), while the y-axis represents $PFind$, which is the negative log of the probability that the adversary finds the real user.

Observe that in case of CDG and EDG approaches, value of $PFind$ is close to zero. This implies that an adversary can find the real user with a very high probability. In case of CDG, the real user is at the center of the cloaking region. Thus, an adversary, which knows the center of CR, would know the user's real location. In contrast, in case of EDG approach, the real user's location always lies at a distance of $R_{max}/2$ from the center of the cloaking region. Hence, an adversary, which knows the center of CR, can select all the locations at a distance of $R_{max}/2$ and find the real user's location with a very high probability.

On the other hand, the ODG approach has non-zero value of $PFind$ because all the candidates are at a fixed distance from the cloaking region center. However, in the ODG approach, value of $PFind$ is not close to the value of k. Because in areas with more infeasible regions, the ODG approach forms dense clusters of candidates, thus degrading the quality. It can be observed from the results in Figure 3.12 that the AGDG approach shows $PFind$ close to the value of k. The AGDG uses a virtual cloaking region to predetermine the real user's location on the map. Hence, it makes the location of real users and dummies similar w.r.t. the distance from the center of the cloaking region. Thus, in AGDG, even if the adversary knows the center of the cloaking region, the k-anonymity property still holds good, thereby protecting the user's location privacy.

3.5.2 Experimental Evaluation

To evaluate the performance of AGDG, we use two metrics namely (a) Effective Cloaking Region (ECR) and (b) Entropy (H). They are defined as follows:

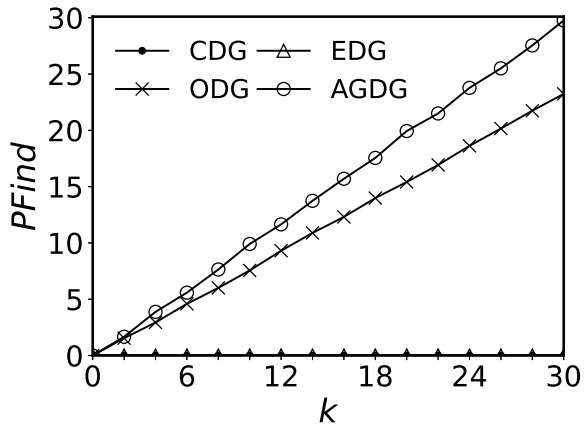

Figure 3.12: $PFind$ Vs k

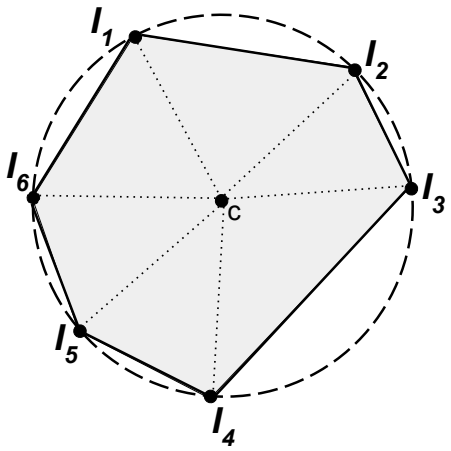

Figure 3.13: Effective Cloaking Region

(a) Effective Cloaking Region (ECR): Effective Cloaking Region is a widely used metric [37] to compare the effectiveness of a privacy preservation algorithm. As shown in Figure 3.13, it measures the maximum area covered by k location points (k-1 dummy locations and a real user location). We have computed ECR based on adding the area of triangles formed by all the adjacent locations and the centre of the cloaking region as follows:

$$ECR = \sum_{i=1}^{k} Area(l_i, l_{(i+1)\%(k+1)}, C) \tag{3.11}$$

Here, the $Area$ function returns the area of a triangle, given three vertices. Thus, ECR is equal to 0 for $k \leq 2$.

(b) Entropy (H): Entropy is widely used to measure the degree of anonymity in location-based services [39]. It indicates the uncertainty in determining the reallocation of an individual from all the candidates. Usually, the query probability (p) of each possible location is used as supplemental information to construct the entropy-based privacy metric. We thus assign each possible location a query probability, denoted by p_i, and the sum of all probabilities p_i is 1. As a result, the entropy of identifying the real user from the k candidate set can be computed as follows:

$$H = \sum_{i=1}^{k} -\frac{p_i}{\sum_{i=1}^{k} p_i} log_2 \left(\frac{p_i}{\sum_{i=1}^{k} p_i} \right) \tag{3.12}$$

Thus, maximum entropy $H_{max} = log_2 k$ is achieved when all the k locations have same probability of $1/k$.

We compare AGDG using these performance metrics with the CDG [53], ODG [9] and EDG [44] approaches, which we had discussed earlier in Section 2.2. We adapt these reference approaches with essentially the same setup as AGDG in the interests of meaningful comparison.

3.5.3 Effect of Varying the Number of Candidates

We first evaluate the relationship between k and the entropy (H). In Figure 3.16, the x-axis represents the number k of candidates, while the y-axis represents entropy H. Results without time-dependent infeasible regions (TIR) are shown in Figure 3.14. In contrast, Figure 3.15 considers the presence of TIR. We can observe from the results in Figure 3.14 that in all the approaches, H increases with an increase in k. Since the greater the number of dummies, the more challenging it will be for the adversary to find the real user. Observe that EDG and CDG have poor performance because they do not consider the query probability. On the other hand, ODG and AGDG have higher H than EDG and CDG. Because in ODG and AGDG, query probability is considered when placing candidates.

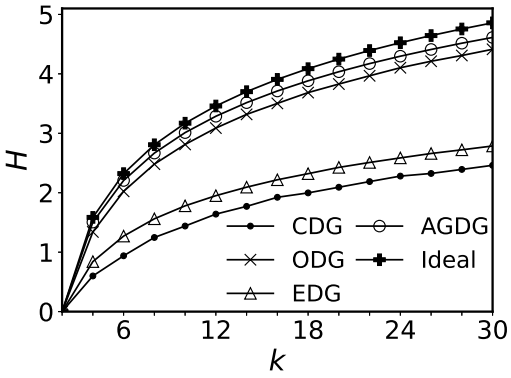

Figure 3.14: H without TIR

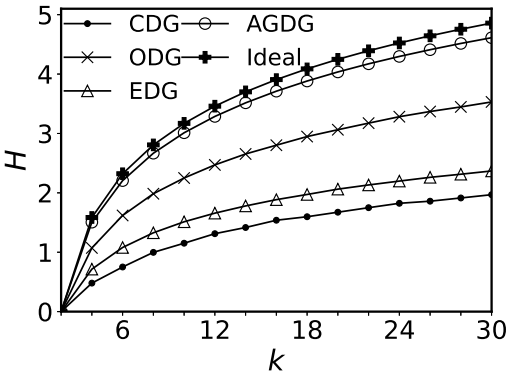

Figure 3.15: H with TIR

Figure 3.16: Effect on entropy H with variations in k

It can be observed that AGDG has better performance than ODG. Moreover, H in AGDG is close to the best possible value because, in AGDG, we use Φ distribution to place dummies such that query probabilities of the locations are close to that of the real user. In practice, we can find many regions that will be closed or not accessible to the public for specific periods, thus forming a time-dependent infeasible region TIR. Results of AGDG in the presence of such time-dependent infeasible regions are shown in Figure 3.15. We can see that for all the previous approaches compared to the results in Figure 3.14, the results in Figure 3.15 have inferior performance. This is because only AGDG considers time-dependent infeasible regions. In contrast, the other approaches may place their dummies on a TIR, thereby rendering them ineffective.

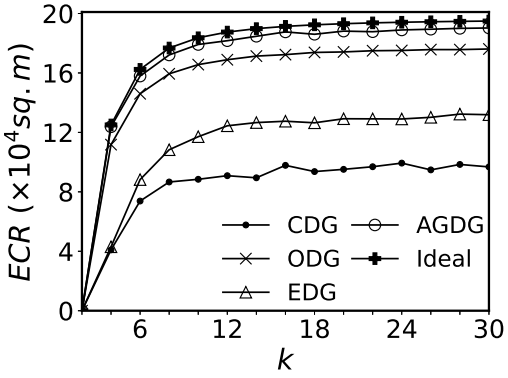

Figure 3.17: ECR without TIR

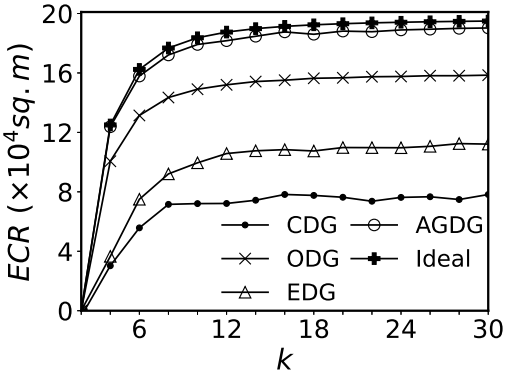

Figure 3.18: ECR with TIR

Figure 3.19: Effect on effective-cloaking region ECR with variations in k

Since the user's location privacy is closely related to the cloaking region, we evaluate the privacy area provided by different schemes with increasing k. Our experimental results at $IRR = 0.3$ are shown in Figure 3.19. In Figure 3.19, the x-axis represents the number of candidates (k), and the y-axis represents the effective cloaking region (ECR) formed by these k candidates. It can be observed in the results from Figure 3.17 that ECR increases with variations in k. This occurs because the greater the number of candidates, the more area they will cover in a given region. It can be observed that CDG has poor performance because CDG does not consider the placement of infeasible regions. Hence, there is a chance of dummies being placed in the infeasible regions, thereby rendering them ineffective for location privacy preservation. Observe that EDG has lower CR than that of ODG at IRR = 0.3 because in EDG, dummies are placed using a random distribution. However, in case of ODG, dummies are placed on the circle's circumference. Since there are fewer infeasible regions (IRR = 0.3), dummies can be placed far from each other on the circumference.

On the other hand, AGDG has better performance than the other approaches because we place dummies using the Ψ distribution, thereby making candidates far apart. In practice, time-dependent infeasible regions can be formed when certain areas are closed or not accessible to the public for some time. Results of AGDG in the presence of such time-dependent infeasible regions are shown in Figure 3.18. We can see that for all the previous approaches compared to the results in Figure 3.17, the results in Figure 3.18 have the worst performance. This is because none of the previous approaches consider the placement of time-dependent infeasible regions in their vicinity. Hence, the existing approaches place dummies on the infeasible regions, thereby rendering them ineffective.

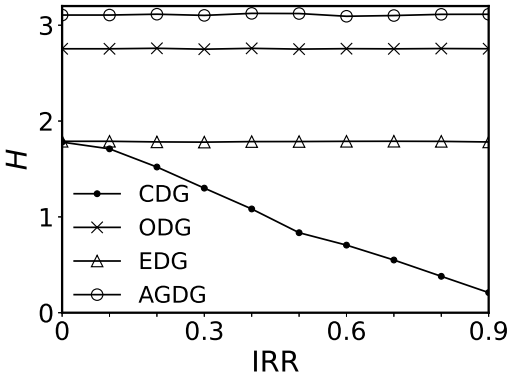

Figure 3.20: H without TIR

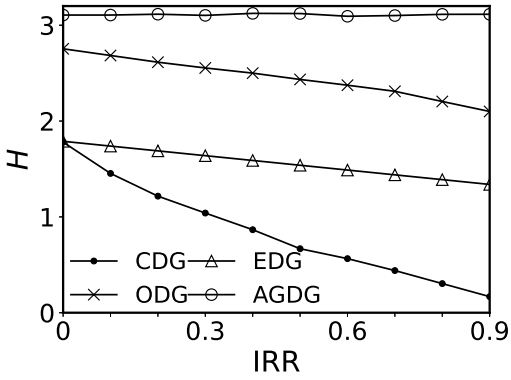

Figure 3.21: H with TIR

Figure 3.22: H with variations in IRR

3.5.4 Effect of Varying the Ratio of Infeasible Regions

Privacy preservation schemes should ensure that their dummy location closely resemble the real user's location, even in locations with a higher number of infeasible regions. Hence, we evaluate the entropy (H) with an increasing infeasible region ratio (IRR). Figure 3.22 depicts the results of our experiments. In Figure 3.22, the x-axis represents the probability of placing infeasible regions on the layout (IRR), while the y-axis represents the system's entropy formed by k candidates. It can be observed from the results in Figure 3.20 that CDG has a significant decrease in H as IRR increases. This occurs because, CDG approach does not consider the placement of infeasible regions, thereby placing dummies in infeasible regions, making them useless. All of the other approaches have nearly constant H throughout the experiment. These schemes consider only feasible regions, thereby making them immune to any changes in IRR. It can be observed that EDG has less H as compared to that of ODG and AGDG because EDG does not consider the query probability of its environment.

On the other hand, AGDG has better performance than ODG because AGDG uses the Φ distribution to place dummies in locations such that their query probabilities are close to that of a real user's location. Since the public cannot access certain regions for some period, time-dependent infeasible regions TIR will be formed. Results of AGDG in the presence of such time-dependent infeasible regions are shown in Figure 3.21. We can observe that for all the previous approaches compared to the results in Figure 3.20, the results in Figure 3.21 have the worst performance. This is because none of the previous approaches consider the placement of time-dependent infeasible regions in its surrounding. Hence, all the previous approaches may place dummies on these time-dependent infeasible regions, thereby rendering them ineffective.

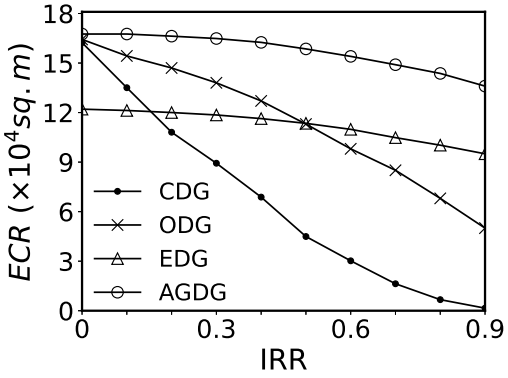

Figure 3.23: ECR without TIR

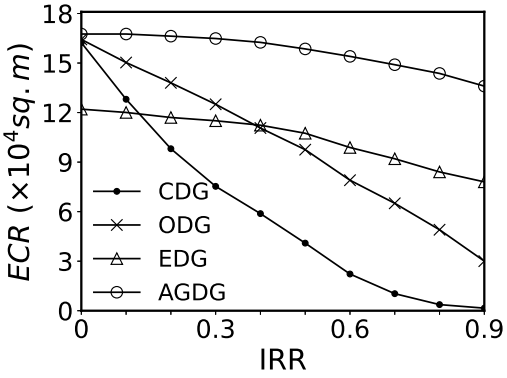

Figure 3.24: ECR with TIR

Figure 3.25: ECR with variations in IRR

Privacy preservation schemes should function in locations with various amounts of infeasible regions. Hence, we evaluate the effective cloaking region with an increase in the infeasible region ratio (IRR). Figure 3.25 shows the results of our experiments. In Figure 3.25, the x-axis represents the IRR, while the y-axis represents the ECR formed by k candidates. It can be observed in Figure 3.23 that AGDG, ODG and CDG have similar ECR when there are no infeasible regions. Because in AGDG, ODG and CDG candidates are placed at the circumference of the cloaking region when there are no infeasible regions. In the case of EDG, at locations with fewer infeasible regions, EDG has less ECR than AGDG, ODG and CDG since in EDG, dummies are placed in the annulus randomly.

Observe that CDG has a significant decrease in ECR as IRR increases. This occurs because CDG does not consider infeasible regions hence, it places dummies in infeasible regions. Similarly, even in ODG, ECR is greater when IRR is less, but with an increase in IRR, ECR decreases. Because in ODG, candidates are placed only on the circumference of the circle, as IRR increases, candidates from clusters on the circumference, thus reducing ECR. On the other hand, both AGDG and EDG show a smaller decrease in ECR as IRR ratio increases. This is because dummies are placed in the annulus in case of AGDG and EDG.

Moreover, AGDG has better ECR than EDG because, in AGDG, we use Ψ probability distribution for the placement of candidates. In contrast in EDG approach, candidates are placed randomly on the annulus. In practice, we can find many regions that will be closed or not accessible to the public for a certain period, thus forming a time-dependent infeasible region TIR. Results of AGDG in the presence of such time-dependent infeasible regions are shown in Figure 3.24. We can observe that for all the previous approaches, the results in Figure 3.23 have better performance than the results in Figure 3.24. This is because only AGDG considers time-dependent infeasible regions. On the other hand, previous approaches may place their dummies on a TIR, thereby rendering them ineffective.

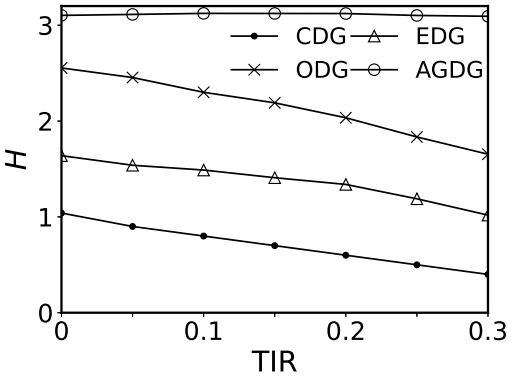

Figure 3.26: H Vs TIR

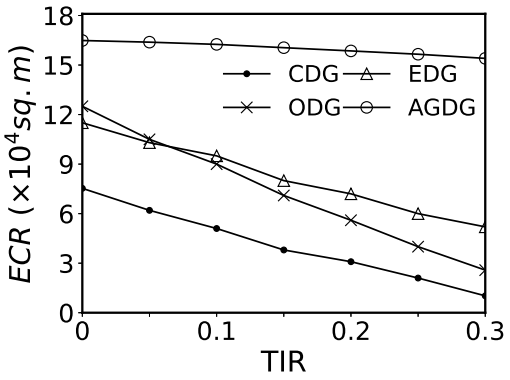

Figure 3.27: ECR Vs TIR

Figure 3.28: variations in TIR

3.5.5 Effect of Varying the Time-dependent Infeasible Regions

Privacy preservation schemes should protect users' location privacy at all times of the day. In different parts of the day, some areas can be considered infeasible regions (i.e., time-dependent infeasible regions). We evaluate the entropy with an increasing time-dependent infeasible region. Figure 3.26 depicts the results of our experiments. In Figure 3.26, the x-axis represents the probability of placing time-dependent infeasible regions on the layout (TIR), while the y-axis represents the system's entropy (H).

It can be observed from the results in Figure 3.26 that in all the previous approaches, entropy decreases as TIR increases. In all the previous approaches, the dummies are placed at locations that might be considered infeasible regions; since these approaches do not consider the time-dependent infeasible region. In contrast, the entropy in AGDG is nearly constant since AGDG considers the placement of time-dependent infeasible regions in its surroundings. AGDG places dummies only in the feasible region, thereby making it immune to any changes in the amount of time-dependent infeasible regions.

Since the privacy preservation scheme should function in locations with various amounts of time-dependent infeasible regions. We evaluate the effective cloaking region with varying amounts of the TIR. Figure 3.27 shows the results of our experiments. In Figure 3.27, the x-axis represents the probability of placing time-dependent infeasible regions on the layout, while the y-axis represents the effective cloaking region (ECR) formed by k candidates.

It can be observed from the results in Figure 3.27 that CDG, ODG and EDG have a substantial decrease in the effective cloaking region. As in these, the location of dummies might be in a time-dependent infeasible region. This decreases the effective cloaking region since dummies at time-dependent infeasible regions can be pruned out from the entire candidate set. On the other hand, AGDG shows only a slight decrease in ECR as the time-dependent infeasible region increases. This is because

AGDG evaluates dummies' placement by considering time-dependent infeasible regions in the user's surroundings.

3.6 Summary

In this chapter, we addressed the issue of location privacy in the context of dummy generation and proposed the Annulus-based Gaussian Dummy Generation (AGDG) approach as a solution for efficiently protecting users' location information. Through theoretical analysis and extensive performance evaluation, we demonstrated that our proposed AGDG approach effectively improves location privacy, including in regions with time-dependent infeasible regions, compared to existing approaches. Our results highlight the potential of AGDG as a promising method for protecting users' location privacy in location-based services. In the next chapter, we propose the improved privacy preserving approach for Spatial Range Queries.

Chapter 4

Enhancing Location and Intent Privacy for Spatial Range Queries

In this chapter, we present our proposed approach to improve the efficiency of distributed spatial cloaking-based approaches. In Section 4.1, we present our problem background. Section 4.2 explains the basic idea of the proposed algorithm. Section 4.3 explains the proposed *ijk*Cloak approach. In Section 4.4, we conduct theoretical analysis to show the superiority of our approach compared to the previous approaches. In Section 4.5, experimental results are reported and show that the *ijk*Cloak is more efficient than the existing approaches. Finally, Section 4.6 concludes the chapter with a summary.

4.1 Problem Background

In this section, we explain the problem background. We will then define essential terms, and provide the problem statement.

Consider a mobile user, present in distributed mobile network environment and has access to other mobile users around it. A distributed location-based privacy-preserving approaches leverage their mobile network [11] to hide the location of the user. We shall explain the following terms relevant to distributed location-based privacy-preserving approaches followed by our problem statement.

Mobile user: A mobile user refers to a GPS-enabled mobile phone user, who is capable of communicating with other mobile users through a multi-hop routing protocol [10, 27] without the need for support from fixed communication infrastructure or centralized servers. A mobile user, who desires to request SRQ from the LBS provider, is termed as the *request originator* or *query originator*, denoted by u_c.

Peer: Let u_c be a mobile user. All the other mobile users are termed *peers* to u_c. The peers are capable of communicating with each other mobile users and LBS provider.

Spatial range query: Spatial Range Query (SRQ) is a location-based query used to retrieve objects within a specific spatial range around a given location [30]. The SRQ can be represented as $\langle l, r, q \rangle$, where l represents the query location. This location can either be the user's current location or a specific location of interest. The notation r is the spatial range around l within which the user wants to retrieve desired objects, and q denotes the intent of the user. For example for a query *"What are the cancer hospitals within 5 km of my current location?"* the user's intent would be a string "cancer hospitals".

Location-based service provider: The Location-Based Service (LBS) provider is an online service that provides responses to given SRQs. The result for the SRQ query $\langle l, r, q\rangle$ from the LBS provider is a set of all the desired objects within r distance around l, represented as $R_{\langle l,r,q\rangle}$. LBS provide can communicate with peers and mobile user.

In this thesis, we consider that the mobile peers and the LBS provider are not trusted entities. As a result, both user query location and intent information can be compromised. Moreover, the mobile peers in the network can also collude with each other and, as a result, both the user's query location and intent information could be compromised. This implies that sending the exact query directly to the mobile peers in its surroundings would violate the user query location and intent privacy.

Problem Statement: Consider a mobile network environment where the LBS provider and mobile peers can potentially act as adversaries, a mobile user u_c seeks to obtain the result of an SRQ $\langle l_c, r_c, q_c\rangle$ from the LBS provider. The problem is to process SRQ by protecting its location and intent privacy.

4.2 Proposed Approach: Basic Idea

In a mobile network environment, the issue is to get the results of user SRQ while protecting the user query location and intent information from the LBS provider and peers.

Existing location-based privacy protection approaches often assume that all peers are trustworthy, which makes the user vulnerable to attacks from malicious peers. Few other approaches often reveal the user's actual location in the final request sent to the peers or LBS provider, which compromises the user's privacy. Some approaches also rely on a central server, which can create a single point of failure. Finally, many existing approaches fail to adequately protect the user's intent adequately. Some works have attempted to address these issues by using large clusters of peers to hide the user's information. Still, these solutions might not be feasible for a highly dynamic mobile network environment.

We now present the notion of ijk-anonymity to preserve the user's query location and intent information while getting results for SRQ.

About ijk-anonymity: We first explain about i, j, and k parameters. Consider the query "What are the top four cancer hospitals within 5 km of my current location?". For such a query to hide the query intent (which is "top four cancer hospitals"), $i-1$ dummy intents (such as "best gyms", "jogging park", and "nearest children's hospitals") are added to the query intent information q. To hide the location of the query (which is "my location" in this case), we fragment the given location into j fragments queries to obscure the user's actual location. During the processing time of the j queries, k denotes the total number of queries received by any adversary. Note that the LBS provider may receive several queries from different users. Some of the queries are the j fragment queries. It can be noted that ($k > j$). The notion of ijk-anonymity is defined as follows.

ijk-anonymity: Let $\mathcal{K}$ be a set of SRQ queries received by LBS provider ($|\mathcal{K}| = k$). Let $\mathcal{J} \subset \mathcal{K}$ and $|\mathcal{J}| = j$. Let $\mathcal{I}$ is the set of intents where $|\mathcal{I}| = i$. We say a spatial range query $\langle l, r, q\rangle \notin \mathcal{K}$ is said

to possess ijk-anonymity with set $\mathcal{K}$ if l can be calculated from set $\mathcal{K}$ if and only if set $\mathcal{J}$ is selected from $\mathcal{K}$. Moreover, the query intent q cannot be distinguished from at least $i-1$ intents in set $\mathcal{I}$.

In the case of ijk-anonymity, it can be noted that the probability of finding the user's query intent and the location from $\mathcal{K}$ is $P_{ijk} \leq \frac{1}{i \cdot \binom{k}{j}}$.

The basic idea is to improve the location and intent privacy of the SRQ based on ijk-anonymity. The user's SRQ is fragmented into j SRQ (referred to as fSRQ) queries to hide the actual location in the user's query. Moreover, $i-1$ dummy intents are added to the fSRQs to hide the actual query intent. We term the proposed approach as ***ijk*Cloak**. Note that, in *ijk*Cloak, the query is processed with significant improvement in privacy due to ijk-anonymity using the concepts of query fragmentation and insertion of dummy user intents. This is because selecting correct j SRQs from k received queries is challenging.

4.3 *ijk*Cloak Approach

In this section, we first explain an overview of network communication in the proposed approach. Next, we present the proposed *ijk*Cloak approach.

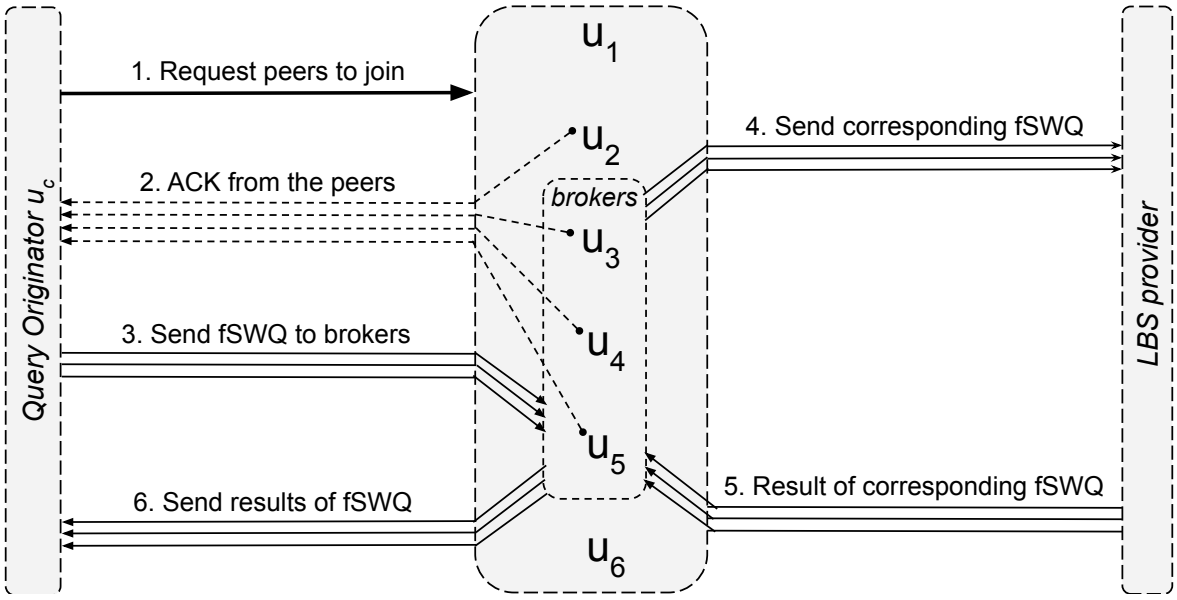

Figure 4.1: Illustrative example of query processing in *ijk*Cloak

Overview of communication: Figure 4.1 depicts an illustrative example of network communications in *ijk*Cloak approach. Consider mobile user u_c, who wants to find the result of its SRQ $\langle l_c, r_c, q_c \rangle$. Firstly, mobile user u_c computes j number of fSRQ $\mathcal{Q} = \{\langle l_1, r_1, q_1 \rangle, \langle l_2, r_2, q_2 \rangle, ...\langle l_j, r_j, q_j \rangle\}$ based on user query location l_c and intent information q_c. Next, mobile user u_c uses a multi-hop routing protocol to request peers in the surrounding environment to join the network (See Figure 4.1 Step 1). Next, the peers in the surrounding environment willing to help user u_c will send an acknowledgement to u_c (See Step 2 here peers $\{u_2, u_3, u_4, u_5\}$ have sent an acknowledgement to u_c). Among the the

acknowledged peers, the user u_c selects j peers $\mathcal{J} = \{u_{a_1}, u_{a_2}...u_{a_j}\}$ at random (In Figure 4.1, we consider j=3, and peers $\{u_3, u_4, u_5\}$ are selected by the user at random). These j peers are termed as the *brokers*. Each of the j *brokers* will receive one fSRQ query chosen at random from $\mathcal{Q}$ (See Step 3). These j brokers send their corresponding fSRQ to the LBS provider (See Step 4). The LBS provider sends results of these queries $\{R_{\langle l_1,r_1,q_1\rangle}, R_{\langle l_2,r_2,q_2\rangle}...R_{\langle l_j,r_j,q_j\rangle}\}$ to corresponding brokers (See Step 5). Next, all the brokers send back their results to user u_c (See Step 6). Finally, the user u_c computes the final result to his SRQ $R_{\langle l_c,r_c,q_c\rangle}$ using the results of the fSRQ sent by the brokers.

Algorithm 1: *ijk*Cloak($l_c, r_c, q_c, i, j, IR, \delta, I_s$)

Input : l_c: query location of u_c; r_c: range of the query; q_c: user's query intent; i: number of intents added in the user's query (i-1 dummy intent and query intent); j: number of fragmented SRQ required; IR: set of infeasible regions in the surrounding; δ: location randomization constant; I_s: set of dummy intents; h: hop distance (initially set to zero);
Output : $R_{\langle l_c,r_c,q_c\rangle}$: result for the SRQ;
Variable : fSRQ: list of fragment spatial range query; ackPeers: peers who have sent acknowledgement; $LBS\_results$: results received from brokers;

1 fSRQ = fSRQ_Computation($l_c, r_c, q_c, i, j, IR, \delta, I_s$);
2 ackPeers = NEED_PEERS_Originator(h, j);
3 $LBS\_results$ = DATA_TRANSFER_Originator(ackPeers, fSRQ);
4 $R_{\langle l_c,r_c,q_c\rangle}$ = Collect($LBS\_results$);

Algorithm 1 depicts the steps involved in the proposed *ijk*Cloak approach. The algorithm takes the following inputs to perform a spatial search while ensuring the location and intent privacy of the query originator's u_c query. The first input is the location, denoted by l_c, which serves as the centre of the spatial search (x_c, y_c). The range of the query is specified by r_c, representing the maximum distance from l_c within which u_c is searching for desired objects. Another input to the algorithm is the intent of the user's query, denoted by q_c, representing the specific place or category of interest for u_c. A user-defined constant i specifies the number of dummy intents added to the user's query intent information to protect the user's intent privacy. The number of fragmented queries generated is specified by j. The algorithm also takes a set of infeasible regions in the surrounding area, denoted by IR. To further protect the user's privacy, a constant, denoted as δ, randomizes the user query location. A set of dummy intents, denoted as I_s, adds noise to the search queries, making it difficult for an attacker to determine the user's query intent. Finally, the hop distance, denoted as h, is

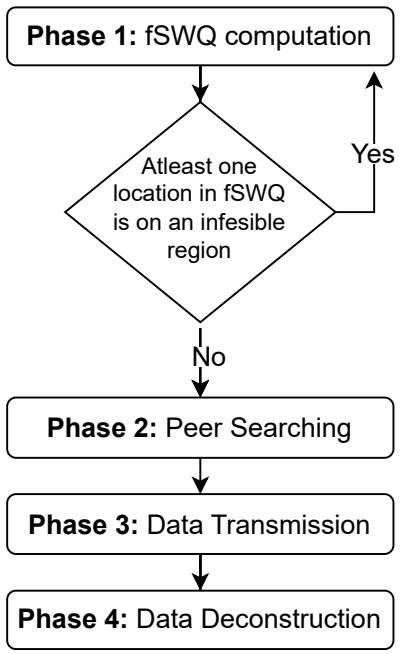

Figure 4.2: Workflow of our approach

the number of hops required to be made by the peers. Initially, the hop distance is set to zero. The output to the algorithm is the set of desired objects that satisfy the user's actual query, denoted as $R_{\langle l_c, r_c, q_c \rangle}$.

The workflow of our approach is depicted in Figure 4.2. It comprises the following four phases as follows:

1. Fragmented SRQ computation phase
2. Peer Searching phase
3. Data Transmission phase
4. Data Deconstruction phase

Now, we shall discuss each phase in detail.

4.3.1 Fragmented SRQ computation phase

The proposed approach incorporates the notion of ijk-anonymity in a mobile network to hide both location and intent of the user's query. In this regard, the mobile user u_c calculates j fragmented spatial range queries (fSRQ) from its actual SRQ. All the calculations in this phase are performed on the user's u_c mobile device. For computing fSFQs, we have to consider the presence of infeasible regions, location randomization, and intent randomization.

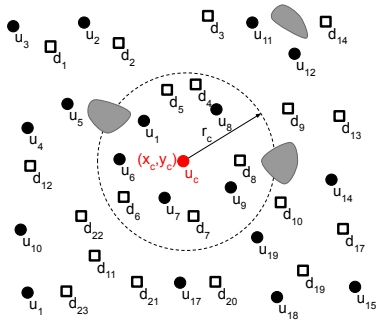

Figure 4.3: Coordinates of u_c

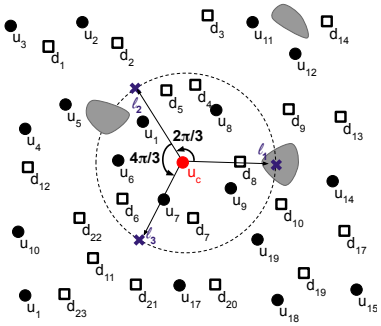

Figure 4.4: Location in fSRQ

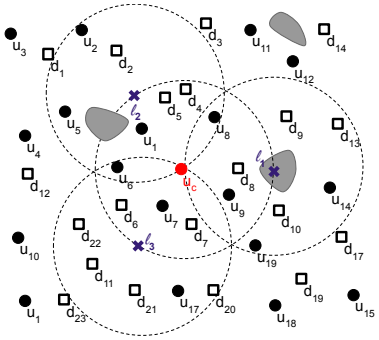

Figure 4.5: Fragmented Spatial Range Queries

As an example, consider a mobile network environment with 23 desired objects $\{d_1, d_2, \ldots, d_{23}\}$ represented as a hollow square in Figure 4.3. Moreover, there are 19 mobile users, $\{u_1, u_2, \ldots, u_{19}\}$ represented as solid circles. The dotted circles represent the spatial range of the query. Suppose the user u_c wants to get the result of her SRQ $\langle l_c, r_c, q_c \rangle$. Here, l_c is the location of the user's query u_c represented as (x_c, y_c), r_c is the spatial query range, and q_c is the user's query intent. The result of SRQ $\langle l_c, r_c, q_c \rangle$ is a set of desired objects $R_{\langle l_c, r_c, q_c \rangle} = \{d_4, d_5, d_6, d_7, d_8\}$.

Methodology to compute fSRQs: The method to compute the j number of fSRQs $\mathcal{Q} = \{\langle l_1, r_c, q_c \rangle, \langle l_2, r_c, q_c \rangle, \ldots \langle l_j, r_c, q_c \rangle\}$ is as follows. First, we calculate the fragmented locations (l_1, l_2, $\ldots l_j$) in

the fSRQs. In this approach, j fragmented locations are set at a distance of r_c units from u_c and are equidistant from each other as shown in Figure 4.4. The fragmented locations are calculated as follows:

$$l_n = \Big(x_c + r_c \cos((n-1)\tfrac{2\pi}{j}), y_c + r_c \sin((n-1)\tfrac{2\pi}{j})\Big)$$

Here, $n \in \{1, 2, \ldots j\}$ corresponding to each fragmented location $\{l_1, l_2, \ldots l_j\}$. Figure 4.5 depicts the results of fSRQs. In the example, the result of fSRQ $R_{\langle l_1, r_c, q_c\rangle}$ is $\{d_8, d_9, d_{10}, d_{13}\}$ because these destinations are at a spatial range of r_c from l_1. Similarly, the result of $R_{\langle l_2, r_c, q_c\rangle}$ and $R_{\langle l_3, r_c, q_c\rangle}$ are $\{d_2, d_4, d_5\}$ and $\{d_7, d_8, d_{11}, d_{21}, d_{22}\}$ respectively. Once user u_c gets all the results of fSRQs, he will select all the intended destinations within a distance of r_c from query location l_c. Here, the destinations d_4, d_5, d_6, d_7, d_8 are within a range of r_c from l_c and thus will be the result of $\langle l_c, r_c, q_c\rangle$. Observe in Figure 4.5 that this is the expected result. This is because the union of all the areas covered by j fSRQs will always cover the user's SRQ range area [51].

Algorithm 2: fSRQ_Computation($l_c, r_c, q_c, i, j, IR, \delta, I_s$)

Input : l_c: query location of u_c; r_c: range of the query; q_c: user's query intent; i: number of intents added in the user's query (i-1 dummy intent and query intent); j: number of fragmented SRQ required; IR: set of infeasible regions in the surrounding; δ: location randomization constant; I_s: set of dummy intents;
Output : R: list of fSRQs;
Variable : *mapIR*: hash map of infeasible regions in the layout;

```
1  mapIR = [];
2  for l in IR do
3      mapIR[l] = true
4  repeat
5      for n in 1, 2, ... j do
6          α_n, β_n ← rand(0, δ);
7          a ← rand(0, 1);
8          l_n calculated using Equation 4.1;
9          r_n calculated using Equation 4.2;
10         q_r ← randomly select i − 1 intents from I_s;
11         q_n ← q_c ∪ q_r;
12         R ← ⟨l_n, r_n, q_n⟩ ∪ R
13 until !(mapIR[l_1] or mapIR[l_2] ... or mapIR[l_j]);
14 return R
```

Methodology to handle infeasible regions: In the preceding case, we have not considered the presence of infeasible regions in the geographical area in which mobile user is located. Recall that a geographic area is considered an infeasible region if it is unlikely that a mobile user will physically be present at the location [43]. Given a set of infeasible regions in a spatial environment, we pre-compute and map them to our spatial environment using the steps in Algorithm 2 Lines 1-3. Now, consider a scenario shown in Figure 4.6, where the fragmented query location l_1 is on an infeasible region. To handle such cases, we recalculate the fragmented locations such that none of the fragmented locations are in the infeasible

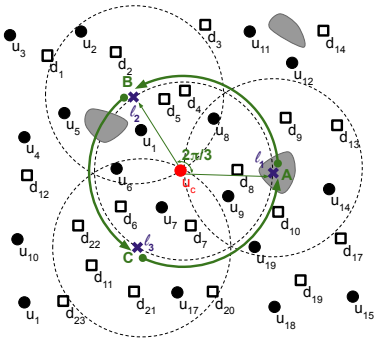

Figure 4.6: Considering Infeasible regions

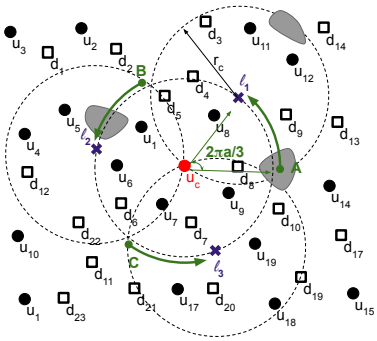

Figure 4.7: Readjusting location in fSRQ

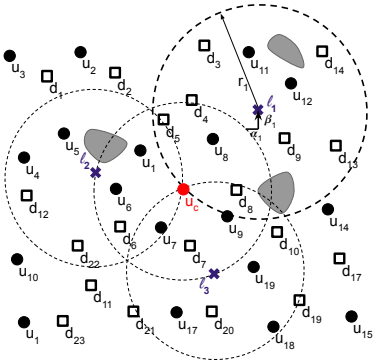

Figure 4.8: Readjusting query range

region. The new fragmented locations can be computed as follows:

$$l_n = \left(x_c + r_c \cos\left((a+n-1)\frac{2\pi}{j}\right), y_c + r_c \sin\left((a+n-1)\frac{2\pi}{j}\right)\right)$$

Here, a is a random number ranging from 0 to 1. As shown in Figure 4.6, the fragmented location l_1 can move in an arc from A to B while a varies from 0 to 1. Similarly, l_2 and l_3 can move from B to C and C to A, respectively. Figure 4.7 depicts the new fragmented query locations. Note that this will change the results of all the fSRQs.

Methodology for location randomization: Note that the fragmented locations computed so far are always on the circumference, as depicted in Figure 4.7. Hence, we add a randomization phase to randomize the fragmented locations further, as depicted in Figure 4.8. Due to the randomization phase, the fragmented location is moved horizontally by α units and moved vertically by β units. Computation of fragmented location with randomization phase is given below.

$$l_n = \left(x_c + \alpha_n + r_c \cos\left((a+n-1)\frac{2\pi}{j}\right), y_c + \beta_n + r_c \sin\left((a+n-1)\frac{2\pi}{j}\right)\right) \tag{4.1}$$

Here, $\alpha, \beta \in (0, \delta)$, where δ is the location randomization constant defined by the user. Since the fragmented location is modified, the query radius in the fSRQs will also change (see Algorithm 2 Lines 6-9). The new query range is calculated as follows:

$$r_n = r_c + \sqrt{\alpha_n^2 + \beta_n^2} \tag{4.2}$$

Because of location randomization, the actual location of user u_c is protected, even in the case of $k = j$ (i.e., when the number of queries received by LBS adversary is equal to j fSWQ). To further improve privacy increasing the value of δ would be beneficial, as it would further randomize the locations of the fSWQs.

Methodology for intent randomization: Recall that it is also important to hide the user's query intent information along with the location information. For this, we assume the data set of dummy intents I_s. The set I_s can have multiple intents such as *the names of all the hospitals*, *the names of all hotels*, *all the parks* and so on. We modify each fSRQ by picking multiple dummy intents within the range of the query to the user's query intent information. The dummy intents are picked randomly from I_s) (Line 10, Algorithm 2). The modified intent in an fSRQ is $\overline{q_n}$, where $\overline{q_n}$ is a set of multiple intents i.e., $\overline{q_j}$ = $\{q_c$, q_1, q_2, $\cdots$, $q_{i-1}\}$). Here, i is the number of intents added in an fSRQ (Refer Line 11-12, Algorithm 2).

Adding dummy intents in all the fSRQs will not increase the number of messages exchanged between peers. However, the size of the message might increase. This is because the result of the fSRQ with all the intents, can be transmitted at a time in a single message. Note that this phase will increase computation costs for the location-based service (LBS) provider, as LBS will now have to respond to fSRQs with multiple intents.

Finally, the new fSRQ after considering the presence of infeasible regions, location randomization, and intent randomization is $\langle l_1, r_1, \overline{q_1}\rangle$, $\langle l_2, r_2, \overline{q_2}\rangle$, $\ldots \langle l_j, r_j, \overline{q_j}\rangle$.

Algorithm 3: NEED_PEERS_Originator(h, j)

Input : h: hop distance; j: number of brokers required;
Output : T: the set of peers that sent acknowledgement;

```
1  T ← {∅};
2  number of peers found j' = |T|;
3  while j' < j do
4      Broadcast NEED_PEERS with ⟨u_c, h, mID, pK_c⟩;
5      T' ← set of peers that responded back;
6      if j' < j-1 then
7          if T = T' then
8              Suspend the request;
9          h ← h + 1;
10         T ← T';
11 return T;
```

4.3.2 Peer Searching phase

After computing fSRQs, u_c searches for the peers in its surroundings spatial environment using a multi-hop routing protocol (refer Algorithm 3. The user u_c acts as the *request originator* and request its surrounding peers to act as brokers. A broker is a peer acting as a mediator between u_c and LBS. For searching the set of brokers, the *request originator* u_c broadcasts the NEED_PEERS request message to its neighboring peers along with a tuple ⟨request originator ID (u_c), hop distance (h), message sequence ID (mID), public key of the originator $(pK_c)\rangle$. Here, u_c iteratively increases the number of hops (h) from 0 until the required number of brokers is found (see Algorithm 3 Lines 1-4). Among all the

responded peers, u_c selects j number of peers at random and such j peers, which have responsed, are referred to as brokers.

Having received the NEED_PEERS request from *request originator*, the surrounding peers take the following steps (Algorithm 4). A mobile peer u_p responds to the NEED_PEERS request from either the request originator u_c or the peer forwarding the request u_r. First, u_p checks if it is a duplicate request based on the message sequence ID. If it is a duplicate request, it simply replies with an ACK message without processing the request (Lines 1-2). Otherwise, u_p processes the request based on the value of h:

Algorithm 4: NEED_PEERS_Receiver(h)

```
   Input  : h: hop distance;
   Output : send tuple T_p to u_r;
1  if request is duplicate then
2      Reply u_r with an ACK message;
3  if h=1 then
4      Send the tuple ⟨u_p, h, mID, pK_p⟩ to u_r
5  else
6      h ← h-1;
7      Broadcast NEED_PEERS with ⟨u_p, h, mID, pK_p⟩;
8      T_p ← set of peers that responded to the above request;
9      for T_i in T_p do
10         T_i.h ← T_i.h + 1
11     T_p ← T_p ∪ ⟨u_p, h, mID, pK_p⟩;
12     Send T_p back to u_r
```

Case I: When h = 1, u_p returns a tuple ⟨request receiver ID (u_p), hop distance (h) (which is set to one in this case), message sequence ID (mID), public key of the receiver $(pK_p)\rangle$ to u_r (see Lines 3-4).

Case II: When $h > 1$, u_p decrements h and broadcasts the NEED_PEERS request with a tuple ⟨request receiver ID (u_p), $(h - 1)$, mID, public key of the receiver $(pK_p)\rangle$. The peer u_p keeps listening to the network until it collects the replies from its neighboring peers. Next, u_p increments the h of each collected tuple, and then it appends its own tuple to the collected tuples T_p. Finally, it sends T_p back to u_r (refer, Lines 6-12, Algorithm 4).

When u_c collects the tuples (T) from its neighboring peers, if u_c cannot find j number of peers within a hop distance of h, it increments h by 1. It re-broadcasts the NEED_PEERS request along with a new message sequence ID and new h. The u_c repeatedly increments h till it finds j number of peers (see Algorithm 3 Lines 3-10). If there are not enough connected peers for u_c or u_c finds the same set of peers in two consecutive broadcasts, i.e., with hop distances h and h + 1, u_c has to relax its privacy profile, i.e., decreases the value of j by 1, or be suspended for a while (refer Lines 7-8, Algorithm 3). Since, in our approach, higher anonymity is achieved using lesser peers, the number of hops required to find peers should be small.

Algorithm 5: DATA_TRANSFER_Originator(T, fSRQ)

Input : T: list of responses from peers by executing Algorithm 4; R: fSRQ by executing Algorithm 2;
Output : $LBS\_results$: results of all the fSRQ;

1 $LBS\_results \leftarrow \{\emptyset\}$;
2 $T \leftarrow$ responses from peers;
3 $R \leftarrow$ set of j fSRQs;
4 $T^{'} \leftarrow$ select j responses randomly from T;
5 **for** u_a *in* $T^{'}$ **do**
6 $\quad R^{'} \leftarrow$ select different fSRQ from R;
7 $\quad E_{u_a}(R^{'}) \leftarrow$ encrypt the SRQ with public key of broker u_a;
8 $\quad$ Broadcast DATA_TRANSFER with tuple $\langle u_c, u_a, h_{u_a}, E_{u_a}(R^{'}), mID\rangle$
9 **repeat**
10 $\quad R^{'} \leftarrow R$ whose results $\notin LBS\_results$;
11 $\quad$ **foreach** *r in* $R^{'}$ **do**
12 $\quad\quad$ **if** $|T| - |T^{'}| < |R^{'}|$ **then**
13 $\quad\quad\quad$ Broadcast NEED_PEERS
14 $\quad\quad R^{'} \leftarrow$ select missing fSRQ from R;
15 $\quad\quad E_{u_a}(R^{'}) \leftarrow$ encrypt the SRQ with public key of broker u_a;
16 $\quad\quad$ Broadcast DATA_TRANSFER with tuple $\langle u_c, u_a, h_{u_a}, E_{u_a}(R^{'}), mID\rangle$
17 **until** *$LBS\_results$ does not have all the results*;
18 $LBS\_results \leftarrow$ response from brokers;

4.3.3 Data Transmission phase

Having received acknowledgment from peers interested in helping u_c hide its query information. Now, u_c selects j brokers from the acknowledged peers and sends fSRQs to these brokers in this phase. Here, u_c acts as the request originator and sets results of fSRQs (LBS_results) to $\{\emptyset\}$.

Let T be the list of peers who responded to the NEED_PEERS request broadcast and R be the list of fSRQ. User u_c selects j mobile peers from T randomly. These j peers ($u_{a_1}, u_{a_2}, \ldots u_{a_j} \in T$) are termed as the *brokers* (refer Lines 1-4, Algorithm 5). User u_c now broadcasts j messages m_1, m_2, $\ldots m_j$ each containing a distinct fSRQ (refer Lines 5-8, Algorithm 5). Each message contains a tuple $\langle$request originator ID (u_c), request receiver (broker) ID (u_a), hop distance (h_{u_a}), fSRQ encrypted with public key of the receiver $E_{u_a}(R)$, message sequence ID (mID) $\rangle$. The n^{th} message is of the form:

$$m_n = \langle u_c, u_{a_n}, h_{u_{a_n}}, E_{u_{a_n}}(\langle l_n, r_n, \overline{q_n}\rangle), mID\rangle$$

Here, $1 \leq n \leq j$ and corresponding to messages in $\{m_1, m_2, \ldots m_j\}$. Although these messages are broadcast to everyone, only the corresponding brokers can decrypt them. This is because SRQ's in the messages are encrypted using the public key of the corresponding broker. A broker u_a responds to a message m = $\langle u_c, u_a, h_{u_a}, E_{u_a}(\langle l, r, \overline{q}\rangle), mID\rangle$ send from the user u_c in the following manner:

1. First, u_a checks if the message m is sent for itself using the receiver ID. If the message is not for u_a (i,e. receiver ID does not match), then the message is ignored.

2. If the message is for u_a (i,e. receiver ID matches) then u_a decrypts the SRQ $D_{u_a}(E_{u_a}(\langle l, r, \overline{q}\rangle))$ and gets the SRQ $\langle l, r, \overline{q}\rangle)$. Here, D_{u_a} is the private key of the broker u_a. The decrypted SRQ is sent to the LBS provider.

3. Once u_a gets the response from the LBS provider $R_{\langle l,r,\overline{q}\rangle}$, it sends the response back to the user u_c by transmitting the message as a tuple $\langle u_a, u_c, E_{u_c}(R_{\langle l,r,\overline{q}\rangle}), mID\rangle$. Here, u_a is the message sender, u_c is the message recipient, $E_{u_c}(R_{\langle l,r,\overline{q}\rangle})$ is the result of the SRQ $R_{\langle l,r,\overline{q}\rangle}$ from LBS provider when encrypted using the public key of message recipient and mID is the message ID.

Finally, u_c gathers all the responses from brokers in LBS_results (see Algorithm 5 Line 18).

Counter failures: Sometimes user u_c might not receive a response from a few brokers due to them moving out of the range of the user, or there might be some error from the LBS provider. In such cases, more brokers are selected from the set T to counter such failure. If T does not have enough brokers, we run a multi-hop routing protocol to pick new brokers. Next, we send SRQ to the new brokers whose results were not found in LBS_results. This is done till we receive results for all the j fSRQs (refer Lines 9-17, Algorithm 5).

Algorithm 6: Collect($LBS\_results$)

Input : $LBS\_results$: result for fSRQs;
Output : $R_{\langle l_c,r_c,q_c\rangle}$: result of u_c's actual query

```
1 R_⟨l_c,r_c,q_c⟩ ← {∅};
2 results ← select query from LBS_results with intent equal to q_c;
3 for result in results do
4     for obj in result do
5         if dist(obj, l_c) ≤ r_c then
6             if obj ∉ R_⟨l_c,r_c,q_c⟩ then
7                 R_⟨l_c,r_c,q_c⟩ ← R_⟨l_c,r_c,q_c⟩ ∪ obj
```

4.3.4 Data Deconstruction phase

In this phase, mobile user u_c finally calculates the result of the SRQ $\langle l_c, r_c, q_c\rangle$ using the results collected from all the brokers. User u_c selects all the SRQ results from LBS_results (refer Algorithm 5). Results with the query intent equal to q_c are pruned out from LBS_results (refer Line 2, Algorithm 6). Next, user, u_c computes the final result $R_{\langle l_c,r_c,q_c\rangle}$ from the LBS_resutls (refer Line 3-7).

Table 4.1: Parameters of our performance study

S.N.	Parameter	Default Values	Variations
1	user density (uD)	250 users/sq.km	10, 20, $\cdots$, 500
2	mobile user's minimum velocity (V_{min})	0 km/hr	–
3	mobile user's maximum velocity (V_{max})	90 km/hr	–
4	Transmission Range (TR)	250 m	–
5	probability of adversarial peers (pA)	0	0, 0.1, $\cdots$, 1
6	Infeasible Region Ratio (IRR)	0.3	–
7	location randomization constant (δ)	0.5 km	–
8	Total queries received by the LBS provider(k)	10	2, 3, $\cdots$, 30
9	Number of fragments required (j)	3	1, 2, $\cdots$, 10
10	Number of intents added (i)	3	1, 2, $\cdots$, 10

4.4 Theoretical Analysis

We conduct theoretical analysis to show that the user query location is more secure by using a privacy protection scheme that adopts ijk-anonymity than a‘ scheme that uses k-anonymity. Let P_k and P_{ijk} be the probability of finding a data point if it followed k-anonymity and ijk-anonymity, respectively.

Lemma 4.4.1. *In dataset $\mathcal{K}$, where $\mathcal{J} \subset \mathcal{K}$, $k = |\mathcal{K}|$, $j = |\mathcal{J}|$ and $i = |\mathcal{I}|$ then, always $P_k \geq P_{ijk}$.*

Proof. Defining ijk-anonymity, we have $P_{ijk} = \frac{1}{i \cdot \binom{k}{j}}$, which can be split as $\frac{1}{ik}\frac{j}{k-1}\frac{j-1}{k-2}\cdots\frac{2}{k-(j-1)}$. Given $\mathcal{J} \subset \mathcal{K}$, we find $j+1 \leq k$, which implies $\frac{j}{k-1} \leq 1$, and by similar logic, $\frac{j-1}{k-2} \leq 1$; $\frac{j-2}{k-3} \leq 1$; $\ldots$ $\frac{2}{k-(j-1)} \leq 1$. Substituting these inequalities, we get $P_{ijk} = \frac{1}{ik}e_1 \times e_2 \times e_3 \ldots e_{j-1}$, where $e_1, e_2, e_3 \ldots e_{j-1} \leq 1$. Let $e = e_1 \times e_2 \times e_3 \ldots e_{j-1}$; we find $e \leq 1$ and $i \geq 1$. Thus, $P_{ijk} = \frac{1}{ik} \times e \implies P_{ijk} \leq \frac{1}{k}$. Knowing that $P_k = \frac{1}{k}$, we finally obtain $P_{ijk} \leq P_k$. □

Thus, the probability of finding a data point that follows ijk-anonymity is always less than or equal to the probability of finding a data point that follows k-anonymity.

4.5 Performance Evaluation

This section reports our performance evaluation. We implement the simulation system in Python 3.0 and conduct experiments on a computer with a fifth-generation Intel Core-i5, 2.7GHz frequency, and 8 GB of main memory. We create a discrete event simulation with event-based modeling [8] to model the simulation. This simulation generates multiple events, and these events are stacked in the event queue. Events with the least execution time are placed on top of the event queue. A central simulator object executes the events from the event queue.

The performance study parameters used in our experiments are summarized in Table 4.1. In our simulation, the size of the layout is 200 km $\times$ 200 km square. The average mobile user density in the layout uD is 250 users/square.km. Initially, all the mobile users are distributed uniformly at the start of the simulation. Each mobile user has a velocity ranging from 0 to 90km/s. Mobile users consider an individual random walk model based on the *random way-point* model [5, 21]. In a random way-point, each mobile client randomly chooses its destination in the space with a randomly determined speed ranging from V_{min} to V_{max}. When the mobile client reaches the destination, it comes to a standstill to determine its next destination and repeats the motion.

In our simulation, each mobile client can issue a query to the LBS provider and the rate of issuing the query is set to one per ten minutes. Every mobile user who requests an SRQ will log all the details of its query, like the number of peers in the surroundings and the total response time for the request. All the experiments show the average result of all the clients who request an SRQ from the LBS provider. The transition range of each mobile client TR is set to 250 meters. The peers in the network can act as malicious entities with a probability of pA. Depending on the infeasible region ratio (IRR), infeasible regions were randomly arranged in the layout. The location randomization constant δ is set to 0.5 km. The value of k defines the number of queries received by any adversary (here, the adversary can be an LBS provider or peers or someone with access to LBS server data) while processing the fragmented queries. The value of k in an individual simulation is the same throughout the simulation. The user's SRQ query is fragmented into j queries. The number of intents sent in an fSRQ is set to i. The default value of i is set to one to study the impact of location anonymization in isolation.

Now, we explain performance metrics used in our experiments and how we mock them in our simulation.

Degree of Anonymity (A): This metric calculates the degree of anonymity, equal to the probability of an adversary finding the user. A lower value indicates better protection from an adversary.

Number of Hops Required (H): This metric calculates the number of hops required to find the desired number of peers from the mobile network to cloak the user's query information. A lower value of H indicates less time for a user to receive the query result.

Communication Cost for total Hops (CH): This metric calculates the total requests generated by all mobile peers to make hops to gather the desired number of peers for the request originator. A lower value of CH indicates fewer resources are required to receive the query result.

Anonymity against compromised LBS (AL): This metric calculates the degree of anonymity when requesting a query from a compromised LBS provider. When LBS is compromised, all the client's requests are accessible to the adversary (it can be some external adversary who has access to the LBS server or the LBS provider itself). In this case, the adversary will know which query maps to which client. Hence, if the query has the client's actual location, then the client's location will be compromised. To simulate this scenario, we log all the queries received by the LBS provider with the details of the query originator. Then an adversary is assumed to have access to this data to find the user query location in the logs.

Anonymity against compromised peers (AP): The peers in the surrounding can be malicious and expose the user query location. Nevertheless, our approach sends only the fragmented SRQ to the peers; hence a single peer will never know the user's exact location. Moreover, peers cannot access the information sent to another peer since it is encrypted with the desired peer's public key. Hence, the only way for a peer to get the user query location information is to collude with other brokers in the network. Hence metric AP calculates the degree of anonymity against malicious peers who can collude with each other. We log all the fSRQs sent to compromised peers to simulate this scenario. The user's anonymity AP is calculated by assuming that an adversary has access to this data.

Peers Colluding Cost (PC): This metric describes the total communication cost required to compromise the user query location via colluding with other compromised peers in the network.

Anonymity against Center of CR attack (AC): This metric describes the user's anonymity against the centre of CR attack. In this attack, the adversary knows the location of all the users in the cloaking region. Here, the adversary assumes that the user close to the centre of the CR is the request originator. Usually, it is true in previous approaches since users propagate requests equally in all directions. Hence there is a higher probability that the request originator will be the user at the centre of the CR. To simulate this attack, we assume that the adversary knows the location of all the users in the cloaking region. Hence, the adversary considers that the user at the centre of CR is the required originator.

We compare our approach with prive [15], MobiHide [14], and mainly cloakP2P [11]. We choose cloakP2P as the baseline for performance comparison because cloakP2P peer gathering and data transfer methods are close to that of our approach. Moreover, cloakP2P does not use other p2p network formations (like Kademlia, Chord). For comparison, many schemes, such as [14, 15, 31] also choose cloakP2P. We compare the performance of *ijk*Cloak with cloakP2P by varying the user density in the layout uD and the number of queries (k) an adversary receives. Also, we evaluate the practicality and effectiveness of our proposed scheme. The adversary can either be the LBS provider or the peers in the surrounding. But since the peers in the surroundings only receive a fragment of the actual user's query in the form of fSWQ, the peers can not calculate the user's actual query location. On the other hand, the LBS provider will receive all the j fSWQ along with queries from other mobile users. Therefore, in our performance study, we consider $k(k \geq j)$ as the number of queries received by the LBS provider while processing j number of fSWQ. We adapt these reference approaches with essentially the same setup as ours in the interests of meaningful comparison. We compare the performance of our approach with the previous approaches against different attacks and variations in k, j, i and uD.

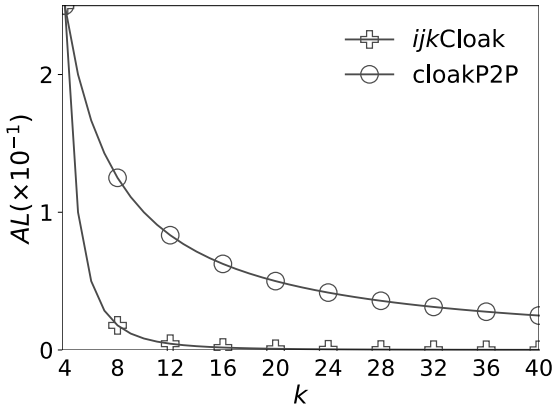

Figure 4.9: k Vs AL

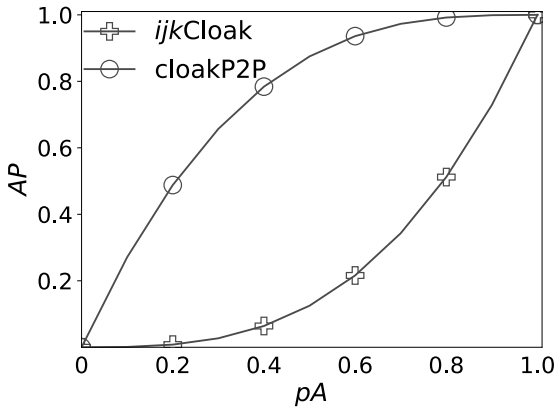

Figure 4.10: pA Vs AP

4.5.1 Effect on Anonymity when LBS is Compromised (AL)

In Figure 4.9 the x-axis represents the number of queries received by the LBS provider k. On the other hand, the y-axis represents AL. We can observe that in both approaches, AL decreases with an increase in k. This is because as k increases, it becomes demanding for an adversary to guess the correct query originator. Moreover, we can observe from the results in Figure 4.9 that AL is always less in *ijk*Cloak than cloakP2P. This is because, in *ijk*Cloak, the adversary must find all the brokers carrying fSRQ's from k peers, which is a more challenging task. Moreover, in *ijk*Cloak, intent randomization is run to hide the user's query intent.

4.5.2 Effect on Anonymity when Peers are Compromised (AP)

In Figure 4.10 the x-axis represents pA. Whereas the y-axis represents AP. In Figure 4.10, as pA increases the value of AP also increase. This is because as more and more peers act as adversaries, it will become harder to preserve user query location privacy. Moreover, from the results in Figure 4.10 it is clear that *ijk*Cloak always performs better than cloakP2P. This is because to compromise the user query location in *ijk*Cloak, all the brokers must be adversaries (hence they can collude to get the actual query location of the user). On the other hand, in cloakP2P, if at least one of the peers helping the user is compromised, then the user query location will be compromised. This is because, in cloakP2P, peers are considered trustworthy, and the actual location is sent to the peers.

In Figure 4.11 the x-axis represents pA. Whereas the y-axis represents the total communication cost that the adversarial peers would require to compromise the user query location PC. In Figure 4.11 the amount of communication PC increases drastically as pA increases. This is because it is not sufficient for adversarial peers to collude only with peers in the user's surroundings. Since the peers do not know the user's surrounding information, they should collude with all the adversarial peers in the layout. As

we can observe from the results, the communication cost for maintaining a network to collude with other adversarial peers is huge. Hence, we can assume it is practically far-fetched.

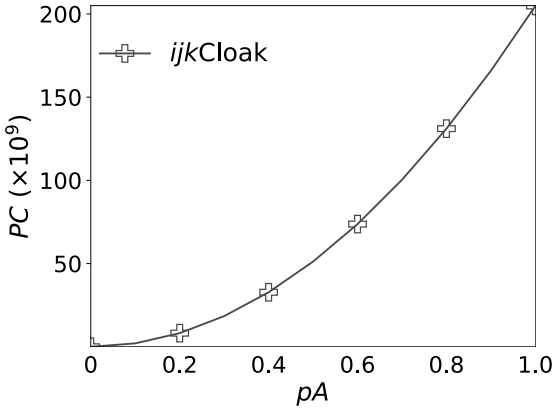

Figure 4.11: pA Vs PC

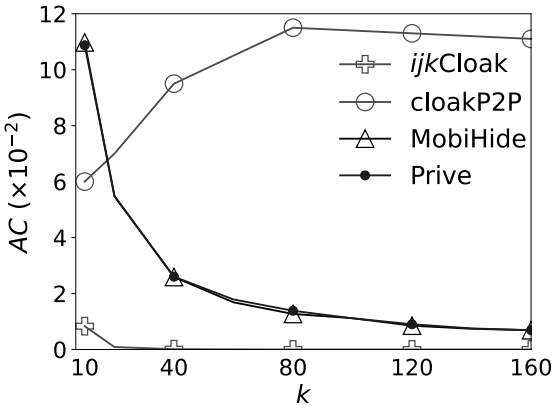

Figure 4.12: k Vs AC

4.5.3 Effect on Anonymity with Center of CR Attack (AC)

In Figure 4.12 the x-axis represents the number of queries received by the LBS provider k and the y-axis represents AC. We can observe from the results in cloakP2P for k = 40, the probability of the query originator being at the centre of CR is 0.1. In contrast, the maximum bound for predicting the location of the query originator using k-anonymity should be 1/40 = 0.025 ($1/k$). Hence, a user query location is 25 times more likely to be compromised than the k-anonymity maximum bound. This is because, in cloakP2P, users are likely to come uniformly from all directions. Hence, the adversary can guess the query originator's location more easily. Therefore cloakP2P has poor performance compared to the other approaches.

On the other hand, approaches like MobiHide and Privé achieve the required k-anonymity degree of $1/k$ at all times. These approaches use data structures like Hilbert's curve to map the layout and create their cloaking region. In the case of *ijk*Cloak, directly finding a cluster of peers (brokers) used to cloak the query result itself is not possible. Since, in *ijk*Cloak, all the locations are not aggregated by one user and sent to the LBS provider. Instead, brokers individually send the request to the LBS provider. Thus, the adversary can not know all the brokers helping the user get the SRQ result. Even if the adversary wants to guess the brokers from the k peers, it will be bound by the same bound as that of ijk-anonymity (i,e. $\frac{1}{iC_k^j}$) as can be seen from the results in Figure 4.12.

4.5.4 Effect of Variation in k Number of Queries Revived by LBS

The results of our experiments with variation in the number queries received by the LBS provider while responding to fragmented queries k are depicted in Figure 4.13. In Figure 4.13, the x-axis represents k. The y-axis represents the anonymity of the users. It can be observed from the results in

Figure 4.13 that cloakP2P, MobiHide and Privé have their anonymity close to that of the baseline value of the k-anonymity (i.e., $1/k$). On the other hand, *ijk*Cloak has anonymity much less than all other approaches. This is because, in *ijk*Cloak, ijk-anonymity is used. For example, to achieve location anonymity of 0.025, cloakP2P would need 40 peers. In contrast, *ijk*Cloak would need eight peers to be in its surroundings to achieve similar anonymity. This is because to find the user query location in *ijk*Cloak adversary has to guess j brokers from all the peers it has received.

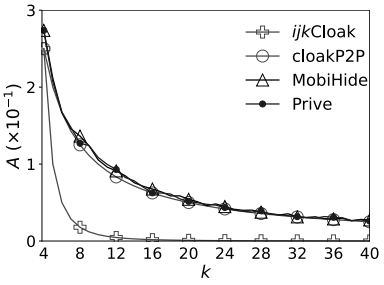

Figure 4.13: k Vs A

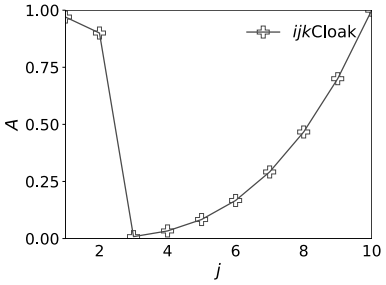

Figure 4.14: j Vs A

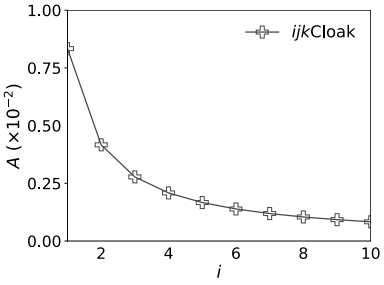

Figure 4.15: i Vs A

4.5.5 Effect of Variation in j Fragments Generated

Figure 4.14 depicts the effect of variation in j. In Figure 4.14, the x-axis represents the number of fSRQ created and the number of brokers required (j). The y-axis represents the user's anonymity (probability of being found). Previous approaches are not compared here since the terms j, and i were proposed in our approach. In the case of $j = 1$ and $j = 2$, A is close to one. This is because when the number of brokers is one or two, they cannot cover all the desired objects of the SRQ unless these brokers are in the same location as the query originator. For other values, as j increases, A also increases. This is because in a circle, if any three points on its circumference are found, then the centre of the circle can be calculated. Hence as j increases, it becomes easier for an adversary to pick three brokers from k peers in its surrounding. Hence an ideal value of j in our approach should be three.

4.5.6 Effect of Variation in i Intents Generated

Figure 4.15 depicts the effect of variation in i. In Figure 4.15, the x-axis represents the number of intents sent to the LBS provider (both actual and dummy intents). The y-axis represents the anonymity of the user. It can be observed from the results in Figure 4.15 that A decreases as i increases. This is because an adversary must further guess the user's query intent from i intents (i-1 dummy and one actual intent) present in a query.

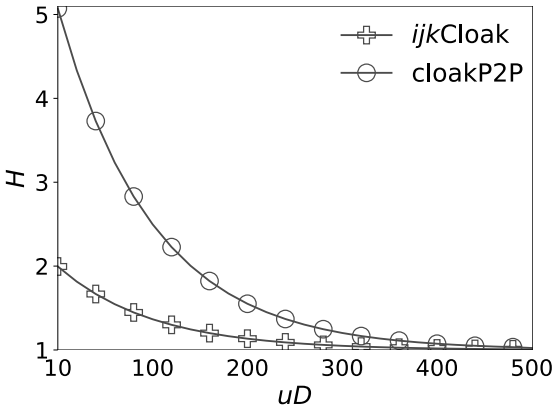

Figure 4.16: uD Vs H

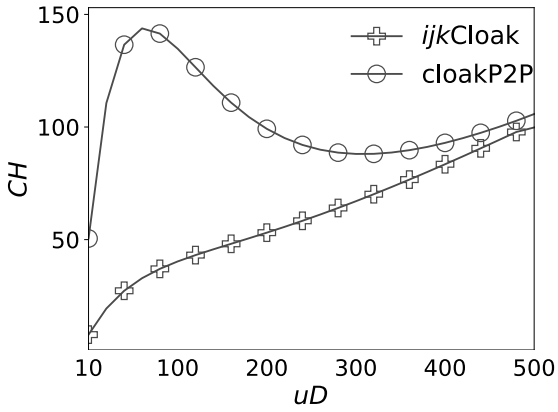

Figure 4.17: uD Vs CH

4.5.7 Effect of Variation in User Density (uD)

Since the privacy preservation scheme should function in regions with different user densities, we evaluate *ijk*Cloak with varying user densities. Figure 4.16, 4.17 shows the results of our experiments with variation in mobile users density in the layout uD.

In Figure 4.16, the x-axis represents uD, while the y-axis represents the number of hops required to reach the required number of peers H. It can be observed in Figure 4.16 that the H is maximum at low density. This is because in regions with less uD, the approach has to make multiple hops to reach the required number of peers to preserve the user query location privacy. Moreover, H of *ijk*Cloak is less than cloakP2P. This is because, in *ijk*Cloak, it is enough to find j brokers from the surrounding (since $j = 3$ in our cases). Whereas in cloakP2P, since it user k-anonymity, the approach has to find k users in the surround. Since k is greater than j, it required more hops to reach the required no of peers in cloakP2P.

In Figure 4.17, the x-axis represents uD. while the y-axis represents the communication cost by the system due to hops CH. It can be observed in Figure 4.17 that CH of *ijk*Cloak is always less compared to cloakP2P. This is because in *ijk*Cloak required number of peers can be achieved with fewer hops. In contrast, in the case of cloakP2P, it takes more hops to find the required number of peers to preserve the user query location privacy. In the case of cloakP2P, CH is high when the uD is less. Since it requires more hops to find k peers, it needs to make more hops, increasing the system's communication cost. However, as uD increases, fewer hops are required to find k users; hence communication cost decreases. After some point, the communication cost increases linearly as it would take at least one hop to get k users. However, since the density is high, more users will be requested to join the network than required. On the other hand, in the case of *ijk*Cloak, the communication cost also increases when uD is low, but the increment is minute. Furthermore, as density increases, the CH increases linearly but still is less than the value of cloakP2P. This is because *ijk*Cloak requires fewer hops to gather brokers.

4.6 Summary

In this chapter, to preserve the location and intent privacy for location-based spatial range queries, we have proposed the notion of ijk-anonymity. Our proposed approach, designated as *ijk*Cloak, adopts ijk-anonymity in a mobile network to preserve the user query location privacy using fewer peers. Hence *ijk*Cloak is more feasible for users in highly mobile and dynamic networks. In *ijk*Cloak, the user's exact location is sent to neither the LBS provider nor the peers. Moreover, in *ijk*Cloak, the intent of the user's query is also preserved. Furthermore, we conduct theoretical analysis, performance evaluation and experiments on resistance to attacks to demonstrate that *ijk*Cloak effectively provides improved location privacy. In the next chapter, we present summary and conclusion.

Chapter 5

Summary and Conclusions

In this chapter, we present summary, conclusions and future work.

5.1 Summary

Location-based services (LBSs) enable mobile users to obtain location-specific information by requesting LBS providers to retrieve the desired data. However, preserving privacy is a major issue due to the possibility of leakage of user location to adversaries. Several cloaking-based privacy preservation approaches have been proposed in the literature to preserve users' location privacy. In this thesis, we have improved the existing dummy generation and spatial cloaking-based distributed approaches. Existing dummy generation approaches fall short in providing adequate security, especially if an adversary discerns the centre of the cloaking region. These approaches also overlook the aspect of *time-dependent infeasible regions*. Furthermore, existing spatial cloaking-based distributed approaches do not protect users' intent privacy and demand substantial network peers for obfuscating a user's location. Acknowledging these limitations, this thesis has proposed improved approaches for protecting user location and intent information.

The first of these approaches, the Annulus-based Gaussian Dummy Generation (AGDG), is built upon existing dummy generation approaches. The AGDG approach considered user query probability and the distribution of infeasible regions in the surrounding area. Unlike previous approaches, AGDG used virtual cloaking regions with a specific distribution to obscure the true user's location. Consequently, AGDG rendered user location information more resilient to attacks, even when an attacker knew the centre of the cloaking region. Additionally, we incorporated the concept of time-dependent infeasible regions to preserve user location privacy further through a multi-layered structure.

The conducted experiments demonstrate the effectiveness of the proposed AGDG approach in enhancing user privacy. By varying the number of candidates, the entropy (H) increases, improving the resistance to potential adversaries. Moreover, AGDG outperforms other methods when considering the ratio of infeasible regions and the presence of time-dependent infeasible regions, maintaining a high system entropy and effective cloaking region. As such, AGDG exhibits resilience and superior perfor-

mance in diverse conditions, affirming its proficiency in providing robust location privacy protection for users. Therefore, AGDG can be utilized in mobile devices where it can intercept any location-based query sent from the device and dispatch multiple dummy queries, thereby safeguarding the user's actual location information.

The second method, *ijk*Cloak, built upon and improved existing spatial cloaking-based distributed approaches. We introduced ijk-anonymity within this method to preserve both location and intent privacy for spatial range queries. In the ijk-anonymity technique, a user's SRQ was fragmented into j SRQ queries to conceal the actual location in the user's query. Furthermore, $i-1$ dummy intents were added to the fSRQs to hide the true query intent. The *ijk*Cloak approach adopted ijk-anonymity in a mobile network to protect users' location privacy using fewer peers, thereby increasing its feasibility for users in highly mobile and dynamic networks. In *ijk*Cloak, the user's information was preserved against both the LBS provider and the peers.

The effectiveness of *ijk*Cloak in preserving user location privacy was demonstrated through theoretical analysis, performance evaluations, and resistance to attack experiments. This study shows that *ijk*Cloak is very good at protecting user privacy, even when location services or peers are compromised. It works better with fewer peers as k changes, keeps the best j value at three, and improves privacy as i increases. Also, *ijk*Cloak works well with different numbers of users, with fewer steps and lower communication costs, showing it can protect privacy in many different situations.

5.2 Conclusion

The conclusions are as follows:

1. Based on the experimental results, we conclude that AGDG effectively enhances user location privacy. It protects location information even if an adversary knows the centre of CR. Moreover, it generates more realistic dummies and considers the presence of time-dependent infeasible regions.

2. We conclude that *ijk*Cloak effectively preserves intent and location privacy even when LBS and peers are compromised. Importantly, it achieves this privacy preservation with fewer peers compared to previous approaches. Thus, making it more secure and practical for highly dynamic mobile networks.

5.3 Future Work

Potential directions for future research include:

1. The proposed AGDG approach effectively preserves the user's location privacy. However, intent privacy also forms an integral aspect of user privacy. Therefore, future research could explore

incorporating intent privacy measures into the AGDG methodology to provide a more comprehensive privacy solution.

2. In the current implementation of *ijk*Cloak, brokers are chosen randomly from surrounding peers. Future work could consider developing a broker selection algorithm that considers factors such as the broker's location, availability, and reliability. Such an enhancement could significantly optimize system operations and improve overall performance.

3. An exciting avenue of future research is the development of practical use cases where our approach can be used to maintain privacy in wireless network-based applications, such as those employed in military operations, security establishments, and disaster response systems.

4. A promising avenue for future work lies in applying the *ijk*Cloak framework within commercial mobile networks. With this approach, mobile companies could create user-centric privacy solutions. This process would involve partnering with Location-Based Service providers, peers, and mobile network operators, potentially enhancing privacy across a range of services.

Related Publications

1. S. Siddiqie, A. Mondal, and P.K. Reddy. "An improved dummy generation approach for enhancing user location privacy." Database Systems for Advanced Applications (DASFAA) (11–14) (2021).

2. S. Siddiqie, A. Mondal, and P.K. Reddy. "An improved dummy generation approach for infeasible regions." Applied Intelligence Journal, Springer, (1-15) (2023).

3. S. Siddiqie, R. Srinivas, and P.K. Reddy. "Location and Intent Privacy Preservation for Spatial Range Queries in a Mobile Network." IEEE Transactions on Information Forensics and Security (2023). [Under Review].

Other Publications (Not related to the thesis)

1. S. Siddiqie, A. Ralla, P.K. Reddy and A. Mondal,"Sensor-based Framework for Improved Air Conditioning under Diverse Temperature Layout", International Conference on Distributed Computing and Networking (ICDCN) (1–6)-(2020).

2. A. Ralla, S. Siddiqie , P. K. Reddy and A. Mondal, "Coverage Pattern Mining Based on MapReduce," ACM India Joint International Conference on Data Science & Management of Data (CODS-COMAD) (209-213)-(2020).

Bibliography

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *ACM SIGSAC Conference on Computer & Communications Security*, pages 901–914, 2013.

[2] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. Di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *IFIP Working Conference on Data and Applications Security*, pages 47–60, 2007.

[3] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati. An obfuscation-based approach for protecting location privacy. *Dependable and Secure Computing*, 8(1):13–27, 2009.

[4] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive computing*, 2(1):46–55, 2003.

[5] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa. Stochastic properties of the random waypoint mobility model. *Wireless networks*, 10(5):555–567, 2004.

[6] S. Boccaletti, G. Bianconi, R. Criado, C. I. Del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[7] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *International Conference on Computer and Communications Security*, pages 251–262, 2014.

[8] P. Bratley, B. L. Fox, and L. E. Schrage. *A guide to simulation*. Springer Science & Business Media, 2011.

[9] T.-Y. Cai, D.-H. Song, J.-H. Youn, W.-G. Lee, Y.-K. Kim, and K.-J. Park. Efficient dummy generation for protecting location privacy. *The Journal of Korea Institute of Information, Electronics, and Communication Technology*, 9(6):526–533, 2016.

[10] C.-Y. Chow, H. V. Leong, and A. T. Chan. Distributed group-based cooperative caching in a mobile broadcast environment. In *International Conference on Mobile Data Management*, pages 97–106, 2005.

[11] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *International Symposium on Advances in Geographic Information Systems*, pages 171–178, 2006.

[12] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *International Conference on Pervasive Computing*, pages 152–170, 2005.

[13] G. Ghinita. *Privacy for Location-based Services*, volume 3. Morgan & Claypool Publishers, 2013.

[14] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In *International Symposium on Spatial and Temporal Databases*, pages 221–238, 2007.

[15] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Prive: anonymous location-based queries in distributed mobile systems. In *International Conference on World Wide Web*, pages 371–380, 2007.

[16] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *International Conference on Mobile Systems, Applications and Services*, pages 31–42, 2003.

[17] M. Gruteser and B. Hoh. On the anonymity of periodic location samples. In *International Conference on Security in Pervasive Computing*, pages 179–192, 2005.

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

[19] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, 2005.

[20] H. Hu and J. Xu. Non-exposure location anonymity. In *International Conference on Data Engineering*, pages 1120–1131, 2009.

[21] E. Hyytiä and J. Virtamo. Random waypoint mobility model in cellular networks. *Wireless Networks*, 13(2):177–188, 2007.

[22] Y. Ji, R. Gui, X. Gui, D. Liao, and X. Lin. Location privacy protection in online query based-on privacy region replacement. In *Annual Computing and Communication Workshop and Conference*, pages 742–747, 2020.

[23] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar. Location privacy preserving mechanisms in location based services: A comprehensive survey. *Computing Surveys (CSUR)*, 54(1):1–36, 2021.

[24] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *International Symposium on Spatial and Temporal Databases*, pages 239–257, 2007.

[25] J. Krumm. Inference attacks on location tracks. In *International Conference on Pervasive Computing*, pages 127–143, 2007.

[26] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[27] W.-S. Ku, R. Zimmermann, H. Wang, and C.-N. Wan. Adaptive nearest neighbor queries in travel time networks. In *International Workshop on Geographic Information Systems*, pages 210–219, 2005.

[28] B. Lee, J. Oh, H. Yu, and J. Kim. Protecting location privacy using location semantics. In *International Conference on Knowledge Discovery and Data Mining*, pages 1289–1297, 2011.

[29] C. Li and B. Palanisamy. ReverseCloak: Protecting multi-level location privacy over road networks. In *International Conference on Information and Knowledge Management*, pages 673–682, 2015.

[30] L. Li, R. Lu, and C. Huang. Eplq: Efficient privacy-preserving location-based query over outsourced encrypted data. *Internet of Things Journal*, 3(2):206–218, 2015.

[31] S. Liu, J. H. Wang, J. Wang, and Q. Zhang. Achieving user-defined location privacy preservation using a p2p system. *IEEE Access*, 8:45895–45912, 2020.

[32] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang. A game-theoretic approach for achieving k-anonymity in location-based services. In *INFOCOM*, pages 2985–2993. IEEE, 2013.

[33] Y. Matsuo, N. Okazaki, K. Izumi, Y. Nakamura, T. Nishimura, K. Hasida, and H. Nakashima. Inferring long-term user properties based on users' location history. In *International Joint Conferences on Artificial Intelligence*, pages 2159–2165, 2007.

[34] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65, 2002.

[35] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *Transactions on Knowledge and Data Engineering*, 13(1):124–141, 2001.

[36] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li. Achieving k-anonymity in privacy-aware location-based services. In *International Conference on Computer Communications*, pages 754–762, 2014.

[37] B. Niu, Z. Zhang, X. Li, and H. Li. Privacy-area aware dummy generation algorithms for location-based services. In *International Conference on Communications*, pages 957–962, 2014.

[38] Y. Qiu, Y. Liu, X. Li, and J. Chen. A novel location privacy-preserving approach based on blockchain. *Sensors*, 20(12):3519–3537, 2020.

[39] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *International Workshop on Privacy Enhancing Technologies*, pages 41–53, 2002.

[40] A. R. Shahid, N. Pissinou, S. Iyengar, and K. Makki. Delay-aware privacy-preserving location-based services under spatiotemporal constraints. *International Journal of Communication Systems*, 34(1):1—-10, 2020.

[41] J. Shao, R. Lu, and X. Lin. Fine: A fine-grained privacy-preserving location-based service framework for mobile devices. In *Conference on Computer Communications*, pages 244–252. IEEE, 2014.

[42] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, pages 247–262, 2011.

[43] S. Siddiqie, A. Mondal, and P. Krishna Reddy. An improved dummy generation approach for enhancing user location privacy. In *International Conference on Database Systems for Advanced Applications*, pages 487–495, 04 2021.

[44] D. Song, M. Song, V. Shakhov, and K. Park. Efficient dummy generation for considering obstacles and protecting user location. *Concurrency and Computation: Practice and Experience*, 33(2):5146–5156, 2021.

[45] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Transactions on Networking*, 11(1):17–32, 2003.

[46] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 10(05):557–570, 2002.

[47] I. Ullah, M. A. Shah, A. Khan, and G. Jeon. Privacy-preserving multilevel obfuscation scheme for vehicular network. *Transactions on Emerging Telecommunications Technologies*, 32(2):4204–4230, 2021.

[48] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, 2014.

[49] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis. Secure kNN computation on encrypted databases. In *Special Interest Group on Management of Data*, pages 139–152. ACM, 2009.

[50] P. Xiong, G. Li, W. Ren, and T. Zhu. LOPO: a location privacy preserving path optimization scheme for spatial crowdsourcing. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–16, 2021.

[51] Y. Xu, J. Peng, W. Wang, and B. Zhu. The connected disk covering problem. *Journal of Combinatorial Optimization*, 35(2):538–554, 2018.

[52] B. Yao, F. Li, and X. Xiao. Secure nearest neighbor revisited. In *International Conference on Data Engineering*, pages 733–744. IEEE, 2013.

[53] H. Zhao, J. Wan, and Z. Chen. A novel dummy-based kNN query anonymization method in mobile services. *International Journal of Smart Home*, 10(6):137–154, 2016.

[54] B. Zhou and J. Xu. Privacy protection in personalized web search: A peer group-based approach. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 424–432. Springer, 2013.