BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi, Bangalore - 560019

A project report on

“LOCATION BASED PRIVACY”

Submitted in partial fulfilment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

By

Niteesh Narayana Hegde (1BM14IS058)

Sameer Raghavendra Katti (1BM14IS079)

Under the guidance of

Mahalakshmi B S

Assistant Professor

Department of Information Science and Engineering
2017-2018

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi,
Bangalore – 560019

Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled "***Location Based Privacy***" is a bonafide work carried out by **Niteesh Narayana Hegde (1BM14IS058)** in partial fulfilment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2017-2018**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Mahalakshmi B S	**Dr.Radhika K R**	**Dr. K. Mallikharjuna Babu**
Assistant Professor	**Professor and HOD**	**Principal**

Examiners

Name of the Examiner **Signature of the Examiner**

1.

2.

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi,
Bangalore – 560019

Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled "***Location Based Privacy***" is a bonafide work carried out by **Sameer Raghavendra Katti (1BM14IS079)** in partial fulfilment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2017-2018**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Mahalakshmi BS
Assistant Professor

Dr.Radhika K R
Professor and HOD

Dr. K. Mallikharjuna Babu
Principal

Examiners

Name of the Examiner **Signature of the Examiner**

1.

2.

BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi,
Bangalore – 560019

Department of Information Science and Engineering

CERTIFICATE

This is to certify that the project entitled "***Location Based Privacy***" is a bonafide work carried out by **Niteesh Narayana Hegde (1BM14IS058), Sameer Raghavendra Katti (1BM14IS079)** in partial fulfilment for the award of degree of Bachelor of Engineering in **Information Science and Engineering** from **Visvesvaraya Technological University, Belgaum** during the year **2017-2018**. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the Bachelor of Engineering Degree.

Mahalakshmi BS **Assistant Professor**	**Dr.Radhika K R** **Professor and HOD**	**Dr. K. Mallikharjuna Babu** **Principal**

Examiners

Name of the Examiner **Signature of the Examiner**

1.

2.

ACKNOWLEDGEMENT

We wish to express my sincere acknowledgement to my guide **Mahalakshmi BS**, Assistant Professor, Department of Information Science and Engineering for valuable advices, suggestions, guidance, continuous support and encouragement throughout the project. I am grateful for her belief upon us for the successful completion of the project.

We sincerely thank **Dr. K Mallikharjuna Babu**, Principal, BMSCE, for providing us necessary infrastructure and facilities and providing this opportunity.

We are also thankful to all the teaching and non-teaching staff and my friends who have contributed directly and indirectly towards my project.

A large amount of material, references has been obtained from various online sources, journals, textbooks. I thank all those who contributed in providing these resources which were invaluable towards the project.

We thank our parents and our family members for providing constant support and their blessings which helped us to translate my efforts into successful achievements

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 – INTRODUCTION

1.1 OBJECTIVE

The objective is to develop a system where the users continue to use the Location based Services without compromising on the needs of location privacy.

1.2 MOTIVATION

The location is the most important key to retail success and is equally important to maintain one's privacy which is why people should be on alert when using the services offered by all those big corporations that are keeping a watchful eye on them -- like Google, Twitter, Facebook, AT&T, Verizon, and Apple, to name a few.

All of these companies have started (or are on the verge of starting) location-centric services. Facebook can now precisely tell the neighbourhood of a person let alone the city. Apple made headlines due to its new iPhone privacy policy which allows it to collect "precise", "real-time geographic location" of its users' iPhones, iPads, etc. In an updated version of it's privacy policy, it was mentioned that the user's location will be shared with partners and licensees. Apple and other companies were entitled to store location information of the user. The problem with not agreeing to the policy was it was must to download and use the services offered.

Why should people care about this? There are a few reasons. One is whether the user likes this idea of getting ads sent to them based on their location as opposed to virtual space. That's a pretty minor issue. Apple will let users opt out of targeted ads (though not out of location tracking). Many people may even find targeted ads a convenience.

The problem with the data collection like this is the unanticipated uses of the data. One example is what happens when a company that collects users' location data goes out of business or is acquired. Any agreement it might have had with users vis-a-vis privacy is essentially moot. And when Internet companies go out of business, their data is often their only tangible, valuable asset. So, conceivably, the company that tracks how often you go to Domino's might end up selling that information to your health insurance company. Expect your rates to rise accordingly.

Sure, location based services make day-to-day life easy. It's more convenient to use phone's GPS to locate the nearest restaurant or gas station. If you're stranded, they could help save your life. But there's a dark side to them most people don't think about. What kind of location information are your service providers storing about you, and how long do they hold onto it? Those are two questions everyone should ask before clicking Yes on a terms and conditions tick box. The user's liberty may depend on it.

The location tracking involves threats to two kind of privacies. The first one is the identity privacy. It is the user details that are acquired by the location service provider. This can be kept a secret by not logging in to most of the systems. The second privacy is the location privacy which contains the exact location details of the user. This isn't protected. The systems require exact location details.

There is a lot of danger associated with sharing the user's location. It might be used to predict the exact time and location of a user, which no longer remains a privacy. Thus, a system in which the user's location can be protected is the need of the hour.

1.3 PROBLEM STATEMENT

With the problems associated with the sharing of location information with the location based service providers, it should be kept secure. The location privacy becomes the topmost priority of any location based service. The suitable measures to maintain the location privacy have to be implemented to make sure that the location of the user is not compromised at any cost.

We propose a two-tier system in which location privacy can be maintained. The system involves a proxy server which helps the users to maintain their location privacy. The user who has to get services from LBS provider, communicates the LBS through the proxy server. The user choose the privacy policy to be used and the communicates with the server which inturn communicates with the LBS. The location privacy policies that can be used are:

1. Location Obfuscation: A technique where location of the users is generalized in order to preserve the location privacy. It is way in which the larger area can be given as the user's location instead of the user's exact location. For example, if the user's exact location is "BMS College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru" then the system disguises the address as "Basavanagudi, Bengaluru".

2. Dummy Location: A method where a fake location in the close vicinity as to the user's real location is selected as a substitute. A dummy location near-by the exact location is used to make sure the services won't differ much. For example, if the user's exact location is "BMS College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru" then the system provides " Bull Temple, Basavanagudi, Bengaluru" as the substitute.

3. k-Anonymity: A group of k users are said to be k-anonymous if they can't be distinguished precisely. The user presents his exact location to proxy server. The proxy then calculates the other k-1 dummy users such that the users can't be distinguished easily. The responses for each of the dummy locations are gotten from the LBS and only the response for the user's exact location is returned back to the user. In this way, the user can still avail all the location based services without the LBS recognizing the exact location.

1.4 MODERN TOOLS

1. Python 2.7.:

Python is a widely used high-level programming language for general-purpose programming. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

2. Postman - An application in Google Chrome for the request and response study.

The Postman Rest Client is a very popular and easy to use HTTP Request composer that makes it easy to call web services. It also provides as an alternative for autogenerating API documentation.

3. Google Geocoding and Reverse Geocoding APIs. :

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.

Reverse geocoding is the process of converting geographic coordinates into a human-readable address.

One can also use the Google Maps Geocoding API to find the address for a given place ID. The Google Maps Geocoding API provides a direct way to access these services via an HTTP request. The following example uses the Geocoding service through the Google Maps JavaScript API to demonstrate the basic functionality.

4. JetBrains PyCharm Community Edition 2017:

IDE for the development of the python code.

CHAPTER 2 - LITERATURE SURVEY

The study as mentioned in [1] provided the obfuscation techniques that proved to be effective in preserving the location privacy. The extent to which mobile devices have been controlling the lives of the individuals, the technical improvements of location techniques are fostering the development of new applications that use the physical position of users to offer location-based services for business, social, or informational purposes. In such a context, the privacy concerns are always on the rise and the need of the hour are the sophisticated solutions that would be able to guarantee different levels of location privacy to the users. In this paper, this problem was addressed and a solution based on different obfuscation operators was presented. The solution when used individually or in combination, protects the location privacy of users. An adversary model was introduced and an analysis of the proposed solution was examined. The paper showed the effectiveness of the location obfuscation techniques.

Location-Based Services (LBSs) have become one of the most popular activities in our daily life. These have eased the life of people over the course of past four-five years. The users are able to send queries to the LBS server and without any difficulties learn about their surroundings. However, these location-related queries which are sent to the server may turn as the potential threats in maintaining the location privacy of the user. They have resulted in some serious privacy concerns since the LBS server is untrusted has all the information about users and may track them in various ways. In the study made in [2], two dummy-locations based solutions to achieve k-anonymity for privacy-area aware users in LBSs were proposed. These were developed considering that side information may be exploited by adversaries. First some candidates were chosen based on the virtual circle or grid method, then these candidates were blurred into the final positions of dummy locations based on the entropy-based privacy metric. The solution was thoroughly analysed by experiments. The security analysis was done. The evaluation results indicated that the proposed V-circle solution could significantly improve the privacy anonymity level. The proposed V-grid solution further enlarged the cloaking region while keeping similar privacy level. This paper established the level of security k-Anonymity can provide with respect to location privacy.

With the advancement in GPS, the position of the user can be given with error less than a few meters. The presence of such highly accurate positioning systems enable use a number of location based services and at the same time it helps the vendors to provide various types of location-based services. With every boon comes a bane. As the position data obtained by such devices include deeply personal information, protection of location privacy becomes one of the most significant issues of location-based services. In [3], a technique to anonymize location data was proposed. In the proposed technique, the user of a location-based service uses several (or even one) fake location data (dummies) and sends them to the server. The location sent to the service is mingled with several other locations and it becomes almost impossible for the provider to detect the true location of the user. As the service provider is unable to distinguish the true location data, the user's location privacy is protected. The performance study was conducted on the proposed technique using practical trajectory data.. These experiments showed that the proposed technique protects the location privacy of users. This study again highlighted the importance of k-Anonymity in correspondence with location privacy.

As has been evident, the Location-Based Services (LBS) have become a vital part of our daily life. The users are at the risk of compromising their location privacy since the server is untrusted. The server can track thee users as and when required. So, this becomes a serious issue. To address this privacy issue, the work in [4] proposed a Dummy-Location Selection (DLS) algorithm to achieve k-anonymity for users in LBS. This was different from the already existing approaches. The proposed DLS algorithm carefully selects dummy locations considering that side information may be exploited by adversaries. The algorithm first chooses the dummy locations based on the entropy metric, and then the proposed enhanced-DLS algorithm made sure that the selected dummy locations are spread as far as possible. The results of the experiments were analysed. The evaluation results showed that the proposed algorithm can significantly improve the privacy level in terms of entropy. The enhanced-DLS algorithm was able enlarge the cloaking region while keeping similar privacy level as the DLS algorithm.

The clustering techniques become important for the above mentioned k-Anonymity techniques. The user location and the dummy locations generated must be similar so that the adversary ,even if it contains side information, won't be able to easily distinguishing between the user location and the dummy locations. There have been several schemes for linear mapping of a multidimensional space proposed for various applications such as access methods for spatio-temporal databases and image compression. In these applications, one of the most desired properties from such linear mappings is clustering. The main aim of clustering is to preserve the similarity between the elements within each cluster. This means that the locality between objects in the multidimensional space is preserved in the linear space. It is widely believed that the Hilbert space-filling curve achieves the best clustering. In the paper [5], the clustering property of these Hilbert Curves were studied. The analysis of the clustering property of the Hilbert space-filling curves was done by deriving closed-form formulas for the number of clusters in a given query region of an arbitrary shape (e.g., polygons and polyhedra). Both the asymptotic solution for the general case and the exact solution for a special case generalize previous work. The results show that the number of clusters depends on the hyper-surface area of the query region and not on its hyper-volume. It was also shown that the Hilbert curves achieves better clustering than the z curve. The paper also had the formulae that provided a simple measure that can be used to predict the required disk access behaviours and hence the total access time. It was proved that the Hilbert curves achieve a better clustering than the z curve in a 2- dimensional space. The average number of clusters for the Hilbert curve is one fourth of the perimeter of a query rectangle, while that of the z curve is one third of the perimeter plus two thirds of the side length of the rectangle in the un-favoured direction. Furthermore, by the simulation experiments, it was shown that the Hilbert curve outperforms both the z and Gray-coded curves in 2-dimensional and 3-dimensional spaces. Thus, it was assumed that his trend will hold even in higher dimensional spaces.

CHAPTER 3

The project aims at developing new new methods of preserving the location privacy. The techniques that eliminate (or make sure there is minimal effect) any chances of location tracking have to developed, tested and used. These techniques must be formed with keeping in mind all the requirements of location privacy. The results of the techniques should be analyzed to predict which technique works better under the given conditions.

CHAPTER 4 – METHODOLOGY

4.1 SOFTWARE REQUIREMENT SPECIFICATION

4.1.1 FUNCTIONAL REQUIREMENTS

a. Register for the new user : The system should provide a simple and unambiguous way to register. It should ask only the minimal information required such as name, contact number, email id, etc. The password should be hashed and hash value should be stored in the database.

b. Authenticate the apps : The system should authenticate the applications that need location based services.

c. Obtain Address : For a given latitude longitude of user's location from the application, the system must be able to correctly determine the address and the locality of the user.

d. Obtain Lat-Long : For a given address of user, the system must be able to correctly determine the latitude longitude of the user.

e. Secure User's actual location : The actual location of the user must be known to the system only and must not be provided to the applications that gives location based services.

f. Make Location Obfuscated: The location of the users should be generalized in order to preserve the location privacy.

g. Able to select Dummy Location: The system must select a location other than user's real location as its substitute.

h. k-Anonymity: Given a location of the user, the system must choose k-1 other random locations and maintain k-anonymity.

i. Communication with LBS : The system must be able to connect to the LBS, send the queries and get the responses.

j. Filter the unwanted responses: The system must be able to filter the unwanted responses and send only the required responses back to the user.

k. Connection to Users : The users must be given a interface to connect with the proxy server.

4.1.2 NON-FUNCTIONAL REQUIREMENTS

a. Performance:- The system must respond to the requests within the time of 5 seconds.

b. Portability:- The system must be compatible with all the environments such as Windows, Linux, Android, IoS, web, etc.

c. Availability:- The system must be always available to the users.

d. User friendly- The system should be unambiguous and clear at all the sections. The minimal information required should be taken.

e. Security :- The system must make sure that the user information shouldn’t be disclosed at any cost.

4.1.3 SOFTWARE REQUIREMENTS

1. Python 2.7
2. Postman Application to analyse the API responses.
3. Pycharm IDE for Python code development.
4. Ubuntu /Windows 10.
5. Google APIs – Geocoding and Reverse Geocoding.

4.1.4 HARDWARE REQUIREMENTS

1. PROCESSOR - Intel core i3 or higher.
2. RAM – 4-8 GB.

4.2 HIGH LEVEL DESIGN

4.2.1 SYSTEM ARCHITECTURE

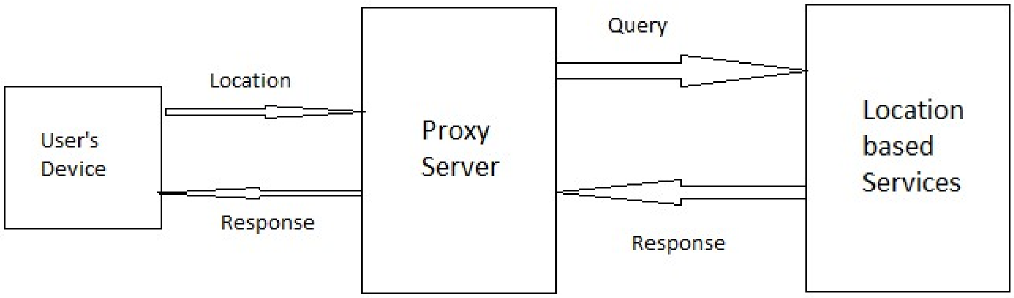


Figure 4.1 – System Architecture

The figure represents the architecture proposed. The user's device connects to the proxy server and submits the latitude and longitude. The proxy server then processes the request using the technique defined by the user and with the given parameters. The obfuscated or dummy or k locations are used as the requests which are sent to the location based servers. The resonses from the location based services are analysed and the most suitable one is sent back to the user. Thus, the user will be able to obtain location based services without actually revealing his exact location to the location based server.

4.2.2 DFD DIAGRAMS:

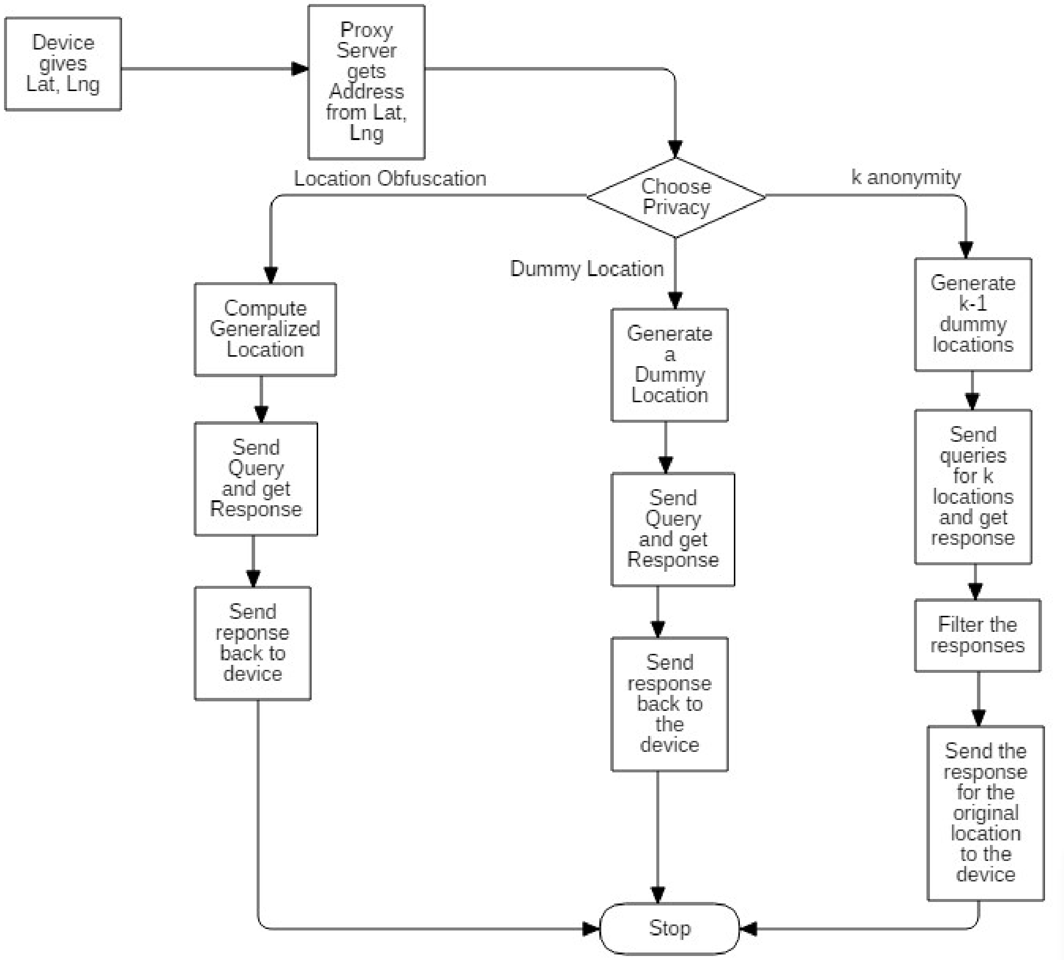

Device gives Lat, Lng
Proxy Server gets Address from Lat, Lng
Choose Privacy
Location Obfuscation
Dummy Location
k anonymity
Compute Generalized Location
Send Query and get Response
Send reponse back to device
Generate a Dummy Location
Send Query and get Response
Send response back to the device
Generate k-1 dummy locations
Send queries for k locations and get response
Filter the responses
Send the response for the original location to the device
Stop

Figure 4.2 – Activity flow diagram

The above figure gives a description about the activity flow in the system. The device gives its location information to the proxy server. The proxy server then chooses the technique specified by the user and computes. It computes the obfuscated location, sends the query to the location based server and gets the response back. Then send the response back to the user. In the second method, it computes the dummy location, sends the query to the location based server, gets the response and send it back to the user device. For k-anonymity, it computes k-1 other dummy locations, send queries for each of them and get the responses. Among the response it'll send back the response corresponding to the user's location.

4.2.3 SEQUENCE DIAGRAM

1. The sequence diagram for the process using Location Obfuscation.

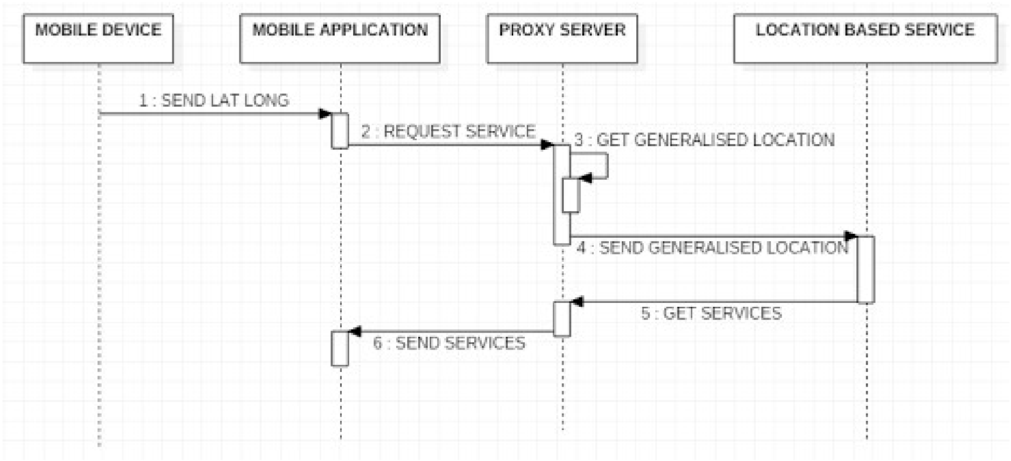


Figure 4.3 – Sequence diagram for Location Obfuscation

The above diagram depicts the sequence of events which take place while computing the obfuscated location. At first the mobile device or the user's device provides the latitude and longitude to the mobile application. The application then sends the request to the proxy server. The proxy server then computes the generalized location. Then using the generalized location, it'll send the query to the location based server. The LBS sends the response back. Then the proxy server returns the response back to the application.

2. Sequence Diagram using Dummy Location:

The figure 4.4 depicts the sequence of events which take place while computing the dummy location. At first the mobile device or the user's device provides the latitude and longitude to the mobile application. The application then sends the request to the proxy server. The proxy server then computes the dummy location. Then using the dummy location, it'll send the query to the location based server. The LBS sends the response back. Then the proxy server returns the response back to the application.

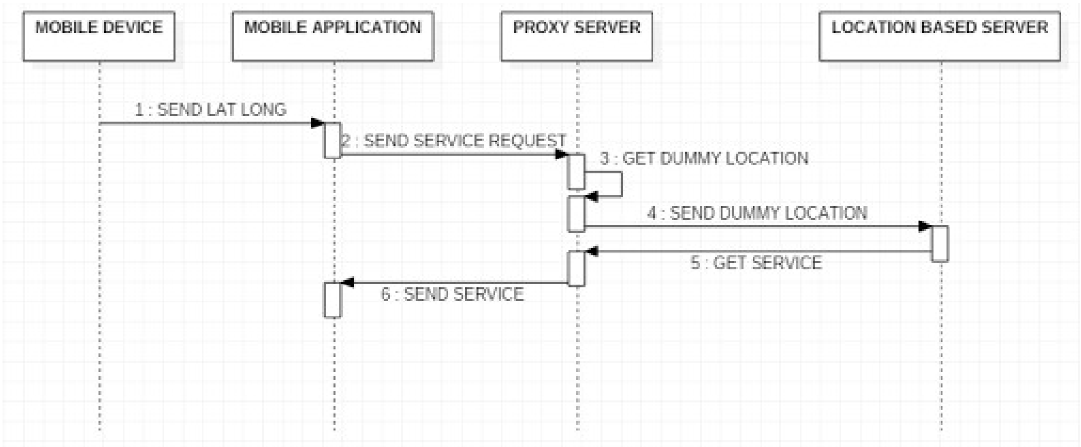


Figure 4.4 – Sequence diagram for Dummy Location Selection.

3. Sequence Diagram for k-Anonymity Approach:

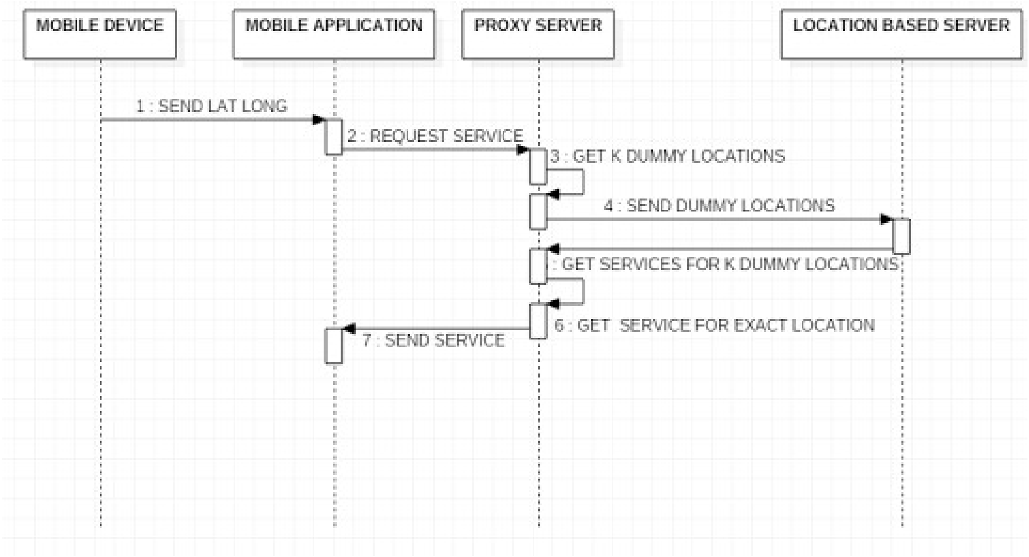


Figure 4.5- Sequence diagram for k-Anonymity.

The above diagram depicts the sequence of events which take place while computing the obfuscated location. At first the mobile device or the user's device provides the latitude and longitude to the mobile application. The application then sends the request to the proxy server. The proxy server then computes the k-1 dummy locations. Then using the k locations, it'll send the query to the location based server for each location. The LBS sends the response back for each of the k locations. Then the proxy server returns the response corresponding to the user's location back to the application.

4.3 LOW LEVEL DESIGN

4.3.1 CLASS DIAGRAM

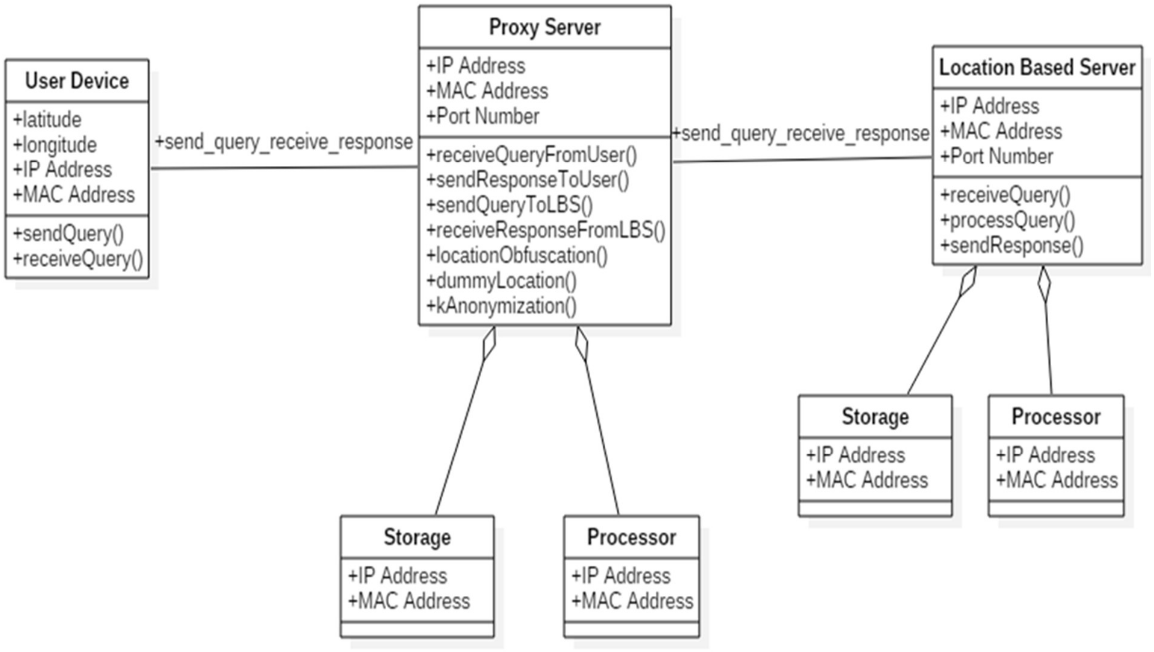


Figure 4.6 – Class Diagram

The class diagram consists of three main classes:

1. User Device: This is the mobile or static device that the user is accessing the internet from. The device has attributes such as the latitude and longitude which form the location, IP Address and MAC Address.

The functionalities of the user device are it can send the query with valid details and receive the appropriate rresponse from the Proxy server;

2. Proxy Server: This is the middle server. It has attributes such as the IP Address, MAC Address, Port Number. The functionalities are to receive query from user, process them, send modified query to LBS, get response and send back the appropriate response back to the user.

It is an aggregation of two units – Storage and Processor. Both have their own IP Address and MAC Address.

3. Location Based Server: This the actual server maintained by the location service provider. It has it's own IP Address, MAC Address, Port Number.

The functionalities are it can receive the query, process it and then send the response back.

It is an aggregation of two units – Storage and Processor. Both have their own IP Address and MAC Address.

4.3.2 DATA STRUCTURE

1. Python List:

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type. Creating a list is as simple as putting different comma-separated values between square brackets.

For example −

```
list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1, 2, 3, 4, 5 ];

list3 = ["a", "b", "c", "d"]
```

2. Python Dictionary:

Each key is separated from its value by a colon (:), the items are separated by commas, and the whole thing is enclosed in curly braces. An empty dictionary without any items is written with just two curly braces, like this: {}.

Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

Example: dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}

3. JSON Format:

When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any object into JSON, and send JSON to the server. We can also convert any JSON received from the server into objects. This way we can work with the data as JavaScript objects, with no complicated parsing and translations.

JSON syntax is derived from JavaScript object notation syntax:

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON data is written as name/value pairs. A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value.

Example:

```
{
 "firstName": "John",
 "lastName": "Smith",
 "isAlive": true,
 "age": 25,
 "address": {
   "streetAddress": "21 2nd Street",
   "city": "New York",
   "state": "NY",
   "postalCode": "10021-3100"
 },
 "phoneNumbers": [
   {
     "type": "home",
     "number": "212 555-1234"
   },
   {
     "type": "office",
     "number": "646 555-4567"
   },
   {
     "type": "mobile",
     "number": "123 456-7890"
   }
 ],
 "children": [],
 "spouse": null
}
```

4.4 IMPLEMENTATION

4.4.1 PROGRAMMING LANGUAGE

Python 2.7

4.4.2 PLATFORM

Windows 10 was used for the project and IDE used was PyCharm Community Edition 2017.

API: Google's Geo-Coding and Google's Reverse Geo-Coding

4.4.3 CODING CONVENTIONS

Comments

Comments are included wherever required. The comments are usually written on a single line or on multiple lines. The whole idea of writing the comments is to indicate functioning of the code written. Comments on single line are prefixed with ‘#’ and comments on multiple lines begin with ''' ''' three single quotes and end with three single quotes as per the comments syntax in Python.

Indentation

The programming language used in the project is Python. The Python strictly requires indentation to differentiate different control structures, iterative statements, loops and as a result. A standard indentation of four spaces was used.

Naming conventions

The code written contains different datatypes, functions, etc which are named according to the functionality or their use in the code. Such a naming is used to reduce the confusions while going through the code. The naming is done using camel case. The words are usually combined with underscore for readability and it enhances the understanding of the code.

Alignment

The code written is aligned properly according to indentation and all similar parameters are aligned vertically to easily find out the parameters used and can be easy to find out the bugs.

Spaces and Tabs

The code is written with appropriate whitespaces wherever required to enhance the readability and understandability of the code. The code is also written with appropriate tabs which provide equal space to parameters and functions written.

4.4.4 STRATEGIES APPLIED

The strategies applied for each type of the location privacy techniques is as follows:

Location Obfuscation: This is technique by which user's location is made general. For example, say that the user's location is "**1a, Bull Temple Rd, Basavanagudi, Bengaluru, Karnataka 560004, India**", it can be generalized as "**Basavanagudi, Bengaluru**".

Steps:

1. The user's location in the form of latitude and longitude.

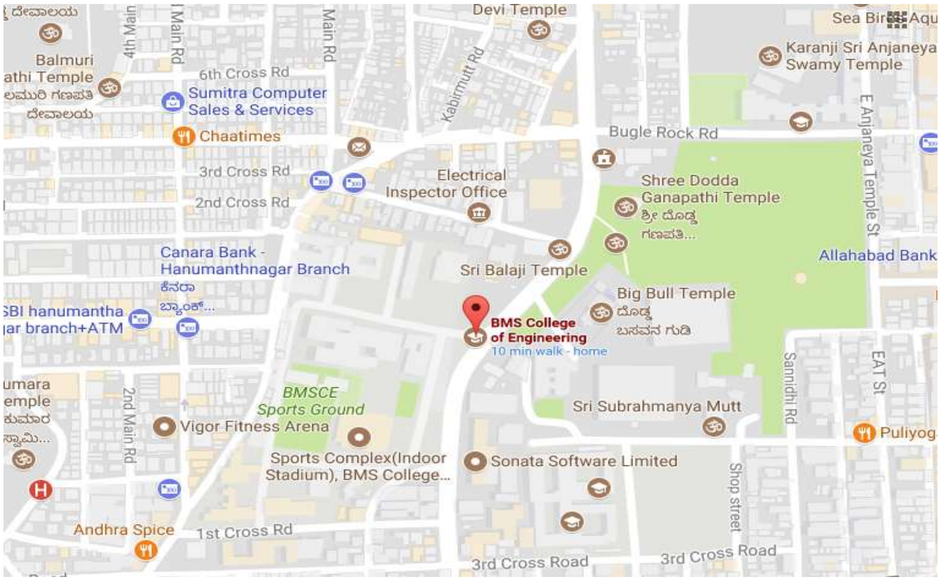


Figure 4.7 – The user location

2. Using the Google Reverse Geo-Coding API, the values of north-east and south-west latitudes and longitudes are obtained.

3. The obtained points are manipulated to get north-west and south-east.

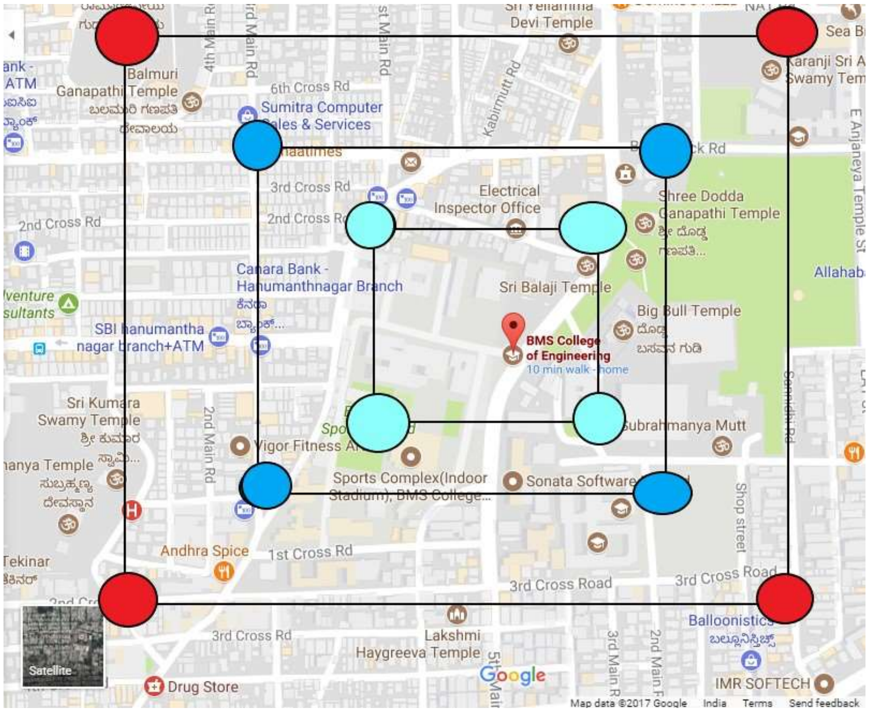


Figure 4.8 – The rectangles constructed around the user's location.

4. Now, an imaginary rectangle with these four points as the vertices is constructed.

5. One more rectangle is obtained by reducing the first rectangle. This is done twice.

6. With three rectangles, there are twelve different points and the location of each of the point is obtained using Reverse Geo-Coding API from Google.

7. The response of the API is used to get the sub-localities. The sub-localities are then mapped to number of repetitions.

8. The sub-locality with maximum number of repetitions is selected as the Obfuscated Area.

Dummy Location Selection: This is a method in which the location of the user is hidden. A fake or a dummy location which is in close proximity to the user is used as a substitute for user's location.

Steps:

1. The user's location in the form of latitude and longitude is obtained.

2. An imaginary circle is drawn with the centre being the user's location.

3. Four imaginary lines are taken. These lines are such that they intersect the circle and intersect each other.

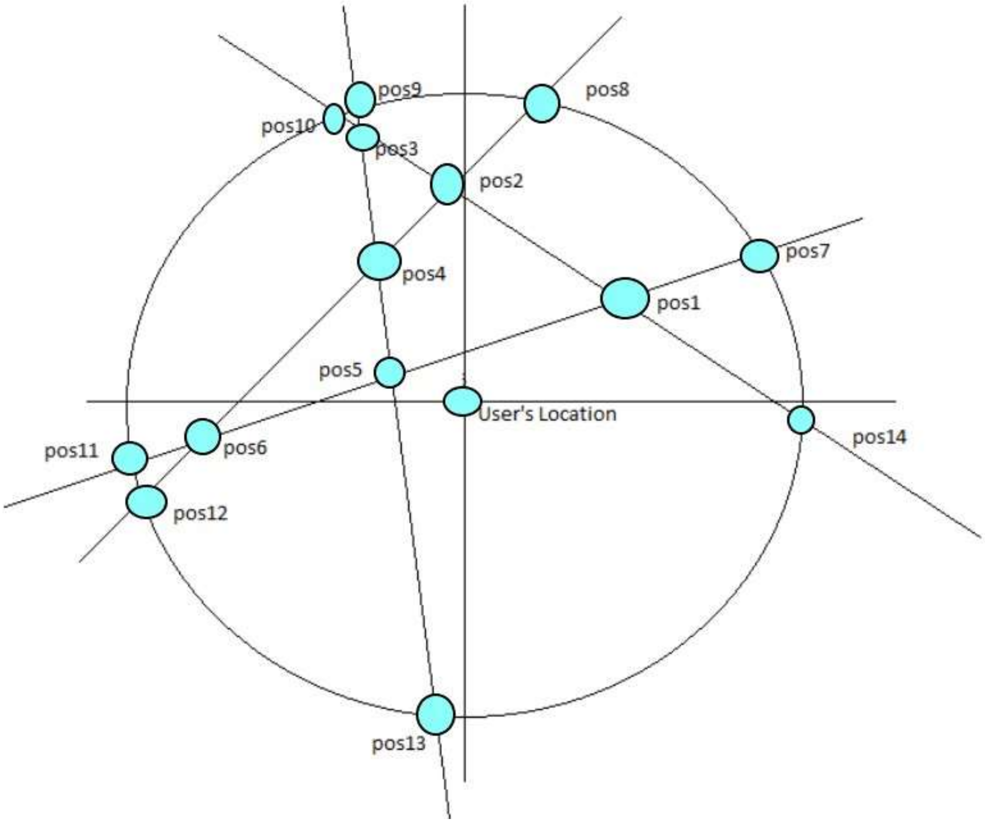


Figure 4.9 – The circle and lines method to form points.

4. The points of intersection outside the circle are neglected.

5. A distance sum is calculated as,

dist_sum = dist_sum + dist of a from from b, for each pair (a, b).

6. The minimum distance sum indicates that the point is closer to all the other points than any other point.

7. This point is choosen as the dummy location.

In the given figure, the point pos5 is the nearest to all other points. So the pos5 is chosen as the dummy location for the user's location.

K-Anonymization:

This is a method in which the user location is hidden among various other locations. For some value k, the user location is merged into a set of k-1 distinct locations which are not too distinguishable from user's location. All these locations are sent to the service provider and responses are obtained. Among the responses, the response for the user's location is selected and it is sent back to the user. This way, the user can actually get the services for his exact location without actually making the server precisely know his location.

Hilbert Curves: A Hilbert curve (also known as a Hilbert space-filling curve) is a continuous fractal space-filling curve first described by the German mathematician David Hilbert in 1891, as a variant of the space-filling Peano curves discovered by Giuseppe Peano in 1890.

Because it is space-filling, its Hausdorff dimension is 2 (precisely, its image is the unit square, whose dimension is 2 in any definition of dimension; its graph is a compact set homeomorphic to the closed unit interval, with Hausdorff dimension 2).

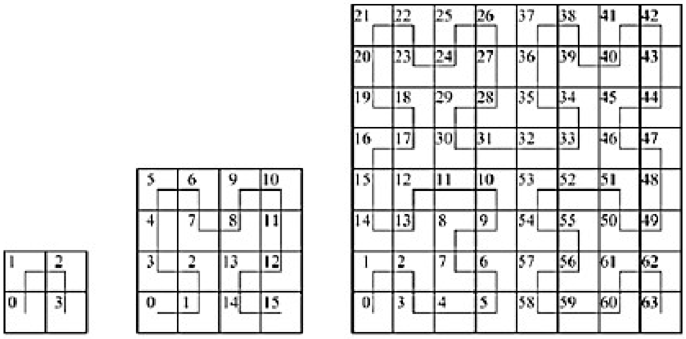


Figure 4.10 – Hilbert Curves

Steps:

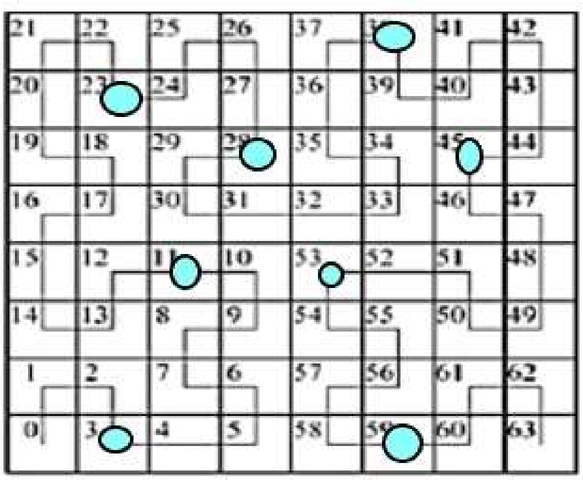
1. A grid of n^4 squares is constructed.

2. A Hilbert curve is made such that it passes through all the squares.

3. A linear function is used which maps a set of values 1,2,3,…,k to 1,2,3,…,n^4.

4. These values in the second set are now the positions where dummy locations are situated.

5. One among the positions is randomly assigned to the user's current location. The positions are chosen as k-1 dummy locations.



User's Location - Box No. 53

Figure 4.11 – Dummy Locations created using Hilbert Curve and Linear Mapping.

The scenario where in k = 8 is depicted. The location in box 53 is the user's exact location. Other locations such as the ones in box 3, 11, 23, 28, 38, 45, 59 are dummy locations.

4.4.5 PSEUDOCODE

The algorithm for the location obfuscation is as shown below.

Location Obfuscation Algorithm:

```
lat, long = getLocation()                                //get location from user
address = getAddress(lat,long)                           //get address using GOOGLE API
north_east, south_west = getPoints(address)              //get the points in address response
points[] = constructRectangle(north_east, south_west)    //construct rectangle
points1[] = minimizeRectangle(points)                    //minimize the rectangle
points2[] = minimizeRectangle(points1)                   //minimize the minimized rectangle
points = merge(points, points1, points2)
for each point in points:
    sublocality1, sublocality2 = getSublocality(point)
    location_info.add(sublocality1, sublocality2)
max_repeated = max(location_info)
return max_repeated.location
```

The algorithm for the dummy location generation is as shown below.

Location Obfuscation Algorithm:

```
lat, long = getLocation()                                   //get location from user

address = getAddress(lat,long)                              //get address using GOOGLE API

north_east, south_west = getPoints(address)                 //get the points in address response

points[] = constructRectangle(north_east, south_west)       //construct rectangle

points1[] = minimizeRectangle(points)                       //minimize the rectangle

points2[] = minimizeRectangle(points1)                      //minimize the minimized rectangle

points = merge(points, points1, points2)

for each point in points:

        sublocality1, sublocality2 = getSublocality(point)

        location_info.add(sublocality1, sublocality2)

max_repeated = max(location_info)

return max_repeated.location
```

The algorithm to compute the k-Anonymity is as shown below.

Location Obfuscation Algorithm:

```
lat, long = getLocation()                                   //get location from user
address = getAddress(lat,long)                              //get address using GOOGLE API
north_east, south_west = getPoints(address)                 //get the points in address response
points[] = constructRectangle(north_east, south_west)       //construct rectangle
points1[] = minimizeRectangle(points)                       //minimize the rectangle
points2[] = minimizeRectangle(points1)                      //minimize the minimized rectangle
points = merge(points, points1, points2)
for each point in points:
        sublocality1, sublocality2 = getSublocality(point)
        location_info.add(sublocality1, sublocality2)
max_repeated = max(location_info)
return max_repeated.location
```

4.5 TESTING

The Testing phase analyses the software testing unit/module, subsystem integration, system, user acceptance, and security -as defined in the test plan. The Test Analysis Report records results of the tests., presents the capabilities and deficiencies for review, and provides a means of assessing software progression to the next stage of development or testing. The results of each type of test wiz. Unit, Integrated and System, are added to the software development document for the module or system being tested. The set of Test Analysis Reports provides a basis for assigning responsibility for deficiency, correction and follow up, and for preparation of a statement of project completion.

Business Requirement Document:

1. Location Obfuscation:

get_location : Given the location in the form of latitude and longitude, thefunction makes an API call using Google API, gets the response and returns the same.

location_generalization : The method gets the location details, converts it to JSON format. The sublocalities present in the response are fetched. If sublocality is not present, then locality is taken.

select_location : This method creates the four corners of imaginary box where the actual location lies and selects, processes all the four corners as individual points.

get_minimized_square : This function creates a minimized rectangle from the present rectangle.

find_obfuscated_area : This function finds the obfuscated area based on the points obtained in the select_location method.

find_coord : The function which returns the latitude and longitude of a location on giving the sub-locality and locality.

calculate_dist : It calculates the distance between two locations whose latitude and longitude are known.

2. Dummy Location Selection:

create_dummies : This function creates the dummy locations. The user location, whose latitude and longitude are given, is used to create dummy locations and then among the set of locations, the appropriate location is returned.

get_address : This function returns the address of location whose latitude and longitude are known.

3. K-Anonymization :

hilbert : This function creates the Hilbert curve which passed through all the individual boxes.

map_range : This function maps the values in the set { 1, 2, 3, …, k } to the set { 1, 2, 3, …, 4^n }.

create_dummy_loc_block : This function creates the blocks within the given square. It also makes sure that mapping values aren't repeated.

create_dummy_loc : This function shuffles the set of locations and makes sure that the probability of finding the exact user location lesser. Then gets the address of the users.

get_correct_response : This function returns the correct response to the user among the k different responses obtained from the server.

get_address : This function returns the address of location whose latitude and longitude are known.

4.5.1 UNIT TESTING

The individual functions (or units) were tested to check their working as per requirements. The test cases were written and the tests were carried out. The test IDs, test description and the test status with any defects present were tabulated.

1. Location Obfuscation :

Test ID	Test Description
Loc_Obs_001	Check if the response is returned for a particular latitude and longitude.
Loc_Obs_002	Check if the response is converted into JSON format.
Loc_Obs_003	Check if the locality and sub-locality are fetched from the response.
Loc_Obs_004	Check if the rectangle is minimized or not.
Loc_Obs_005	Check if the four locations are created for each rectangle.
Loc_Obs_006	Check if the list that maps the sub-localities to localities is created.
Loc_Obs_007	Check if obfuscated location is formed.
Loc_Obs_008	Check if the latitude and longitude are returned for a particular address passed.
Loc_Obs_009	Check if the distance is calculated for a pair of latitude and longitude.

Table 4.1 – The table of tests and their description for Location Obfuscation

2. Dummy Location Selection:

Test ID	Test Description
Dummy_001	Check if the random slopes and intercepts are created.
Dummy_002	Check if the intersection points of each of the lines with the circle are found.
Dummy_003	Check if the intersection points of each pair of lines are found.
Dummy_004	Check if the intersection points which are outside the circle are discarded.
Dummy_005	Check if the sum_dist is calculated.
Dummy_006	Check if the point with the minimum dist_sum is selected.
Dummy_007	Check if the dummy location is returned.

Table 4.2 – The table of tests and their description for Dummy Location Selection.

3. k- Anonymization :

Test ID	Test Description
Anon_001	Check if the Hilbert curve is created.
Anon_002	Check if the blocks are created.
Anon_003	Check if the **K** value is lesser than the total blocks.
Anon_004	Check if the values are mapped.
Anon_005	Check if the repeated values are present in the mapped values.
Anon_006	Check if the requests are sent for each of the **K** locations.
Anon_007	Check if the responses are received.
Anon_008	Check if the response for the user's location is identified.
Anon_009	Check if the requests are made in a random order.
Anon_010	Check if the dummy location is not created within same block of already existing location.

Table 4.3 – The table of tests and their description for k-Anonymization.

TRACEABILITY MATRIX

Test ID	Status	Reason of irregular failures (if any)
Loc_Obs_001	PASSED	Gets slow because of interaction with the external server. If server is not responding, throws error. That's rectified by calling the function multiple times using **@retry.**
Loc_Obs_002	PASSED	
Loc_Obs_003	PASSED	
Loc_Obs_004	PASSED	
Loc_Obs_005	PASSED	
Loc_Obs_006	PASSED	
Loc_Obs_007	PASSED	
Loc_Obs_008	PASSED	Gets slow because of interaction with the external server. If server is not responding, throws error. That's rectified by calling the function multiple times using **@retry.**
Loc_Obs_009	PASSED	
Anon_001	PASSED	
Anon_002	PASSED	
Anon_003	PASSED	
Anon_004	PASSED	
Anon_005	PASSED	
Anon_006	PASSED	
Anon_007	PASSED	Gets slow because of interaction with the external server. If server is not responding, throws error. That's rectified by calling the function multiple times using **@retry.**
Anon_008	PASSED	

Anon_009	PASSED	
Anon_010	PASSED	
Dummy_001	PASSED	
Dummy_002	PASSED	
Dummy_003	PASSED	
Dummy_004	PASSED	
Dummy_005	PASSED	
Dummy_006	PASSED	
Dummy_007	PASSED	

Table 4.4 – Traceability Matrix for Unit Testing

4.5.2 INTEGRATION TESTING

Once the units were tested and made sure they were working as per requirements, the units where then integrated into their respective modules. The new integrated modules were tested and the results are as shown below.

Test ID	Description	Input	Expected Output	Actual Output	Comments
Int_001	Location Obfuscation	Valid Data (Latitude and Longitude)	Show the O/P	Show the O/P	
		Invalid Data (Latitude/Longitude)	Throw exception	Exception	If latitude and longitude are out of proper range, an exception must be thrown.
Int_002	Dummy Location Selection	Valid Data (Latitude and Longitude)	Show the O/P	Show the O/P	
		Invalid Data (Latitude/ Longitude)	Throw an exception	Exception	If latitude and longitude are out of proper range, an exception must be thrown.
		Invalid Data (radius value <= 0)	Throw an exception	Exception	The radius of the area in which the dummy locations has to be generated should be more than 0 km.
Int_003	k-Anonymization	Valid Data (Latitude and Longitude)	Show the O/P	Show the O/P	
		Invalid Data (Latitude/ Longitude)	Throw an exception	Exception	If latitude and longitude are out of proper range, an exception must be thrown.
		Invalid Data (**K** value <= 1)	Throw an exception	Exception	The **K** value which orresponds to number of similar locations, must be atleast 2 (user location and one other location).

Table 4.5 – Table indicating test scenarios for Integration Testing

4.5.3 SYSTEM TESTING

Test ID	Test Description
Sys_001	The system was checked with the unit testing and integration testing.
Sys_002	The portability of the system was checked.
Sys_003	Performance of the system during execution for efficiency.
Sys_004	Reliability of the system for providing location privacy.

Table 4.6: Test Scenarios Table of System Testing

System testing provides the working of system as a whole when all the components of the system are combined. The table contains various test scenarios to be tested for proper working of the system and scenarios are given ID and descriptions for each scenario are tested.

The traceability matrix for the System Testing.

BRD	TEST ID	Description	Status	Defects
System	Sys_001	Validate the unit and the integrated testing.	PASSED	
	Sys_002	Portability of the system.	PASSED	
	Sys_003	Performance of the system for efficiency	PASSED	
	Sys_004	Reliability of the system for providing location privacy.	PASSED	

Table 4.7: Requirement Traceability Matrix of System Testing

The test cases of various scenarios are tested for proper working of the system in various cases such as portability issues, performance of the system, user interface provided to the user, reliability of the system and recovery of data in case of failures where it can fail or pass. All the tests PASSED.

CHAPTER 5 - RESULTS AND DISCUSSION

5.1 RESPONSE TIME

It is the time interval between the time a user sends a query with location and the time the response is received. This is divided into four parts:

1. The time it takes to deliver the query from the user to the server

2. The time the server uses to compute

3. The time for the server to send the private location or manipulated location to service provider and receive responses from the service provider.

4. The time it takes to send back the appropriate response to user.

The time taken by server is the main criteria of evaluation for any system. Our experiments show that the server computes

1. Location Obfuscation in less than 3 seconds.

2. Dummy Location Selection in less than 1-2 milliseconds.

3. k-Anonymization in less than 0.125 seconds for **K** = 100. With the experiments, it has been found that this time is dependent on the value of **K**.

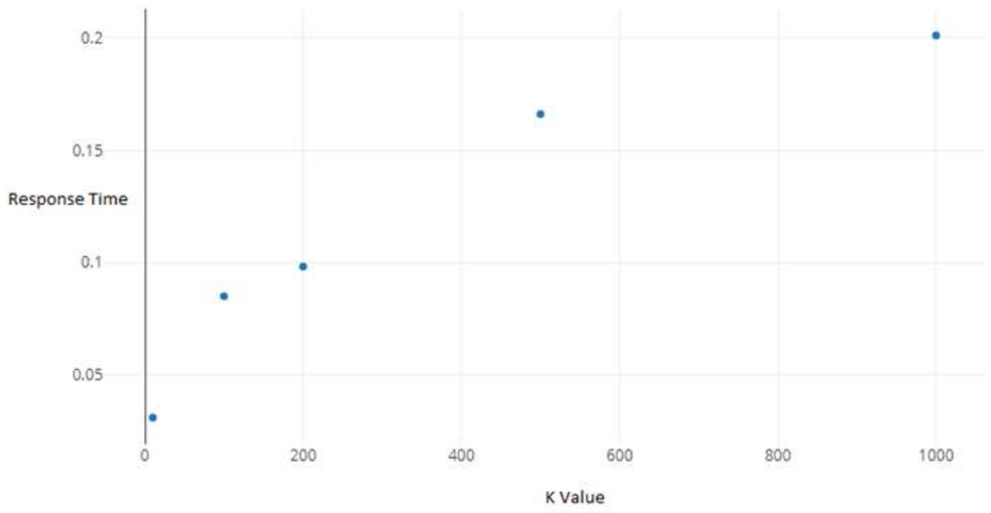


Figure 5.1 – The graph of **K** vs **Response Time** for k-Anonymization technique.

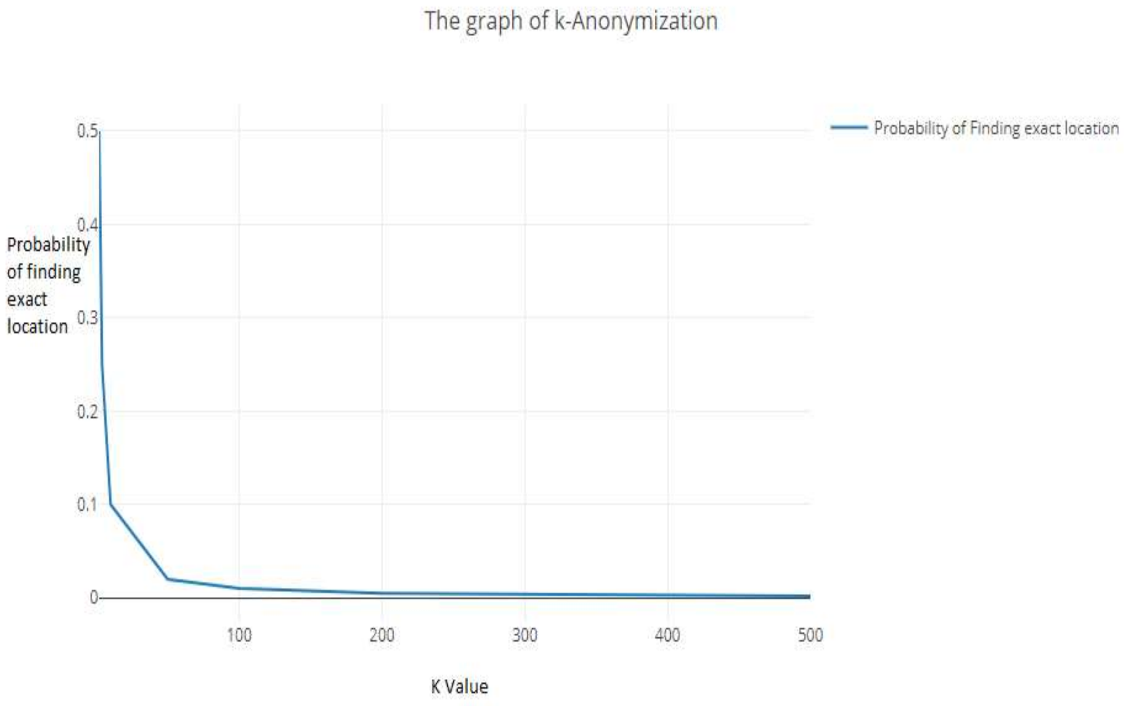


Figure 5.2 - The graph of **K** vs **Probability of finding exact Location** for k-Anonymization technique.

5.2 SECURE LOCATION

It was verified from the experiments that the system helped in maintaining the location privacy. The methods secured the location in different manner.

Location Obfuscation:

This method gave a generalized area or the locality of the user's location for all the locations.

Some of the examples are:

1. User's Location: Kempe Gowda tower atop twin boulders., B M S College of Law Road, Basavanagudi, Bengaluru, Karnataka 560004, India.
   Obfuscated Location generated: Basavanagudi, Bengaluru.
2. User's Location: Railway Station Rd, Malmaddi, Dharwad, Karnataka 580001, India
   Obfuscated Location generated: Malmaddi, Dharwad.
3. User's Location: Hanamsagar Road, Hanamsagar, Karnataka 583281, India
   Obfuscated Location: Hanamasagar.

The location obfuscation will work at all kinds of places – City, Town, Village, etc.

Dummy Location Selection:

This technique generates a fake or dummy location. This dummy location can be used as the substitute to the user's actual location. The location of the user and the fake location won't differ much geographically.

Tests were conducted to generate the dummy location for the user location at various places and each dummy location repeats once in about 50-100 attempts. Thus, the system can be said secure enough and maintains the location privacy.

Examples for Dummy Location Selection:

User Location: BMS College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru.

Some of the dummy locations generated were:

i) 5, Hanumanthnagar, Banashankari Stage I, Banashankari, Bengaluru, Karnataka 560019, India.

ii) 2, 4th Cross St, Srinivasnagar, Banashankari, Bengaluru, Karnataka 560050, India.

iii) 107, 2nd Main Rd, Hanumanthnagar, Banashankari Stage I, Banashankari, Bengaluru, Karnataka 560050, India.

k-Anonymity :

The method generates k-1 dummy locations and merges those locations with the user's location to form a set of k similar locations. Then these locations can be used to query the LBS.

Try 1:

User's Location: B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India

k=10 locations:

1. Sri Maasti Venkateshiyangar Rd, Gavipuram Extn, Gavipuram Extention, Kempegowda Nagar, Bengaluru, Karnataka 560019, India
2. Megha Residency, Basavanagudi, Bengaluru, Karnataka 560004, India
3. Panchavati Niwas, Jain Temple Cross Rd, Parvathipuram, Vishweshwarapura, Shankarapura, Bengaluru, Karnataka 560004, India
4. 3rd Cross Rd, Gavipuram Extn, Gavipuram Extention, Kempegowda Nagar, Bengaluru, Karnataka 560019, India
5. Temple, Govindappa Road, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
6. **B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India**
7. BP Wadia Rd, Basavanagudi, Bengaluru, Karnataka 560004, India
8. Rathna Niwas, 4th Cross Rd, Shankarapura, Bengaluru, Karnataka 560004, India
9. 137, Kanakapura Rd, 2nd Block, 7th Block, Jayanagar, Bengaluru, Karnataka 560011, India
10. 50, Vanivilas Rd, Basavanagudi, Bengaluru, Karnataka 560004, India

Try 2:

User's Location: B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India

k = 10 locations:

1. NN Plaza, Subbarama Chetty Rd, Shankarapura, Bengaluru, Karnataka 560004, India
2. Shyampuri Residency, Conservancy Ln, Basavanagudi, Bengaluru, Karnataka 560004, India
3. **B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India**

4.. Steps to Bull Temple, Basavanagudi, Bengaluru, Karnataka 560004, India

5. 85, Gavipuram Extn, Gavipuram Extention, Kempegowda Nagar, Bengaluru, Karnataka 560019, India
6. Pampa Mahakavi Rd, Shankarapura, Bengaluru, Karnataka 560004, India
7. Rathna Niwas, 4th Cross Rd, Shankarapura, Bengaluru, Karnataka 560004, India
8. E Anjaneya Temple St, Basavanagudi, Bengaluru, Karnataka 560004, India
9. 2, DVG Road, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
10. 6/152, DVG Road, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India

Try 3:

User'Location: B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India

k = 10 locations:

1. 14/3, Bugle Rock Rd, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
2. 115, Surveyor St, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
3. 35, 1st Cross Rd, Gavipuram Extn, Gavipuram Extention, Kempegowda Nagar, Bengaluru, Karnataka 560019, India
4. 4, Krishna Rd, Basavanagudi, Bengaluru, Karnataka 560004, India

5. Prathiba Complex, Karnic Rd, Shankarapura, Bengaluru, Karnataka 560004, India
6. 99/6, Mahantara Lay Out, Kempegowda Nagar, Bengaluru, Karnataka 560019, India
7. 4, Krishna Rd, Basavanagudi, Bengaluru, Karnataka 560004, India
8. 24, Bugle Rock Rd, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
9. 66, Vanivilas Rd, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
10. **B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India**.

Try 4:
User's Location: B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India

k = 10 locations:
1. 29, HB Samaja Rd, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
2. 104, Surveyor St, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
3. 93, Kanakapura Rd, Basavanagudi, Bengaluru, Karnataka 560004, India
4. 14, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
5. **B M S College of Engineering, Bull Temple Road, Basavanagudi, Bengaluru, Karnataka 560019, India**
6. Siddaiya Complex, Mount Joy Rd, Gavipuram Extn, Basavanagudi, Bengaluru, Karnataka 560019, India
7. Soundarya Nilayam, Market Rd, Basavanagudi, Bengaluru, Karnataka 560004, India
8. 113, Diagonal Rd, Parvathipuram, Vishweshwarapura, Shankarapura, Bengaluru, Karnataka 560004, India
9. 38/2, Gandhi Bazar Main Rd, Gandhi Bazaar, Basavanagudi, Bengaluru, Karnataka 560004, India
10. North Public Square Road, Basavanagudi, Bengaluru, Karnataka 560004, India

The different tries conducted show that at k=10, a user location is well hidden among 10 locations and the position of the user location never is the same. Also, the other 9 locations keep on changing from try to try depicting the effectiveness of the algorithm.

CHAPTER 6 - CONCLUSION AND FUTURE WORK

The project aimed at providing the location security to maintain the location privacy of the user. We presented privacy-enhanced techniques that helped in protecting privacy related to the location of the user based on location obfuscation, dummy location selection and k-Anonymization. Our proposed techniques aims at achieving a solution that considers the accuracy of location measurements, which is an important feature of location information, while preserving the location privacy of users. In addition to privacy preserving techniques, we also presented and defined a way to express users privacy preferences. The mentioned techniques for the preservation of location privacy have been proven to be effective enough to maintain the privacy. The experimental results and analysis showed the same. The usage of the Hilbert curve for the generation of the k-1 dummy users was effective since the dummy users were always selected within the vicinity of the user and is effective in not providing any chances of backtracking the user's location based on other k-1 locations.

REFERENCES

1. An Obfuscation-Based Approach for Protecting Location Privacy, Claudio A. Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, Pierangela Samarati, IEEE Transactions on Dependable and Secure Computing (Volume: 8, Issue: 1, Jan.-Feb. 2011).

2. Privacy-area aware dummy generation algorithms for Location-Based Services, Ben Niu; Zhengyan Zhang; Xiaoqing Li; Hui Li, 2014, IEEE International Conference on Communications (ICC).

3. Protection of Location Privacy using Dummies for Location-based Services, H. Kido, Y. Yanagisawa, T. Satoh, 21st International Conference on Data Engineering Workshops (ICDEW'05).

4. Achieving k-anonymity in privacy-aware location-based services, Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, Hui Li, IEEE INFOCOM 2014 - IEEE Conference on Computer Communications.

5. B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. IEEE Trans. Knowledge and Data Engineering, 13(1):124-141, 2001.