A Web Application for Protecting Users Location Information

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The School of Engineering & Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Hima Lahari Marneni
Spring 2015

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David Thomas
Committee Member
ABSTRACT

Now-a-days every application wants the user to share their location in order to retrieve any information. To find a location user has to send their location to the Location Based Services (LBS) and then they are allowed to query the database. Let’s take an example of a user trying to find the nearest coffee shop he has to share his location to find the shop. The user doesn’t like to report location because of the privacy issues. Data is the only asset of the server, which it won’t give away without having control of the data.

In this paper a system is proposed with a two stage approach in which the user’s privacy and content of the LBS won’t get compromised. In the first stage using oblivious transfer user share his/her location in a public grid and in the second stage based on the location private information retrieval is executed to get the users' preferred point of interest.
# TABLE OF CONTENTS

Abstract.......................................................................................................................... ii

Table of Contents.......................................................................................................... iii

List of Figures................................................................................................................ iv

1. Background and Rational............................................................................................ 1
   1.1 Introduction.............................................................................................................. 1
   1.2 Attacks in Real Life.................................................................................................. 2
   1.3 Related Work.......................................................................................................... 5
       1.3.1 Mix Zone.............................................................................................................. 5
       1.3.2 Path Confusion.................................................................................................. 6
       1.3.3 Dummy Location.............................................................................................. 7
       1.3.4 Cloaking............................................................................................................ 8
       1.3.5 Private Information Retrieval........................................................................... 9

2. Narrative...................................................................................................................... 11
   2.1 Problem Statement................................................................................................. 11
   2.2 Motivation.............................................................................................................. 11
   2.3 Project Objective.................................................................................................... 11

3. Proposed System......................................................................................................... 12
   3.1 System Model........................................................................................................ 12
   3.2 Threat Model......................................................................................................... 13
   3.3 Proposed System.................................................................................................... 13
       3.3.1 Private Information Retrieval........................................................................... 14
       3.3.2 Oblivious Transfer Phase................................................................................. 15
3.3.3 AES Encryption ................................................................. 16
3.4 Flow Execution ................................................................. 17
3.4.1 Use Case ................................................................. 19
3.4.2 Sequence Diagram ......................................................... 20
3.5 Environment ................................................................. 20
3.5.1 NetBeans IDE .............................................................. 20
3.5.2 Tomcat 7.0 .............................................................. 21
3.5.3 SQLyog Community ...................................................... 21
4. Functionality of the Application .............................................. 22
4.1 Home Page ................................................................. 22
4.1.1 User Functionalities .................................................... 23
4.1.2 Admin Functionalities .................................................. 31
4.1.3 Database Tables ....................................................... 34
5. Testing and Evaluation ......................................................... 36
5.1 Test Case ................................................................. 36
6. Conclusion and Future Work ................................................. 44

Bibliography and References ..................................................... 46
Appendix-A ................................................................. 49
Appendix-B ................................................................. 54
Appendix-C ................................................................. 61
LIST OF FIGURES

Figure-1.1: Interaction between LBS and user.........................................................1

Figure 1.2.1 News Post about the woman being stalked due to GPS.................3

Figure 1.2.2 Posts about the woman being stalked due to GPS.......................4

Figure 1.2.3 Posts about the employees being stalked due to GPS.....................4

Figure 1.3.1: Mix-Zone arrangement.................................................................6

Figure-1.3.2: How dummy locations are created and send to LBS.....................7

Figure 1.3.3: Cloaking based on k-anonymity technique..................................8

Figure 1.3.4: framework for K-anonymity location privacy................................9

Figure-3.1: System Model.................................................................................12

Figure 3.3.1 System Architecture.................................................................14

Figure 3.4.1 Flow of Execution......................................................................17

Figure 3.4.2 Work Flow of Proposed System...............................................18

Figure 3.4.3 Use Case Diagram......................................................................19

Figure 3.4.4 Sequence Diagram.....................................................................20

Figure 4.1 Home Page.....................................................................................22

Figure 4.1.1 User Login Page..........................................................................23

Figure 4.1.2 User Registration Page..............................................................24

Figure 4.1.3 Mail received after activating user account................................24

Figure 4.1.4 User home page.........................................................................25

Figure 4.1.5 Selecting initial position.............................................................26

Figure 4.1.6 Option to name the location.......................................................26
Figure 4.1.7 the get key option is selected the master key is displayed......................27
Figure 4.1.8 Selecting the Destination.................................................................28
Figure 4.1.9 Initial Point is encrypted and secret key is displayed.........................29
Figure 4.1.10 Directions displayed in maps.........................................................30
Figure 4.1.11 Directions clearly shown.................................................................30
Figure 4.1.12 Admin home page............................................................................31
Figure 4.1.13 new user details at admin page for getting activated.......................32
Figure 4.1.14 New POI generated by user waiting for security key generation..........33
Figure 4.1.15 Security Key generated for given user details..................................33
Figure 4.1.16 SQL Connector.................................................................................34
Figure 4.1.17 User Details Table...........................................................................35
Figure 4.1.18 Encrypted latitude and longitude values in database.....................35
Figure 5.1.1 Alice registration page.................................................................36
Figure 5.1.2 Alice selecting her initial location......................................................37
Figure 5.1.3 Secret key generated for Alice on server side...................................38
Figure 5.1.4 Alice selecting final destination.......................................................38
Figure 5.1.5 The initial point of Alice is encrypted...............................................39
Figure 5.1.6 Alice gets directions from initial point to final point.......................40
Figure 5.1.7 Alice location encrypted in the database..........................................40
Figure 5.1.8 Initial point of the user.................................................................42
Figure 5.1.9 User Select the final destination......................................................42
Figure 5.1.10 Initial point encrypted.................................................................43
Figure 5.1.11 Directions from initial to final destination.......................................43
1. BACKGROUND AND RATIONALE

1.1 Introduction:

Users are always connected and have access to the internet with the development in technology and because of the omnipresence of the mobile devices. Users have many benefits of the online services. In the online services the LBS (location based system) are getting high popularity. The LBS a service that provides location-aware information based on the user’s current location. Examples like close-by public resources, localized news, maps, emergency services and even the personal search for the marketing or a friend finder. The working of LBS is shown in Figure-1.1.

![Figure-1.1: Interaction between LBS and user [19]](image)

Privacy assurance is the major problem in the application in which the location of the user is being disclosed. Due to the sequence or continuous queries done by the user, LBS can reveal the private information about the user’s like health information, organizations, properties, financial status, political and religious orientation, social
security number and also the information about the persons who are more intimate to the user. So the user is directly or indirectly revealing his/her private information. The query request can reveal the location of a person at a particular time, a request for the personal point of interest (POI i.e., Shops, clubs, etc...), the request done for the sensitive information and even the routes taken by the user at a particular instance.

Whenever a query is done by the user the Location Server saves the details of the user like mobile number, location, username etc…. in a table in its own database. These Location Servers have some backdoor from where the details of the user can be retrieved. This would be a great threat to the user’s private life and there would be no privacy.

The location Server (LS) which provides the Location based Services (LBS), uses all its resources to get the data about various point-of-interests [18]. So, the LS would not give its data without any fee. It is also important to see that the unauthorized user should not be allowed to access the data for which they have not paid [17]. So a solution must be derived to protect the user’s privacy when querying the database and also to prevent unauthorized users from accessing the data.

1.2 Attacks in Real Life:

Here are few real life examples of why user location has to be preserved and what consequences occur when location is shared:

1) A Canadian reporter was defamed through a false profile which has misleading posts and intellectually inconsistent political positions. This was because his personal information was hacked and used to defame him.
2) An American soldier’s bank account was drained because of his name, email-id and Facebook profile was replicated by a security expert.

3) A man was accused of stalking his ex-girlfriend by installing GPS in her car. There are also cases where women were stalked using the GPS system. The new for this attack is shown in the figure 1.2.1 and figure 1.2.2.

![News Post about the woman being stalked due to GPS](https://www.foxnews.com)

**Figure 1.2.1** News Post about the woman being stalked due to GPS [21]

4) A company used GPS to track their employees and an alarm was set to ring if the employee goes to bars or a pub, which violates the privacy of the employees and there would be no privacy left as shown in figure 1.2.3.
In Location Privacy Act 2001, privacy threat due to location share was identified in the United States. A privacy threat occurs when such sensitive information about the user is disclosed due to one or more location queries. Every request done will have the
time, user identity, location and requested service. So this a serious privacy issue of the user that has to be considered. The privacy threats that can occur:

So based on the situations and study there is a need to find a method in which the user privacy is not compromised and also the location server’s data should also be protected from the being accessed by the attackers.

1.3 Related Work:[23]

Previously some methods are proposed by many people to protect user location privacy, but there were drawbacks in those papers. Here a summary of all the work is shown.

1.3.1 Mix-Zone: (Beresford and Stajano 2003)

The mix - zone is the region in space where the user’s movement and identity cannot be traced by any application. In this approach the user’s name and identity are changing constantly in the mix-zone to safeguard the user’s privacy [7]. The information is being exchanged between user and location sever, the continuous change of user name protects the user’s privacy to some level. A user refuses to send any location update if the mix zone has less than k users. Upon emerging from the mix zone, an adversary cannot know which one of the user has come out of the zone. The mix zone is shown in Figure-1.3.1.
When a mix-zone was applied to road network, their investigation showed that the required number of users like 6-7 to satisfy unlink ability property for repeated queries over an interval. Due to this reason this approach requires careful control of the users who are present in the mix-zone, which is hard to achieve in practice [11].

1.3.2 Path Confusion: [1]

If the adversary’s location is near to the user’s location, he can easily find out the identity of a person close to his location. To prevent this there is an approach implemented based on the perturbation algorithm [1]. In this approach what the algorithm does is

i) If any two users meet at one location, this algorithm can cross paths in the area.
   So the attacker gets confused.

ii) If any two users are moving in parallel, then the algorithm shows the path as they are going in opposite ways.
This approach is good as it is preventing the attacker from knowing the actual location of the user, but it is not able to protect the time-series of the user's location information.

1.3.3 Dummy Location: (Gedik and Liu 2005)

In this method the user’s original location is replaced by dummy location in order to confuse the attacker [2]. The location and parameters of the user are converted into another query having the user’s real location and k-1 dummy locations whenever a user queries the server which is shown in Figure-1.3.2.

![Figure-1.3.2: How dummy locations are created and send to LBS [2]](image)

In this the privacy of the user is not protected because in order to give a correct reply to the user the server requires user’s original location [5]. Due to too-many queries with the user’s dummy location the server is unable to differentiate the actual location from dummies. Due to a lot of input flow to the user device because of not differentiating the location the user device will have communication overhead and the device may fail to work.
1.3.4 Cloaking: (Hashem and Kulik 2007)

The cloaking is based on the K-anonymity. K-anonymity is a privacy concept not much related to location privacy [3] [4]. It hides the user location in k-1 neighbour’s location as shown in Figure-1.3.3. By using the k-anonymity it guaranteed that the user’s location is not determined from the other K-1 fake locations [9] [10].

The techniques which use k-anonymity will have a trusted third party called the anonymizer, for creating the cloaking region (CR), which will have the set of k-1 neighborhood locations and the original user’s location. [13]

![Diagram of Cloaking based on k-anonymity technique](image)

**Figure-1.3.3:** Cloaking based on k-anonymity technique [13]

In this approach they assume that the user is trying to locate the nearest Point of Interest (POI). So whenever the user sends the anonymizer the query, it authenticates the user and searches for the neighbors of the user, who are active at that point of time so that it can create the CR. After the CR is created the anonymizer sends the request to the LBS. Once the response is received by the anonymizer the results are sent back to the authenticated user as shown in Figure-1.3.4.
Figure 1.3.4: framework for K-anonymity location privacy. Con stands for k-nearest neighbor query, ASR is anonymizing spatial region and POI is the point of interest [14]

The user’s location preserved using this approach, but the drawbacks of this approach are:

i) When sending the CR to the LBS the location of the anonymizer is sent to the LBS, the anonymizer can be tracked by the attacker and the attacker can directly get the access of the anonymizer. By doing so the privacy of all the user’s is compromised.

ii) To create the CR the anonymizer uses the locations of the users who are registered to it. So there should be a minimum number of users to create the CR.

iii) If a user in the CR sends out a query the location of the user can be tracked easily as the location of that user is sent in all the CR’s.

1.3.5 Private Information Retrieval: (Ghinita et al. 2008 and Ghinita et al 2009) [6][8]

The Private Information Retrieval (PIR) approach uses the Grid cells (GC) concept [6]. The PIR is used to get the data from the LBS without compromising the user’s location [8]. The GC’s is used to query the LBS for a reduced set which will have the POI of the user.

This approach requires TTP and CR and the privacy is achieved using the cryptographic technique. If the only cryptographic technique is used the user privacy is
protected, but the LBS data will be at risk, as the PIR approach will retrieve information about whole place instead of some particularly interesting point. In this approach the server will generate the regions regarding to POI and when it answers the user query, the server sends the regions to the user. The user recognizes his region and using the PIR user requests all the places within that region.

The drawbacks of this approach are that the technique is more expensive and has a high CPU cost. The user is made to go through many tests in order to process the query which makes the execution time more. It will not protect the database’s privacy.
2. NARRATIVE

2.1 Problem Statement:

The main problem with the location-based query is that the user wants to find the location of his/her point of interest (POI) but whenever a query is sent to the LBS the user's location is being shared which will result in the privacy threat. The Location Server (LS) uses all its resources and assets to gather the information about various point-of-interests. The LS will not disclose its data without paying some fee. So, the user’s privacy and LBS data are at threat because of the location-based queries.

2.2 Motivation:

Many techniques and solutions were proposed to protect the user’s privacy in location-base queries. There were various encryption methods used to protect the data in the LBS. But the previous techniques and solutions had their own drawbacks. Due to these reasons a new solution has to be proposed to increase the security of user and server in protecting their privacy and data. The performance of the system must also be increased computationally and communicational. The solution proposed should be an intuitive solution i.e.; it must be clear from the logical point of view.

2.3 Project Objective:

The drawbacks of the previous work done on Private Information Retrieval by Ghinita et al in 2009 has to be reduced. So in the proposed system the Private Information Retrieval concept is used and an idea from the cloaking by Hashem and Kulik in 2007 are used.
3. PROPOSED SYSTEM

3.1 System Model:

The service providers are getting a huge profits due to the location-based services that they are providing. The network operators outsource the data supplied by the location-based service to the third party service provider. For example, in the UK, the network providers Vodafone and orange give the location information of their customers to the third party applications such as mapAmobile or World-Tracer.com [22]. So, in the system model the network operator and the service provider are considered as two separate entities. The network operator implements the API used by the service provider to offer the location-based services.

In the system model that is developed, the network provider will always have the location and the customer identity of the user. In the existing system the details about the user location and identity will be sent to the service provider.

Figure-3.1: System Model [11]
3.2 Threat Model:

The service provider should not be aware of the user’s identity and the location, this is the main threat that is addressed in our model. The identity of the user who is using the services provided by the service provider can be revealed at some times, but the location of the user should never be revealed to the service provider.

It is very important to consider this threat due to two reasons:

1) If the service provider is a malicious one, then user details are at risk [22]. The information may be used by the stalker or an attacker to attack the user. The disclosure of the user details is not a required action to be done by the service provider.

2) If the service provider is not a malicious one then also the user details may be disclosed because of the bugs in the software installed by the attacker to get unauthorized access to the database and a query may be done on the network operator for location information of the user [22].

Here a system is proposed which can handle such type of attacks by the service provider. And also the attacks done on the application.

3.3 Proposed System:

Consider a scenario in which Alice is searching for gold deposits in California. She finds a place with a small trace of gold and follow those traces to find the deposit. But there are many deposits which have patents in California. Now she has to find a gold deposit which has no patent on it. She travels throughout California with GPS and a computer. When she finds a gold deposit she searches to see if there is a patent on that
land and she has to pay for each query done on the database. Here the problem is that her rival Bob is also searching for the gold deposit. Instead of searching for the deposit by himself, Bob is keeping a watch on the queries done by Alice and location. So if Alice finds a deposit Bob will also know about that and he will claim patent before Alice does.

So Alice has to find a method by which Bob will not be able to know about her queries and location. She can send a lot of queries to the database in order to confuse Bob but this is expensive as she will have to pay for each query. She can buy the database and then query on her own database.

By considering the problem of Alice a system is designed (shown in figure 3.3.1) with Private Information Retrieval (PIR) and Oblivious Transfer. Let’s see what Private Information Retrieval (PIR) is:

3.3.1 Private Information Retrieval (PIR):

Private Information Retrieval (PIR) protocol allows users to query a database D with size N and retrieve a block of data from a database without revealing to the server
which block is retrieved. To protect the privacy of the user PIR is used, as the service provider should not know what query is sent by the user and which block is retrieved from the database. Using the PIR user requests about their location interests without revealing the location of the user. To do this a server database has to be transferred or scanned to query about the user's interest, which would lead to server knowing about the data being retrieved and also the process is too costly.

To get the exact data from the service provider a private grid, with the location interests, is superimposed on the public grid, with the GPS coordinates of the user location. Due to this every block in public grid will have a POI in the block superimposed on the private grid.

PIR will not restrict the data being retrieved i.e., it can retrieve one block or a bit of data from the database. User can select the size of the block in the public grid [11]. Due to this the data in the database on the server is at risk because the user can select a bigger grid which will retrieve a big block of data. By using the PIR the user privacy is guaranteed but the data of the service provider will be at risk because there is no restriction of how much data to be retrieved.

So a method has to be implemented in order to provide safety to the server’s database because the server allots all its resources to gather the information in the database. To do this a method is implemented called Oblivious Transfer Phase protocol.

3.3.2 Oblivious Transfer Phase:

In general Oblivious Transfer Phase protocol is a method in which a sender sends the information to the receiver, but remains unaware of what information is received [11]. This is done by sending a number of pieces of data to the receiver but the receiver selects
only one from the large amount of data that is received. The sender will not know which
data is selected by the receiver. As the user sends the coordinates the server will select
one coordinate and create a key using that selected coordinate.

Here this protocol is used to extract only one block of data from the huge amount
of data that is retrieved using PIR i.e., only one from the public grid [11]. The Public grid
will have ‘m’ columns and ‘n’ rows. So each block of the public grid will have a cell Id
and the key used to encrypt the data that enters the block.

The user will mark his location coordinates which will help PIR retrieve a single
block of data from the service provider’s database. Once the coordinates are determined,
at the network provider, the transfer phase is executed and a matrix with the symmetric
key is generated for that particular coordinates. Here the user is sending his coordinates,
but he is not aware about which coordinates are selected to generate the symmetric key
and where they are stored in the grid.

The symmetric key generated is used by the PIR to encrypt the data of POI that is
retrieved from database. So, when a user queries for some point-of-interest (POI) the data
that is retrieved is encrypted [11]. The Id of the user and the symmetric key are stored
one block of the public grid, which are used to decrypt the data encrypted by PIR.

3.3.3 AES Encryption:

AES encryption is the standard encryption method to encrypt the electronic data.
AES encryption is a symmetric key encryption as the encryption and decryption is being
done on a database. AES is a symmetric-key block cipher that can use 128, 192 and 256
bits and decrypts the data in blocks of 128 bits [15]. Asymmetric key encryptions use
public and private keys to encrypt and decrypt the data, as the data is being shared
between two people. In AES only a single key is used as the data is encrypted while saving and decrypted in the same place when retrieving.

The matrix prepared by the AES will be a 4x4 matrix which is the symmetric key that is stored in the block of the public grid. This 4x4 matrix is used to encrypt the data when storing the data retrieved by PIR and decrypting the data when user retrieves the block.

3.4 Flow of Execution:

Figure 3.4.1 shows the flow of execution of the application which is using oblivious transfer and Private information retrieval.

![Flow of Execution Diagram](image-url)
Firstly a user is registered to the application by giving his name, address, phone number etc. Once registered, the user will login into the application and enters his location coordinates. This location will act as the initial location for the user to get the directions.

Now the user location coordinates are sent to the application server to create an Id and the symmetric key matrix. This is the Transfer Phase where the ID of the user location and the corresponding key are added to the public grid by the server. The work flow of the proposed system is shown in figure 3.4.2.

![Figure 3.4.2 Work Flow of Proposed System [11]](image)

Now users can access the location created by him as the initial location and search for directions. In the application he can select the final destination where he wants to go. Once the selection is done the PIR phase is run to retrieve the data. For this the initial location is encrypted using the symmetric key matrix that is generated at the server for that particular location. Once the data is retrieved, using the same key matrix the data will be decrypted and direction information will be shown in the application.
3.4.1 Use Case:

Use cases are used to determine the actions done by the actors. The figure 3.4.3 shows the actions performed by the user and the administrator. In the figure 3.4.3 user performs sign up, login, creates POI’s, get Direction and logout and the server performs activating new users, adding POI’s and log out.

![Use Case Diagram](image-url)

Figure 3.4.3 Use Case Diagram
3.4.2 Sequence Diagram:

Figure 3.4.4 shows the sequence diagram shows the combination of objects based on time sequence. The functionalities for the user and the server is shown in the figure in the sequence in which they are executed.

![Sequence Diagram](image)

Figure 3.4.4 Sequence Diagram

3.5 Environment:

3.5.1 NetBeans IDE:

NetBeans is a software development tool which is written in Java. Modules which are the modular software components used by the user to develop an application. This is
an application in which the new versions and the Integrated Development Environment (IDE) are developed by the third party developers. This is mainly developed to support Java, but it can also support PHP, c/c++, HTML etc..

3.5.2 Tomcat 7.0:

It is an open source web server used to run the Java EE, Java server and JavaScript pages.

3.5.3 SQLyog Community:

It is a Graphical User Interface (GUI) tool used for the RDBMS MySQL. It is a combination of features like MySQL Workbench, phpMyAdmin and other MySQL front end and GUI tools. The features make it the most powerful SQL manager.
4. FUNCTIONALITY OF THE APPLICATION

The proposed system described above is implemented as a web application. The application is written using the JSP and Java. This application is designed in a way to protect the user location and the data at the server. The google maps are integrated to the application in order to provide the point-of-interests (POI) and to get the directions.

The location hiding system which is developed has the following user interface and the steps for execution:

4.1 Home Page:

The home page shown in the Figure 4.1 has the user login, Location Server login and a home page. The user has to login or signup to continue with the application.
From the home page a user can login or an administrator. So first let us see the user side functionalities and working.

4.1.1 User Functionalities:

Figure 4.1.1 shows the user login page with the “username” and “password”. For the new users, there is a “sign up” option at the bottom.

![User Login Page](image)

Figure 4.1.1 User Login Page

If a new user wants to register to the application users have to enter their details on the registration page shown in figure 4.1.2. Once all the details are entered user clicks on the register button. The details entered by the user are stored in the database. If the registration is successful a “successful registration” is displayed.
Once the user is registered admin of application will activate the user profile so that the user can login with their username and password. Once the account is activated a mail is send to the user with the unique registration ID as shown in figure 4.1.3.
The user will login into his profile and the user home page will have the options to add POI, get directions and logout as shown in figure 4.1.4.

![User home page](image)

**Figure 4.1.4 User home page**

Now the user will add a POI which is nothing but his present location by dragging and dropping the pin on the map as shown in figure 4.1.5.

This place can be given a name like “home”, “office” and save it using the ID that is generated during registration (shown in figure 4.1.6).
Figure 4.1.5 Selecting initial position

Figure 4.1.6 Option to name the location
There is a get key option which will fetch the master key from the database shown in figure 4.1.7. The latitude and longitude values of the user location are saved in the database and a copy is sent to the admin so that he can run the oblivious transfer phase. Once the admin runs the transfer phase and generates the symmetric key the location will be good to use.

Figure 4.1.7 the get key option is selected the master key is displayed

Now the user goes to the “Get direction” option and selects the final destination in the maps as shown in figure 4.1.8.
Figure 4.1.8 Selecting the Destination

After selecting the destination, click on Next option which will display a page with the encrypted initial location and the symmetric key that is used for encryption which is shown in figure 4.1.9. The AES encryption class is used to encrypt the data by using the 4x4 matrix generated by the admin.
Figure 4.1.9 Initial Point is encrypted and secret key is displayed

Once the user clicks on Decrypt button the PIR phase is executed and the direction from the initial location to the destination are displayed as shown in figure 4.1.10 and figure 4.1.11 shows the clear direction display.
Figure 4.1.10 Directions displayed in maps

Figure 4.1.11 Directions clearly shown
### 4.1.2 Admin Functionalities:

The Administrator will login into his account from the home page and he will have options considering the “new users”, “new POI’s” and “logout” as shown in the figure 4.1.12.

**Figure 4.1.12 Admin home page**

When a new user is registered to the application the admin will get the user details and the admin have to activate the user as shown in the figure 4.1.13.
Now when a new POI is added by the user the admin has to run an oblivious transfer query in order to generate the key and Id as shown in figure 4.1.14 and figure 4.1.15. To generate the secret key a matrix class is executed which will create a 4x4 matrix which will be used by AES encryption to encrypt the initial location.
Figure 4.1.14 New POI generated by user waiting for security key generation

Figure 4.1.15 Security Key generated for given user details
4.1.3 Database Tables:

The user details are stored in the database using SQLyog. Once the SQLyog application is run, it asks for the root and password as shown in the figure 4.1.16.

![Figure 4.1.16 SQL Connector](image)

There are two tables in database: one with the user details shown in the figure 4.1.17 and another for the users POI and the secret keys generated for that particular POI location. The latitude and the longitude details entered by user are encrypted and stored in the database as shown in the figure 4.1.18. For this the given location is converted from bytes to bits and then stored in the database.
Figure 4.1.17 User Details Table

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
<th>phone</th>
<th>email</th>
<th>un</th>
<th>pwd</th>
<th>status</th>
<th>pk</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>abhay</td>
<td>08944654</td>
<td>2243657</td>
<td><a href="mailto:abhay@gmail.com">abhay@gmail.com</a></td>
<td>ed</td>
<td>ed</td>
<td>ok</td>
<td>17149411</td>
</tr>
<tr>
<td>5</td>
<td>hyd</td>
<td>9231213</td>
<td>95229391105</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>ali</td>
<td>ali</td>
<td>ok</td>
<td>23982709</td>
</tr>
<tr>
<td>5</td>
<td>hyd</td>
<td>9231213</td>
<td>95229391105</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>ali</td>
<td>ali</td>
<td>ok</td>
<td>35007146</td>
</tr>
<tr>
<td>10</td>
<td>avinash</td>
<td>784112</td>
<td>569461464446</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>ali</td>
<td>ali</td>
<td>ok</td>
<td>68691441</td>
</tr>
<tr>
<td>12</td>
<td>AFS</td>
<td>696565</td>
<td>6964622316</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>AFS</td>
<td>AFS</td>
<td>non</td>
<td>non</td>
</tr>
<tr>
<td>4</td>
<td>hina</td>
<td>523143</td>
<td>8522991105</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>hina</td>
<td>hina</td>
<td>ok</td>
<td>34028872</td>
</tr>
<tr>
<td>7</td>
<td>keer</td>
<td>523143</td>
<td>8522991105</td>
<td>keer@kp@<a href="mailto:4@gmail.com">4@gmail.com</a></td>
<td>keer</td>
<td>keer</td>
<td>ok</td>
<td>32023977</td>
</tr>
<tr>
<td>17</td>
<td>fudu</td>
<td>56658</td>
<td>514541484545</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>fudu</td>
<td>fudu</td>
<td>ok</td>
<td>fudu</td>
</tr>
<tr>
<td>9</td>
<td>mura</td>
<td>784112</td>
<td>469134656</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>mura</td>
<td>mura</td>
<td>ok</td>
<td>52666270</td>
</tr>
<tr>
<td>6</td>
<td>sai</td>
<td>784112</td>
<td>469134656</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>sai</td>
<td>sai</td>
<td>ok</td>
<td>33592130</td>
</tr>
<tr>
<td>1</td>
<td>sojid</td>
<td>2nd Street</td>
<td>null</td>
<td>sojid@kp@<a href="mailto:4@gmail.com">4@gmail.com</a></td>
<td>soj</td>
<td>soj</td>
<td>ok</td>
<td>35700935</td>
</tr>
<tr>
<td>6</td>
<td>siddi</td>
<td>523143</td>
<td>5022991105</td>
<td><a href="mailto:himalshari.narreni@gmail.com">himalshari.narreni@gmail.com</a></td>
<td>siddi</td>
<td>siddi</td>
<td>ok</td>
<td>15155100</td>
</tr>
</tbody>
</table>

* (NULL) (NULL) (NULL) (NULL) (NULL) (NULL) (NULL) (NULL)

Figure 4.1.18 Encrypted latitude and longitude values in database

<table>
<thead>
<tr>
<th>lat</th>
<th>lon</th>
<th>pk</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>75</td>
</tr>
<tr>
<td>(NULL)</td>
<td>(NULL)</td>
<td>0 Kb</td>
</tr>
</tbody>
</table>

35
5. TESTING AND EVALUATION

To discover the errors in the project the designed system is tested. It is a way to check the functionality, integration and outputs of the designed system. There are many kinds of testing and each test will address a specific requirement of the designed system.

5.1 Test Cases:

Test Case-1:

Consider the Alice and Bob scenario where Alice is searching for the gold deposit without a patent on it and Bob is watching her locations so that he can take the patent of the gold deposit before Alice does.

Here Alice registers with the application developer and enters her location as Corpus Christi when she is actually in California as shown in figure 5.1.1.

![Figure 5.1.1 Alice registration page](image-url)
Now the details of Alice are with the application server. Alice enters her initial location as her POI as shown in figure 5.1.2 once activated and a master key is mailed to her.

![Transfer Phase](image)

Figure 5.1.2 Alice selecting her initial location

The admin runs the oblivious transfer phase and generate the secret key which is shown in the figure 5.1.3. Now Alice selects a destination where she wants to go as shown in figure 5.1.4. The initial location is encrypted using AES encryption and the secret key generated as shown in figure 5.1.5.
Figure 5.1.3 Secret key generated for Alice on server side

Figure 5.1.4 Alice selecting final destination
The PIR phase is executed and the route to the location is retrieved from the service provider database which is shown in figure 5.1.6.

Bob will not be able to find what all places Alice is traveling as the location details in the packet transformed to the server are encrypted and only decrypted using the matrix selected. Now if bob tried to hack the application server then all the details about the location of Alice are encrypted and stored in database so he cannot get any information which is shown in figure 5.1.7.
Figure 5.1.6 Alice gets directions from initial point to final point

Figure 5.1.7 Alice location encrypted in the database
**Test Case-2:**

Consider a scenario where the employer was stalking on his employees. Employer of a company has given his employees a mobile phone with GPS. The employer has installed an application on the phone by which he will know where his employees are going and what they are doing. Because of this the employee’s privacy was breached and they had to work for extra hours and had lots of work to do. If an employ goes on a vacation he has to carry his company mobile because of which his personal privacy was at threat.

Now, as the employee has gone to a new place he has to use GPS in order to get direction, but he doesn’t want to do that as his location would be tracked. So, he is registered in the developed application where he will get his direction and also his privacy will not be at risk as the employer will get the location details of the employee what was used to register in the application and not his present location.

The user enters his initial point as shown in figure 5.1.8 and selects the destination where he wants to go as shown in figure 5.1.9. His initial location is encrypted using the AES encryption and the secret key that is generated as shown in the figure 5.1.10. Now PIR phase is executed and directions are shown on maps as shown in the figure 5.1.11.
Figure 5.1.8 Initial point of the user

Figure 5.1.9 User Select the final destination
Figure 5.1.10 Initial point encrypted

Figure 5.1.11 Directions from initial to final destination
6. CONCLUSION AND FUTURE WORK

The scope of the project is to develop a system in which user can determine and acquire the location data without losing his/her privacy. An enhancement to the Ghinita’s system has been made to protect the database of the server, which was the drawback of the system. The user location is safeguarded and the server’s data is also protected using the Private Information Retrieval and Oblivious Transfer protocol.

The drawbacks of the cloaking method, i.e., if the anonymizer or the middle server gets hacked then user details are at risk, this problem is also solved in the proposed system as the locations and time travelled by users are all encrypted in the database. The proposed system is developed as a web application and is tested. The system is tested using the scenarios as shown in the testing section. The main intention is to provide the user with the system which will provide privacy for the user location and secure his details.

In the system the initial location of the user has to be set before going to the directions and a key has to be generated by the administrator. So, in future the selection of the user’s initial point can be made dynamic. The proposed method can be extended to phones and social networks because when a picture is taken in a phone the location where the picture is taken will be saved in the metadata and even it’s same with the social networks as the location from where a status is kept or a picture is uploaded will be saved.

When the proposed system is applied to google or the mobile network providers then the location of the user will not be available for the third party applications. These third party applications get the users location and flood the user with the unwanted
location based advertisements, coupons and a lot of promotional mails which can be spam. When applied to the browsing sites only authorized users with permission to access the details about the user location, interests etc… as the google has a backdoor where anyone can access the user details. This can be prevented using the method proposed.

When implementing on real time applications the details have to be encrypted at the server and only be decrypted at the user. In between if there is any attack the encrypted details can only be visible to the attacker and he will not know which key was used to decrypt the data as a new key will be generated for every request.
BIBLIOGRAPHY AND REFERENCES


[14] P. Tao, L. Qin and W. Guojun, “Enhanced Location Privacy Preserving Scheme in Location-Based Services,” this article has been accepted for inclusion in a future issue of this journal.


[21] F. Mohamed Mokbel, “Privacy- Preserving Location Service,”

[22] R. David, U. Hengartner, “Hiding Location Information from Location-Based Services,”

APPENDIX-A

AES Encryption:

The overall encryption process of AES. The input of AES encryption and decryption is a single 128-bit block of data. This data will be arranged in the form of a matrix of bytes. The matrix will be stored in a state array and the array of data is modified at each stage of encryption and decryption [16]. After the final stage the state array will be the output of the encryption or the decryption process.

The matrix prepared by the AES will be a 4x4 matrix. The bytes of data are arranged in the column of the matrix. So the first 4 bytes of data will be placed in the first column of matrix and next four bytes in the second column and so on. In the same way the key matrix is also prepared using the column arrangement. The data computation consists of “add round key” steps and a final step with byte substitute, shift rows, mixes columns and round key. This process can be said as an alternative process for XOR and scramble data byte operation so all steps can be easily reversed using XOR and table lookup methods.

Each round will take 4x4 matrix as input and will give 4x4 matrix as an output. The output of the final round will be the cipher text of the given data. The output of each round will act as an input to the next round. This process is shown in the figure 3.4.2 for 128bit data. At each stage of Add round key there will be different key which will be used. The key for encryption is used only by the Add round Key.
Now to see in depth about the four transformations used in AES.

1) Substitute Bytes:

This stage uses an S-box to do the byte-by-byte substitution of the block. The S-box is permutation of all the 256 8-bit values constructed using a defined transformation values in $2^8$. This table is fixed for all AES encryption and, for decryption it is just the inverse of the table. This table is designed in such a way that it can resist all the attacks [20]. There is a low correlation between input and output bits because of which the output cannot be determined using the simple mathematical functions as shown in figure 3.4.3.
2) Shift Rows:

The shift row is a permutation of the data. In this step the values are diffused between columns. This mainly performs a circular rotate on each row of 0, 1, 2, 3, places for respective rows. When decrypting, the rows have to be moved in the opposite direction [20]. The row shift moves every individual byte from one column to another column as shown in figure 3.4.5.

3) Mix Columns:
Every column is processed separately and all bytes are replaced by a value which depends on all values in the column or the function of all the four bytes in that column which is shown in figure 3.4.7.

Figure 3.4.7 Mix Column process [20]

The mix column is designed as a multiplication matrix in which every byte is considered as the polynomial in S-box. For decryption to inverse this mix column, a different set of constants is used. These constants are calculated from the linear code having the maximal distance between code words.

4) Add Round Key:

This is a simple bitwise XOR of the current block with a portion of the expandable key. This is the only step which utilizes the key and gives results. It has to be the method to start and end of each round which is shown in figure 3.4.9.
Figure 3.4.9 Add Round key process [20]
APPENDIX-B

Code for generating the secret key

```java
package matrix;
import java.util.Random;
import java.util.Vector;
import java.text.DecimalFormat;

public class Matrix {

    Random r = new Random();
    double[][] randomIntA(int m, int n, int max, int min) {
        double[][] A = new double[m][n];
        for (int i = 0; i < A.length; i++)
        {
            for (int j = 0; j < A[i].length; j++)
            {
                A[i][j] = r.nextInt(max - min + 1) + min;
            }
        }
        return A;
    }

    double[][] randomIntB(int m, int n, int max, int min) {
        double[][] B = new double[m][n];
        for (int i = 0; i < B.length; i++)
        {
            for (int j = 0; j < B[i].length; j++)
            {
                B[i][j] = r.nextInt(max - min + 1) + min;
            }
        }
        return B;
    }
}
```
public double[][] inverz_matrice(double[][] in) {
    int st_vrs = in.length, st_stolp = in[0].length;
    double[][] out = new double[st_vrs][st_stolp];
    double[][] old = new double[st_vrs][st_stolp * 2];
    double[][] new1 = new double[st_vrs][st_stolp * 2];

    for (int v = 0; v < st_vrs; v++) {
        for (int s = 0; s < st_stolp * 2; s++) {
            if (s - v == st_vrs)
                old[v][s] = 1;
            if (s < st_stolp)
                old[v][s] = in[v][s];
        }
    }

    for (int v = 0; v < st_vrs; v++) {
        for (int vl = 0; vl < st_vrs; vl++) {
            for (int s = 0; s < st_stolp * 2; s++) {
                if (v == vl)
                    new1[v][s] = old[v][s] / old[v][v];
                else
                    new1[vl][s] = old[vl][s];
            }
        }
    }
}
old=prepisi(new1);
for (int v1=v1+1;v1<st_vrs;v1++){
    for (int s=0;s<st_stolp*2;s++){
        new1[v1][s]=old[v1][s]-old[v][s]*old[v1][v];
    }
}
old=prepisi(new1);

for (int s=st_stolp-1;s>=0;s--){
    for (int v=s-1;v>=0;v--){
        for (int s1=0;s1<st_stolp*2;s1++){
            new1[v][s1]=old[v][s1]-old[s][s1]*old[v][s];
        }
    }
    old=prepisi(new1);
}
for (int v=0;v<st_vrs;v++){
    for (int s=st_stolp;s<st_stolp*2;s++){
        out[v][s-st_stolp]=new1[v][s];
    }
}
return out;
public static Vector main(int t) {
    String key="";
    Vector v=new Vector();
    double c[][]=new double[t][t];
    Matrix m=new Matrix();
    double a[][]=m.randomIntA(t,t,5,1);
    double b[][]=m.randomIntA(t,t,5,1);
    for (int i = 0; i < a.length; i++)
    {
        for (int j = 0; j < b.length; j++)
        {
            c[i][j]=a[i][j]*b[i][j] ;
        }
    }
    for (int i = 0; i < c.length; i++)
    {
        for (int j = 0; j < c.length; j++)
        {
            System.out.print(c[i][j]+" ");
        }
        System.out.println(\n") ;
    }
}
double inv[][] = m.inverz_matrike(c);

for (int i = 0; i < inv.length; i++)
{
    for (int j = 0; j < inv.length; j++)
    {
        DecimalFormat df2 = new DecimalFormat("###.##");
        System.out.print(df2.format(inv[i][j]) + " ");
        key = key + df2.format(inv[i][j]) + " ";
    }
    System.out.println("\n") ;
    v.add(key);
    key = "";
}

return v;

}

public static void main(String[] args)
{
    Vector vv = null;
    vv = main(5);
    System.out.println("\n"+vv.size()) ;
    for (Object o:vv)
    {
        System.out.println(o);
    }
}
Code for AES Encryption

```java
package crypt;
import javax.swing.*;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Random;

public class AES {
    byte[] skey = new byte[1000];
    String skeyString;
    static byte[] row;
    static byte key[] = null;

    static byte data[] = null;
    String inputMessage, encryptedData, decryptedMessage;

    public AES() {
    }

    public static byte[] generateSymmetricKey() {
        byte[] knumb = null;
        try {
            Random r = new Random();
            int num = r.nextInt(10000);
            String knum = String.valueOf(num);
            knumb = knum.getBytes();
        } catch (Exception e) {
            System.out.println(e);
        }
        return knumb;
    }
}
```
public static byte[] getRawKey(byte[] seed) throws Exception {
    KeyGenerator kgen = KeyGenerator.getInstance("AES");
    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    sr.setSeed(seed);
    kgen.init(128, sr); //
    SecretKey skey = kgen.generateKey();
    raw = skey.getEncoded();
    return raw;
}

public static byte[] encrypt(byte[] raw, byte[] clear) throws Exception {
    SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
    byte[] encrypted = cipher.doFinal(clear);
    return encrypted;
}

public static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception {
    SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");
    Cipher cipher = Cipher.getInstance("AES");
    cipher.init(Cipher.DECRYPT_MODE, skeySpec);
    byte[] decrypted = cipher.doFinal(encrypted);
    return decrypted;
}

public void setKey(byte[] key){
    this.key=key;
    System.out.println("key="+key);
}
public void setKey(byte[] key){
    this.key=key;
    System.out.print("key="+key);
}

public void setData(byte[] data){
    this.data=data;
}

public byte[] getKey(){
    System.out.print("key="+key);
    return key;
}

public byte[] getData(){
    return this.data;
}
APPENDIX-C

Class Diagram:
Clear Class Diagram of Flow: