International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

Preserving Location Privacy in Geo-Social Applications

Sankalp Mehta[1], Assistant Professor
Mayuri Y. Mahekar [2], Jyoti M. Kerur [3], Malashri S. Sangolli [4], Madhavi M. Patil[5]

[1]Associate Professor, Department of Computer Science and Engineering
[2,3,4,5]Students, Department of Computer Science and Engineering
K. L. E. College of Engineering and Technology,
P. B. No. 16 Banantikodi Road Chikodi-591201.
DIST: BELGAUM, STATE: KARNATAKA, COUNTRY: INDIA

Abstract - *Geo-social application deals with constantly sharing user's geographic information on their current whereabouts. For example FourSquare, is used by many people to get information about their surroundings, through their friends and their recommendations. But, without any privacy protection these systems can be easily misused by tracking the users or target them for home invasion.*

LocX is an alternative for such problems which provides significantly-improved location privacy without adding uncertainty into query results or relying on strong assumptions about server security. The key insight is to apply secure user-specific, distance-preserving coordinate transformations to all location data shared with the server. Only the friends of a user share this user's secrets so they can apply the same transformation. Servers can evaluate all location queries correctly, but LocX privacy mechanisms guarantee that servers are unable to see or infer the actual location data from the transformed data or from the data access. LocX privacy mechanism is successful than a powerful adversary model, where prototype measurements used to show that it provides privacy with very little performance overhead, making it suitable for today's mobile devices.

Key Words: **Location privacy. Security, Location based social applications, Location transformation, Efficiency.**

1. INTRODUCTION

Geo-social applications operate on fine-grain, time-stamped location information, so it is also called as Location-Based Social Applications (LBSAs). Geo-social networking is a type of social networking in which geographic services and capabilities such as geo coding and geo tagging are used to enable additional social dynamics. The location data submitted by user can allow social networks to connect and coordinate users with local people or events that match their interests. LBSAs are permission-based applications that use real-time location intelligence from a customer's mobile device.

Unfortunately, there is a significantly increased risk to personal privacy. For current services with minimal privacy mechanisms, this data can be used to infer a user's detailed activities, or to track and predict the user's daily movements. Unauthorized use of location information has been misused for economic gain [6], physical stalking, and to gather legal evidence. Group on and Living Social are some example companies that are leading the thriving business of local activities.

To take care of user's data and location, here designed mechanisms provide efficiently improved user privacy protection without sacrificing the accuracy of the system, or making strong assumptions about the security or trustworthiness of the application servers.

2. PROBLEM STATEMENT

Design and development of a complete geo- social android application that shares one user location or likely location to another user. This architecture uses two servers, one is the proxy server and another is the index server. Proxy server is used to store co-ordinate values in encrypted format. And index server is used to store content values of user. In this proposed system both user have to register first. And only register user can access their notification. User one shares particular longitude value and latitude value to another user. Due to proxy server these values stores in encrypted format. Hence proxy server stores the coordinate values and index server stores data in content format that means whatever the description of the location. For recovery of this encrypted data the secret key is sent on another user's mail id. Because of this second user can easily get location without any threat and without loss of information.

3. OBJECTIVE

Main purpose of this project is to build a system that enables users to query for friends data based on locations, while preserving their location privacy. This system have to support:

- Point query to query for data associated with a particular location.

- Circular range query to query for data associated with all locations in a certain range (around the user).
- Nearest-neighbour query to query for data associated with locations nearest to a given location.

4. EXISTING SYSTEM

Existing systems have mainly taken three approaches to improving user privacy in geo-social systems:

- Introducing uncertainty or error into location data.
- Relying on trusted servers or intermediaries to apply anonymization to user identities and private data.
- Relying on heavy-weight cryptographic or private information retrieval (PIR) techniques.

Limitations of Existing System:

In the first approach, it requires both users and application providers to introduce uncertainty into their data. There is a central tradeoff between the measure of mistake brought into the time or area. space, and the amount of protection allowed to the client. But this way degrades the quality of application results returned to the user. Clients dislike the loss of exactness in results, and application providers have a characteristic disincentive to conceal client information from them, which reduces their capacity to adapt the information.

The second approach depends on the trusted intermediaries (proxies) or servers in the system to ensure client protection. This is a dangerous presumption, since private information can be exposed by either software bugs, configuration errors at the trusted servers or by malicious administrators.

The third approach uses heavy-weight cryptographic mechanisms to obtain provable privacy guarantees. This approach is too expensive to deploy on mobile devices and even on the servers in answering queries such as nearest-neighbor and range queries.

5. PROPOSED SYSTEM

In proposed system, LocX method is used to give significantly high user privacy protection and also maintains the accuracy in LBSAs. User's location coordinates are transformed by using "secrets", which are shared securely among friends only. Coordinate transformations preserve distance metrics, so the application server can perform both point and nearest-neighbor queries correctly on transformed data. Thus the application server is unknown to the exact location of user.

Advantages of Proposed System:

- It gives flexibility to support point, circular range, and nearest-neighbour queries on location data efficiently, suitable for today's mobile devices.
- It provides strong location privacy. The servers processing the data (and the administrators of these servers) should not be able to learn the history of locations that a user has visited.

6. REQUIREMENT ANALYSIS

6.1 USER REQUIREMENTS

The security problem is considered here as the location data is updated every unit of time to the application provider. Users registered to geo-social applications, can be tracked by unauthorized people for dangerous purposes. Thus, protection of information is required against possible violations that can compromise its secrecy (or confidentiality). Secrecy is compromised if information is disclosed to users not authorized to access it.

6.2 SYSTEM REQUIREMENTS:

The system's services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

6.2.1 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software system or its units. A function is described as resource of inputs, its behaviour and outputs. They might be specialized points of interest, information control, estimations and processing and other specific functionality that defines what a system expected to achieve.

Functional requirements of proposed system include following things:

- Strong Location Privacy: The servers processing the data (and the administrators of these servers) should not be able to learn the history of locations that a user has visited.
- Location and user unlinkability: The servers hosting the services should not be able to link if two records belong to the same user, or if a given record belongs to a given user, or if a given record corresponds to a certain real-world location.
- Location data privacy: The servers should not be able to view the content of data stored at a location.
- Flexibility to support point, circular range, and nearest neighbour queries on location data.

6.2.2 NON-FUNCTIONAL REQUIREMENTS:

A non functional requirement specifies criteria that can be used to judge the operation of the system, rather than specific behaviours. It basically defines how the system is supposed to be. These are often called as qualities of the system.

Non functional requirements are not directly concerned with the specific functions delivered by the system. They are related to emergent system properties such as scalability, reliability, modularity, response time and store occupancy. Alternatively, the non functional requirements define constraints on the system such as the capabilities of I/O devices and the data representations used in system interface. The requirements specify or constraint the emergent properties of the system. Therefore, they may

specify system scalability, portability, robustness, security and modularity.

The non functional requirements are:

- Efficiency in terms of computation, bandwidth, and latency, to operate on mobile devices.
- The applications implemented on LocX are lightweight and suitable for running on today's mobile devices by using 2d-Translation, 2d-Scaling and 2d-Rotation algorithm.
- Response time: In proposed system, location information split into two parts and stored on two different servers. The application gives great response time while storing or retrieving the location data from the servers.
- Scalability: The proposed system is scalable for dynamically changing environments with huge data in database. Simulators are capable of simulating the behavior of proposed system, so it is necessary to incorporate in the simulators to get accurate results.

7. BASIC DESIGN

This section clears out the need of LocX system. To support different types of queries on location data (circular range, nearest neighbour) LBSA server need to reveal the location coordinates of user in plaintext. But, malicious server can break the user's location privacy. So, here coordinate transformation method is introduced.

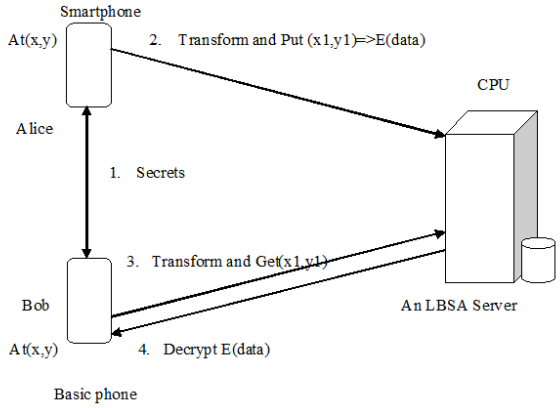

Figure 7.1 Basic Design of LBSA

In Coordinate Transformation, each user 'u' in the system chooses a set of secrets that they reveal only to their friends. These secrets include a rotation angle θ_u, a shift b_u, and a symmetric key sym m_u. The exchange of secrets takes place via a separate trusted channel, such as email, phone etc. The secret angle and shift are used by the users to transform all the location coordinates they share with the servers. Similarly, the secret symmetric key is responsible for encryption all the location data they store on the servers. These secrets are known only to the friends, and hence only the friends can retrieve and decrypt the data.

In Figure 7.1, when a user 'u' wants to store a review 'r' for a restaurant at (x, y), she would use her secrets to transform(x, y) to (x', y') and store encrypted review E(r) on the server. When a friend 'v' wants to retrieve u's review for the restaurant at (x, y), she would again transform (x, y) using u's secret(previously shared with v), retrieve E(r), and then decrypt it using u's symmetric key to obtain 'r'. Similarly, 'v' would transform (x, y) according to each of her friends' secrets, obtain their reviews, and read them. In this design only focus is on point queries.

A Limitation Of Basic Design:

The server can uniquely identify the client devices (for e.g. using the IP address). Using this, the server can associate different transformed coordinates to the same user (using the IP). Sufficient number of such associations can break the transformations.

7.1. INPUT DESIGN

The input design is the link between the information system and the user. This is the method of converting a user-oriented description of the input into a computer-based system. Designing correct input data is an important step to avoid errors in the data input process and this will help for getting correct information from the computerized system. The main goal of input designing is to control the amount of input required, prevent the errors, avoid delay, and avoid extra steps and keep the process as simple as possible. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Often the collection of input data is considered as the most expensive part of the system, but this helps a lot in development of project correctly.

Always, Input Design considers the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Here, Input Design method starts by creating user-friendly screens for the data entry to handle large volume of data. Here the goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that it will help users at a right time. Thus the objective of input design is to create an input layout that is easy to follow.

7.2. OUTPUT DESIGN

The output is said to be quality one when meets the end user requirements and presents the information clearly. Processing results of any system are communicated to the users and to other system through outputs. Output Design is the direct source of information for the user. To improve the system's relationship, efficient and intelligent output design is most important one with source and destination machine and to help user in decision-making.

The process of designing a computer output should be well organized, well mannered. Every output element is designed in such a way that users will find the system can use easily and effectively. A very correct output should be developed for user compatibility. While designing computer output, the idea of specific output that is needed to meet the user requirements, is to be clear in analysis step. The output design selects appropriate methods for presenting information. This helps in creating document, report, or other formats that contain information produced by the system.

Following are the objectives of output form of an information system:

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

8. SYSTEM ARCHITECURE

System architecture is the conceptual design that defines the structure and behavior of the system. An architecture description is a formal description of the system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The system architecture is shown below:

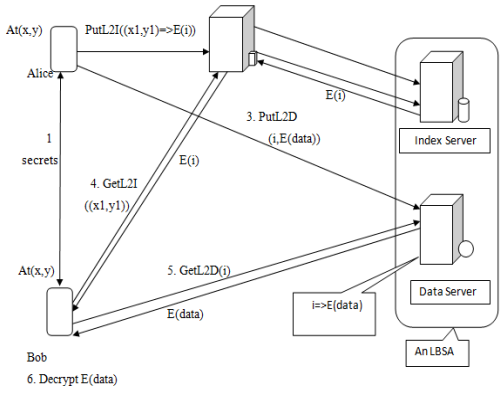

Figure 8.1 : Architecture Diagram.

LocX Mechanism can be described as:

- First, Alice and Bob true friends exchange their secrets via trusted channel,
- Then Alice generates a review for the restaurant (at (x, y)), and stores the L2I on the index server via a proxy.
- Then she stores the I2D on the data server directly.
- Bob later visits the restaurant and fetches for L2Isfrom his friends by sending the transformed coordinates via a proxy.
- He decrypts the L2I obtained and then queries for the corresponding I2D,

Finally Bob decrypts Alice's review using shared secrets.

9. DETAILED DESIGN

In system design, architecture diagram and mechanisms of LocX have explained. In this detailed design section, main modules of LocX technique, algorithms used for encryption and decryption, formula used to process nearest-neighbour queries these details are discussed briefly.

9.1 SYSTEM MODULES

To overcome from the limitations of basic design two new mechanisms are introduced:

First, LocX method splits the mapping between the location and its data into two pairs: a mapping from the transformed location to an encrypted index (called L2I), and a mapping from the index to the encrypted location data (called I2D). This splitting causes to make the system efficient.

Second, users store and retrieve the L2Is via untrusted proxies. This redirection of data via proxies, together with splitting, significantly improves privacy in LocX. For efficiency, I2Ds are not proxied, yet privacy is preserved.

9.1.1. Location Transformation

1. Decoupling a location from its data

In today's systems, location data data(x,y) corresponding to the real-world location(x, y) is stored directly under (x, y) on the server. But in LocX, the location (x, y) is first transformed to (x', y'), and the location data is encrypted into E(data(x,y)). Then the transformed location is decoupled from the encrypted data using a random index 'i' via two servers as follows:

- L2I = [(x', y'),E(i)],which stores E(i) under the location coordinate (x', y').
- I2D = [i,E(data(x,y))], which stores the encrypted location data E(data(x,y)) under the random index i.

User's secret random number generator generates the index. The server which stores L2Is is referred as the index server and the server which stores I2D is referred as the data server. They can be on the same server, privacy properties still hold. This separation of location information into two components (L2Iand I2D) helps to efficiently run different types of location queries on L2Is and retrieve only relevant I2Ds.The key interfaces used by the applications to store and retrieve data on the LocX servers are listed in API's and their functions in LocX.

2. Proxying L2Is for Location Privacy

Users store their L2Is on the index server via untrusted proxies. These proxies can be any of the following: PlanetLab nodes, corporate NATs and email servers in a user's work places, a user's home and office desktops or laptops, or Tor nodes. The only need is there should be one-hop indirection between the user and the index server. A user can store her L2Is via different types of proxies without restricting herself to a single proxy, thus it provides tremendous flexibility in proxying L2Is. Furthermore, compromising these proxies by an attacker does not break users' location privacy, as

- The proxies also only see transformed location coordinates and hence do not learn the users' real locations.
- Due to the noise added toL2Is.

To simplify the description, for now, assume that the proxies are non-malicious and do not collude with the index server.

9.1.2. Data Storage

1. Storing L2I on the Index Server

In this transformation distances between points are preserved, so circular range and nearest- neighbour queries for a friend's location data can be processed in the same way on transformed coordinates as on real-world coordinates. Then the user generates a random index 'i', using her random number generator and encrypts it with her symmetric key to obtain at the transformed coordinate on the index server via a proxy. The L2I is small in size and is application independent, as it always contains the coordinates and an encrypted random index. Thus the overhead due to proxying is very small.

2. Storing I2Ds on the Data Server

The user can directly storeI2Ds (location data) on the data server. This is both secure and efficient.

- This is secure because the data server only sees the index stored by the user and the corresponding encrypted blob of data. In the worst case, the data server can link all the different indices to the same user device, and then link these indices to the retrieving user's device. But this only reveals that one user is interested in another user's data, but not any information about the location of the users, or the content of the I2Ds, or the real-world sites to which the data in the encrypted blob corresponds to.
- The content of I2Dis application dependent. For example, a location-based video or photo sharing service might share multiple MBs of data at each location. Since this data is not proxied, LocX still maintains the efficiency of today's systems.

9.1.3.Data Retrieval

1. Privacy while querying the index server

To prevent attacks while querying the index server, users add noise to the query. This Noise is a few (N) additional, randomly selected points, $[(x1'_1, y1'_1),(x1'_2, y1'_2)...(x1'_N, y1'_N)]$, added to each query sent to the index server. By adding noise, server will not be able to derive the real location of the user. In addition, the user can easily filter out the L2Is corresponding to the noise. So that accuracy is still maintained.

Adding noise and coupled with routing the index server queries via proxies, provides strong location privacy during querying. The queries only contain a list of points in the transformed coordinate space, without any user identifier or actual location information. Due to proxying, the server cannot identify the client. And finally the noise prevents derivation of user's location based on transformed coordinate. Putting noise and proxying schemes together, the server cannot link multiple different queries to the same user.

2. Querying the data server and decrypting location data

After obtaining the L2Is from the index server corresponding to a point (x', y'), which is transformed with friend u's secrets, the client user identifies the L2Is using this secret, and then encrypts the returned L2Is with u's symmetric key. Then the user directly queries the data server for the I2Dscorresponding to all the decrypted indices (i_1, i_2 ...). She then obtains the I2Ds from the data server, decrypts them using the symmetric key of the friend whose key was used to decrypt the corresponding index. And then the user consumes the data as per the application. There is no need for a proxy in this step as the index and the encrypted data on the data server cannot link a user to her location. Since the decrypted index is sent to the data server, it cannot even be linked to an encrypted index on the index server. The "Rijndael AES algorithm -128 bits" is used here for encryption and description.

9.1.4. Query Processing

Supporting circular range and nearest-neighbour queries. Earlier approaches only support point queries, where a user fetches data at a given location coordinate. To support more complex queries like circular range and nearest-neighbour queries, change necessary is for the index server to return data around a query point instead of returning data at a query point. Since the location transformation is distance preserving, using 'Haversine Formula' on the L2Is input by the users can support both circular range and nearest-neighbour queries. Finally, the user should mention the type of the query she wants to run, and also specify a query range along with the query while querying the index server. The rest of the steps in querying remain the same.

10. IMPLEMENTATION

Main Modules: There are four main modules

1. LocX module
2. Proxy server
3. Index server
4. Data Server

LocX Module:

LocX builds on top of the basic design, and introduces two new mechanisms to overcome its limitations. First, in LocX, we split the mapping between the location and its data into two pairs: a mapping from the transformed location to an encrypted index (calledL2I), and a mapping from the index to the encrypted location data (called I2D). This splitting helps in making our system efficient. Second, users store and retrieve the L2Is via untrusted proxies. This redirection of data via proxies, together with splitting, significantly improves privacy in LocX. For efficiency, I2Ds are not proxied, yet privacy is preserved.

Proxy Server:

Users store their L2Ison the index server via untrusted proxies. These proxies can be any of the following: Planet Lab nodes, corporate NAT sand email servers in a user's work places, a user's home and office desktops or laptops, or Tor [34] nodes. We only need a one-hop indirection between the user and the index server. These diverse types of proxies provide tremendous flexibility in proxying L2Is, thus a user can store her L2Is via different proxies without restricting herself to a single proxy. Furthermore, compromising these proxies by an attacker does not break users' location privacy, as

- The proxies also only see transformed location coordinates and hence do not learn the users' real locations.
- Due to the noise added to L2Is.

To simplify the description, for now, we assume that the proxies are non-malicious and do not collude with the index server. But we will later describe our solution in detail to even defend against colluding, malicious proxies. With this high-level overview, we now describe our solution to store and query data on the servers in detail. We also explain the challenges we faced, and the tradeoffs we made in making our solution secure and efficient.

Index Server:

First consider storing L2I on the index server. This transformation preserves the distances between points1, so circular range and nearest neighbour queries for a friend's location data can be processed in the same way on transformed coordinates as on real-world coordinates. Then the user generates a random index using her random number generator and encrypts it with her symmetric key to obtain at the transformed coordinate on the index server via a proxy. The L2I is small in size and is application independent, as it always contains the coordinates and an encrypted random index. Thus the over head due to proxying is very small.

Data Server:

The user can directly storeI2Ds (location data) on the data server. This is both secure and efficient.

- This is secure because the data server only sees the index stored by the user and the corresponding encrypted blob of data. In the worst case, the data server can link all the different indices to the same user device, and then link these indices to the retrieving user's device. But this only reveals that one user is interested in another user's data, but not any information about the location of the users, or the content of the I2Ds, or the real-world sites to which the data in the encrypted blob corresponds to.
- The content of I2Dis application dependent. For example, a location-based video or photo sharing service might share multiple MBs of data at each location. Since this data is not proxied, LocX still maintains the efficiency of today's systems.

11. APPLICATION

In this paper we implement LocX in Java. We used AES with 128 bits keys for encryption and decryption. We measured LocX's performance on both desktops and on android mobile phones. The index and data servers were run on the same Dell Power Edge server. Clients were run on another machine with the same configuration. We used the same code base for both desktop and mobile tests. But we had to modify the code slightly for Android OS to deal with some missing libraries. In addition, we had to make certain optimizations to limit the memory usage to under 16 MBs for LocX process in Android. Here we sketch how to build LBSAs using LocX.

We demonstrate the usage of our APIs by building three applications. In today's systems that provide these services, the data are entrusted to the server in plain text, which performing the computations in the application logic. But since we do not trust the server in LocX, the application logic that computes on the plain-text location data is moved to the client. Location-based reminders. This application users place.

12. RESULTING APPLICATION

An android application called LocX is created, which is geo-social application. This application gives idea about nearby places from current location, based on recommendations given by friends. As numbers of users are using this application they require their respective secret key. The particular secret key is generated by using random key generator algorithm. Users should first register themselves in the network for accessing LocX application. Application provides various functions to make the system user friendly like:

- Finding location: User can see the location on google maps and the data is stored on that particular location with the help of transformed coordinates.
- Location update: User can update their respective location and all other remaining users will be notified about it. Providing all privacy aspects user can retrieve the data by using the decrypted index.

Location Recommendations: Users can give their recommendations which will get stored at respective

location. All other users can see recommendations if they are authorized.

13. RESULT ANALYSIS

Threats of existing system

Location-based applications promise safety and convenience, they threats the privacy and security of their customers. In order to get a location-based service, a user has to report her private location information to the server. With untrustworthy servers, such model provides several privacy threats. For example, an employer may check on her employee behaviour by knowing the places she visits or the personal medical records can be inferred by knowing which clinic a person visits. Hence the main threat is privacy and security of information.

Proposed System

Our existing system one user can share his location to the another user by using only one server. This system is no up to the mark secure. So now we proposing new architecture a new method like one user can share his details like whatever his location details to another user by using two servers. These are proxy server and index server. Proxy server is used to data maintaining in coordinate values in encrypted format. And index server is used to store actual content values of users. Whatever methodologies we used is two users are registered and two servers are used. Due to use of encrypted format data or whatever longitude or latitude values are encrypted and stores in the proxy server. Because of this encryption method in proxy server data is keep at high security. And this data is send to index server at this stage secret key is forwarded to another users id. Hence another user can get actual content by decryption process. That monitors the location, movement.

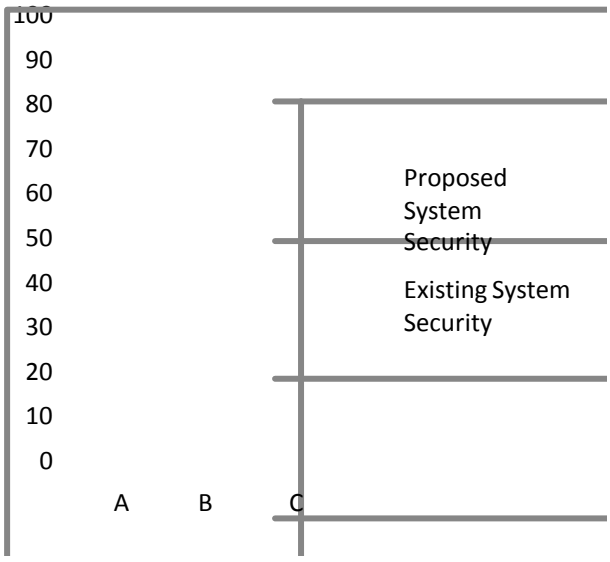

Figure 8.1 Graphical Representation of Security

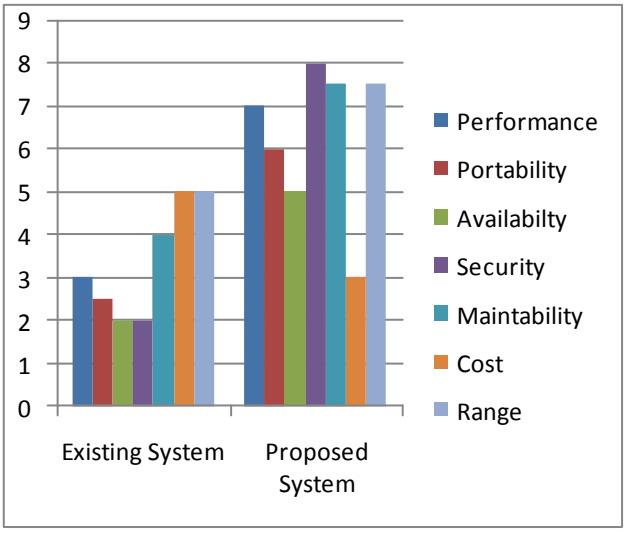

Figure 8.2 Graphical Representation Based On Various Parameters

Figure 8.2 shows various parameters such as,

- Performance: The performance of proposed system is higher than the existing system.
- Portability: In the proposed system the authorised person can carry android application so that he can get sufficient security of his location as introduced in the system.
- Availability: Due to the portability of the security of the proposed system, the registered person can get information any time.
- Security: The main purpose of proposed system is to provide more security. It is much higher than the existing system.
- Maintainability: The system should be optimized for supportability or ease of maintenance as far as possible.
- Cost and Range: Range can find anywhere and it is independent of cost.

FUTURE WORK

The project ensuring distributed data sharing and security in android. After uploading data on database this project will maintain all the records about user who have used the data. Also bundling of the file with its information and accessing that data or location by getting that particular key & through that we can preserve our location is the scope of the system.

CONCLUSION

In LocX, users efficiently transform all their locations shared with the server and encrypt all location data stored on the server using inexpensive symmetric keys. Only friends with the right keys can query and decrypt a user's data. LocX is responsible for some amount of computational and communication overhead to the existing system. But otherwise, LocX prototype runs efficiently even on resource constrained mobile phones. Overall, LocX is a big step

toward making location privacy practical for a large class of emerging geo-social applications.

ACKNOWLEDGEMENT

We are thankful to **Prof. Sankalp Mehta** Asst. Professor, Dept. of Computer Science & Engineering who gave unending support right from the stage. We also thank **Prof. Chetan Bulla** Asst. Professor, Dept. of Computer Science & Engineering who has played an important role in carrying this work.

REFERENCES

[1] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, lester and pierre: Three protocols for location privacy," in Proc. of PET, 2007.

[2] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbour querie using space transformation to preserve location privacy," in Proc. of SSTD, 2007.

[3] J.Manweiler, R.Scudellari, L.P.cox, "SMILE:Encounter-Based Trust forMobile Socia Services", Proc. 16th ACMConf.Computer Comm.Security (CCS '09),2009.

[4] Krishna P. N. Puttaswamy, Shiyuan Wang, Troy Steinbauer, Divyakant Agrawal, Amr El Abbadi, Christopher Kruegel and Ben Y. Zhao, "Preserving Location Privacy In Geo-Social Applications", Department of Computer Science, UC Santa Barbara, 2014.

AUTHORS

Prof. Sankalp Mehta,
Associate Professor
Dept. of Computer Science & Engineering,
KLE Society's KLE College of Engineering & Technology Chikodi.

Mayuri Y. Mahekar,
Student of Computer Science & Engineering,
bearing USN: 2KD12CS017
KLE Society's KLE College of Engineering & Technology Chikodi.

Jyoti M. Kerur,
Student of Computer Science & Engineering,
bearing USN: 2KD12CS013
KLE Society's KLE College of Engineering & Technology Chikodi.

Malashri Sangolli,
Student of Computer Science & Engineering,
bearing USN: 2KD12CS015
KLE Society's KLE College of Engineering & Technology Chikodi.

Madhavi Patil,
Student of Computer Science & Engineering,
bearing USN: 2KD12CS014
KLE Society's KLE College of Engineering & Technology Chikodi.