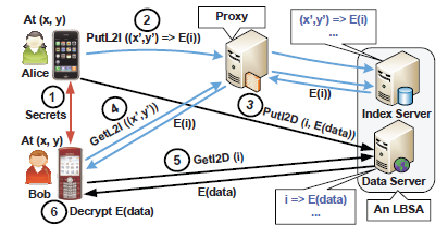
Preserving Location Privacy in Geo-Social

Applications

**Abstract**:

Using geo-social applications, such as FourSquare, millions of people interact with their surroundings through their friends and their recommendations. Without adequate privacy protection, however, these systems can be easily misused, *e.g.,* to track users target them for home invasion. In this paper, we introduce *LocX*, a novel alternative that provides significantly-improved locationprivacy without adding uncertainty into query results or relying on strong assumptions about server security. Our key insight is to applysecure user-specific, distance-preserving *coordinate transformations* to all location data shared with the server. The friends of a usershare this user’s secrets so they can apply the same transformation. This allows all location queries to be evaluated correctly by theserver, but our privacy mechanisms guarantee that servers are unable to see or infer the actual location data from the transformeddata or from the data access. We show thatLocX provides privacy even against a powerful adversary model, and we use prototypemeasurements to show that it provides privacy with very little performance overhead, making it suitable for today’s mobile devices.

**Architecture Diagram:**



**Existing System:**

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

(a) introducinguncertainty or error into location data .

(b) relying on trusted servers or intermediaries to apply anonymization to user identities and private data.

(c) relying on heavy-weight cryptographic or private information retrieval (PIR) techniques.

The challenge, then, is to design mechanisms that efficiently protect user privacy without sacrificing the accuracy of the system, or making strong assumptions about the security or trust worthiness of the application servers. More specifically, we target geo-social applications, and assume that servers (and any intermediaries) can be compromised and, therefore, are untrusted.

**Proposed System:**

To address this challenge, in this paper, we propose *LocX* (short for location to index mapping), a novel approach to achieving user privacy while maintaining full accuracy in location-based social applications (LBSAs from here onwards).

Our insight is that many services do not need to resolve distance-based queries between arbitrary pairs of users, but only between friends interested in each other’s locations and data. Thus, we can partition location data based on users’ social groups, and then perform *transformations* on the location coordinates before storing them on un trusted servers. A user knows the transformation keys of all her friends, allowing her to transform her query into the virtual coordinate system that her friends use. Our coordinate transformations preserve distance metrics, allowing an application server to perform both point and nearest-neighbor queries correctly on transformed data. However, the transformation is *secure*, in that transformed values cannot be easily associated with real world locations without a *secret*, which is only available to the members of the social group. Finally, transformations are efficient, in that they incur minimal overhead on the LBSAs. This makes the applications built on LocX lightweight and suitable for running on today’s mobile devices.

**Main Modules:**

**1.Locx module**

**2.proxy server**

**3.index server**

**4.Data Server**

**LOCX Module:**

Loc X builds on top of the basic design, and introduces two new mechanisms to overcome its limitations. First, in Loc X, we split 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 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 (as explained later).

**Proxying L2Is for location privacy:**

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 (a) the proxies also only see transformed location coordinates and hence do not learn the users’ real locations, and (b) due to the noise added toL2Is (described later). 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.

**Storing L2I on the index server:**

First consider storing L2I on the index server. This transformation preserves the distances between points1, so circular range and nearest neighbor 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 over head due to proxying is very small.

**Storing I2Ds on the data server:**

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

*1)* 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.

*2)* 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.

**Mechanisms:**

In this we use Locx Mechanisms is used in this project.

1) Alice and Bob exchange their secrets,

2) Alice generates and L2I and I2D from her review of the restaurant (at (x, y)), and stores the L2I on the index server via a proxy.

3) She then stores the I2D on the data server directly.

4) Bob later visits the restaurant and fetches for L2Is from his friends by sending the transformed coordinates via a proxy.

5) he decrypts the L2I obtained and then queries for the corresponding I2D, 6) finally Bob decrypts Alice’s review.

# System Configuration:-

# H/W System Configuration:-

# Processor - Pentium –III

**Speed - 1.1 Ghz**

**RAM - 256 MB(min)**

**Hard Disk - 20 GB**

**Floppy Drive - 1.44 MB**

**Key Board - Standard Windows Keyboard**

**Mouse - Two or Three Button Mouse**

**Monitor - SVGA**

# S/W System Configuration:-

* Operating System :Windows95/98/2000/XP
* Front End : java, jdk1.6
* Database : My sqlserver 2005
* Database Connectivity : JDBC.