Rochester Institute of Technology

Efficient Location Privacy

Submitted By: FNU Kanchan
MS in Computer Science

Submitted To: Dr. Piezhao Hu
Motivation:

Today, there are tons of social media applications, many of which provide the users with attractive features such as: nearby friends, profile matching within the locality, profile display of the people who cross path a user etc. These applications seem interesting to users but if seen from the security perspective, these seem pretty intrusive. It is because these applications use GPS coordinates to get the location of the user. Releasing location data can lead to security concerns such as location tracking by service providers. This data stored in the servers may lead to privacy threats including intentional and unintentional insider attacks, e.g. employees of a company exploiting private data.
The project, Efficient Location Privacy, aims to develop an efficient location masking technique that offers both the privacy and usability of location data. The approach used is the spatial cloaking approach. The idea behind this approach is: instead of sharing the exact GPS location of the user, an area in the form of a geometrical figure is shared which encloses the user within it. There are many algorithms and approaches to secure the privacy of the user. This project covers some new approaches and interesting ideas which provide not only more privacy than the current technique, but also the usability to the user. It also introduces the novel concept of giving the user some control to its privacy.

For Privacy: Reveal the box, not the location point

For Usability: Allow users to control the level of details (LoD)

Background:

The technique followed in the project for Geohashing is called Z order Geohashing. This technique is similar to the one used in the Bing Map Tile system. It which maps two dimensional coordinates into a binary string.

The figure below explains the Z order system:
This mapping preserves locality and hence, the proximity of the users. The closer the two users, the smaller the enclosing bounding box (shown in the diagram below)
**Trusted Server Technique:**

- Common solution for location privacy uses spatial cloaking
- GPS location data is sent to a trusted server.
- Server operates prefix matching to compute bounding box to be sent back as result. This box encloses all the communicating users.

**Architecture:**

![Diagram showing Trusted Server Technique]

- GPS Coord -> Trusted Server
- Bounding Box -> Trusted Server
- Bounding Box -> Trusted Server
- Trusted Server -> U1
- Trusted Server -> U2
Example:

Problem in Current Approach:

With the increase in the size of the bounding box, the error increases exponentially. Error is significant, hence, contains lot of noise.
Versions of the Improved GeoHashing Technique:

During the explanations, we assume that the two users participate in the communication process.

For examples and architecture,

**BK**: Binary Key (e.g., 1010110), **LoD**: Level of details represented as binary mask (e.g., 111100)

1) Improved Trusted Server Version:

- The GPS coordinates are encoded as quad key and sent to the server by both the users along with the level of details the users want to have.
- The server performs the rotation operations to the quad keys got from both the users.
- The server follows some heuristics along with the prefix matching based on the common level of detail or relationship between the two users, to compute the improved bounding box to be sent back to the user.
- If any third party asks for the location data from the user, user can follow the above technique and hands over the task to the trusted server. The trusted server then performs the encryption operations and sends the bounding box results to the asking party. The third party can then perform its normal feature operations by communicating to the user.
Architecture:
Roles and Responsibilities:

• **Server Roles:**
  1. Prefix Matching
  2. Identifying the level of privacy of both the parties and choose the optimal one
  3. Computing the results based on the relationship status of the two parties
  4. Computing the efficient bounding box
  5. Refining the bounding box
  6. Encrypting the results/bounding box
  7. Sending the encrypted results to either third party server or to the clients

• **Client Roles:**
  1. Encoding of GPS coordinates to binaryKey/quadKey
  2. To assign the level of privacy based on the relationships
  3. Decoding the encrypted bounding box results received from the server
Examples:

The red bounding box shows the output box with the current Trusted Server approach and the blue box shows the result from the improved Trusted Server Approach.

**Server View**

**Client view with corresponding Server View:**
Server View for the above Clients

**Issues:**

- Need to trust the server
- The results are improved and has less noise than the current system, but they are not accurate enough
- Some heuristics may need us to send multiple bounding boxes, which not increases communication traffic.
- More the output boxes, harder the encryption.

2) **Two-Party Computation version:**

- Infrastructureless & ad hoc approach
- Masked bounding box or coordinates are exchanged or shared among communicating parties without any server or third party intervention
- **Issues:** scalability
Exchange of Location data

**P1:** 101101010
**U1 Mask:** 111110000

**U1:** 10110

**U2 Mask:** 11110000

**U2:** 1011

Exchange of Location data

Bounding box

**P1:** 101101010
**U1 Mask:** 111110000

**U1:** 10110

Request of Location data

**U1 requests for bounding box**

**U1 shares masked encoded location data**

**U2 computes the box and sends back**

**P1:** 101101010

**U1 Mask:** 111110000

**U1:** 10110

Request of Location data
Examples:

Client View

Server View for the above Clients
3) Location Privacy with Control Version

- Location encoded as binary key is sent to trusted server
- Level of detail is only shared initially or on user’s request
- If there is a third party request for a user’s details, encoded location of the user, masked to the level of detail, is sent back
- Less noise than the basic GeoHashing technique
- **Issues:** trusted server required

![Diagram showing the interaction between U1, U2, and the Trusted Server]

- LOD is shared
  - initially
  - up on user’s request
4) Homomorphic Encryption (HE) without Trusted Server Version:

- Very efficient and secure approach in terms of HE
- Level of detail is only shared initially or on user’s request
- Encoded location data is encrypted and shared with server
- The encrypted binary key is masked with the level of detail and sent to the user interested or exchanged between two users
- The result is decrypted on the user’s side
Results Comparison:

<table>
<thead>
<tr>
<th>Two Party Comp. Version</th>
<th>User1</th>
<th>User2</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.jpg" alt="Map" /></td>
<td><img src="image2.jpg" alt="Map" /></td>
<td><strong>N/A No Server</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trusted Server Version</th>
<th><img src="image3.jpg" alt="Map" /></th>
<th><img src="image4.jpg" alt="Map" /></th>
<th><img src="image5.jpg" alt="Map" /></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Loc. Privacy w Control Version</th>
<th><img src="image6.jpg" alt="Map" /></th>
<th><img src="image7.jpg" alt="Map" /></th>
<th><img src="image8.jpg" alt="Map" /></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>HE w/o Trusted Server Version</th>
<th><img src="image9.jpg" alt="Map" /></th>
<th><img src="image10.jpg" alt="Map" /></th>
<th><img src="image11.jpg" alt="Map" /></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>N/A Server View Not Possible</strong></th>
<th><strong>N/A No Server</strong></th>
<th><strong>N/A Server View Not Possible</strong></th>
</tr>
</thead>
</table>
Application View:
Future Work:

- Incorporate the new improved GeoHashing techniques into Homomorphic Encryption. Location data will be encrypted, thus don’t need a trusted server
- Prepare a technical paper for publication

References:
