Location Oblivious Privacy Protection for Group Nearest Neighbor Queries

A. K. M. Mustafizur Rahman Khan
Tanzima Hashem
Egemen Tanin
Lars Kulik
Introduction

- Increased popularity of Location Based Services (LBS).
- The users risk their privacy by exposing locations while receiving such services.
- Demand of group oriented LBSs, such as, group nearest neighbor query.
A group nearest neighbor (GNN) query returns the location of a point of interest (POI) that minimizes the aggregate distance from a spread out group of users.

In privacy preserving GNN queries users’ locations are not revealed.
Motivation

- Most of the techniques in privacy are tailored for an individual user.
- Obfuscation based techniques such as spatial cloaking or providing imprecise location do not provide enough privacy.
- Cryptographic techniques are usually not implementable on mobile phones with limited computation resources.

Develop technique for privacy preserving GNN query that does not use obfuscation and cryptography.
Challenges

- Preserving a user’s privacy from other group members.
- Necessary information are distributed among multiple users and these information must be aggregated/used in a privacy preserving manner.

Advantages

- Services from an LSP are demanded as a group using aggregated information. Thus individual location information can be kept hidden from an LSP.
Work of Hashem et al.

- Users send their imprecise locations anonymously to the location service provider (LSP).
- The LSP returns a set of candidate POIs.
- Each candidate's aggregate distance is computed with "private filter" algorithm.
- The candidate with smallest aggregate distance is returned as the GNN.

- Attackers may guess users' exact locations using background information.
- Ashouri-Talouki et al. showed that "private filter" algorithm is vulnerable to partial collusion attacks.
Work of Huang et al.

- Treats the problem as a Secure Function Evaluation problem.
- Uses Garbled Circuit as a solution.

- Computationally extremely expensive. Experiment with 10 users and 2000 POIs took 4 seconds for each user in a distributed model and 20 seconds in a centralized model. Smart phones are on average 1000 times slower than desktop computers.*

- High communication cost. The number of Oblivious Transfers performed by each user is equal to the number of POIs.

Work of Ashouri-Talouki et al.

- Uses cryptographic approach to calculate centroid of the users without revealing their coordinates.
- Returns the nearest point to the centroid as the GNN.

Ar + Br < Aq + Bq

Answers are approximation.
Framework of Our Proposed Methods

- Privacy preserving protocols to aggregate information:
  - PAD: Calculates aggregate distance
  - CWMD: Compare aggregate distances

- Searching algorithms:
  - LOOP: Basic algorithm
  - H-LOOP: Advanced algorithm
Private Aggregate Distance (PAD) Protocol

- Compute aggregate distance without disclosing individual distance.
- Use distributed secure multi-party computation protocol*.
- Does not use cryptography: Computationally easy.
- A user’s individual distance remains private if there is at least one honest user other than the target.

An Example of PAD

- Suppose there are four users A, B, C and D and a coordinator X.
- A coordinator can be a third party or one of the users.
An Example of PAD

- Suppose there are four users A, B, C and D and a coordinator X.
- A coordinator can be a third party or one of the users.
- Each user has a secret number.
- The users want to calculate the total of these numbers.
An Example of PAD

- Each user divides the secret number randomly and send to other users.
An Example of PAD

- Each user divides the secret number randomly and send to other users.
An Example of PAD

- Each user divides the secret number randomly and send to other users.
An Example of PAD

- Each user divides the secret number randomly and send to other users.
An Example of PAD

- Each user updates her number by deducting sent numbers and adding received numbers.
An Example of PAD

- Each user reports the updated number to the coordinator.
- The coordinator publishes the total.

Total: 54
Comparison With Minimum Disclosure (CWMD) Protocol

Find the POI that has the smallest aggregate distance, when the set of POI is distributed to multiple users.

\[ S_1 = \{ P_{11}, P_{12}, \ldots \} \]
\[ S_j = \{ P_{j1}, P_{j2}, \ldots \} \]
\[ S_n = \{ P_{n1}, P_{n2}, \ldots \} \]

Step 1

Find the POI that has the smallest aggregate distance, when the set of POI is distributed to multiple users.
LOOP: Location Oblivious Private Algorithm

LOOP does not require a user to provide imprecise location and thus is location oblivious.
Limitations of LOOP

• An LSP needs to disclose all POI’s locations to the user.
• The information of aggregate distances are leaked:
  Cannot resist MPAD attack from users.
  (MPAD attack is a mass target attack, where an attacker attacks the group collectively and does not distinguish between individual users.)
H-LOOP: Heuristic LOOP

- Four users A, B, C and D.
- * represents POIs.
- Suppose division threshold = 4.
H-LOOP: Heuristic LOOP

- Number of POIs in the search space is larger than the division threshold (4).
- Divide the search space into non-overlapping equal-sized squares.
- Calculate aggregate minimum distance of each square.
- Select the square with the smallest aggregate minimum distance as new search space.
H-LOOP: Heuristic LOOP

- Number of POIs in the search space is larger than the division threshold (4).
- Divide the search space into non-overlapping equal sized squares.
- Calculate aggregate minimum distance of each square using PAD.
- Select the square with the smallest aggregate minimum distance as new search space.
H-LOOP: Heuristic LOOP

- Number of POIs in the search space is **smaller** than the division threshold (4).
- Calculate aggregate distance of each POI using PAD.
- Find the POI with the smallest AD using CWMD.
- Publish the POI as the GNN.
Experimental Setup

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>User distribution</td>
<td>Uniform, Zipfian</td>
<td>Uniform</td>
</tr>
<tr>
<td>POI distribution</td>
<td>Uniform, Zipfian</td>
<td>Uniform</td>
</tr>
<tr>
<td>Group size</td>
<td>3, 5, 10, 15, 20</td>
<td>10</td>
</tr>
<tr>
<td>Number of POIs</td>
<td>2K, 5K, 10K, 20K</td>
<td>2K</td>
</tr>
<tr>
<td>Number of attackers</td>
<td>1,2,3,4,5,6,7,8</td>
<td>1</td>
</tr>
</tbody>
</table>

- We also used parks and flats in California as locations of POIs and users.
- Considered 10,000 private GNN queries for each set of experiments.
- Used a desktop computer with Intel(R) Core(TM) i7-2600 3.40GHz processor and 8GB RAM.
- Measured computation time, error rate and error size.
  - Error rate is the percentage of wrong answers
  - Error size = 1 - Aggregate distance of answer/ Aggregate distance of true GNN.
Both error rate and error size of H-LOOP are at least three orders of magnitude smaller than those of Talouki et al.’s method.
Cumulative Error Probability

Group size: 10
Number of POI: 2000
Number of colluder: 1

Cumulative error probability of H-LOOP is at least three orders of Magnitude smaller than that of Talouki et al.’s method.
Effect of Location Distribution on Computation Time

<table>
<thead>
<tr>
<th>User distribution</th>
<th>POI distribution</th>
<th>LOOP</th>
<th>H-LOOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>Uniform</td>
<td>0.76</td>
<td>0.53</td>
</tr>
<tr>
<td>Zipfian</td>
<td>Uniform</td>
<td>0.79</td>
<td>0.54</td>
</tr>
<tr>
<td>Uniform</td>
<td>Zipfian</td>
<td>0.75</td>
<td>0.68</td>
</tr>
<tr>
<td>Zipfian</td>
<td>Zipfian</td>
<td>0.74</td>
<td>0.70</td>
</tr>
</tbody>
</table>

In all cases our proposed methods took less than 1 ms, whereas Hung et al reported 2s as computation time in compatible experiments.
Effect of the Number of POIs on Computation Time

User distribution: Uniform
Group size: 10

POI distribution: Uniform
Number of colluder: 1

Computation time is logarithmic to the number of POIs.
**Effect of the Number of POIs on Accuracy**

Mean error size did not show any trend and $1.37E-4\%$ was the largest mean error size.

User distribution: Uniform  
Group size: 10  
POI distribution: Uniform  
Number of colluder: 1

Although error rate increases linearly with the number of POIs, both error rate and error size remains small.
Both error rate and error size decreases sharply with the increase of group size.
Effect of Colluder Group Size

Although both error rate and error size increases with the increase of number of colluders, they remain small.

User distribution: Uniform
Group size: 10

POI distribution: Uniform
Number of POI: 2K
Conclusion

- We proposed a heuristic privacy preserving GNN query algorithm.
- Our method does not require a user to provide any location information and thus location oblivious.
- Our method is strong against attacks, highly accurate and fast enough to be deployed in real world application.