Increasing numbers of people are subscribing to location-based services, and as the popularity grows so do the privacy concerns. Varieties of research exist to address these privacy concerns. Each technique tries to address different models with which location-based services respond to subscribers. In this work, we present ideas to address privacy concerns for the two main models, the snapshot nearest neighbor query model and the continuous nearest neighbor query model.

First, we address snapshot nearest neighbor query model where location-based services response represents a snapshot of point in time. In this model, we introduce a novel idea based on the concept of an open set in a topological space where points belong to a subset called neighborhood of a point. We extend this concept to provide anonymity to real objects where each object belongs to a disjointed neighborhood such that each neighborhood contains a single object. To help identify objects, we implement a database which dynamically scales in direct proportion with the size of the neighborhood. To retrieve information secretly and allow the database to expose only requested information, private information retrieval protocols are executed twice on the data. Our study of the implementation shows that the concept of a single object neighborhood is able to efficiently scale the database with the objects in the area.

The size of the database grows with the size of the objects and area covered by the location-based services. Typically, creating neighborhoods, computing distances between objects in the area and running private information retrieval protocols causes the CPU to respond slowly with this increase in the size of the database. In order to handle a large number of objects, we explore the concept of kernel and parallel computing in GPU. We develop GPU parallel implementation of the snapshot query to handle large number of objects. In our experiment, we exploit parameter tuning. The results show that with parameter tuning and parallel computing power of GPU we are able to significantly reduce the response time as the number of objects increases. To determine response time of an application without knowledge of the intricacies of GPU architecture, we extend our analysis to predict GPU execution time. We develop the runtime equation for an operation and extrapolate the run time for a problem set based on the equation, and then we provide a model to predict GPU response time.

As an alternative, the snapshot nearest neighbor query privacy problem can be addressed using secure hardware computing which can eliminate the need for protecting the rest of the system and minimize resource usage and network transmission time. In this approach, a secure coprocessor is used to provide privacy. We process all information inside the coprocessor to deny adversaries access to any private information. To obfuscate access pattern to external memory location, we use oblivious random access memory methodology to access the server. Experimental evaluation shows that using a secure coprocessor reduces resource usage and query response time as the size of the coverage area and objects increases.

Second, we address privacy concerns in the continuous nearest neighbor query model where location-based services automatically respond to a change in object’s location. In this model, we present solutions for two different types known as moving query static object and moving query moving object. For the solutions, we propose plane partition using a Voronoi diagram, and a continuous fractal space filling curve using a Hilbert curve order to create nearest neighbor relationship through a path. Specifically, space filling curve results in multi-dimensional to 1-dimensional object mapping, and values are then assigned to the objects based on proximity. To prevent subscribers from issuing a query each time there is a change in location and to reduce the response time, we introduce the concept of transition and update time that indicate where and when the nearest neighbor changes. We also introduce a database that dynamically scales with the size of the objects in a path to help obscure and relate objects. By executing the private information retrieval protocol twice on the data, the user secretly retrieves requested information from the database.
The results of our experiment show that using plane partitioning and a space filling curve with transition time between objects reduces the total response time.

Major: Computer Engineering

Educational Career:
Bachelor's of Computer Engineering, BS, 2007, University of Central Florida
Master's of Computer Engineering, MS, 2009, University of Central Florida

Committee in Charge:
Ratan Guha, Chair, EECS
Damla Turgut, UCF
Mostafa Bassiouni, UCF
Cliff Zou, UCF
Ram Mohapatra, UCF

Approved for distribution by Ratan Guha, Committee Chair, on February 10, 2016.

The public is welcome to attend.