Anonymous and Privacy-Sensitive Collection of Sensed Data in Location-Based Applications

James Fogarty
HCI Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
jfogarty@cs.cmu.edu

Pedram Keyani
HCI Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
pkeyani@cs.cmu.edu

Jason I. Hong
HCI Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
jasonh@cs.cmu.edu

Karen P. Tang
HCI Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
kptang@cs.cmu.edu

CHI 2006, April 22–27, 2006, Montreal, Canada.

Abstract
Existing approaches to privacy in location-based applications generally treat people as the entity of interest. Anonymity and privacy are then addressed through a fidelity tradeoff, obscuring either a person's identify or location. But the intentional obfuscation of location can interfere with many potential applications. This position paper discusses hitchhiking, our new that treats locations as the entity of interest. Taking this new perspective allows applications that preserve personal privacy and anonymity while collecting sensed data from people who visit locations of interest.

Keywords
Hitchhiking, privacy, anonymity, location-based computing, mobile social software

ACM Classification Keywords
H5.2. Information interfaces and presentation: User Interfaces; H1.2. Models and Principles: User/Machine Systems.

Introduction
The emergence of location-based computing and mobile social software promises many new and compelling applications, but raises very real privacy risks. For example, consider the ZipDash approach to using

location-enabled mobile phones to collect live traffic flow information [2]. ZipDash uses continuous precise location disclosures to infer traffic congestion by monitoring the rate at which people are moving on major roads. But this continuous disclosure introduces the potential for abuse. A malicious operator could determine where an otherwise anonymous person lives by observing where most of their trips begin and end. The person's identity can then be obtained from existing address databases, providing a malicious operator with a detailed and non-anonymous history of a person's movement.

Our work on hitchhiking, scheduled for presentation at CHI 2006 [1], enables the protection of personal privacy and anonymity while using existing mobile device networks to build these types of applications. Hitchhiking uses the physical constraints of location to address threats to the anonymity and privacy of people reporting sensed information about a location.

The Hitchhiking Approach

Hitchhiking is a software-based implementation targeted at existing commodity devices, including laptop computers and mobile-phones. While the providers of existing networks can already track a device's location, hitchhiking allows applications to be built on these networks without introducing any additional threats.

In a typical scenario, a person's location-aware mobile phone would have a set of GPS coordinates defining areas of interest to a traffic monitoring application, such as bridges, tunnels, and other bottlenecks. When a phone detects that a person is driving through one of these areas of interest, it checks to see if the person has approved the disclosure of information about this location. If so, it begins reporting the GPS coordinate and velocity of the vehicle, allowing the application server to model traffic flow in the location of interest. If the person has not previously approved reporting from the location, a client-generated map will later be used to request approval.

The proper implementation a hitchhiking approach has seven requirements:

Location is computed on the client. Significant prior work has developed client-side computation of a device's location. If a device must communicate with an application server to compute its location, the server will always know the device's location.

Only the client device is trusted. While it is fairly easy to design a client that does not intentionally reveal a person's identity or support tracking, hitchhiking sets the higher standard of assuming that the servers used by an application are maliciously attacking a client in an attempt to induce identity or tracking violations.

Each person must approve reporting from a location. Because applications do not intentionally reveal identity, a malicious server can attack an individual by requesting information about a relatively private location, such as a person's home. Hitchhiking therefore requires the explicit approval of each person who reports from a location.

Physical constraints prevent location spoofing. A malicious server can be expected to attempt to trick a person into approving the disclosure of information about a potentially sensitive location. For example, the

server might request approval for a sensitive location, but give the location an innocent name and description. Hitchhiking therefore uses the physical constraints of location to ensure it is clear what location is being approved for disclosure. An application might require that a person physically be in the location being approved, or a client might use the GPS coordinates defining a location to generate a trusted map.

Location identifiers are based in the physical location. Reporting about a location using an arbitrary identifier, such as a unique ID in a database, allows malicious servers to track clients by giving them different identifiers to describe the same location. Identifying a location with a sensed physical property, such as a GPS coordinate or the identifiers of detectable WiFi access points, precludes this attack.

Location identifiers are generated by the client. If a server provides a GPS coordinate defining a location of interest, and the client reports on that location using the provided GPS coordinate, the low-order bits of the coordinate could hide a tracking identifier. Client devices must generate location identifiers without using information provided by an application server. If a client reports a person's current GPS coordinate or the set of currently detectable WiFi access points, it is then up to the server to determine what location of interest a person is reporting from.

Sensed identifiers are not reported to a server. Some clients will sense unique identifiers, such as the MAC addresses of nearby Bluetooth-enabled phones. Such identifiers may be used on a client, perhaps to determine how many people are nearby, but they must not be reported to an application server. Reporting such information to an application server would allow a malicious operator to track the movement of the people associated with those identifiers.

Discussion

While participating in this workshop on mobile social software, we hope to discuss how the hitchhiking approach might suggest other technical solutions to protecting privacy in location-aware applications and social software. We have so far focused on applications that attempt to determine when locations are relatively empty (coffee shop space availability, conference room availability, traffic monitoring, and bus tracking). But the hitchhiking approach could also be applied to the "what's hot" problem. Mobile devices of people in a popular destination could anonymously share that a location is currently busy, letting others join the party. It is also interesting to consider whether we can maintain the anonymity of hitchhiking while focusing "what's hot" applications on a person's social network.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010, by an AT&T Labs fellowship, and by the National Science Foundation under grants IIS-0121560 and IIS-032531.

References

[1] Tang, K.P., Keyani, P., Fogarty, J. and Hong, J.I. (2006) Putting People in their Place: An Anonymous and Privacy-Sensitive Approach to Collecting Sensed Data in Location-Based Applications. *To Appear, ACM Conference on Human Factors in Computing Systems (CHI 2006)*.

[2] Zipdash - Mobile Map and Traffic App. http://www.zipdash.com