USENIX Association

Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems

Lihue, Hawaii, USA
May 18–21, 2003

Privacy-Aware Location Sensor Networks

Marco Gruteser, Graham Schelle, Ashish Jain, Rick Han, and Dirk Grunwald
Department of Computer Science
University of Colorado at Boulder
Boulder, CO 80309
{gruteser,schelleg,ajain,rhan,grunwald}@cs.colorado.edu

Abstract

Advances in sensor networking and location tracking technology enable location-based applications but they also create significant privacy risks. Privacy is typically addressed through privacy policies, which inform the user about a service provider's data handling practices and serve as the basis for the user's decision to release data. However, privacy policies require user interaction and offer little protection from malicious service providers. This paper addresses privacy through a distributed anonymity algorithm that is applied in a sensor network, before service providers gain access to the data. These mechanisms can provide a high degree of privacy, save service users from dealing with service providers' privacy policies, and reduce the service providers' requirements for safeguarding private information.

1 Introduction

Sensor network technology promises a vast increase in automatic data collection capabilities through efficient deployment of tiny sensing devices. Arrays of sensors could be deployed alongside roads to monitor traffic patterns or inside buildings to sense contextual information for adaptive computing services. In particular, there is great interest in location tracking systems, which determine the position of users for location-based services. We foresee that sensor network technology decreases the cost of such systems by replacing cables with multi-hop radio communications and allowing in-network processing of data.

While these technologies offer great benefits to users, they also exhibit significant potential for abuse.[1] Particularly relevant are privacy concerns, since sensor network technology provides greatly expanded data collection capabilities.

A common approach addresses privacy concerns at the database or location server layer—after data has been collected. For example, privacy policies govern who can use an individual's data for which purposes [2, 3, 4]. Furthermore, data perturbation [5] or anonymity mechanism [6, 7] provide access to data without disclosing privacy sensitive information. However, data is difficult to protect once it is stored on a system. In the past, private data has been inadvertently disclosed over the Internet and companies have distributed data in violation of their own privacy policies. In addition, data theft and distribution through company insiders poses a serious challenge. Such approaches also do not address the risks that an adversary circumvents the location server and directly collects data from the location tracking system.

This paper leverages sensor nodes' data processing capabilities to enhance privacy through distributed, in-network anonymity mechanisms. These mechanisms are applied before data leaves the sensor network and can be stored in a location server; thus, databases and locations servers are removed from the trusted computing base, meaning users only need to trust the sensor network itself. A third party, independent from the data consumers, could install and service the network to establish user trust. The paper concentrates on location sensor networks, since location information is especially privacy sensitive and potentially specific enough to reveal the identity of individuals. Specifically, the paper contributes the following key ideas:

- a discussion of privacy risks and attacks for location sensor networks
- a distributed privacy algorithm that cloaks location information to preserve anonymity
- a complimentary routing scheme and election algorithm that chooses leaders for hierarchically organized entities in physical space

2 Related Work

Privacy concerns in location-based application scenarios are typically addressed in a location broker residing in the middleware layer. To our knowledge, Spreitzer and Theimer [8] pioneered the development of such an architecture. In this work, each user owns a trusted user agent that acts as an intermediary. It collects location information from a variety of sensors and controls application access to this data.

More recent research addresses the specifics of privacy policies, on which access control decisions are based. For instance, Myles and colleagues [9] describe an ar-

[1] Indeed, at least in one case a man stalked his former girlfriend aided by a GPS device and digital cellular transmitter mounted on her car [1].

chitecture for a centralized location server that controls access from client applications through a set of validator modules that check XML-encoded application privacy policies. In the automotive telematics domain, Duri and colleagues [4] present a policy-based framework for protecting sensor information, where an in-car computer can act as a trusted agent. Hengartner and Steenkiste [10] point out that access control decisions can be governed by either room or location policies; thus, such systems should be able to resolve conflicts between several different policies. Snekkenes [3] presents advanced concepts for specifying policies in the context of a mobile phone network. These concepts enable access control based on criteria such as time of the request, location, speed, and identity of the located object. However, the author concludes by expressing doubt that the average user will specify such complex policies. In addition, privacy policies mainly serve as a vehicle for establishing trust in a service provider—they cannot guarantee that the provider adequately protects the collected data from in- or outside attacks.

Anonymity mechanisms present an alternative to privacy policy-based access control through depersonalization of data before its release. Specifically, Gruteser and Grunwald [11] analyze the feasibility of anonymizing location information for location-based services in an automotive telematics environment. In addition, Beresford and Stajano [12] independently evaluate anonymity techniques for an indoor location system based on the Active Bat. These approaches address the problem of too precise location information that enables identification of a user or continued tracking of movements. However, access control or anonymity mechanisms in the middleware offer little protection when the location tracking system (the sensors) are owned by an untrusted party, such as in a shopping mall.

The Cricket Location-Support System [13] incorporates privacy concern in the design of the location sensor system itself. The system comprises a set of beacons embedded into the environment and receiving devices that determine their location through listening for the radio and ultrasound beacons. This approach enhances user privacy over previous systems, such as the Active Badge [14] and the Active Bat [15], because device location information is initially only known to the devices themselves. The owner can then conceivably decide to whom this data should be released. Therefore, users do not need to trust the embedded sensors or a location server. However, it requires the user to carry a device that is compatible with the beacons and powerful enough to make access control decisions, to delegate them to the user (via a suitable interface), or to communicate the request to another trusted agent. It does not cover other classes of location-tracking systems, where the user carries no device (e.g., infrared cameras) or the device is not powerful enough to allow such decision-making (e.g., RFID or the Active Bat).

3 Design Considerations

One usage example of a location sensor network is an in-building occupant movement tracking system. Such a location system would be useful for architectural and interior design, since it would deliver data on the popularity and usage of different building areas such as conference rooms, alcoves, individual offices, or supermarket aisles.[2] However, employees or customers might be concerned about their privacy. We will revisit this example throughout the paper.

These applications require aggregate statistics on the popularity of certain locations but not necessarily precise information about a person's location at any given time. Therefore, we argue that this problem can reasonably be addressed through anonymity mechanisms that reduce data quality within known bounds to maintain a well-defined level of anonymity in different situations.

We do not restrict the system to a specific location sensing technology but make the following assumptions. The location tracking system comprises an array of sensor nodes, one or more base stations, and a location server. The sensor nodes are resource limited computing devices with wireless communication capabilities (e.g., [17, 18]). The sensors itself should be capable to determine the number of individuals in an area and monitor changes in real-time. Base stations bridge the wireless sensor communications into the wired network, where the location server collects the sensor data and publishes it to applications.

The sensor system periodically reports location information as a set of tuples (c, a) where a labels an area and c the count of data subjects, who visited the area during the period. Areas are hierarchically organized; therefore, the network can present an overall count for a certain area in addition to counts for smaller sub-areas within.

3.1 Privacy Threats and Attack Model

We define a location privacy threat as an instance in which an adversary can obtain an individual's (the data subject's) location information through the location system *and* can identify the individual. For example, through the location system an adversary could obtain the current position of every individual. Continuous access to this information would allow him to track movements of an unknown user. However, for this to constitute a location privacy threat, the adversary must also be able to link identities to the reported user locations.

To identify individuals, the adversary can have prior information about the people and space that are monitored. For example, knowing who owns a particular office would most likely correctly identify a person that is monitored in this office [12]. The adversary can simply

[2] In fact, the IBM Footprint research project [16] developed an inexpensive $10 infrared sensor. An array of such sensors allows stores to measure the effectiveness of their store design by tracking the path of customers through the store. For example, it reveals whether promotional items are effectively placed, whether customers stopped to look at promotions, or how long customers had to search for a specific item.

link these two pieces of information and conclude that with very high probability the identified individual is in his office. Once identified, he can then track the individual's movements to other areas of the building by monitoring the location updates. Through adaptively changing data precision, the sensor network seeks to prevent (or at least make sufficiently difficult) that an adversary can link prior information with the information obtained through the sensor system. The network should only reveal precise locations of groups of people, but not of individuals and their paths. Inspired by Samarati and Sweeney [19, 6, 7], we consider the data k-anonymous, if every location reported from the network is indistinguishable from at least k 1 other subjects.

This work also considers a more sophisticated adversary, with local access to the sensor network, who attacks the network to gain more precise location information. In particular, the adversary could mount the following attacks:

- Passive Attacks

  Eavesdropping. The adversary could simply listen to data and control traffic. Control traffic conveys information about the sensor network configuration. Data traffic contains potentially more detailed information than accessible through the location server.

  Traffic analysis. An increase in the number of transmitted packets between certain nodes could signal that a specific sensor has registered activity.

- Active Attacks

  Insert false data. A malicious node could trick the system into reducing data distortion (privacy protection) through spoofing subjects.

  Change routing behavior. An inserted or compromised node could drop packets, forward them incorrectly, or advertise itself as the best route to all nodes (blackhole effect) in an attempt to gain information.

This paper focuses on user privacy; hence, we do not consider attacks such as denial of service, where the adversary does not learn any private information.

4 System Design

A network is needed that provides near real-time location information with the properties that it preserves k-anonymity with respect to the described attack model while still delivering useful data. To achieve this goal, we take the following approach.

4.1 Approach

Data cloaking. The sensor network perturbs the sensed location data so that it meets the k-anonymity criterion. Ideally, the network applies only the minimum necessary perturbation so that the data retains its usefulness for a large number of applications.

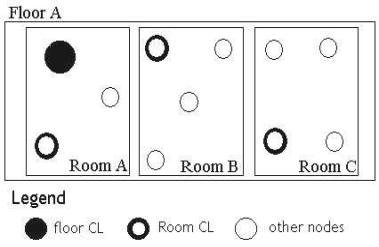

Figure 1: The desired result of coordination leader election is one CL at each hierarchical level

Hierarchical aggregation. Network nodes organize distribution of sensed location data through a spanning tree. Multiple nodes throughout the spanning tree, the coordination leaders (CL), cloak data so that no single entity has a complete view of the original data. The hierarchy should reflect the spatial characteristics of the area. For example, it could be organized into cubicles, rooms, floors, and buildings.

Secure and unobservable communications. Nodes communicate with encrypted and authenticated data packets (e.g., using the SPINS protocols [20]) to prevent eavesdropping and active attacks. In addition, data transmissions are periodic and independent from sensor readings to protect against traffic analysis.

4.2 Coordination Leader Election and Spanning Tree Construction

The decentralized cloaking and aggregation mechanism requires one coordination leader for every level of hierarchy; for example, one CL for every room, for every floor, and for the building. Figure 1 depicts such a configuration. All data flows from the individual nodes first to the room CL and then to the floor CL, which sends the data to a location server. Since CLs can be outside the single-hop radio range, network nodes need to establish routes to higher-level coordination leaders. The node routing tables will hold an entry for each of the hierarchical CLs that this node belongs to. Referring to Fig. 1, a node in Room B may be required to forward packets to its room CL, while also forwarding packets from Room C bound for the floor CL. For n levels of hierarchy, n routing table entries will be required. Notice that the size of the routing table scales with the total number of hierarchies, rather than the number of nodes in the network.

A hierarchical node ID assignment that mirrors the characteristics of the physical area simplifies coordination leader election. To this end, the ID describes where the node is physically located (e.g., in which room). The node ID is subdivided into several bitfields that determine its identification at every level within the hierarchy.

Every node will have a unique ID, but nodes within the same room will share the same room ID, while nodes on the same floor will all share the same floor ID. The IDs and the hierarchy are statically configured during system installation.

Coordination leader election and routing table setup uses a 3-way handshake protocol. The process starts with a root node, such as a base station or the location server to elect coordination leaders for the top level (e.g. floors). The selected coordination leaders than recursively apply the protocol to find CLs for their sublevels until leaders are elected for all levels. The handshake involves the three packet types CL_REQUEST, CL_REPLY, and CL_CONFIRM, which simply contain the sender and receiver node ID, a hop count, and the packet type.

CL_REQUEST A CL broadcasts a CL_REQUEST packet to discover subordinate CLs. This packet is flooded through the network, but is dropped at all nodes that are not subordinates of the request CL (this is determined by comparing sender and node ID). For example, when the CL of Floor A sends out a CL_REQUEST, nodes on Floor B would drop the packet. As this packet is propagated through the network, nodes can set up routing tables to the originating CL. Therefore, a CL_REQUEST from a CL at hierarchical level n, will result in filling all the routing tables for level n at all nodes who recognize the sender as their CL.

CL_REPLY Every node that receives a CL_REQUEST packet from a parent CL answers with a CL_REPLY packet. This indicates that the replier is a potential candidate to be a CL at the sublevel. Reply packets use the routes to the CL established through the CL_REQUEST packet. The parent CL then chooses as an unique CL for each direct sublevel the quickest replier among all candidates with the lowest hop count. Other metrics such as signal strength for single hop candidates are also plausible. Intermediate nodes forwarding CL_REPLY packets will increment the hop count field. They can also drop packets from same-level nodes, if they have already sent a reply packet that is superior according to the well-known selection metric. As an example, a node in Room B can drop reply packets from other nodes in Room B, if they have a higher hop count than a previously forwarded packet.

CL_CONFIRM Finally, the parent CL sends a CL_CONFIRM packet to each chosen CL, which is flooded until it reaches the new CL. A confirmed CL then restarts the process by sending out its own CL_REQUEST to the next lower level of hierarchies.

4.3 Data Cloaking

Nodes employ two basic techniques to increase anonymity: Provide less spatial accuracy and perturb the count of subjects in the covered area. In our hierarchical organization, less spatial accuracy can be achieved by omitting a range of the less significant bits of the sender node ID (ID blurring); thus, the two approaches are:

1. Cloak ID, provide precise data
2. Cloak data, provide precise ID

The data cloaking algorithm combines both approaches. Each node stores the desired anonymity level k, which is preconfigured. If the number of subjects meets or exceeds k the algorithm cloaks data and provides a precise node ID (which describes the area); otherwise, it provides precise data with a cloaked ID.

Data cloaking is achieved through smart rounding. We define the smart rounding function as follows:

$$y = \begin{cases} x & \text{if } x \bmod k = 0 \\ round_k(x \quad (0.5 * r)) & \text{otherwise} \end{cases}$$

where $round_k$ rounds to the nearest multiple of k and r is a random variable that contains 0 or 1 with equal probability. At higher event counts, smart rounding will allow for more precision of data location, rather than actual data values. Smart Rounding would occur only once if no aggregation occurred with other data packets and then be passed directly up to the highest level.

Having the ID blurred at lower subject counts will allow aggregation of these small numbers to occur at higher hierarchical levels. Eventually the blurred ID will get to a hierarchical level where it can be aggregated and exceeds k. Otherwise, it will get passed up to the highest hierarchical level with that highest level's ID.

In order to defend against traffic analysis attacks, the network will follow a near constant rate of data traffic. Already, nodes are sending events to CLs at a constant rate. Of course, lack of traffic could also possibly give away information. In the office usage space example, lack of traffic would relate to no movement meaning that no one is in the room, floor, or even building. This is information we do not want an attacker to gain. Therefore all nodes are required to send at least one packet per data gathering interval (with an event count of zero). Even if a node has seen no events, it still must send a packet. CLs are also required to send at least one packet.

We require *at least* one packet, because CLs may have to send multiple packets due to the size of the data incoming versus the buffer size available on the node. By allowing more than one packet to be sent, our design is also much more scalable to larger networks, where data incoming to a node may completely overwhelm the resources available in a node.

5 Preliminary Conclusions

We have outlined a potential solution to the challenge of integrating privacy-enhancing mechanisms into sensor systems. This approach promises to strengthen user privacy protection compared to solutions at the database level because it *prevents collection* of privacy-sensitive data. From our ongoing work, we draw the following experiences and preliminary conclusions:

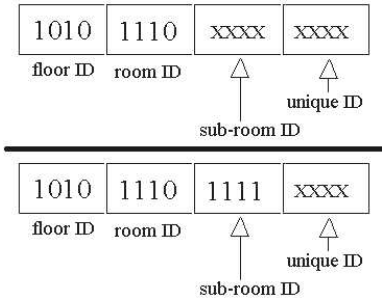

Figure 2: From the lower ID, a receiver would only know an event occurred at the sub-room level. The upper ID shows even more blurring to the room level

- Designing privacy protection into sensor systems seems feasible albeit the current design suffers from a substantial communication overhead to defend against traffic analysis. This is especially concerning, if sensors have a very restricted energy budget.
- Privacy concerns influence system design especially in the area of networking protocols.
- Needed is a formal, likely probabilistic, model for location anonymity that captures the notion of a continuous stream of data. This would enable a better evaluation of the privacy protection afforded by such systems.
- Finally, a better understanding of the location data accuracy requirements for different classes of applications would enable an analysis of the level of anonymity that can be sustained for such applications.

References

[1] CNN. Police: Gps device used to stalk woman. `http://www.cnn.com/2002/TECH/ptech/12/31/gps.stalk.ap/index.html`, December 31 2002.

[2] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *4th International Conference on Ubiquitous Computing*, 2002.

[3] Einar Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 48–57. ACM Press, 2001.

[4] Sastry Duri, Marco Gruteser, Xuan Liu, Paul Moskowitz, Ronald Perez, Moninder Singh, and Jung-Mu Tang. Framework for security and privacy in automotive telematics. In *2nd ACM International Worksphop on Mobile Commerce*, 2002.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[6] Pierangela Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6), 2001.

[7] Latanya Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[8] Mike Spreitzer and Marvin Theimer. Providing Location Information in a Ubiquitous Computing Environment. In *Proceedings of the Fourteenth ACM Symposium on Operating System Principles*, pages 270–283, 1993.

[9] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[10] Urs Hengartner and Peter Steenkiste. Protecting Access to People Location Information. In *Proceedings of First International Conference on Security in Pervasive Computing (to appear)*, LNCS. Springer, Mar 2003.

[11] Marco Gruteser and Dirk Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services (to appear)*, May 2003.

[12] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[13] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pages 32–43. ACM Press, 2000.

[14] Roy Want, Andy Hopper, Veronica Falco, and Jonathan Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

[15] Andy Ward, Alan Jones, and Andy Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, Oct 1997.

[16] IBM Research Exploratory Computer Vision Group. Footprint: Infrared person tracking. `http://www.research.ibm.com/ecvg/misc/footprint.html`.

[17] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System Architecture Directions for Networked Sensors. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 93–104. ACM Press, 2000.

[18] Hector Abrach, Jim Carlson, Hui Dai, Jeff Rose, Anmol Sheth, Brian Shucker, and Richard Han. MANTIS: System Support For MultimodAl NeTworks of In-situ Sensors. Technical Report CU-CS-950-03, University of Colorado, Department of Computer Science, April 2003.

[19] P. Samarati and L. Sweeney. Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement through Generalization and Suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International, 1998.

[20] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, pages 189–199. ACM Press, 2001.