Die Implikationen des ubiquitous Sensing am Beispiel von Location Privacy

Melanie Imhof

Departement für Informatik, ETH Zurich

imhofme@ethz.ch

Zusammenfassung

Durch die Verbreitung von mobilen Sensorkonten entstehen neue Herausforderungen, diese sind nicht nur technischer, sondern vermehrt auch sozialer Natur. In dieser Ausarbeitung wird die Problematik der Privatsphäre bezüglich des Aufenthaltortes (Location-Privacy) besprochen. Dazu werden die Anforderungen, welche an ein solches System gestellt werden, erläutert.
Zudem werden zwei Lösungsansätze aufgezeigt und diskutiert. Einerseits der Ansatz der Mix-Zonen, der mit Applikationen, welche temporäre Pseudonyme verwenden, arbeitet und diese unbemerkt verändert. Andererseits der Ansatz der Path Confusion, welcher für Applikationen mit vollständiger Anonymität verwendbar ist. Dieser verfälscht die Pfade um die Anonymität zu garantieren und gleichzeitig die Datenintegrität möglichst gut zu bewahren.

1 Einführung

In naher Zukunft können Sensordaten überall erfasst und kommuniziert werden, dies wird nicht nur Vorteile haben. Neue Herausforderungen, wie in [4] beschrieben, entstehen. Beispielsweise möchte man eine gewisse Interoperabilität erreichen, das heisst verschiedene Sensoren sollen miteinander kommunizieren können, was neue standardisierte Protokolle voraussetzt. Zudem muss beachtet werden, dass nicht jeder Benutzer das Know-how eines Systemadministrators hat und dennoch gewisse Dienste in Anspruch nehmen möchte. Zu den technischen Herausforderungen kommen immer mehr auch soziale Bedenken hinzu, die nicht vernachlässigbar sind. Vor allem Datenschutz ist für die Benutzer wichtig und wurde deshalb auch schon in vielen Staaten im Gesetz festgehalten. Dabei spielt der Schutz der Information über den Aufenthaltsort eine besonders wichtige Rolle, da selbst bei anonymisierter Erfassung auf die Identität der Benutzer geschlossen werden kann. Dadurch können zudem weitgehende Implikationen auf die Persönlichkeit eines Benutzers gemacht werden. Der Ort ist gerade deshalb bei den meisten Applikationen mit ubiquitous Sensing eine massgebende Komponente zur Implementierung der Funktionalität. Die Messungen, die mehr und mehr durch mobile Geräte vorgenommen werden, können erst in den richtigen Kontext gebracht werden, wenn die Position bekannt ist. Beispielsweise ist keiner an der Temperatur interessiert, wenn der Ort nicht bekannt ist.
Es ist deshalb wichtig Methoden zu haben um die Informationen über den Ort kontrolliert abzugeben und damit eine gewisse Privatsphäre zu gewährleisten. Die Notwendigkeit diese Daten zu schützen, möchte ich am Beispiel der Predestination Methode[6] zeigen.

Für das System wurden 169 Personenwagen mit einem GPS-Sender ausgestattet, dabei kamen innerhalb von zwei Wochen 1,228,237 Positionen zusammen. Diese Daten wurden dann dazu verwendet das Ziel einer Person möglichst genau vorauszusagen.
Um die Wahrscheinlichkeit eines Ziels zu berechnen wurde ein Gitter über die Karte gelegt, wobei 1600 Zellen mit einer Seitenlänge von einem Kilometer entstanden. Da die Wahrscheinlichkeit, eine Zelle als Ziel zu haben, von der Topologie abhängt, wurde jeder Zelle auf Grund von der United States Geological Survey(USGS) Karte eine gewisse Grundwahrscheinlichkeit zugeordnet. Weiter ging man von der Open-World-Assumption aus, welche besagt, dass die Wahrscheinlichkeit für ein Ziel an dem ein Benutzer noch nie war nicht gleich null ist. Somit werden auch Ziele vorausgesagt, die noch nie besucht wurden. Ausserdem tendieren Menschen dazu, Ziele in der Nähe von Orten zu wählen, an denen sie schon einmal waren oder an denen sie gerade sind. Auch dem wurde Rechnung getragen, wie auch der Tatsache, dass meist effiziente Routen gefahren werden. Unter all diesen Vorbedingungen haben sie die Wahrscheinlichkeit einer Zelle berechnet und lagen dabei im Schnitt nur 3 Kilometer daneben. Dies ist je nach Applikation ein gutes oder auch schlechtes Ergebnis. Es wird jedoch klar, dass aus Ortsinformationen sehr viel hergeleitet werden kann, wie in diesem Beispiel einen zukünftigen Aufenthaltsort. Es muss deshalb sichergestellt werden, dass solche Daten nicht unverfälscht abgegeben werden.

2 Anforderungen

In diesem Kapitel möchte ich aufzeigen, welche Anforderungen an Systeme, die Location-Privacy erlauben, gestellt werden und welche Probleme daraus entstehen. Dazu werde ich anhand von zwei Beispielen zeigen, dass die Art der Anwendung der Lokalisierungsinformationen eine ganz entscheidende Rolle spielt und dies auch zu ganz unterschiedlichen Anforderungen führt.

Bei der ersten Beispielapplikation handelt es sich um eine Staumeldungssoftware. Bei der Zweiten um eine bei der man verschiedene Dienste abonnieren kann, die dann je nach Aufenthaltsort angeboten werden. Beispielsweise wird der Kaffeepreis angezeigt, wenn man in der Nähe eines Cafes ist oder es werden die Nachrichten abgerufen, sobald man im Zug sitzt.

2.1 Niveau der Privatsphäre

Um verschiedene Systeme zu vergleichen ist es wichtig eine Grösse zu haben die etwas über die Qualität aussagt. Das Maximum der Privatsphäre wird in diesem Zusammenhang dann erreicht, wenn die Entropie maximiert wird, denn dann ist die Unsicherheit beim Gegner am grössten, was der Fall ist, wenn jeder Ort gleich wahrscheinlich ist. Falls sich zwei Personen am gleich Ort aufhalten, wird die Entropie maximal, da beide möglichen Zuordnungen gleich wahrscheinlich sind. Nach Baik Hoh und Eric Horvitz [5] ist die Anonymität jedoch nicht garantiert, da es meist keine Rolle spielt wie die Zuordnung ist, denn beide Möglichkeiten enthalten die gleiche Information, nämlich die Aufenthaltsorte der beiden Personen.

2.2 Anonymität und Pseudonymität

Anonymität ist dann gewährleistet, wenn ein Individuum nicht von den anderen unterscheidbar ist. Pseudonymität hingegen ist, wenn man eine fixe Identität (ein Pseudonym) hat, von welchem aber nicht auf die richtige Identität geschlossen werden kann [3].

Vollkommene Anonymität schränkt die Applikationen extrem ein und es können keine Dienste mehr angeboten werden, bei denen eine Authentifizierung nötig ist. Die erste Beispielapplikation kann mit anonymisierten Daten arbeiten, da man zur Bestimmung eines Staus nur die Anzahl der Personen benötigt und nicht deren Identitäten. Für unsere zweite Beispielapplikation kommt jedoch eine vollkommene Anonymisierung nicht in Frage, da nur die Benutzer informiert werden sollten, welche den Dienst abonniert haben. Dies ist jedoch nur möglich wenn die Benutzer unterschieden werden können, was mit Pseudonymität erreicht werden kann. Pseudonymität ist jedoch schwer zu erreichen, denn die Lokalisierung erlaubt es zu bestimmen wo jemand wohnt und arbeitet, wodurch die Identitätsbestimmung leicht fällt. Bei diesem Problem ist vor allem die Methode der Mix-Zonen, welche im Kapitel 3.1 besprochen wird, hilfreich.

2.3 Integrität der Daten

Die Integrität der Daten und welche Fehler tolerierbar sind, hängt ebenfalls stark von der Anwendung ab. So sind die Daten eines GPS-Systems für eine Applikation, die auf wenige Zentimeter genaue Daten benötigt, schon von Beginn an unbrauchbar. Meist ist es aber eher der Fall, dass die Daten zu Beginn genug genau sind, man diese aber verfälschen möchte um die Privatsphäre zu gewährleisten. Dabei muss man sehr genau beachten, wie viel Verfälschung noch tolerierbar ist. Beim Kaffeebeispiel führen Ungenauigkeiten von wenigen Metern nicht dazu, dass die Applikation nutzlos wird. Aber wenn man die Kaffeepreise von allen Cafes, der nächsten zehn Häuserblocks bekommt, ist dies bestimmt nicht wünschenswert.

2.4 Wo ist die Information?

Der Ort an dem die Information gespeichert wird, ist fast gleich wichtig wie die Ortsinformation selbst. Wird die Information ohne Verfälschung auf einen Server geschickt, dem man nicht traut, so macht es auch keinen Sinn mehr diese nachträglich zu verfälschen. Deshalb sind die meisten Systeme darauf aufgebaut, dass die Information entweder schon beim Benutzer verfälscht wird oder auf einem vertrauenswürdigen Server. Die Applikationen erhalten, dann nur die verfälschte Information.

2.5 Benutzerinteraktion

Prinzipiell gibt es bei Location-Privacy-Systemen zwei Ansätze. Beim ersten versucht man die Information automatisch zu verfälschen, beim zweiten gibt man dem Benutzer die Möglichkeit, diese selbst zu verfälschen.
Beim manuellen Ansatz liegt es beim Benutzer zu entscheiden wie viel er preisgeben will. Dies kann beispielsweise durch eine benutzerspezifische Einschränkung oder durch temporäre Zurückhaltung der Daten kontrolliert werden. Um dadurch einen effektiven Schutz zu gewährleisten wird, jedoch eine regelmässige Anwendung dieser Möglichkeiten erfordert. Dies ist jedoch für den Benutzer zeitaufwändig und mühsam, wodurch möglicherweise auf Datenschutz verzichtet wird oder die Anwendung nur teilweise oder gar nicht verwendet werden kann.
Der automatische Ansatz reduziert die Benutzerinteraktion auf ein einmaliges Einstellen des Grades der Privatspäre und ist demnach viel leichter benutzbar, auch wenn der Benutzer dadurch weniger Kontrolle hat und auf den Anbieter vertrauen muss.

3 Lösungsansätze

3.1 Mix-Zonen

Die Idee Mix-Zonen für Location-Privacy einzusetzen entstand an der Universität von Cambridge [2]. Es ist für Applikationen, welche temporäre Pseudonyme erlauben, geeignet. Beispielsweise für die Kaffeeapplikation, die in der Einführung angesprochen wurde. Da ein einziges Pseudonym keine Sicherheit gibt, basiert die Idee darauf, dass in Zonen in denen keine Dienste abonniert wurden, sogenannte Mix-Zonen, das Pseudonym unbeobachtet gewechselt werden kann. Damit dies von der Applikation versteckt geschehen kann, findet die Kommunikation ausschliesslich über einen Proxy statt. Ein weiterer Schutz entsteht durch sogenannte „update periodes“. Dabei wird nicht stetig die neue Position gesendet, sondern nur von Zeit zu Zeit. Somit hat man einen wählbaren Parameter über welchen der Grad der Privatsphäre reguliert werden kann.
Mix-Zonen sind also die Regionen in denen keine Dienste angefordert werden und die Benutzer anonym sind, sie haben zwar ein Pseudonym, welches aber nicht zugeordnet werden kann. Es entsteht ein „anonymity set“ (eine Menge von anonymen Benutzern), vorausgesetzt es befinden sich mehr als ein Benutzer in dieser Region. Um maximale Sicherheit zu gewährleisten, sollte die Mix-Zone also möglichst gross sein. Dabei ist aber zu beachten, dass die benötigte Zeit, eine Mix-Zone zu durchqueren, nicht grösser als eine Update-Periode sein darf. Falls die benötigte Zeit kleiner als die Update-Periode ist, können zwei Benutzer, welche die Mix-Zone betreten, beim Verlassen unterschieden werden, wodurch eine Zuordnung der alten zu den neuen Pseudonymen vorgenommen werden kann. Als Beispiel sieht man in der Abbildung 1 eine Mix-Zone und drei abonnierbare Dienste, eine Bank, ein Kaffee und einen Flughafen. Dabei ist die Distanz vom Kaffee zur Bank viel grösser als die vom Flughafen zur Bank. Angenommen es betreten zwei Benutzer A und B gleichzeitig die Mix-Zone. Benutzer A geht vom Kaffee zur Bank und Benutzer B vom Flughafen zur Bank. Falls die Update-Periode kleiner ist, als die benötigte Zeit um vom Kaffee zur Bank zu gehen, wird die Applikation zuerst ein Update vom Benutzer B erhalten und erst später eines von Benutzer A, sie werden also unterscheidbar. Ist die Update-Periode jedoch grösser, wird die Applikation von beiden Benutzern erst eine Nachricht erhalten, wenn beide die Bank erreicht haben, weshalb es dann nicht möglich ist sie auseinander zu halten.

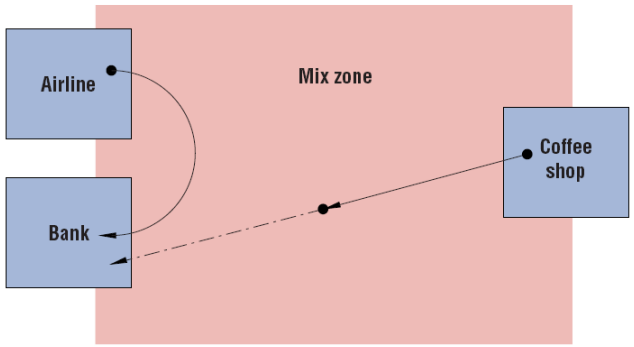

Abbildung 1: Mix-Zone und drei Dienste (Bank, Kaffee und Flughafen)

Durch dieses System wird es sogar möglich schon beim Registrieren für einen Dienst anhand von alten Daten zu berechnen wie gross die Anonymität im Schnitt sein wird.

Um das System zu testen wurde folgendes Experiment durchgeführt. Im AT&T Lab, welches mit einem Active Bat System [1] ausgestattet ist, trugen beinahe alle Angestellten während zwei Wochen einen Bat, wobei mehr als 3.5 Millionen Stichproben entstanden. In Abbildung 2 ist der Grundriss eines Stockwerkes des Cambridge Labors. Für das Experiment wurden drei Mix-Zonen gebildet:

- z_1: Korridor im 1.Stock
- z_2: Korridor im 1.Stock und Hauptkorridor
- z_3: Korridor, Hauptkorridor und Treppenhaus in allen drei Stockwerken

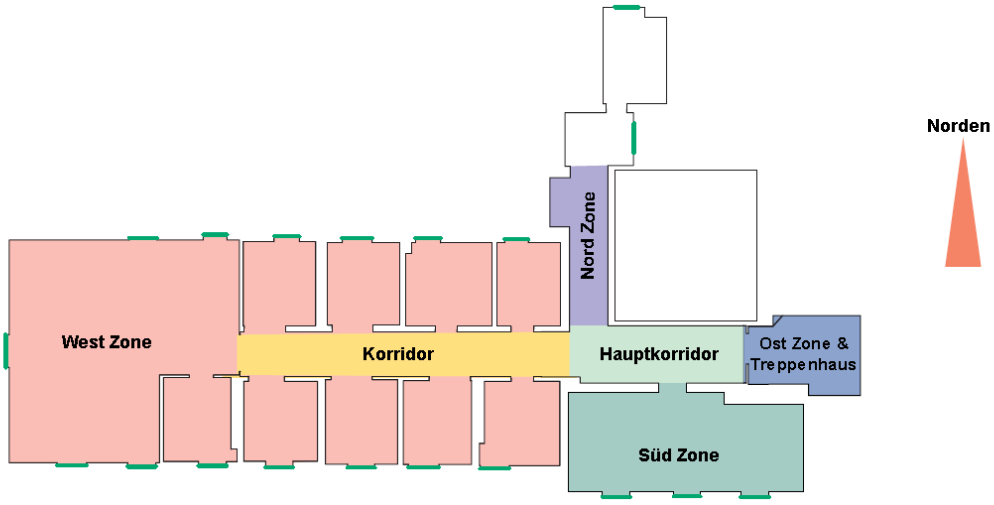

Abbildung 2: Grundriss eines Stockwerkes des AT&T Labors in Cambridge [2]

Dabei sind vor allem zwei Probleme aufgetreten. Das Erste entstand dadurch, dass nur gerade 50 Sekunden benötigt werden um von der einen Seite von z_3 zur anderen zu gelangen. 50 Sekunden sind jedoch für die dadurch resultierende minimale Update-Periode relativ gross, da zwischen den meisten Räumen ohne Update gewechselt werden kann. Das zweite Problem war, dass es sehr wahrscheinlich ist, dass jemand der im dritten Stock ist auch im dritten Stock bleibt und deshalb keine echt anonyme Menge entsteht.
Ausserdem wurde im Ansatz die Richtung in die jemand geht bis anhin vernachlässigt. In der Realität ist es ziemlich unwahrscheinlich, dass sich jemand plötzlich wieder umdreht und wieder in die Richtung geht von welcher er gekommen ist. Dies führt dazu, dass die Applikation relativ gut raten kann, wer jetzt wer ist.

Es ist zudem noch zu klären, was passieren soll, wenn der Anonymitätsgrad unter eine gewisse Grenze fällt. Eine Lösung dafür wäre keine Positionsaktualisierungen mehr zu senden, dadurch wird die Applikation aber unzuverlässig und dies ist nicht wünschenswert. Meistens wird man unter die Grenze fallen, weil zu viele Dienste abonniert wurden, das heisst diese müssen wieder reduziert werden. Man könnte die kritischsten Dienste löschen, dies muss aber dem Benutzer mitgeteilt werden, sonst denkt dieser er habe den Dienst

noch immer und er bekommt dennoch keine Informationen mehr. Dieser Ansatz wird jedoch sicherlich dazu führen, dass der Benutzer mit Warnungen bombardiert wird.
Eine zusätzliche Idee ist die Einführung von „dummy user“, also solchen Benutzern die es in der Realität nicht gibt, sondern die nur eingefügt werden um mehr Anonymität zu garantieren. Dadurch werden jedoch Ressourcen für diese Benutzer verschwendet. Zudem ist es ein schwieriges Problem die Benutzer so realitätsnah zu bewegen, dass sie nicht von den anderen unterschieden werden können.
Als letztes bleibt die Ungewissheit über die Skalierbarkeit des System. Das Experiment wurden innerhalb eines Gebäudes gemacht, es müsste draussen, zum Beispiel anhand von GPS-Daten, noch einmal durchgeführt werden.

3.2 Path Confusion

Biak Hoh und Marco Gruteser [5] haben an der Universität von New Jersey einen Algorithmus entwickelt welcher auf der Störung eines Pfades basiert (Path Confusion) und dadurch die Location-Privacy für ein gegebenes Niveau der Privatsphäre maximiert. Im Gegensatz zum Ansatz der Mix-Zonen, ist er vor allem für Applikationen, die keine Identifizierung eines Benutzers verlangen, verwendbar, wie beispielsweise für Staumeldungsapplikationen. Dabei reicht es aus, periodisch von einer grossen Zahl von Fahrzeugen die Position festzustellen. Eine Anonymisierung durch einen Proxy ist nicht ausreichend, da Multi-Target-Tracking Algorithmen [7], trotz allem eine Zuordnung vornehmen können.
Die Hauptidee des Algorithmus, basiert darauf, dass wenn sich zwei Pfade kreuzen, der Gegner gezwungen ist, eine Zuordnung zu erraten und damit seine Unsicherheit steigt. Deshalb werden jedes Mal, wenn zwei oder mehrere Benutzer in einem gewissen Radius R bei einander sind, alle auf den Mittelpunkt gesetzt und diese Information wird dann an das Lokalisierungssystem weiter gegeben. Dadurch werden Pfade künstlich gekreuzt. In der Abbildung 3 sieht man, dass x und y im zweiten Schritt im gleichen Kreis sind und ihre Pfade verfälscht werden.

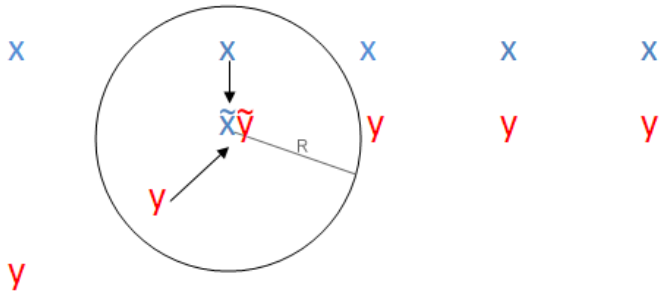

Abbildung 3: Path Confusion

Danach ist für den Gegner nicht mehr klar wer x und wer y ist. Wird dies bei jedem Schritt und mit jedem Benutzern gemacht, so ist es für einen Gegner nahezu unmöglich die zugrundeliegende Information herauszufinden. Die Information die jedoch tatsächlich gebraucht wird für die oben erwähnten Applikationen wurde nur soweit wie nötig verfälscht und ist deshalb noch immer für die gleichen Zwecke einsetzbar. Die dabei entstehende Verfälschung ist ausserdem über den Parameter R zu regulieren, welcher die maximale erlaubte Verfälschung bestimmt. Dabei gilt für jeden Benutzer n zu jeder Zeit k und jede verfälschte Position

$\tilde{x_n}(k), \tilde{y_n}(k)$:

$$(\tilde{x_n}(k) - x_n(k))^2 + (\tilde{y_n}(k) - y_n(k))^2 \leq R^2$$

Das Niveau der Privatsphäre ist als Summe aller durchschnittlichen Fehler für eine Menge von N Benutzer und Pfaden der Länge K definiert.

$$Niveau = \frac{1}{NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \sqrt{(\tilde{x_n}(k) - x_n(k))^2 + (\tilde{y_n}(k) - y_n(k))^2}$$

Nicht selten kommt es jedoch vor, dass keine eindeutige Zuordnung von den Benutzern auf die entstehenden Kreise möglich ist, das heisst ein Benutzer ist genug nahe zu mehreren anderen Benutzern. Dabei entsteht die Frage, welche Kombination dazu führt die Unsicherheit des Gegners zu maximieren. Es wird vorgeschlagen das Optimierungsproblem in 1 zu lösen.

$$\max_{\forall \tilde{x}_n(k), \forall \tilde{y}_n(k)} \frac{1}{N} \sum_{i=1}^{I} p_i(k) d_i(k) \tag{1}$$

Dabei ist N die Anzahl aller Benutzer, I die Anzahl der Hypothesen (das heisst die Anzahl aller möglichen Zuordnungen), $(\tilde{x}_n(k), \tilde{y}_n(k))$ die verfälschte Position des n-ten Benutzers im k-ten Schritt, p_i die Wahrscheinlichkeit der i-ten Hypothese und d_i die Summe der Fehler der Hypothese i.
Mit diesem Algorithmus und einem Location-Privacy Niveau von $R = 150$ konnte eine grössere Privatsphäre erreicht werden, als mit zufälliger Verfälschung der Pfade. Zudem hat sich gezeigt, dass die Privatsphäre über die Zeit variierte, jedoch mit der Zeit zu nahm. Der Algorithmus ist jedoch zu komplex um ihn für viele Benutzer in einem Echtzeitsystem einzusetzen. Es muss also eine vereinfachte Form des Algorithmus gefunden werden.

4 Schlussfolgerungen

In dieser Arbeit wurde aufgezeigt, welche Anforderungen ein System, das Location-Privacy anbietet, erfüllen muss. Dabei hat sich herausgestellt, dass verschiedene Anwendungen unterschiedliche Ansprüche stellen.

Anwendungen, die eine vollkommene Anonymisierung nicht zulassen, da ihr Dienst nur erbracht werden kann, wenn ein temporäres Pseudoym vorhanden ist, können vom Lösungsansatz der Mix-Zonen profitieren. Probleme haben sich bei kleinen Distanzen ergeben, da eine Zuordnung noch relativ einfach erratbar ist. Ausserdem ist unklar, was passiert, wenn man unter einen gewissen Anonymitätsgrad fällt und wie gut das System skaliert. Eine zusätzliche Idee ist die Einführung von „dummy usern“, was jedoch nur schwer realisierbar ist.

Im Gegensatz dazu standen die Applikationen, die mit einer kompletten Anonymisierung arbeiten können. Diese können den Ansatz der Pfadverfälschung (Path Confusion) verwenden, bei welchem die Pfade so manipuliert werden, dass bei einem begrenzten Grad der Verfälschung eine maximale Privatsphäre garantiert werden kann. Da die vorgeschlagene Verfälschung zu einem Optimierungsproblem führt, dass für eine grosse Anzahl von Benutzer in einem Echtzeitsystem nicht effizient berechenbar ist, muss noch eine optimierte Version des Algorithmus gefunden werden.

Zudem haben alle Arten von Applikationen, die auf Lokalisierungsinformationen basieren auch gemeinsame Anforderungen an das Privacy-System. Es wird gefordert, dass die Daten eine gewisse Integrität behalten, die Informationen an einem sicheren Ort gespeichert sind und dass der Benutzer bestimmen kann wie viel er preisgeben will.

Literatur

[1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001.

[2] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46–55, 2003.

[3] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *In Pervasive*, pages 152–170, 2005.

[4] K. W. Edwards and R. E. Grinter. At home with ubiquitous computing: Seven challenges. In *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, pages 256–272, London, UK, 2001. Springer-Verlag.

[5] B. Hoh and E. Horvitz. Protecting location privacy through path confusion. In *In SECURECOMM 05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205. IEEE Computer Society, 2005.

[6] J. Krumm and E. Horvitz. Predestination: Inferring destinations from partial trajectories. In *In Ubicomp*, pages 243–260, 2006.

[7] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.