Universität Stuttgart
Fakultät 5: Informatik, Elektrotechnik und Informationstechnik
Institut für Parallele und Verteilte Systeme
Abteilung Verteilte Systeme

Diplomarbeit Nr.: 3308

Algorithmen zur sicheren Verwaltung und Fragmentierung von Benutzertrajektorien

Jan Barocka

Privacy Aware Management and Fragmentation Algorithms
for Movement Trajectories

Studiengang: Informatik

eingereicht von: Jan Barocka ‹jan@barocka.de›

begonnen am: 03. April 2012
eingereicht am: 03. Oktober 2012

Prüfer: Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Betreuer: Herr Dipl.-Inf. Marius Wernke

CR-Klassifikation: C.2.0

Zusammenfassung

Durch die zunehmende Verbreitung von mobilen Endgeräten, welche in der Lage sind, ihren Standort zu erfassen, erhöht sich zugleich die Zahl der ortsbasierten Anwendungen (wie z.B. Facebook Places, Foursquare etc.). Die Positionsinformationen, welche von diesen Anwendungen genutzt werden, stellen persönliche Daten dar und müssen geschützt werden, um die Privatheit eines Benutzers sicherzustellen. Viele ortsbasierte Anwendungen benötigen aktuelle und kontinuierliche Positionsinformationen in Form von Benutzertrajektorien, z.B. um Verkehrsinformationen zu berechnen. Im Rahmen dieser Arbeit wird ein Verfahren zur sicheren Verwaltung und Fragmentierung dieser Benutzertrajektorien vorgestellt, mit dem es möglich ist, die Privatheit eines Benutzers bei der Veröffentlichung seiner Bewegungsdaten zu gewährleisten. Hierzu wird eine Trajektorie in mehrere Fragmente geteilt, die hinsichtlich ihres Informationsgehalts bewertet und auf unterschiedlichen Lokationsservern verteilt werden. Ortsbasierten Anwendungen, die entsprechende Berechtigungen vom Benutzer erhalten haben, sind dennoch in der Lage, die Trajektorien zu rekonstruieren. Um die Effektivität des Verfahrens zu prüfen, wurden ein Prototyp entwickelt. Mit dessen Hilfe konnten die Möglichkeiten eines Angreifers betrachtet werden, Informationen aus Daten zu gewinnen, in deren Besitz er durch das Kompromittieren eines Lokationsservers gelangen könnte. Es wird gezeigt, dass es durch das beschriebene Verfahren möglich ist Benutzertrajektorien unabhängig von vertrauenswürdigen Instanzen mit einem garantierten Maß an Privatheit zu schützen.

Abstract

Along with the increasing number of mobile devices being able to detect their location various new location based applications (like Facebook Places, foursquare etc.) are arising. The location data used by these applications contains sensitive information about the user and therefore has to be protected to assure the users' privacy. Most location based applications need continuous, current locations in form of movement trajectories, e.g. to calculate information about traffic circulation. In the scope of this work a system for the privacy aware management and fragmentation of these movement trajectories is introduced, which ensures the users' privacy while processing his location information. To achieve this goal, the trajectory will be split into fragments which are rated according to the sensitivity of their information and distributed over different location servers. Location based applications that are authorized by the user are still able to reconstruct the original trajectory. To demonstrate the effectivity of the developed algorithms, a prototype has been implemented. With this prototype the opportunities of an attacker to exploit sensitive information by compromising location servers can be evaluated. It is proven through this prototype that the discussed approaches can provide a way to guarantee a defined degree of privacy without using trusted third parties.

Inhaltsverzeichnis

Algorithmenverzeichnis

Abbildungsverzeichnis

1 Einleitung

Die Verbreitung mobiler Geräte wie Smartphones, Navigationssystemen oder integrierten Systemen nahm in den letzten Jahren stark zu. Viele dieser Geräte enthalten Sensoren, um ihren Standort zu bestimmen und diese Information zu verarbeiten. Dabei werden die Genauigkeit der Sensoren und die Leistungsfähigkeit der mobilen Geräte immer besser. Außerdem sind diese mobilen Geräte in der Lage, sich über schnelle Datenverbindungen mit dem Internet zu verbinden. Diese technischen Weiterentwicklungen und die steigende Akzeptanz der Benutzer treiben die Verbreitung ortsbasierter Anwendungen stetig voran.
Ortsbasierte Anwendungen, wie beispielsweise foursquare oder Layar, bieten Funktionen, die es einem Benutzer ermöglichen, Freunde in der Nähe, spezielle Örtlichkeiten oder kontextabhängige Informationen abzufragen. Gleichzeitig können die Sensoren der Geräte genutzt werden, um Daten zur Stauerkennung oder anderen Zwecken des öffentlichen oder kommerziellen Interesses, wie zum Beispiel zur Marktforschung, zu erfassen. Durch ortsbasierte Anwendungen ergeben sich also sowohl Vorteile für den einzelnen Benutzer als auch für die Gemeinschaft.

Allerdings enthalten Positionsinformationen auch sensible Informationen, welche bei Missbrauch oder Zugriff von Dritten eine ernst zu nehmende Bedrohung für die Privatsphäre eines Benutzers darstellen. Sensibel sind diese Informationen, da sie, neben der Identifikation eines Benutzers, eine große Menge an Rückschlüssen, zum Beispiel auf dessen Angewohnheiten, Beziehungen und Kontakte erlauben. Es könnten beispielsweise Krankenhausbesuche, Treffen bestimmter Personen oder die Abwesenheit einer Person ermittelt werden.
In verschiedenen Studien (vgl. Barkhuus (2003)) wurde festgestellt, dass der Großteil der Bevölkerung nicht vorsichtig mit seinen Lokationsdaten umgeht, sondern sie besonders für nützlich erscheinende Dienste bereitwillig zur Verfügung stellt. Dabei kann ein Missbrauch dieser Positionsinformationen negative Konsequenzen nach sich ziehen, angefangen bei der Überwachung einer Person über ortsbasierte

Werbeeinblendungen bis hin zu kriminellen Delikten. Gelangen diese Informationen beispielsweise in die Hände von Kriminellen, wissen diese, wann eine Person ihr Haus verlässt und wohin sie sich begibt. So könnte ein günstiger Zeitpunkt für einen Einbruch oder einen Raubüberfall geplant werden.

Es ist also wenig verwunderlich, dass die „Sicherheit und Privatheit von Informationen“ immer häufiger Thema der öffentlichen Diskussion in Politik und Medien wird. Auf der einen Seite möchte man von den bereitgestellten Diensten profitieren, andererseits aber so wenig sensible Informationen wie möglich preisgeben.

Laut Barkhuus (2003) befindet sich die Identität einer Person im Mittelpunkt der Privatheit. Sie hängt von mehreren Attributen ab, unter anderem dem Namen, dem Geburtsdatum, weiteren persönlichen Daten oder dem aktuellen Aufenthaltsort. Der Aufenthaltsort einer Person nimmt dabei eine besondere Stellung ein, da er sich kontinuierlich ändern kann und neben der Identität weitere Rückschlüsse erlaubt. Werden die Informationen über eine Person mit deren Identität verknüpft, so entsteht das Risiko, dass die Informationen zum Schaden der Person verwendet werden können. Um eine Verbindung von Information und Identität durch die Position zu verhindern, sollte diese besonders geschützt werden. Wir sprechen in diesem Zusammenhang von Lokationssicherheit (engl. location privacy).

Es wurden bereits verschiedene Ansätze, mit unterschiedlichen Schwerpunkten, zur Sicherung der Privatheit im Umgang mit ortsbasierten Anwendungen entwickelt. Häufig setzen diese eine vertrauenswürdige Instanz voraus, welche sicher stellt, dass Daten an eine Anwendung übertragen werden, ohne die Privatheit eines Benutzers zu gefährden. Doch genau diese vertrauenswürdige Instanz existiert in dieser Art nicht. Es ist möglich, dass eine Angreifer die Instanz kompromittiert und so in den Besitz des gesamten dort verfügbaren Wissens gelangt. Darüber hinaus könnte beim Anbieter einer solchen Instanz ebenfalls kommerzielles Interesse bestehen, das ihn veranlasst die gespeicherten Daten an unbefugte Dritte weiter zu geben.

Das in dieser Arbeit vorgestellte Verfahren soll sicherstellen, dass selbst im Falle eines erfolgreichen Angriffs der Angreifer nur eine minimale Menge an sensiblen Informationen erhält. Um dies zu gewährleisten, werden die Positionsdaten aufgeteilt und über mehrere Server verteilt. Somit müsste ein Angreifer unter mehreren Servern, diejenigen finden, welche Teile der Positionsinformationen seiner Zielperson enthalten. Außerdem müssen die Informationen mehrere Server kombiniert werden.

Damit sollen Rückschlüsse auf weitere Informationen erschwert oder verhindert werden. Die Nutzbarkeit der Daten für befugte Parteien soll dabei jedoch möglichst hoch bleiben. Die Berechnungen des Verfahrens sollen auf dem Endgerät des Benutzers durchgeführt werden, damit das Verfahren nicht auf eine vertrauenswürdige Instanz angewiesen ist, welche nicht unter der unmittelbaren Kontrolle des Benutzers steht.

Um die Privatsphäre eines Benutzers sicherzustellen, sollen die kontinuierlichen Positionsinformationen geschützt werden, da eine Positionserfassung nicht nur Einzelpositionen, sondern längere Strecken erfasst. Wir betrachten hierfür den Weg der Positionsinformationen von der sensorischen Erfassung auf dem Endgerät, über die Datenübermittlung auf und -haltung an einem Lokationsserver, bis hin zur Rekonstruktion bei einer Anwendung. Dabei soll ein bestimmtes Maß an Privatheit zugesichert werden, sobald die Informationen das Endgerät verlassen. Gleichzeitig soll der Nutzen der Daten für eine ortsbasierte Anwendung nicht beeinträchtigt werden. Um dies zu gewährleisten, wollen wir die Auswirkungen unseres Verfahrens auf die Datenqualität bei der Rekonstruktion einer zurückgelegten Strecke ebenfalls betrachten.

Die zentrale Idee des Verfahrens besteht also in der Auf- und Verteilung der kontinuierlichen Positionsinformationen. Damit wird sichergestellt, dass nur berechtigte Instanzen Zugriff auf die vollständigen Positionsinformationen eines Benutzers erhalten. Für das Verfahren sind alternative Vorgehen denkbar, welche in dieser Arbeit diskutiert werden sollen.

Das Verfahren soll auf einem mobilen Endgerät ausführbar sein, weshalb die Kosten hinsichtlich Berechnungs- und Kommunikationsaufwand durch unser Verfahren wenig beeinflusst werden sollen. Weitere mögliche Kosten, wie beispielsweise durch den Einsatz zusätzlicher Server, werden nicht beachtet.

Um ein Maß für die Privatheit zu finden, wird eine Metrik entwickelt, welche den Informationsgehalt von bestimmten Positionsinformationen hinsichtlich eines bestimmten Schutzzieles ermittelt. Mittels dieser Metrik ist es möglich, durch das Verfahren, ein bestimmtes Maß an Privatheit zu garantieren und die möglichen Vorgehen bei der Auf- und Verteilung zu vergleichen.

Im Folgenden wird der Aufbau der Arbeit kurz beschrieben. Um das Thema in einen Gesamtkontext einordnen zu können, werden zunächst bekannte Ansätze beschrieben und deren Vor- und Nachteile erfasst.

Mit diesem Vorwissen werden anschließend die Grundlagen für das vorgeschlagene Verfahren erläutert. Dazu soll zunächst das verwendete Systemmodell beschrieben werden, um im späteren Verlauf die Anforderungen an das zu entwickelnde Verfahren in Abhängigkeit dazu definieren zu können. Anschließend wird das Modell eines Angreifers erläutert, um gewisse Schutzziele formulieren zu können.
Im Kapitel 5 wird das Verfahren in seinen Grundzügen und mit unterschiedlichen Vorgehen bei der Fragmentierung und Verteilung beschrieben. Danach erfolgt eine Sicherheitsanalyse unseres Verfahrens und schließlich eine Evaluation anhanddes implementierten Prototyps.
Anschließend werden noch mögliche Erweiterungen beschrieben, um die das Verfahren ergänzt werden kann.

2 Übersicht vorhandener Ansätze

Es existieren bereits Verfahren, um die Privatsphäre von Benutzern im Umgang mit deren Daten zu schützen. Solche Verfahren finden sich in beinahe jedem Bereich der Informatik, der mit Benutzerdaten arbeitet. Zum Beispiel im Bereich der Datenbanken und Informationssysteme, bei der Kommunikation über E-Mail und anderen Nachrichtensystemen oder der Netzwerkkommunikation. Dabei unterscheiden sich die Verfahren auch in der Art der Daten, die betroffen sind. Im Bereich der ortsbasierten Anwendungen stehen Positionsinformationen im Mittelpunkt. Zum Teil lassen sich hier Verfahren aus anderen Bereichen der Informatik anpassen und anwenden. Hier müssen immer die speziellen Eigenschaften der Positionsinformationen beachtet werden, wie z.B. die kontinuierlichen Veränderungen der Position und die ableitbaren semantischen Informationen.

Laut Duckham u. Kulik (2006) können wir die Strategien zum Schutz der Privatsphäre in verschiedene Kategorien unterteilen:

- Regulatorische Strategien: Dabei werden Regeln und Gesetze entworfen, um den Umgang mit persönlichen Informationen zu regulieren.
- Privatheitsrichtlinien (Policies): Dienste und Anwendungen, die Zugriff auf Lokationsdaten erhalten sollen, erhalten eine Berechtigung und müssen bestimmte Richtlinien einhalten. Ein Beispiel für einen solchen Ansatz ist das P3P (platform of privacy preferences project) des World Wide Web Consortium (W3C). Diese Plattform soll eine einfache Möglichkeit bieten, webbasierte Privatheitsrichtlinien auszutauschen.
- Anonymisierung: Die Lokationsdaten werden von der eigentlichen Identität eines Benutzers getrennt. Ein häufig verwendetes Prinzip ist die Pseudonymität, die trotz Anonymisierung eine Identifikation über ein stellvertretendes Pseudonym ermöglicht. Eine Verbesserung hinsichtlich der Privatheit bietet hier zum Beispiel die k-Anonymität, die sicherstellt, dass ein Benutzer nicht von $k-1$ anderen unterschieden werden kann.

- Verschleierung: Indem die Genauigkeit einer Position verringert wird, versucht man die Privatheit eines Benutzers zu erhöhen. Dabei handelt es sich um räumliche und zeitliche Transformationen der Daten.

Die verschiedenen Strategien und einzelne Verfahren werden im Folgenden detaillierter betrachtet.

2.1 Regulatorische Strategien und Privatheitsrichtlinien

In der „Universal Declaration of Human Rights“ (dt. Erklärung der Menschenrecht) von 1948, Artikel 12, steht:

> „No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.“

Damit wird die Privatsphäre zu einem Menschenrecht und muss gegen Eingriffe durch Dritte geschützt werden. Die Privatsphäre bezieht sich auf persönliche Informationen aus dem Umfeld eines Menschen, die ihm in irgendeiner Weise schaden könnten.
Im deutschen Grundgesetz wird der Schutz der Privatsphäre im allgemeinen Persönlichkeitsrecht (Art. 2 Abs. 1 und Art. 1 Abs. 1 GG) geregelt. Im Bereich der mobilen Datenübertragung gilt zusätzlich das Post- und Fernmeldegeheimnis (Art. 10 GG), das die Sicherheit der Kommunikationsverbindung gesetzlich festlegt.
Diese regulatorischen Ansätze zur Sicherung der Privatsphäre eines Benutzer durch die Gesetzgebung oder Privatheitsrichtlinien können laut Duckham u. Kulik (2006) nur schwer auf die dynamischen Charakteristika von Lokationsinformationen reagieren. Außerdem können solche Regeln nicht vor illegalen Zugriffen auf private Daten schützen.

Auch die Verwendung von Richtlinien kann die Privatheit nicht ausreichend schützen, denn Zertifikate können gefälscht werden und garantieren darum keine Sicherheit.
Darüber hinaus würde ein auf Richtlinien basierter Schutzmechanismus eine vertrauenswürdige dritte Instanz voraussetzen. Wie in Privacy Rights Clearinghouse (2012) dargestellt, nimmt die Zahl der erfolgreichen Angriffe auf solche Instanzen

stark zu, weshalb davon auszugehen ist, dass eine vollständig vertrauenswürdige Instanz nicht existiert.

Da die bisher betrachteten Strategien keinen Schutz vor illegalem Zugriff bieten, sollen im Folgenden die Anonymisierung und die Verschleierung genauer betrachtet und vorhandene Ansätze auf diesen Gebieten diskutiert werden. Die Ansätze lassen sich laut Wernke u. a. (2012) in zwei Dimensionen vergleichen, zum einen den geschützten Informationen und zum anderen in der Effektivität gegen bestimmte Angriffe. Die geschützten Informationen stehen in engem Zusammenhang mit den Schutzzielen eines Verfahrens. Steht bei der Anonymisierung hauptsächlich die Identität im Vordergrund, so versucht die Verschleierung hauptsächlich, die Positionsinformationen zu schützen. Außerdem muss zwischen Ansätzen unterschieden werden, die den Schutz einzelner Positionen zum Ziel haben und denen, die versuchen, Privatheit bei kontinuierlicher Positionsinformationen sicherzustellen.

2.2 Anonymisierung

Ein naiver Ansatz, um die Identität eines Benutzers zu schützen, ist die Verwendung eines Pseudonyms. Ein solches Pseudonym soll eine Rückverfolgung unterbinden. Allerdings lassen andere Daten, wie die hier im Mittelpunkt stehenden Positionsinformationen, Rückschlüsse auf die tatsächliche Identität eines Benutzers zu. Verfolgt ein Angreifer zum Beispiel den Beginn der Positionsinformationen eines Benutzers jeden Morgen mit, so kann er auf dessen Heimatadresse schließen und darüber die Identität des Benutzers feststellen. Die Positionsinformationen können also als sogenannter Quasi-Identifikator, wie in Sweeney (2002) beschrieben, verwendet werden. Um die Anonymität eines Benutzers im Umgang mit dessen Positionsinformationen dennoch zu gewährleisten, wurden verschiedene Konzepte vorgestellt. Das Modell der k-Anonymität ist dabei eines der populärsten und soll die eindeutige Identifikation eines Benutzers weiter erschweren.

2.2.1 K-Anonymität

Der k-Anonymitäts-Ansatz nach Sweeney (2002) sichert die Informationen einer Person, indem dafür gesorgt wird, dass diese nicht von den Informationen $k - 1$ anderer Personen unterschieden werden können.

Es müssen sogenannte Quasi-Identifikatoren gefunden werden, Attribute innerhalb der Information, welche die Identität einer Person aufdecken könnten, falls sie kombiniert werden. Die k-Anonymitätsanforderung ist erfüllt, wenn alle Quasi-Identifikatoren innerhalb einer Menge von freigegebenen Informationen zusammen mit mindestens $k-1$ gleichen Quasi-Identifikatoren vorkommen.
Das ursprünglich für Datenbanken entwickelte Modell kann zur Anonymisierung von Daten in ortsbasierten Systemen genutzt werden. Einige Ansätze hierzu werden in Gkoulalas-Divanis u. Kalnis (2010) diskutiert. Dabei wird versucht, k-Anonymität sowohl auf einzelne als auch auf kontinuierliche Positionsinformationen anzuwenden.
Der am weitesten verbreitete Ansatz ist dabei die Verschleierung durch Regionen, in denen sich zum Zeitpunkt einer Anfrage $k-1$ weitere Benutzer befinden. Die Anfrage wird an einen vertrauenswürdigen Server gestellt, dieser entfernt sämtliche Identifikatoren, wie zum Beispiel den Namen oder die Adresse eines Endgeräts. Aus der Position in der Anfrage baut er eine räumliche und zeitliche Region auf, in der sich k Benutzer befinden, welche sich an diesem Server registriert haben. Die eigentliche Anfrage an eine ortsbasierte Anwendung wird dann mittels dieser Region gestartet. Die Antwort der Anwendung wird anschließend auf dem vertrauenswürdigen Server gefiltert und an das Endgerät des Benutzers weitergeleitet. Dabei achtet der Server darauf, dass Anfragen aller k Benutzer mit derselben Region gestartet werden, um dem Angreifer nicht die Möglichkeit zu geben, bestimmte Benutzer auszuschließen und so einer Identifikation näher zu kommen.

Im Umgang mit Trajektorien, also kontinuierlichen Updates der Position wie etwa bei einem Navigationssystem, werden vom Benutzer B auch kontinuierlich Anfragen an eine ortsbasierte Anwendung gestellt. Angenommen bei einer ersten Anfrage befinden sich die Benutzer A, B, C, D innerhalb einer Region, die zur Anonymisierung zum Zeitpunkt t genutzt wird. Nun bewegen sich die Benutzer weiter. Zum Zeitpunkt $t+1$ wird erneut eine Region gewählt, in der sich B, D, G, I befinden. Ein Angreifer könnte daraus schließen, dass entweder B oder D die Anfrage gestellt hat. So erhöht sich die Wahrscheinlichkeit für einen Bruch der Anonymisierung mit jedem weiteren Positionsupdate der Trajektorie. Um dieses Problem zu lösen, wurden laut Gkoulalas-Divanis u. Kalnis (2010) verschiedene Ansätze entwickelt, welche zum Beispiel frühere Bewegungsmuster von Benutzern mit in Betracht ziehen oder andere Gruppierungen der Positionsdaten vornehmen. In Nergiz u. Atzori (2008) wurde die Trajektorien k-Anonymität definiert, wobei eine Trajektorie innerhalb

einer Anfrage an eine ortsbasierte Anwendung nicht von $k-1$ anderen Trajektorien zu unterscheiden sein soll. Um dies mittels eines naiven Ansatzes zu gewährleisten, müsste man die Regionen, mit denen Anfragen gestellt werden, so groß wählen, dass k Trajektorien enthalten sind. Dies würde jedoch die Genauigkeit der Ergebnisse und die Laufzeit des Verfahrens stark verschlechtern. Ein vielversprechender Lösungsansatz wird in Shin u. a. (2010) beschrieben: Eine Trajektorie wird hier in Partitionen unterteilt, um die Anzahl der Regionen zu minimieren und dennoch die k-Anonymitätsanforderung für Trajektorien zu erfüllen. Durch das Finden einer optimalen Partitionierung soll so die Geschwindigkeit und die Sicherheit des Verfahrens verbessert werden.

Die meisten k-Anonymitäts Ansätze setzen eine vertrauenswürdige Instanz voraus, welche die Anonymisierung vornimmt. Sämtliche Informationen, die ein Benutzer an eine ortsbasierte Anwendung geben möchte, werden über diese Instanz ausgetauscht. Außerdem wird in existierenden Ansätzen für kontinuierliche Lokationsupdates die Genauigkeit der Trajektorien verringert, was sie für manche Anwendungen unbrauchbar macht. Andere Ansätze erhalten zwar die Genauigkeit der Positionsinformationen, können die Privatheit eines Benutzers aber nur gewährleisten, wenn viele andere Benutzer ebenfalls im System und in der der Nähe sind. Der k-Anonymitäts Ansatz garantiert jedoch für jeden Teilnehmer einen definierten Grad an Anonymität, der durch den Parameter k beeinflusst werden kann.
Unser Verfahren soll ebenfalls eine definierte Privatheit garantieren, dabei jedoch unabhängig von einer vertrauenswürdigen Instanz agieren. Außerdem soll die Genauigkeit der Positionsinformationen den Anforderungen der Anwendungen genügen.

2.2.2 Mix Zones

Das Modell der Mix Zones wird in Beresford u. Stajano (2004) beschrieben. Bei einer Mix Zone handelt es sich um einen vom Benutzer vorgegebenen geografischen Bereich, in dem die Position eines Benutzers nicht von einer Anwendung verfolgt werden kann. Betritt er eine Mix Zone, so wird sein Pseudonym mit einem noch nicht verwendeten getauscht, unter dem die Daten nach dem Verlassen der Mix Zone herausgegeben werden. Das Pseudonym schützt die Identität eines Benutzers, indem es bei jedem Eintritt in eine Mix Zone geändert wird und so eine Korrelation der Abschnitte vor und nach einer Mix Zone verhindert. Dies gelingt nur, wenn sich

in einer Mix Zone zur gleichen Zeit mehrere Benutzer befinden, die beim Verlassen mit dem verfolgten Benutzer verwechselt werden können. Um die Sicherheit des Verfahrens zu verbessern, werden mehrere Mix Zones verwendet.
Der Ansatz versucht so, das Verfolgen eines Benutzers auf kurze Zeiträume zu beschränken. Die Trajektorie eines Benutzers wird in verschiedene Segmente zwischen den Mix Zones geteilt, wobei jedes Segment ein eigenes Pseudonym erhält, das nicht eindeutig mit den Pseudonymen der anderen Segmente in Verbindung gebracht werden kann.
Um die Korrelation zwischen zwei Pseudonymen weiter zu verringern, können Daten innerhalb der Mix Zone ausgewertet werden. Damit kann dann die Wahl des nächsten Pseudonyms beeinflusst werden. Diese Daten können zum Beispiel frühere Bewegungsinformationen von Benutzern sein, die eine zukünftige Fahrtrichtung bestimmen lassen oder Aufschluss über viel verwendete Routen geben.
Wie bei der k-Anonymität wird auch bei diesem Ansatz die Existenz einer vertrauenswürdigen Instanz vorausgesetzt, welche zwischen den mobilen Objekten und den nicht vertrauenswürdigen Anwendungen vermittelt und die Pseudonymänderung durchführt. Von der Größe einer solchen Mix Zone hängt ab, wie viele Benutzer sich zur selben Zeit in der Zone befinden bzw. sie wieder verlassen und dadurch miteinander verwechselt werden könnten. Allerdings beeinträchtigt die Größe eines Bereichs auch zugleich die Genauigkeit der Positionserfassung, da innerhalb einer Mix Zone keine Positionsinformationen herausgegeben werden.
Entscheidend bei diesem Verfahren ist die Anzahl der Benutzer, die sich zur selben Zeit in einer Mix Zone befindet. Im Idealfall lässt sich ein Benutzer beim Verlassen einer Mix Zone nicht mehr von n anderen Benutzern, die sich zur selben Zeit dort befunden haben, unterscheiden. Die Zahl der anderen Benutzer ist allerdings nicht vorhersehbar. Die Sicherheit eines Benutzers kann also nicht garantiert werden, ohne die Qualität der Positionsinformationen stark einzuschränken.
MobiMix, eine Umsetzung des Mix Zones Ansatzes, findet sich bei Palanisamy u. Liu (2011). Sie übertragen das Konzept auf Straßennetze, statt einfache geometrische Bereiche zu bilden und verwenden zusätzliche Kontextinformationen, um das Verfahren weiter zu verbessern.

In den grundlegenden Anonymisierungsverfahren soll die Identität eines Benutzers unter allen Umständen geschützt werden. Erreicht das Verfahren dieses Schutzziel, so kann eine Anwendung die Positionsinformationen keinem Benutzer zuordnen. Es werden also Anwendungen ausgeschlossen, die einer Identifikation bedürfen. Man

stelle sich z.B. eine Anwendung vor, welche die Fahrstrecken seiner Außendienstfahrzeuge für ein Fahrtenbuch automatisch aufzeichnet. Eine solche Anwendung erfordert eine Zuordnung der Positionsinformationen zu den jeweiligen Fahrzeugen und würde darum nicht mit anonymisierten Daten arbeiten können.

2.3 Positionsverschleierung

Sowohl die Anonymisierungs- als auch die Verschleierungsansätze versuchen, durch das Verbergen von Informationen die Privatheit eines Benutzers sicherzustellen. Der Unterschied liegt darin, dass die Anonymisierung versucht, die Identität eines Benutzers zu verbergen, während eine Verschleierung lediglich die räumlichen und zeitlichen Informationen betrifft. Die Identität eines Benutzers ist bekannt und kann verwendet werden, um gezielte Anfragen zu bearbeiten.
Die Grundidee der Positionsverschleierung versucht Privatheit zu schaffen, indem die Genauigkeit der Positionsinformationen verringert wird. Es wird also die Qualität der Daten reduziert. Dies kann auf verschiedene Arten geschehen:

- Durch Erweiterung der Position auf einen Bereich,
- durch eine Transformation der Positionsinformation,
- die Herausgabe von mehreren unvollständigen Teilinformationen
- oder das Hinzufügen zusätzlicher Daten

In der Literatur werden hierbei verschieden Ansätze verfolgt, die zum Teil auf Trajektorien übertragen werden können.

2.3.1 Verschleierung durch Bereiche

Ein intuitiver Ansatz, um die räumlichen und zeitlichen Informationen zu schützen ist, sie nur ungenau als Bereich zur Verfügung zu stellen. Dabei wird also nicht die genaue Position, sondern lediglich ein möglicher Bereich herausgegeben, in dem sich der Benutzer befindet. Die von einer Anwendung zurückgelieferten Daten müssen anschließend gefiltert werden. Im Unterschied zu k-Anonymität werden die Bereiche jedoch nicht abhängig von der Anzahl der Benutzer gewählt. In Ardagna u. Cremonini (2007) werden hierfür runde Bereiche statt der eigentlichen Position eines Benutzers herausgegeben.

Eine klassische Verschleierung durch Bereiche kann vom Angreifer durch Abbildung der Positionsinformationen auf eine Karte, einem sogenannten Map Matching, angegriffen werden. Dies ermöglicht ihm die Anzahl der alternativen Positionen auf einige wenige, durch das Straßennetz vorgegebene zu beschränken. Um dies zu verhindern, verwenden Duckham u. Kulik (2005) sogenannte *Verschleierungsgraphen*, die den Bereich anhand des Straßennetzes wählen. Damit sind in dem Bereich nur tatsächlich alternative Positionen und ein Angreifer kann mittels eines Map Matchings keine alternativen ausschließen.

Im Gegensatz zu den bereits vorgestellten Verfahren kann dieses Verfahren komplett auf dem mobilen Objekt ausgeführt werden und ist nicht auf eine vertrauenswürdige Instanz angewiesen. Das mobile Endgerät muss die Ergebnisse von Anfragen filtern. Anwendungen, die nicht auf dem mobilen Gerät ausgeführt werden, sind dazu allerdings nicht in der Lage und können nur mit den ungenauen Bereichsinformationen arbeiten.

2.3.2 Verschleierung durch Transformation

Um die räumlichen Informationen zu schützen, können auch einfache Transformationen ausgeführt werden. Es werden also nur Informationen übermittelt, die in Abhängigkeit zur eigentlichen Position stehen. Beschrieben wird ein solches Vorgehen in Gutscher (2006). Andere Ansätze übertragen die Positionen relativ zu einem bestimmten *Anker*, der benötigt wird, um die Positionsinformationen zu rekonstruieren.
Bei diesen Ansätzen sind jedoch Anfragen auf bestimmte Bereiche problematisch, da sie durch die Transformation verfälschte Daten enthalten können. Einer Anwendung ist es jedoch möglich, mit Hilfe einer Zusatzinformation die tatsächliche Position zu rekonstruieren.

Um die bisher beschriebenen Ansätze auf Trajektorien zu übertragen, muss zusätzlich zu den räumlichen Informationen auch die zeitliche Komponente betrachtet werden.

2.3.3 Position Sharing

In Wernke (2012) wird ein Verfahren beschrieben, das sich an der krytographischen Methode des Multi-Secret Sharings orientiert. Hierbei werden präzise Positionsin-

formationen auf verschiedene sogenannte Shares aufgeteilt. Diese Shares werden dann auf unterschiedliche Server verteilt. Jeder der Shares hat nur eine begrenzte Genauigkeit. Möchte eine ortsbasierte Anwendung die exakte Position rekonstruieren, so muss sie dafür mehrere Shares kombinieren. Der Vorteil des Position Sharing gegenüber anderen Verschleierungen ist, dass keine Informationen verloren gehen und die tatsächliche Position durch Kombination der Shares rekonstruiert werden kann. Gleichzeitig wird keine vertrauenswürdige Instanz in Anspruch genommen, denn die Aufteilung einer Positionsinformation in Shares wird direkt auf dem mobilen Gerät durchgeführt.
Dieser Ansatz kann auch auf mehrere Positionen erweitert werden, es werden also nicht nur Einzelpositionen betrachtet. Allerdings betrachtet dieser Ansatz keine kontinuierlichen Positionsinformationen und lässt sich nicht auf Trajektorien anwenden. So wäre es möglich, durch das Nachverfolgen der Shares und unter Nutzung von Karteninformationen die möglichen Positionen einzuschränken und die tatsächliche Position anzunähern. Darüber hinaus müssten für jede neue Position mehrere Shares erzeugt und diese auf mehrere Server verteilt werden, was je nach Anzahl der Shares einen hohen Kommunikationsaufwand verursachen kann.

2.3.4 Dummy-Ansatz

In den sogenannten Dummy-Ansätzen werden gemeinsam mit der eigentlichen Position des Benutzers zusätzlich berechnete Positionsinformationen übertragen. Es wird demnach mit ungenauen Informationen gearbeitet, wobei ein Angreifer nicht zwischen der tatsächlichen Position eines Benutzers und den berechneten unterscheiden kann. Zur Berechnung realistischer Dummies können zum Beispiel frühere Dummy Trajektorien oder Bewegungsmuster anderer Benutzer verwendet werden. In vielen Ansätzen wird dann eine vertrauenswürdige Instanz verwendet, welche die Berechnung der Dummies und die Herausgabe der Daten an die entsprechenden Anwendungen übernimmt, denn mobile Geräte wären mit den erforderlichen Berechnungen stark belastet. Die Ergebnisse einer Anwendung werden später beim mobilen Gerät oder der vertrauenswürdigen Instanz gefiltert, um das Resultat der Anfrage für die eigentliche Position zu erhalten.

Werden die Bewegungsdaten jedoch über einen längeren Zeitraum hinweg erfasst, können die tatsächlichen Trajektorien eines Benutzers von den Dummies unterschieden werden. Mit jedem ausgeschlossenen Dummy verringert sich die Privat-

heit eines Benutzers gravierend, da der Angreifer mit höherer Wahrscheinlichkeit die tatsächliche Trajektorie identifiziert und so in den Besitz der exakten Positionsinformationen gelangt. Es gibt allerdings Mechanismen (vgl. You u. Peng (2007)), die versuchen, realistische Bewegungsmuster für Dummies zu generieren und so eine Identifikation der Dummies zu verhindern.

2.4 Kryptographische Verfahren

Neben den bereits besprochenen Verfahren wäre es denkbar Verschlüsselungsmechanismen zu nutzen, um Positionsinformationen vor unberechtigten Dritten zu schützen. Hierfür eignen sich vor allem asymmetrische Kryptosysteme, bei denen für die Ver- und Entschlüsselung unterschiedliche Schlüssel verwendet werden. Ein öffentlicher Schlüssel wird für die Verschlüsselung genutzt, anschließend kann die verschlüsselte Information nur mittels des dazugehörigen privaten Schlüssels wieder entschlüsselt werden. Mit einer solchen kryptographischen Methode ist es möglich, Positionsinformationen mit dem öffentlichen Schlüssel einer ortsbasierten Anwendung zu verschlüsseln. Anschließend kann die Information direkt oder über eine dritte Instanz an die ortsbasierte Anwendung übergeben werden. Diese ist nun in der Lage, die Positionsinformation mittels des privaten Schlüssels zu entschlüsseln. Die Verschlüsselung schützt dabei die Information vor unbefugtem Zugriff. Um die Ressourcen des Endgeräts zu schönen, welches die Positionsinformationen aufzeichnet, ist es wünschenswert, ein Lokationsupdate nur einmalig an einen Server zu übertragen. Es wird also einmalig verschlüsselt und sämtliche Instanzen, die in Besitz des zugehörigen privaten Schlüssels sind, können die Informationen entschlüsseln. Es muss also sichergestellt werden, dass der private Schlüssel nur berechtigten Instanzen bekannt ist.
Unter Verwendung eines solchen Verfahrens ist es allerdings nicht mehr möglich, Berechnungen auf den verschlüsselten Daten durchzuführen. Infolgedessen können die Positionsinformationen nicht anonym verwendet werden, um beispielsweise Verkehrsinformationen zu gewinnen. Es müssen außerdem immer die vollständigen Daten abgerufen werden, da es einer dritten Instanz, die zur Datenhalten verwendet wird, nicht mehr möglich ist, diese zum Beispiel nach geographischen Bereichen zu filtern.

3 Systemmodell

In diesem Kapitel sollen die Rollen innerhalb des Systems geklärt werden. In der Praxis kann das System unterschiedliche Ausprägungen aufweisen, weshalb ein möglichst allgemeines Modell verwendet wird, in dem Rollen von unterschiedlichen Instanzen übernommen werden können. Jede Rolle übernimmt dabei spezifische Aufgaben und muss hierfür bestimmte Eigenschaften erfüllen.

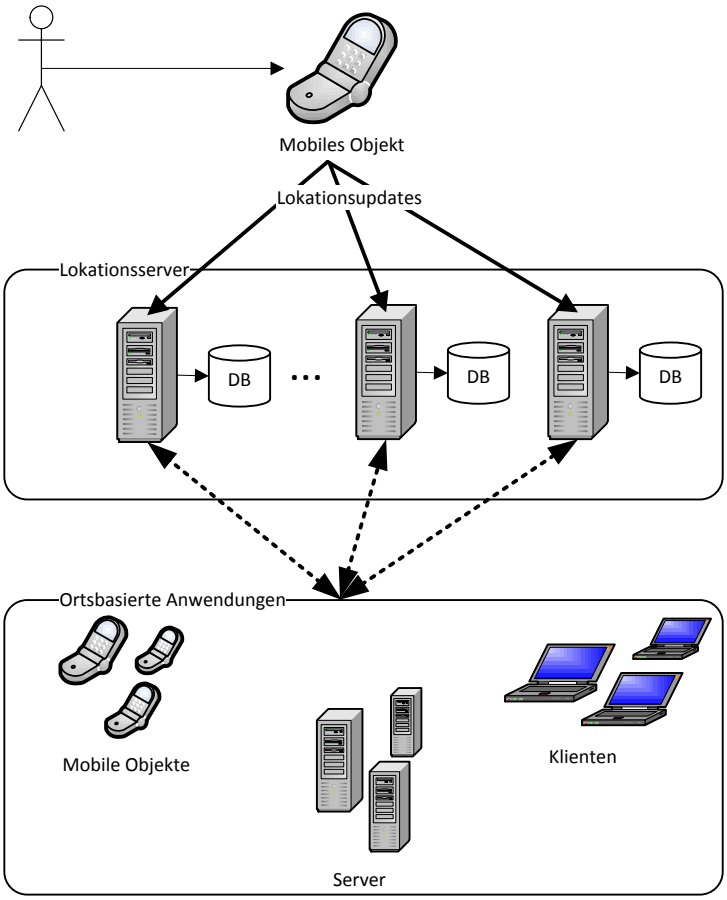

Abbildung 3.1: Architektur des Systems

Zwischen den unterschiedlichen Rollen des Systems werden Lokationsupdates ausgetauscht. Ein *Lokationsupdate* ist ein Datum, das vom mobilen Objekt an verschiedene Lokationsserver gesendet und dort gespeichert wird. Es besteht aus einer Position p mit einer geographischen (lat, lon) und einer Zeitkomponente t, $p = (lat, lon, t)$ und einem Identifikator für das mobile ObjektID_{MO}. Für die Identifikation eines mobilen Objekts wird im späteren Verfahren ein Pseudonym verwenden, um eine direkte Identifikation auszuschließen. Ein Lokationsupdate ist also ein Tupel (ID_{MO}, p).
Das System besteht hauptsächlich aus drei Komponenten:

- Das *mobile Objekt* sammelt Positionsinformationen und gibt sie in Form von Lokationsupdates an mehrere Lokationsserver weiter.
- Die *Lokationsserver* halten die Lokationsupdates persistent in einer Datenbank und ermöglichen den Zugriff darauf.
- Unterschiedliche *ortsbasierte Anwendungen* verwenden die Lokationsupdates, um eine bestimmte Funktion zu erfüllen.

Abbildung 3.1 zeigt die einzelnen Rollen und ihre zusammenhänge, die im Folgenden näher beschrieben werden.

3.1 Mobiles Objekt

Die Rolle eines mobilen Objekts können verschiedene Endgeräte einnehmen. Dies können zum Beispiel Smartphones, Navigationssysteme oder integrierte Systeme sein. Das mobile Objekt kann von einem Benutzer zu verschiedenen Zwecken eingesetzt werden. Es werden folgende Funktionen eines mobilen Objekts angenommen:

- Positionsbestimmung: Das mobile Objekt muss in der Lage sein, seine Position zu bestimmen. Hierzu können unterschiedliche Verfahren verwendet werden. Die höchste Genauigkeit lässt sich über ein Globales Navigationssatellitensystem (GNSS) erzielen, weshalb diese Methode am häufigsten genutzt wird. Das am weitest verbreitete System, ist das Global Positioning System (GPS) der Vereinigten Staaten von Amerika.

- Mobile Datenverbindung: Einem mobilen Objekt muss es möglich sein, die gesammelten Positionsdaten über eine Datenverbindung an einen Server zu senden. Den meisten mobilen Geräten ist es möglich, eine TCP/IP Verbindung aufzubauen.
- Rechenleistung: Um unser Verfahren durchzuführen, muss das mobile Objekt in der Lage sein, einfache Berechnungen auszuführen. Smartphones, Navigationssysteme und andere mobile Geräte sind heute mit starken Prozessoren ausgestattet, die es ihnen ermöglichen, mit möglichst geringem Ressourcenverbrauch auch aufwendige Berechnungen durchzuführen. Außerdem sind diese Endgeräte mit ausreichend Speicher ausgestattet, um die für die Berechnungen notwendigen Datenstrukturen aufzubauen.

Damit ist ein mobiles Objekt in der Lage, seinen Standort zu bestimmen und eigene Berechnungen auf dieser Positionsinformation durchzuführen. Es führt das Verfahren aus und sendet schließlich ein Lokationsupdate an einen Lokationsserver. Außerdem bietet es die Möglichkeit, mit dem Benutzer zu interagieren, um beispielsweise an zusätzliche Informationen zu gelangen.

3.2 Lokationsserver

Um die Positionsupdates der mobilen Objekte zu speichern und bereitszustellen, werden spezielle Server benötigt. Auf einem solchen Server werden die Positionsupdates in einer Datenbank verwaltet. Möchte eine Anwendung Zugriff auf Positionsinformationen erlangen, müssen diese über eine Query angefragt werden. Der Lokationsserver verarbeitet diese Anfrage, indem die gespeicherten Positionsinformationen gefiltert werden und sendet die Ergebnisse zurück an den Absender der Query. Ein Lokationsserver wird eingesetzt, um mehreren Anwendungen gleichzeitig dieselben Positionsinformationen zu Verfügung zu stellen und die Kommunikation mit dem mobilen Gerät zu verringern. Somit werden dessen Ressourcen geschont.
Ein Beispiel für die Umsetzung dieses Prinzips ist Yahoo Fire Eagle. Die Server dieser Plattform verwalten nicht nur die Positionsinformationen, sondern auch die Zugriffsberechtigungen für verschiedene Anwendungen auf die Daten eines Benutzers.

3.3 Ortsbasierte Anwendungen

Die Lokationsupdates eines mobilen Objekts sollen von Anwendungen verarbeitet oder durch bestimmte Dienste genutzt werden können. Dazu fragen sie die entsprechenden Daten bei den Lokationsservern ab. Diese Rolle des Systems wird in der Literatur häufig als LBS (engl. location based service), also ortsbasierter Dienst, bezeichnet. Im weiteren Verlauf der Arbeit soll jedoch der allgemeinere Begriff der ortsbasierten Anwendung verwendet werden. Eine ortsbasierte Anwendung oder LBA (engl. location based application) kann entweder direkt vom Benutzer eines mobilen Objekts genutzt werden oder von Dritten. Dabei lassen sich verschiedene Arten von LBAs anhand der von ihnen verwendeten Informationen unterscheiden:

- Anonyme Anwendungen: Eine LBA, die nur auf Basis der Positionsinformationen arbeitet und dabei nicht die Identität eines mobilen Objekts kennen muss, wird als anonym bezeichnet. Ein Beispiel für einen solchen Dienst ist eine Stauerkennung, die nur Positions- und Geschwindigkeitsinformationen nutzt, um Staus vorherzusagen. Ein solcher Dienst benötigt keinen Zugriff auf die gesamte Trajektorie, sondern lediglich auf einzelne Punkte bzw. Teile davon.
- Anwendungen mit Identifikation: Um eine Identifikation eines mobilen Objekts durch eine LBA zu ermöglichen, bedarf es meist einer Berechtigung durch das mobile Objekt bzw. den Benutzer. Erhält ein Dienst diese Berechtigung, wurde er autorisiert, die Daten des mobilen Objekts direkt abzufragen und zu verarbeiten. Ein Beispiel ist die Überwachung von Taxis eines Unternehmens. Will das Unternehmen die Bewegungen des Taxis mitverfolgen, so muss die entsprechende Anwendung in der Lage sein, die Positionsinformationen den jeweiligen Fahrzeugen zuzuordnen.

Eine LBA kann die Positionsinformationen je nach Verwendungszweck unterschiedlich abfragen. Wir unterscheiden zwischen der Anfrage über eine bestimmte ID, unter der ein mobiles Objekt seine Lokationsupdates sendet und einer ortsbasierten Abfrage, in der eine LBA die Lokationsupdates für einen bestimmten räumlichen bzw. zeitlichen Bereich anfragt.

Es ist möglich, eine LBA auf unterschiedlichen Instanzen zu betreiben. Sie kann direkt auf dem Lokationsserver, dem mobilen Objekt oder einer beliebigen dritten Instanz ausgeführt werden. Um einer LBA Berechtigungen für eine Identifikation

zu erteilen, muss es einem mobilen Objekt möglich sein, zu der jeweiligen Instanz Kontakt aufzunehmen.

4 Angreifermodell

Wie in Wernke u. a. (2012) beschrieben, sind die Annahmen darüber, wie ein Angreifer vorgeht und welche Hilfsmittel ihm zur Verfügung stehen, unbedingt beim Entwurf eines Verfahrens zu beachten. Von diesen Annahmen hängen schließlich die Schutzziele ab, die ein Verfahren erfüllen soll.

4.1 Annahmen über einen Angreifer

Angenommen, dass einem Angreifer das Vorgehen unseres Verfahrens bekannt ist, so ist er in der Lage, Entscheidungen, die aufgrund feststehender Größen innerhalb unseres Verfahrens getroffen werden nachzuvollziehen. Des weiteren verfügt der Angreifer über Karteninformationen, die es ihm ermöglichen, den Bewegungsbereich eines Zielobjektes stark einzuschränken und eventuell Vorhersagen über den zukünftigen Verlauf einer Trajektorie zu tätigen. Dazu lassen sich zum Beispiel Tempolimits für Abschätzungen der Geschwindigkeit nutzen. Der Angreifer ist außerdem in der Lage dieselben Lokationsserver zu erreichen wie ein Benutzer. Es ist einem Angreifer außerdem möglich diese Server durch gezielte Angriffe zu kompromittieren, um an die dort verfügbaren Informationen zu gelangen.
Darüber hinaus wäre es einem Angreifer möglich, Informationen aus zusätzlichen Quellen, wie beispielsweise die eines Telefonbuchs, Öffnungszeiten von öffentlichen Einrichtungen oder Verkehrsinformationen zu nutzen. Um auf solche zusätzlichen Informationen reagieren zu können, kann unser Basisverfahren erweitert werden.

Es wird davon ausgegangen, dass ein Angreifer nicht in der Lage ist, Informationen direkt vom mobilen Gerät des Benutzers zu erlangen, da die Möglichkeit besteht, das Gerät entsprechend vor Zugriffen von außen zu schützen (siehe Gilbert u. a. (2010)).
Sowohl die Lokationsserver als auch die LBAs werden zunächst als nicht vertrauenswürdig betrachtet. Eine LBA kann jedoch Zugriffsberechtigungen erhalten und

so vom Benutzer als vertrauenswürdig eingestuft werden. Vertrauenswürdig sind demnach das mobile Objekt, welches unter der Kontrolle des Benutzers steht und die durch den Benutzer ausgewählten LBAs.
Ein Schwachpunkt vieler Verfahren ist die Annahme, dass ein Lokationsserver als vertrauenswürdige Instanz verwendet werden kann, der zum Beispiel eine Anonymisierung oder Verschleierung durchführt. Ein Lokationsserver kann vom Angreifer kompromittiert werden, dies belegen Berichte des Privacy Rights Clearinghouse (2012) über zahlreiche erfolgreiche Angriffe, bei denen private Daten entwendet wurden. Ein Lokationsserver wird darum als nicht vertrauenswürdig betrachtet. Er muss sowohl für Anwendungen als auch für mobile Objekte zugänglich sein und kann somit auch von einem Angreifer erreicht werden. Wird ein Lokationsserver schließlich kompromittiert, ist der Angreifer in der Lage, sämtliche dort gespeicherten Daten zu verwenden. Darüber hinaus nimmt er Kenntnis von jeder an den Lokationsserver gestellten Anfrage, was ihm eventuell zusätzliche Informationen, wie die von einem Benutzer verwendeten Anwendungen, bringt. Es ist deshalb notwendig, alle Daten, die ein mobiles Objekt verlassen, kritisch zu betrachten.

Gelangt der Angreifer über einen Lokationsserver an dessen Positionsinformationen, kann er diese mittels eines Map Matching Algorithmus auf die Karte abbilden. Durch den Verlauf der kontinuierlichen Positionsinformationen kann ein Angreifer auf die Identität des Benutzer schließen und Annahmen über den bisherigen und weiteren Verlauf einer Trajektorie treffen. Erhält er zum Beispiel mindestens zwei aufeinanderfolgende Positionsupdates, so kann er aus diesen die ungefähre Geschwindigkeit, die Bewegungsrichtung und die gefahrene Straße ermitteln. Darüber hinaus könnte er eine Vermutung anstellen, in welchem Bereich der Trajektorie sich die Positionsinformationen befinden. Während sich ein mobiles Objekt, das in ein Wohngebiet fährt, eventuell in der Nähe eines möglichen Ziels befindet, wird ein mobiles Objekt, das sich auf einer Autobahn bewegt, noch eher darauf eine längere Strecke zurücklegen. Hierbei hilft dem Angreifer die Annahme, dass sich mobile Objekte immer auf schnellsten Wegen bewegen. Über diese Annahme lassen sich alternative Ziele ausschließen, bei denen die bekannten Positionen nicht auf einem schnellsten Weg zur jeweiligen Alternative befinden. Diese Informationen ermöglichen es dem Angreifer, Abschätzungen über die gesamte Trajektorie anzustellen. Je nach Stärke des Angreifers kann er diese Abschätzungen unter Verwendung zusätzlicher Informationen weiter verbessern.

4.2 Schutzziele

Bei Verfahren zum Schutz der Lokationssicherheit werden laut Wernke u. a. (2012) immer gewisse Schutzziele festgelegt, diese hängen von den Annahmen über einen möglichen Angriff ab. Die Schutzziele stehen in engem Zusammenhang zueinander und können nach den betreffenden Informationen kategorisiert werden.

- Schutz der Identität: Die Identität eines Benutzers ist das Schutzziel sämtlicher Anonymisierungsverfahren.
- Schutz der zeitlichen Information: Über die zeitlichen Informationen kann ein Angreifer viele zusätzliche semantische Informationen erhalten. Die zeitlichen Informationen sind bei kontinuierlichen Positionsinformationen schwer zu schützen, wenn es sich um immer aktuelle Daten handelt. Sie können z.B. auch aus Zeitstempeln der Datenübertragung abgeleitet werden.
- Schutz der räumlichen Information: Die geografische Lage einer Information steht im Mittelpunkt der Positionsverschleierung. Das angestrebte Verfahren wird sich ebenfalls auf dieses Schutzziel konzentrieren.

Die einzelnen Informationen stehen in einem engen Zusammenhang. So können beispielsweise von zeitlichen Informationen genauere räumliche Informationen hergeleitet werden, da eine obere Schranke für die zurückgelegte Strecke gegeben ist und damit eine Menge von alternativen Aufenthaltsorten. Ein weiteres Beispiel für diesen engen Zusammenhang ist das Herleiten der Identität aus bestimmten räumlichen und zeitlichen Informationen. Der Ort, an dem sich ein Benutzer nachts aufhält, kann zum Beispiel mit großer Wahrscheinlichkeit als seine Heimatadresse angenommen werden.
Im Umgang mit kontinuierlichen Positionsinformationen muss zusätzlich die Korrelation der einzelnen Informationen betrachtet werden. Durch frühere Positionen lassen sich zum Beispiel unter der Annahme, dass sich Benutzer auf schnellsten Wegen bewegen, mögliche zukünftige Positionen ausschließen oder mit unter unterschiedlichen Pseudonymen veröffentlichte Positionsinformationen in Verbindung bringen.

Der Schwerpunkt des späteren Verfahrens liegt auf dem Schutz der räumlichen und zeitlichen Informationen. Beide Informationen müssen in kombination betrachtet werden, da sie unter Verwendung kontinuierlicher Positionsinformationen nicht ohne weiteres zu trennen sind.

Zum Schutz der Identität eines Benutzers existieren bereits einfache Anonymisierungsverfahren, die als Erweiterung für das Verfahren verwendet werden können, jedoch sollen diese nicht im Mittelpunkt stehen.

5 Verfahren

In diesem Kapitel wird das grundlegende Verfahren erläutert, um die räumlichen Informationen einer Trajektorie zu schützen. Zunächst wird der Begriff Trajektorie definiert:

Eine *Trajektorie* (lat. Bahnkurve) beschreibt in der Physik die Bewegung eines Punktes und wird als Funktion in Abhängigkeit der Zeit verstanden. In unserem Fall unterscheiden wir zwischen der tatsächlichen Trajektorie, welche die exakte Bewegung eines mobilen Objekts beschreibt, und der in unserem Verfahren verwendeten diskreten Trajektorie. Die Position wird nicht permanent aufgezeichnet, sondern lediglich an einigen Punkten, an denen die Sensoren des mobilen Objekts die Position bestimmen. Es ist also nur möglich, die tatsächliche Trajektorie anhand der Punkte der diskreten Trajektorie zu approximieren.
Ein Punkt bzw. eine Position p_i ist gegeben durch seine geographischen Koordinaten, bestehend aus geographischer Breite (engl. latitude) lat_i und Länge (engl. longitude) lon_i. Zusätzlich erhält jeder Punkt einen Zeitstempel t_i.
Ein Punkt ist also gegeben durch: $p_i = (lat_i, lon_i, t_i)$. Wir definieren nun eine Trajektorie T wie unter anderem in Chow (2011) als geordnete Menge von n Punkten p, die zeitlich geordnet sind:

$$T = \{p_i | t_i < t_{i+1}, i \in 1, \ldots, n\}$$

Durch unser Verfahren wird eine Trajektorie in mehrere Fragmente geteilt. Wir definieren ein *Fragment* F als geordnete Teilmenge einer diskreten Trajektorie T, wobei alle Fragmente einer Trajektorie disjunkt sind:

$$T = \bigcup_{j=1}^{m} F_j, \qquad F = \{p_k, \ldots, p_l | p_k, \ldots, p_l \in T \wedge t_k < t_{k+1} < \ldots < t_l\}$$

$$\forall F_q, F_p \subset T : F_q \cap F_p = \emptyset$$

Ziel des Verfahrens ist es zunächst sicherzustellen, dass ein Angreifer, welcher in Besitz eines Fragments gelangt ist, nicht mehr als die Informationen dieses Fragments mit einer definierten Wahrscheinlichkeit rekonstruieren kann. Damit wollen wir einen gewissen Grad an Privatheit garantieren.
Die Fragmente einer Trajektorie werden auf unterschiedliche Lokationsserver übertragen. Da wir voraussetzen, dass mehrere Lokationsserver existieren und dem Angreifer unbekannt ist, auf welchen ein mobiles Objekt seine Positionsdaten speichert, müsste ein Angreifer eine hohe Anzahl an Lokationsserver kompromittieren, um die Wahrscheinlichkeit, alle Fragmente einer Trajektorie zu erlangen, maximieren zu können. Abbildung 5.1 zeigt die Grundidee des Verfahrens. Die Trajektorie wird in

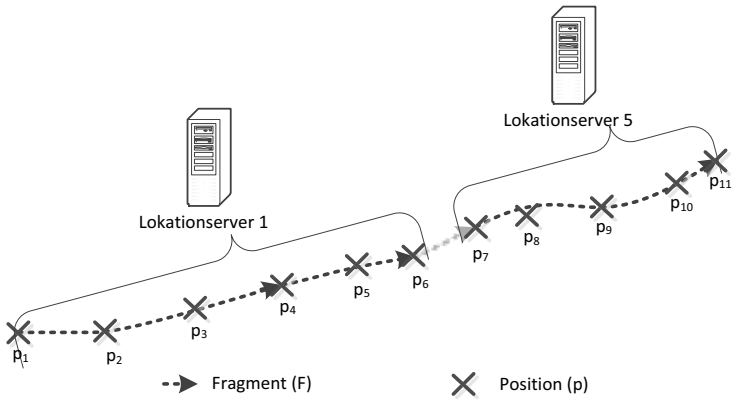

Abbildung 5.1: Prinzip der Fragmentierung und Verteilung einer Trajektorie

Fragmente zerlegt und die unterschiedlichen Fragmente werden auf verschiedene Lokationsserver verteilt. Dabei ist es prinzipiell auch möglich, dass mehrere Fragmente auf dem selben Lokationsserver abgelegt werden. Dies ist abhängig von der Art der Fragmentierung. Ein Serverwechsel wird jedoch immer am Ende eines Fragments durchgeführt, damit erhalten die Lokationsserver immer abgeschlossene Fragmente.
Die Positionen in Abbildung 5.1 repräsentieren tatsächliche Lokationsupdates. Es werden also nicht zwangsläufig alle, von den Sensoren erfassten, Positionen auch an einen Lokationsserver gesendet. Die hieraus entstehenden Abweichungen werden in der späteren Evaluierung betrachtet und als Kriterium für die Qualität des Verfahrens verwendet.
Um die Privatheit eines Benutzers zu gewährleisten, werden sowohl unterschiedliche Fragmentierungs- als auch Verteilungsstrategien betrachtet. Dabei wird die Privatheit durch den Wechsel des Lokationsservers oder das zurückhalten sensibler Positionsinformationen erhöht.

5.1 Anforderungen an das Verfahren

Da einige Anwendungen, wie beispielsweise Stauerkennungen, möglichst aktuelle Daten benötigen, ist es das Ziel, ein Verfahren zu entwickeln, welches die Lokationsupdates nur mit geringer Verzögerung an den Lokationsserver sendet. Aus diesem Grund muss es sich bei den verwendeten Algorithmen um Online-Algorithmen handeln. Es sind also zum Zeitpunkt eines Lokationsupdates keine vollständigen Daten über die Trajektorie vorhanden. Das Verfahren kann also nur auf zuvor gegebenen Informationen, der aktuellen Position und den zuvor besuchten Positionen arbeiten. Außerdem können zusätzlich bestimmte Annahmen verwendet werden, wie dass sich mobile Objekte in den meisten Fällen auf schnellsten Wegen bewegen. Sollte ein Ziel bekannt sein, kann eine Route vorberechnet werden. Allerdings muss dann auf Abweichungen von der Route geprüft und reagiert werden. Könnten wir ein Offline-Verfahren auf eine bestehende Trajektorie anwenden, wäre es möglich, eine optimale Fragmentierung im Bezug zum Informationsgehalt der einzelnen Fragmente zu finden. Dies ist in einem Online-Algorithmus aufgrund des unbekannten, zukünftigen Verhaltens eines Benutzers nicht möglich. Wir wollen jedoch versuchen, eine obere Schranke für den Informationsgehalt von Fragmenten und Mengen von Fragmenten, die sich auf einem Lokationsserver befinden, zu garantieren.
Eine weitere Anforderung, die aus einem Online-Algorithmus resultiert, ist eine akzeptable Geschwindigkeit bei den Berechnungen des Verfahrens. Eine Verzögerung durch die Berechnungen soll demnach im Rahmen von wenigen Sekunden bleiben, sodass die Benutzbarkeit einer LBA nicht durch das Verfahren eingeschränkt wird.

5.2 Gegebenheiten

Dem Verfahren liegen unterschiedliche Informationen vor, die als gegeben betrachten werden. Eine wichtige Informationsquelle sind dabei Kartendaten. Um innerhalb des Verfahrens Karteninformationen nutzen zu können, müssen diese in eine Struktur gebracht werden, welche die Zusammenhänge deutlich und Berechnungen möglich macht.
Wir betrachten eine Karte hierfür als gerichteten Graphen $G = (V, E)$, wobei V die Menge der Knoten ist und E die Menge der Kanten bezeichnet, welche die Knoten verbindet. Eine Straße ist dabei eine Teilmenge der Kanten $W \subseteq E$. Ein Knoten ist

ein Punkt, an dem sich mindestens zwei Straßen treffen und ein Wechsel der Straße möglich ist, also eine Kreuzung von Straßen. Darüber hinaus befinden sich Knoten am Anfang und am Ende einer Straße. Um den realen Verlauf einer Straße besser abbilden zu können, enthält eine Kante zusätzliche Stützstellen. Über diese kann mittels linearer Interpolation die Distanz zwischen zwei Knoten berechnet werden. Eine Straße hat zusätzlich Attribute, wie beispielsweise ob es sich um eine Einbahnstraße, einen Kreisverkehr oder eine Auf- bzw. Abfahrt handelt. Außerdem enthält sie Attribute, für das Tempolimit und die Straßenkategorie, über die, mithilfe der Distanz, eine Fahrzeit abgeschätzt werden kann.
Abbildung 5.2 veranschaulicht einen Ausschnitt eines solchen Graphen. Die roten Punkte repräsentieren dabei Knoten, während die gelben Punkte Stützstellen markieren.

Abbildung 5.2: Beispiel für Kartengraph

Laut Systemmodell (siehe Kapitel 3) nehmen wir an, dass ein mobiles Objekt in regelmäßigen Abständen über die ihm zur Verfügung stehenden Sensoren seine Position abfragt und sie an unser Verfahren übergibt. Diese Position kann eine gewisse Ungenauigkeit aufweisen. Es ist aber möglich, sie mittels eines sogenannten Map-Matching-Verfahrens auf den Kartengraph abzubilden und so den Standort des mobilen Objekts auf der Karte zu bestimmen.

Die Adressen der zur Verfügung stehenden Lokationsserver werden entweder durch den Benutzer verwaltet oder von einer zentralen Stelle bereitgestellt. Über die Adresse des Lokationsservers können anonym Lokationsupdates gesendet und bestimmte Funktionen auf dem Server aufgerufen werden.

5.2.1 Benutzerabhängige Faktoren

Die vom Benutzer abhängigen Faktoren sind zum einen sensible Orte, zum anderen Zusatzinformationen. Sensible Orte können festgelegt werden, um die Fragmentierung und Verteilung von Lokationsupdates zu beeinflussen. Zusatzinformationen können vom Benutzer zur Verfügung gestellt werden, um das Verfahren weiter zu verbessern. Da diese Informationen lediglich auf dem mobilen Objekt verwendet werden, besteht keine Gefahr, dass ein Angreifer direkt in deren Besitz gelangen könnte. Es muss jedoch gewährleistet sein, dass ein Angreifer diese Daten nicht aufgrund der Fragmentierung herleiten kann. Wenn zum Beispiel die Distanz zum Ziel unmittelbar Einfluss auf die Auf- und Verteilung hat, erlaubt dies den Ausschluss von Ziel-Alternativen und führt damit zu mehr Informationen, welche vom Angreifer verwendet werden könnten.
Es wird zunächst davon ausgegangen, dass ein Benutzer die sensiblen Orte und Zusatzinformationen während des laufenden Verfahrens nicht verändert. Sie können also von Beginn an verwendet werden.

5.2.2 Zusatzinformationen

Zur Verbesserung des Verfahrens können zusätzliche Informationen genutzt werden. Eine wertvolle Information ist dabei zum Beispiel das Ziel eines Benutzers. Liegt ein Zielort vor, so kann relativ genau abgeschätzt werden, wie lange die Trajektorie sein wird. Außerdem ist es, durch eine Schnellste-Wege-Berechnung möglich, eine Route vorherzusagen. Auf Basis dieser vorberechneten Route lässt sich dann eine mögliche Fragmentierung bestimmen. Sollte das mobile Objekt im späteren Verlauf von der Route abweichen, werden die Route und entsprechende Parameter der Fragmentierung neu berechnet. Es existieren bereits Algorithmen, die eine solche Routenberechnung in kurzer Zeit und mit geringem Ressourcenaufwand durchführen können.
Die sensiblen Orte sind ebenfalls wertvolle Informationen, die der Benutzer zur Verfügung stellen kann. Dies können zum Beispiel Heimatadresse oder die Arbeitsstelle sein. Wie in Golle u. Partridge (2009) beschrieben, kann ein Angreifer aus häufig besuchten Orten und deren Korrelation einfach Rückschlüsse auf den Wohnort und die Arbeitsstelle eines Benutzers ziehen. Ist er in den Besitz dieser Informationen gelangt, ist es ihm möglich, Anonymisierungsmechanismen zu umgehen und den Benutzer mit hoher Wahrscheinlichkeit zu identifizieren.

Außerdem gibt es weitere spezielle Orte. Wir definieren diese Orte als Points of Interest. Ein Point of Interest (POI) repräsentiert eine Position, die besondere semantische Informationen enthält. Der Wert beziehungsweise die Bedeutung eines solchen Ortes hängt stark vom Kontext ab, also davon, ob er besucht wurde, zu welcher Uhrzeit, wie lange und wie häufig. Es muss also nicht jeder POI für jeden Benutzer auch zwangsläufig als sensibler Ort gelten. Bei einem POI kann es sich zum Beispiel um Krankenhäuser, Bars oder öffentliche Gebäude handeln. Meist sind sie noch zusätzlich mit Informationen wie beispielsweise Öffnungszeiten verknüpft. Mit Hilfe der vom Benutzer besuchten POIs ist es einem Angreifer möglich zusätzliche Informationen herzuleiten, welche die Privatheit des Benutzers gefährden können. So könnte er vom Besuch eines bestimmten Arztes auf eventuelle Krankheiten des Benutzers schließen.
Die Points of Interest können in der Datenstruktur der Karte als spezielle Stützstellen auf die Kanten abgebildet werden, von denen aus sie erreichbar sind. Damit vereinfacht sich die spätere Verwendung dieser Informationen. POIs sind oft in Kartendaten enthalten oder können von öffentlichen Datenbanken abgefragt werden. Sie können mit höherer Wahrscheinlichkeit als mögliche Ziele in Betracht gezogen werden.

5.3 Fragmentierungsarten

Im ersten Schritt des vorgeschlagenen Verfahrens soll die Trajektorie in Fragmente aufgeteilt werden. Dies kann auf unterschiedliche Weise geschehen, wobei man sich verfügbare Informationen der Trajektorie oder ihres Kontextes zunutze machen kann. Es sollen im Weiteren verschiedene Ansätze diskutiert und gegeneinander abgewogen werden. Zunächst wollen wir kurz auf einfache Vorgehensweisen eingehen, hierbei wird anhand der Zeit oder der zurückgelegten Entfernung entschieden, wie ein Fragment gewählt und wann der Lokationsserver gewechselt werden soll. Anschließend werden Fragmentierungsarten betrachtet, welche zusätzliche Informationen verwenden, um einem Angreifer weniger Rückschlüsse zu erlauben.

5.3.1 Zeitbasierte Fragmentierung

Ein naiver Ansatz könnte ein Fragment nach Ablauf einer bestimmten Zeit beenden und ein Neues beginnen. Das Schutzziel hinter diesem Ansatz könnte lauten: Ein

Angreifer, soll ein mobiles Objekt nurf für eine bestimmtes Zeitintervall verfolgen können.
Im Folgenden wird davon ausgegangen, dass eine Positionsbestimmung durch das mobile Gerät in festen Zeitintervallen durchgeführt wird. Somit erhält jedes Fragment eine vorgegebene Anzahl an Punkten der Trajektorie. Da in der Realität aber für dieselbe Strecke zum Beispiel je nach Verkehrslage unterschiedlich viel Zeit benötigt wird, führt dies für dieselbe Strecke zu unterschiedlichen Fragmentierungen. Dabei wäre es denkbar, dass ein Fragment einen äußerst geringen Teil der Strecke erhält, weil der Benutzer sich beispielsweise im Stau befand, während ein anderes einen sehr großen Teil umfasst.
Die Aussagekraft verschiedener Fragmente wäre demnach sehr variabel und kann nicht zuvor bestimmt werden. Es ist also nicht möglich, eine bestimmten Grad an Privatheit zu garantieren.

Abbildung 5.3: Beispiel für Fragmentierung nach Abfahrt

Ein Angreifer, der Karteninformationen verwendet, ist in der Lage, die von ihm kompromittierten Fragmente auf die Karte abzubilden. Durch den Verlauf der Fragmente ist es ihm dann möglich, diese eindeutig weiterzuführen, bis die Trajektorie zu einem Knoten kommt, an dem alternative Entscheidungen möglich sind. Endet ein Fragment, wie in Abbildung 5.3 gezeigt, nachdem das mobile Objekt eine Autobahnausfahrt passiert hat, sich jedoch noch auf der Autobahn bewegt, kann davon ausgegangen werden, dass es sich auch bis zur nächsten Ausfahrt auf dieser Straße bewegt. Einem Angreifer wäre es also möglich, wenn er im Besitz von F_1 ist, F_2 bis zur nächsten Ausfahrt sicher herzuleiten, da es keine alternativen Wege bis dorthin gibt. Das Schutzziel lässt sich nicht einhalten, da selbst unter zusätzlicher Verwen-

dung von Karteninformationen im Verfahren ein Angreifer ein größeres Zeitintervall herleiten kann, als im Fragment enthalten ist.
Darüber hinaus könnte ein Angreifer Points of Interests identifizieren, an denen sich das mobile Objekt längere Zeit aufgehalten hat. Wenn ein mobiles Objekt zum Beispiel an einem Krankenhaus längere Zeit gehalten hat, wird das Fragment je nach gewähltem Zeitintervall an diesem Ort beendet und ein neues gestartet. Der Angreifer könnte also prüfen, ob sich ein Start- bzw. Endpunkt eines Fragments an einem solchen Ort befindet und wie lange sich ein mobiles Objekt dort aufgehalten hat. Damit könnte er den Besuch des Krankenhauses erkennen.

5.3.2 Entfernungsbasierte Fragmentierung

Fragmente können anhand von Entfernungen in der Trajektorie festgelegt werden. So könnte beispielsweise immer dann ein neues Fragment begonnen werden, wenn eine bestimmte Distanz zurückgelegt wurde. Durch dieses Vorgehen wäre es möglich zu garantieren, dass jedes Fragment genau einen festen Anteil der Strecke enthält. Auch bei diesem Ansatz erhalten wir eine Variabilität in der Aussagekraft der Fragmente, da zum Beispiel je nach Art der befahrenen Straße der Informationsgehalt stark variieren kann. Auf einer Autobahn können größere Entfernungen mit weniger Aussagekraft gebildet werden, als auf einer Straße in der Stadt, wo viele Abzweigungen und mögliche Ziele in Betracht gezogen werden könnten.
Außerdem besteht das im vorherigen Ansatz beschriebene Problem (Abbildung 5.3) hier ebenfalls, da Knoten in unterschiedlichen Abständen zueinander vorkommen. Es kann also nicht sichergestellt werden, dass ein Angreifer nur eine festgelegte Distanz durch ein Fragment herleiten kann.

5.3.3 Kartenbereichsbasierte Fragmentierung

Eine Möglichkeit, Karteninformationen für die Fragmentierung zu nutzen, wäre es, Fragmente nach vorgegebenen Bereichen auf der Karte zu bilden. Diese Bereiche können geometrisch oder aufgrund geografischer Gegebenheiten gewählt werden. So könnte man die Karte in Bereiche wie Städte, Landkreise, Bundesländer etc. aufteilen. Ein Serverwechsel wird immer an einer Bereichsgrenze durchgeführt. Ein Fragment befindet sich demnach immer innerhalb eines Bereichs. Abbildung 5.4 zeigt ein Beispiel für eine geometrische Kartenaufteilung mit Quadranten. Es wäre dann

Abbildung 5.4: Beispiel für Bereichsbasierte Fragmentierung

auch vorstellbar, dass nur gewisse Bereiche einer Trajektorie freigegeben werden, in denen eine Nachverfolgung seitens des Benutzers erwünscht ist. Schwierigkeiten entstehen bei diesem Ansatz vor allem an Übergängen von verschiedenen Bereichen. Findet solch ein Übergang nicht an einem Punkt statt, an dem alternative Wege oder Bereiche möglich wären, so kann der Angreifer, wie bei der entfernungs- und zeitbasierten Fragmentierung, eine sichere Annahme über den zukünftigen Bereich und damit den Verlauf der Trajektorie machen.
In unserem Beispiel (siehe Abbildung 5.4) könnte ein Angreifer so aus den Informationen von Fragment F_3 den Anfang von F_4 und das Ende von F_2 herleiten. Außerdem könnten die so entstehenden Fragmente, abhängig von den geografischen Gegebenheiten, unterschiedlich aussagekräftig sein. Während Fragment 2 in unserem Beispiel viele Abzweigungen zu alternative Wege hat und einen großen Teil der Strecke abdeckt, enthält Fragment 3 keine Alternativen und nur einen geringen Teil der Strecke. Es ist also schwer, eine garantierte Privatheit zu bestimmen, da sie von der jeweiligen Strecke und den geografischen Gegebenheiten abhängt.

5.3.4 Knotenbasierte Fragmentierung

Im knotenbasierten Ansatz werden Fragmente an Knoten getrennt. Befinden sich die Grenzen eines Fragments an einem Knoten, kann ein Angreifer keine sichere Aussage darüber treffen, von wo ein mobiles Objekt gekommen ist bzw. wohin es seine Trajektorie fortsetzen wird. Im Unterschied zu den zuvor beschriebenen Arten der Fragmentierung, gewinnt ein Angreifer bei einer knotenbasierten Fragmentierung, durch ein Map Matching keine eindeutigen zusätzlichen Informationen. Abbildung

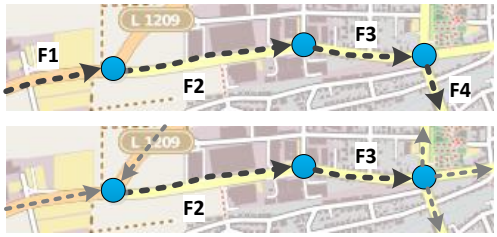

Abbildung 5.5: Fragmentierung an Knoten

5.5 zeigt ein Beispiel für eine Aufteilung nach diesem Ansatz. Der obere Teil der Abbildung zeigt die eigentliche Trajektorie, der untere einen Ausschnitt mit zwei Fragmenten der Trajektorie, in deren Besitz ein Angreifer gelangt ist und mögliche Schlüsse, die er damit über den Verlauf der Trajektorie ziehen könnte.
Die Anzahl und Größe der so entstehenden Fragmente ist von der zurückgelegten Strecke abhängig. Führt sie durch städtische Gebiete, sind die Fragmente kurz, verläuft sie hauptsächlich Überland, werden sie länger. Bei einem naiven knotenbasierten Ansatz entspricht die Anzahl der Fragmente, in die eine Trajektorie aufgeteilt wird, der Anzahl der benutzten Kanten. Diese kann abhängig von der länge der Trajektorie und den verwendeten Verkehrswegen sehr groß werden. Übersteigt die Anzahl der Fragmente, die Anzahl der Lokationsserver, so ist es notwendig mehrere Fragmente an denselben Server zu senden. Um zu entscheiden, welche Fragmente zusammengefasst werden können sind wiederum unterschiedliche Vorgehen denkbar. Ein einfacher Ansatz wäre die Bewertung und Auswahl der Knoten, an denen ein Serverwechsel durchgeführt wird. Diese Knoten können beispielsweise Anhand der sich treffenden Wege bewertet werden, da ein Angreifer ebenfalls die alternativen Wegen betrachten muss, um weitere Teile der Trajektorie herzuleiten. Es ist möglich Straßenkategorien in die Bewertung mit einfließen zu lassen, um eine feinere Unterscheidung der Knoten zu erreichen. So könnten Autobahnen höher bewertet

werden, als Landstraßen, da sie viel befahren sind und auf ihnen größere Strecken zurückgelegt werden können. Es wäre auch möglich die geltenden Geschwindigkeitslimits für eine solche Bewertung zu verwenden.

5.3.5 Bewertungsbasierte Fragmentierung

In dem von uns entwickelten Verfahren wird auf dem Knotenbasierten Ansatz aufgebaut. Jede Kante zwischen zwei Knoten wird dafür zunächst als mögliches Fragment betrachtet und bewertet. Dazu wird eine Bewertungsfunktion verwendet, welche den Informationsgehalt eines Fragments bestimmt. Dieser Informationsgehalt stellt ein Maß für die Sensibilität der Fragmentdaten dar. Überschreitet die Bewertung einen gegebenen Grenzwert, so wird der Lokationsserver gewechselt oder die Positionsinformationen eines Fragments werden nicht gesendet.
Im nächsten Abschnitt soll das konkrete Vorgehen bei der Fragmentierung beschrieben werden.

5.4 Fragmentierung

Die Fragmentierung der Trajektorie wird in zwei Phasen durchgeführt. In der ersten Phase wird bestimmt, ob die aktuelle Position noch innerhalb des bisherigen Fragments liegt. In einem solchen Fall werden die nachfolgenden Fragmente gesucht und einer Bewertung unterzogen. Sollte das nicht der Fall sein, wird das Folgefragment gewählt, auf dem sich das mobile Objekt in diesem Moment befindet. Je nach Bewertung dieses Fragments wird entschieden, ob es herausgegeben werden kann, ohne einen gewissen Grenzwert an sensiblen Informationen preiszugeben. Anschließend wird ein Lokationsserver ermittelt, an den die Lokationsupdates des Fragments gesendet werden können, ohne damit weitere Rückschlüsse zu ermöglichen. Ist für das aktuelle Fragment ein Lokationsserver gewählt und bewertet, können die erfassten Positionsinformationen zu diesem gesendet werden.

Ein wesentlicher Bestandteil des Verfahrens ist die Bewertungsfunktion, welche es ermöglicht, den Informationsgehalt von Abschnitten einer Trajektorie zu bestimmten. Sie trifft eine Aussage darüber, mit welcher Wahrscheinlichkeit ein Angreifer einen sensiblen Ort identifizieren kann. Dabei werden die vom Beginn bzw. Endpunkt eines Fragments erreichbaren Alternativen zu den sensiblen Orten gesucht.

Bei diesen Alternativen handelt es sich um Orte, die nicht aufgrund der Fragmentinformationen von einem sensiblen Ort unterschieden werden können. Die Alternativen können gefunden werden, indem Orte gesucht werden, welche als Alternative zu einem sensiblen Ort infrage kommen und den Anfangs- und Endpunkt des Fragments auf ihrem kürzesten Weg enthalten. Das Fragment würde also komplett auf einem kürzesten Pfad zu diesen Orten liegen. Die alternativen Orte können zum Beispiel Points of Interest (POI) sein. Es können aber auch generell Gebäude an einer Straße sein, die als mögliches Ziel infrage kommen.

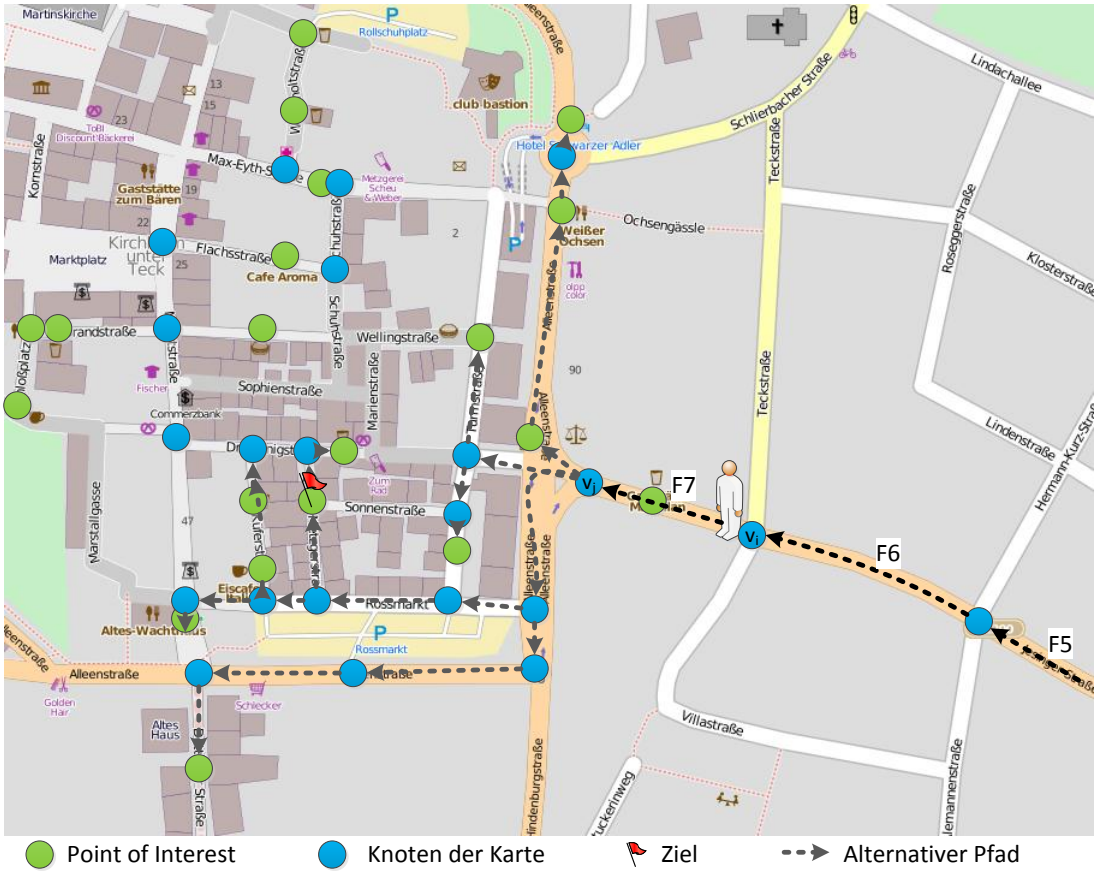

Abbildung 5.6: Beispiel für die Bewertung

Bewertungsfunktion

Die *Bewertungsfunktion* $I(F, CI) = I(v_i \dots v_j, CI)$ mit den Knoten $v_i, \dots, v_j \in V$ und der Menge sensibler Orte CI, wird über die Wahrscheinlichkeit definiert, mit der ein Angreifer, unter der Bedingung, dass er im Besitz der Positionsinformationen zwischen den Knoten $v_i \dots v_j$ ist, einen der sensiblen Orte identifizieren kann. Im Falle eines Fragments wird dabei der Anfangs- und Endknoten des Fragments als Parameter der Funktion verwendet. Bei mehreren Fragmenten wird eine geordnete Menge von Knoten übergeben. Wir betrachten in unserem Verfahren zunächst einen sensiblen Ort als Startposition CI_sder Trajektorie und einen weiteren als Ziel CI_d. Es können aber ebenso sensible Orte $CI_{m_1,\dots,m_n} = \{CI_{m_1}, \dots, CI_{m_n}\}$ von der Trajektorie passiert werden. Der Informationsgehalt bezüglich eines bestimmten sensiblen Ortes berechnet sich durch die bedingte Wahrscheinlichkeit: $Pr(CI_d|v_i \dots v_j) = \frac{1}{Anzahl(A)}$, wobei $A = \{a|a \in POI \wedge d(v_i, a) \leq \lceil d(v_i, CI_d) \rceil\}$.
Wir betrachten demnach alle Alternativen A, die in kürzerer oder der selben Distanz wie der betrachtete sensible Ort erreichbar sind. Als Distanzfunktion nehmen wir die geschätzte Zeit zum Ort und nicht die Entfernung an. Damit werden implizit verschiedene Straßenkategorien und Geschwindigkeitslimits mit einbezogen, denn über Autobahnen ist es möglich, in der selben Zeit weiter entfernte Ziele zu erreichen als zum Beispiel über Landstraßen. Die Distanz bis zum sensiblen Ort wird aufgerundet, damit wird nicht die genaue Distanz als Horizont für die Alternativen gewählt. Der Horizont der Alternativen wird lediglich aufgrund dieser Information abgeschätzt und lässt damit keine exakten Rückschlüsse zu.
Im Falle mehrerer sensibler Orte, die zwischen zwei Fragmenten liegen können, beziehen wir diese durch

$$Pr(CI_{m_1,\dots,m_n}|v_i \dots v_j) = \frac{\sum_{i=1}^{n} Pr(CI_{m_i}|v_i \dots v_j)}{n}$$

in die Bewertung mit ein. Die Gesamte Bewertungsfunktion setzt sich also folgendermaßen zusammen:

$$I(v_i \dots v_j, CI) = max\{Pr(CI_d|v_i \dots v_j), Pr(CI_{m_0,\dots,m_n}|v_i \dots v_j), Pr(CI_s|v_j \dots v_i)\}$$

Wir verwenden die Maximumsfunktion, damit Punkte mit wenig Alternativen, die von einem Angreifer mit höherer Wahrscheinlichkeit als besucht erkannt werden könnten, für die Bewertung des Fragments die größte Rolle spielen.

Abbildung 5.6 zeigt ein Beispiel einer Bewertung in Richtung Ziel. Die grauen Pfeile beschreiben dabei mögliche Alternativen zum sensiblen Ort (rotes Fähnchen). Das Fragment zwischen v_i und v_j wird bewertet. Gehen wir davon aus, dass die Trajektorie keine als sensibel gekennzeichneten Punkte enthält und der Start der Trajektorie viele Alternativen hat, so wird die Bewertung des Ziels durch die Maximumsfunktion gewählt. Es wäre also $Pr(CI_d|v_i \dots v_j) = \frac{1}{11}$, da der Angreifer aus mindestens 11 Alternativen wählen müsste, wenn es ihm gelingen würde, den Horizont richtig zu wählen. Wählt der Angreifer einen zu kleinen Horizont, schließt er das Ziel aus seiner Betrachtung aus, wählt er einen zu großen, so muss er mehr Alternativen betrachten.
Der Start und das Ziel einer Trajektorie können als sensibel angenommen werden, da diese meist höhere Bedeutung haben, als andere im laufe einer Trajektorie besuchte Orte. Weitere sensible Orte können besuchte POIs sein oder anderweitig vom Benutzer festgelegt werden.

Der Wechsel eines Servers und damit die Aufteilung der Trajektorie in unserem Verfahren hängt von zwei Schwellwerten ab. Der erste, das sogenannte *PrivacyLimit*, begrenzt die Bewertung des Informationsgehalts aller auf einem Lokationsserver abgelegten Fragmente nach oben. Damit garantiert das *PrivacyLimit* einen gewissen Grad an Privatheit. Außerdem werden Fragmente, welche höher als dieser Wert bewertet wurden, nicht herausgegeben, da sie zu viele Informationen enthalten. Das *PrivacyLimit* kann als Parameter betrachtet werden, dieser kann beispielsweise verwendet werden um dem Benutzer unmittelbaren Einfluss auf das Verfahren und die Privatheit zu geben. Damit wäre einem Angreifer dieser Wert nicht bekannt, was die Wahrscheinlichkeit, dass eine Entscheidung für einen Serverwechsel auf die Bewertung eines Fragments zurückgeführt werden kann, weiter verringert.
Der zweite Schwellwert in unserem Verfahren ist der globale Fragmenten Zähler, kurz *GFC*. Er nimmt ebenfalls direkten Einfluss auf die Fragmentierung, indem er die Anzahl der auf einem Lokationsserver gespeicherten Fragmente nach oben beschränkt. Würde man ausschließlich ein fest definiertes *PrivacyLimit* zur Prüfung eines Serverwechsels verwenden, könnte ein Angreifer aus der Entscheidung zum Wechsel eines Servers Rückschlüsse auf die Bewertung und damit den Informationsgehalt eines Fragments ziehen. Durch die Einführung des *GFC* ist diese Entscheidung nicht mehr eindeutig nachvollziehbar.

Im Basisverfahren wählen wir den *GFC* in Abhängigkeit von der Anzahl der zur Verfügung stehenden Lokationsserver L' und der geschätzten Anzahl der Fragmente

F der aktuellen Trajektorie T.

$$GFC = \frac{Anzahl(F \in T)}{Anzahl(L')}$$

Auf diese Art und Weise ändert sich der Wert des *GFC* mit jeder befahrenen Route, da sich die geschätzte Anzahl an Fragmenten ändert. Außerdem wird durch Einbeziehung der Anzahl der Lokationsserver versucht, eine ausgewogene Verteilung zu erreichen. Je größer die Anzahl der Lokationsserver, desto geringer wird der Anteil an Fragmenten, die auf einem einzelnen Server gespeichert werden.
Gehen wir davon aus, dass ein Angreifer versucht, eine Trajektorie abzuschätzen und damit eine Anzahl der Fragmente, so könnte er versuchen, den GFC ebenfalls zu ermitteln. Dabei müsste er den Fehler in kauf nehmen, den er bei dieser Abschätzung macht. Außerdem entstehen Fehler bei der Vermutung, an welcher Position sich das Fragment innerhalb der Trajektorie befindet, also wie viele Lokationsserver zu diesem Zeitpunkt in der Menge L' enthalten sind.
Möchte man die Wahrscheinlichkeit, dass ein Angreifer die Entscheidung eines Serverwechsels auf den *GFC* zurückführen kann, weiter reduzieren, wäre es auch möglich, einen Zufallswert mit einzubauen.

Algorithmus 1 zeigt das Vorgehen bei der Fragmentierung. Das Lokationsupdate wird vom Lokationssensor an den Fragmentierungsalgorithmus übergeben und zunächst mittels eines *MapMatching*-Algorithmus auf eine Kante des Kartengraphen abgebildet. Es wird geprüft, ob die so entstandene Position auf der Karte noch innerhalb des aktuellen Fragments $F_{current}$ liegt. Befindet sich die aktuelle Position außerhalb eines Sicherheitsabstandes zu Kreuzungen und Abfahrten kann gegebenenfalls ein Lokationsupdate mittels *SendUpdate* an den aktuell gewählten Lokationsserver $LS_{current}$ verschickt werden. Das Verfahren orientiert sich damit an der Idee der Virtual Trip Lines nach Hoh u. a. (2008), die ein solches Vorgehen und dessen Notwendigkeit zum Schutz gegen Angriffe auf die Lokationssicherheit beschreiben. Aufgrund von Geschwindigkeitsunterschieden an den Knoten im Kartengraphen kann die Wahrscheinlichkeit bestimmter Pfade besser abgeschätzt werden. So wird ein mobiles Objekt vor einer Autobahnausfahrt seine Geschwindigkeit stark reduzieren, wenn es die Autobahn verlassen möchte. In gleicher Weise kann eine Auffahrt durch große Beschleunigung erkannt werden. Um diese Informationen zu verbergen, prüft *InBufferDistance* den Abstand zu den nächsten Knoten und gibt bei Unterschreitung eines gewissen Schwellwerts *true* zurück. Der Schwellwert

Algorithm 1 FragmentationManagement

```
1: function FragmentationManagement(locationUpdate)
2:     mapPos ← MapMatching(locationUpdate)
3:     if mapPos ∉ Route then
4:         ReCalculateRoute(mapPos)
5:     end if
6:     if mapPos ∈ F_current then
7:         if !InBufferDistance(mapPos) then
8:             SendUpdate(mapPos, LS_current)
9:         end if
10:        Calculate I(F_next)
11:        Calculate I(LS_current.Fragments ∪ F_current)
12:    else
13:        if I(F_next) < PrivacyLimit then
14:            F_current ← F_next
15:            if LS_current.FragmentCount < GFC then
16:                if I(LS_current.Fragments ∪ F_current) < PrivacyLimit then
17:                    LS_current ← ChangeLocationServer(LS_current, F_current)
18:                    FragmentationManagement(mapPos)
19:                end if
20:            end if
21:        end if
22:    end if
23: end function
```

wird in Abhängigkeit des Tempolimits ermittelt, da die Beschleunigungs- beziehungsweise Verzögerungsbereiche von der Kategorie der Straße und der Ziel- oder Ausgangsgeschwindigkeit des mobilen Objekts abhängen. Außerdem können hier bestimmte Fragmente wie längere Ausfahrten ausgeschlossen werden. Denn mit Hilfe eines solchen Fragments könnte das vorhergehende oder nachfolgende Fragment eindeutig identifiziert werden.
Während sich das mobile Objekt auf einem bereits bewerteten Fragment befindet kann bereits das nächste Fragment und die möglichen Lokationsserver bewertet werden. Damit wird eine eventuelle Verzögerung aufgrund der Bewertungsberechnung verkürzt. Das nachfolgende Fragment kann eindeutig bestimmt werden, falls ein Ziel bekannt ist und die Route im Vorfeld berechnet wurde. Ist dies nicht der Falle kann eine Bewertung für alle alternativen Fragmente, die am nächsten Knoten möglich sind, durchgeführt werden.
Befindet sich die Position nicht mehr innerhalb des aktuellen Fragments, so wird die Bewertung des nächsten Fragments geprüft. Unterschreitet diese das *PrivacyLimit*, so können Lokationsupdates gesendet werden. Anschließend wird der Lokationsserver für die Updates innerhalb dieses Fragments gewählt. Dazu wird die Anzahl der Fragmente auf dem aktuellen Server auf eine Überschreitung des *GFC* hin untersucht und die Bewertung, unter Berücksichtigung des aktuellen Fragments, gegen das *PrivacyLimit* getestet. Sollte eine der beiden Bedingungen für den aktuellen Lokationsserver nicht zutreffen, wird mittels *ChangeLocationServer* ein neuer gewählt. Anschließend wird mit dem Algorithmus und dem neuen aktuellen Fragment fortgefahren. Die Funktion *ChangeLocationServer* ist dabei für die Verteilung der Fragmente auf den Lokationsservern zuständig. Diese wird in Algorithmus 2, im nachfolgenden Abschnitt detailliert beschrieben.

Um die Bewertung eines Fragments oder der auf einem Server befindlichen Menge von Fragmenten durchzuführen, kann ein Kürzeste-Wege-Algorithmus verwendet und modifiziert werden. Dabei werden kürzeste Wege zwischen zwei Knoten gesucht, auf denen eine bestimmte Abfolge von Knoten besucht wird. Diese Knoten sind durch die Grenzen der zu bewertenden Fragmente vorgegeben. Während der Suche werden die Alternativen gezählt, welche sich auf den Kanten eines solchen kürzesten Weges befinden. Zur Optimierung kann die Suche bereits abgebrochen werden, wenn das *PrivacyLimit* erreicht wurde. Damit wird die Suche nur vollständig durchgeführt, wenn die Lokationsupdates eines Fragments nicht gesendet werden oder ein anderer Lokationsserver gewählt werden muss.

Wird der Lokationsserver nicht gewechselt, so muss auch kein Sicherheitsabstand an Knoten eingehalten werden. Die hier zurückgehaltenen Positionsinformationen könnten problemlos aus den Informationen der angrenzenden Fragmente approximiert werden. Um die Genauigkeit bei der Rekonstruktion seitens einer ortsbasierten Anwendung zu erhöhen, wird ein Sicherheitsabstand an Knoten nur vor und nach einem Serverwechsel eingehalten.

5.5 Wahl eines Lokationsservers

Algorithm 2 ChangeLocationServer

```
1:  function ChangeLocationServer(LS_current, F_current)
2:      L' ← L \ LS_current
3:      for all L'_j ∈ L' do
4:          if I(F_current ∪ F_{L'_j}) > PrivacyLimit then
5:              L' ← L' \ L'_j
6:          else
7:              L'_j.I ← I(F_current ∪ F_{L'_j})
8:              if L'_min = null ∨ L'_min.I > L'_j.I then
9:                  L'_min ← L'_j
10:             end if
11:         end if
12:     end for
13:     if L' = ∅ then
14:         L'_min ← GetNewLocationServer()
15:         if L'_min = null then
16:             NotifyPrivacyLimitReached
17:         else
18:             L ← L ∪ L'_min
19:         end if
20:     end if
21:     return L'_min
22: end function
```

Für die Wahl eines geeigneten Lokationsservers werden die auf dem Lokationsserver bereits vorhandenen Fragmente in Bezug zum aktuellen Fragment bewertet. Ziel der Serverwahl ist es, die Fragmente so auf die Server zu verteilen, dass ein Angreifer durch Korrelation der auf dem Server befindlichen Fragmente die sensiblen Orte nur mit einer durch das *PrivacyLimit* beschränkten Wahrscheinlichkeit herleiten kann.

Algorithmus 2 zeigt das Vorgehen zur Auswahl des Servers für das übergebene Fragment $F_{current}$. Zunächst wird der bisher verwendete Lokationsserver $LS_{current}$, welcher bereits in Algorithmus 1 als Möglichkeit ausgeschlossen wurde, aus der Menge der zur Verfügung stehenden Server L entfernt. Hieraus entsteht die Menge L'. Danach werden iterativ sämtliche Server $L'_j \in L'$ ausgeschlossen, welche unter Hinzunahme des aktuellen Fragments $F_{current}$ das *PrivacyLimit* überschreiten würden. Hierfür wird die bereits beschriebene Bewertungsfunktion I verwendet. Unter den übrigen Lokationsservern wird derjenige ausgewählt, welcher die geringste Bewertung erhalten hat. Sollten alle Lokationsserver ausgeschlossen worden sein, so wird mittels der Funktion *GetNewLocationServer* ein neuer Lokationsserver zur Menge L hinzugefügt und zugleich ausgewählt. Die Menge der zur Verfügung stehenden Server wird also nur erweitert, wenn dies unbedingt notwendig ist. Wir gehen davon aus, dass zu diesem Zweck beispielsweise auf eine Datenbank aller oder der vom Benutzer erwünschten Lokationsserver zugegriffen werden kann. Befindet sich in dieser Datenbank kein weiterer Server, so kann das *PrivacyLimit* nicht eingehalten werden. Sollte dies der Fall sein, so kann der Benutzer benachrichtigt werden oder andere Maßnahmen ergriffen werden. In jedem Fall sollte die Herausgabe der Positionsinformationen zunächst gestoppt werden, damit keine sensiblen Informationen das mobile Objekt verlassen. Wir wollen zu Beginn des Verfahrens zwei Lokationsserver für die Menge L wählen, damit für kurze Trajektorien mindestens zwei Lokationsserver verwendet werden und mindestens eine Aufteilung aufgrund des *GFC* vorgenommen wird.

5.5.1 Anonymisierung und Berechtigungen

Um die Identität eines Benutzers zu schützen und die Korrelation zwischen Fragmenten unterschiedlicher Server zu erschweren, werden die Daten unter einem Pseudonym an die Lokationsserver übertragen. Eine Anwendung, die berechtigt ist Teile der Trajektorie zu verwenden, kann diese Pseudonyme mit der Identität des Benutzers in Verbindung bringen. Bei der Verwendung von Pseudonymen orientieren wir uns an Jorns u. Quirchmayr (2007). Mittels einer kryptographischen Hash-Funktion $sh()$ wird ein Pseudonym erzeugt. Die Funktion $sh()$ bekommt als Eingabe eine Zeichenfolge beliebiger Länge und generiert daraus eine Zeichenfolge fester Länge, einen sogenannten Hash-Wert, der als Pseudonym verwendet wird. Als Hash-Funktion kann zum Beispiel ein secure hash algorithm, kurz *SHA*, verwendet

werden. Diese Algorithmen sind standardisiert in der RFC6234 Spezifikation (Eastlake (2011)). Das so generierte Pseudonym wird zur Identifikation durch berechtigte Instanzen an ein Positionsupdate angehängt. Da wir verschiedene Lokationsserver verwenden wollen, werden wir für jeden Server ein anderes eindeutiges Pseudonym verwenden. Dies erschwert es einem Angreifer, zwei Fragmente auf unterschiedlichen Servern in Verbindung zu bringen. Die Fragmente müssten dann, im Falle einer Kompromittierung, über zeitliche und räumliche Informationen korreliert werden. Diese Korrelation wird ungenauer, je größer dabei die Abstände zwischen den Fragmenten werden.

Ein Pseudonym $ID_{MO}(l)$ für einen Lokationsserver $l \in L$ setzt sich aus einem öffentlichen Schlüssel des Lokationsservers k_{LS} und einem Schlüssel k_{MO}, welchen das mobile Objekt für den jeweiligen Lokationsserver bestimmt, zusammen:

$$ID_{MO}(l) = sh(k_{LS}(l) + k_{MO}(l))$$

Die beiden Schlüssel werden konkateniert und durch die Hash-Funktion in einen Hash-Wert übersetzt, aus dem sich die Schlüssel nicht oder nur mit erheblichem Aufwand rekonstruieren lassen. Man spricht bei einer solchen Funktion auch von einer Einwegfunktion, welche zwar einfach zu berechnen, jedoch nur schwer umzukehren ist. Die Sicherheit eines so erzeugten Pseudonyms ist demnach stark abhängig von der Wahl einer geeigneten Hash-Funktion.
Um die verwendeten Pseudonyme eindeutig zu halten, wird bei der Auswahl eines Schlüssels $k_{MO}(l)$ eine Anfrage mit $ID_{MO}(l)$ vom mobilen Objekt an den Lokationserver l gesendet. Dieser überprüft, ob bereits Daten unter diesem Pseudonym abgelegt wurden und teilt es dem mobile Objekt mit. Sollte das Pseudonym bereits in Verwendung sein, wählt das mobile Objekt ein neues $k_{MO}(l)$ und versucht es erneut. Anderenfalls wird das Pseudonym auf l reserviert und das mobile Objekt kann unter diesem Pseudonym Positionsupdates an den Lokationsserver senden.
Sollen die Positionsdaten eines Lokationsserver l einer Anwendung zur Verfügung gestellt werden, erteilt das mobile Objekt dieser eine Berechtigung, indem es ihr den geheimen Schlüssel $k_{MO}(l)$ zusammen mit der Adresse des Servers übermittelt. Damit ist es der Anwendung möglich, die unter Pseudonym $ID_{MO}(l)$ abgelegten Positionsupdates mit dem mobilen Objekt in Verbindung zu bringen und so zu identifizieren.
Die Anonymisierung wird innerhalb der *SendUpdate* Funktion (siehe Algorithmus 1) durchgeführt. Will ein Benutzer einer ortsbasierten Anwendung identifizierenden

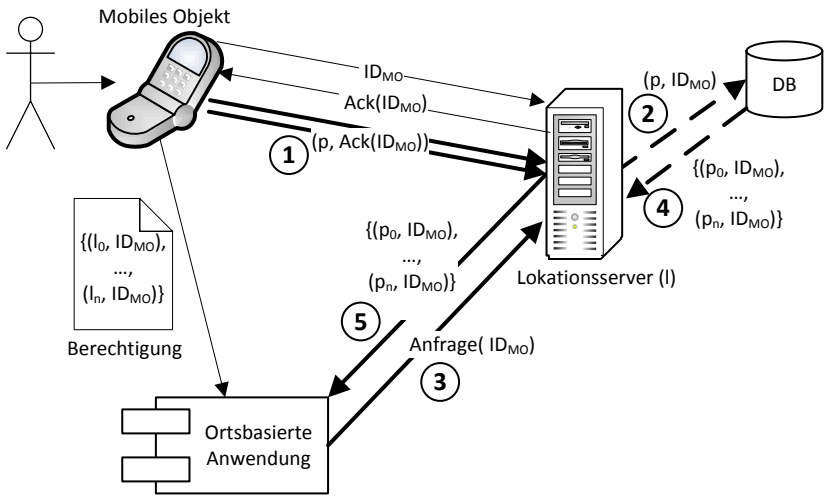

Abbildung 5.7: Ablauf der Berechtigung

Zugriff auf seine Daten geben, sendet er diesem einmalig die Adresse der entsprechenden Lokationsserver und das zugehörige Pseudonym $ID_{MO}(l)$. Anschließend ist es dem Dienst möglich, die Daten, die unter diesem Pseudonym bei l abgelegt wurden, abzufragen.

5.6 Rekonstruktion durch berechtigte LBA

Einer berechtigten LBA sind die Lokationsserver und die entsprechenden Pseudonyme des mobilen Objekts bekannt, auf denen die Fragmente der Trajektorie abgelegt wurden. Es ist einer solchen LBA also möglich, die Fragmente eines bestimmten mobilen Objekts bei den betroffenen Lokationsservern, mittels des Pseudonyms, abzufragen. Anschließend können die Fragmente von der LBA anhand der Orts- und Zeitinformation geordnet und gefiltert werden, um die Fragmente einer einzelnen Trajektorie zu erhalten. Aus diesen rekonstruiert die LBA dann die Trajektorie, indem beispielsweise über die Positionen interpoliert wird.

Möchte man die Güte der Rekonstruktion bewerten, so kann man die tatsächliche Trajektorie mit der rekonstruierten vergleichen und die Abweichung messen. Durch unser Basisverfahren können diese Abweichungen an verschiedenen Stellen auftreten: Zum einen im Sicherheitsbereich eines Knotens, um Ausfahrten und Kreuzun-

gen zu schützen. Hier werden unter Umständen mehrere Lokationsupdates ausgelassen, die allerdings durch einfache Interpolation zwischen der vorhergehenden und nachfolgenden Position, entlang der befahrenen Straße, angenähert werden kann. Außerdem ist es möglich, dass gesamte Fragmente nicht herausgegeben werden, falls sie mit ihrer Bewertung das *PrivacyLimit* überschreiten. Diese Fragmente lassen sich auch anhand des Kartengraphs interpolieren, da der Anfangs- und Endknoten bekannt ist. In jedem Fall gilt jedoch: die Abweichung wird größer, je größer die Distanz der bekannten Informationen ist.

5.7 Anonyme Nutzung der Daten

Auch nicht zur Identifikation berechtigte LBAs können Positionsinformationen von Lokationsservern anfragen. Allerdings sind ihnen die Pseudonyme und Identitäten der mobilen Objekte nicht bekannt. Wir sprechen hier von anonymem Zugriff. Dieser geschieht meist in Form einer Bereichsanfrage, indem sämtliche Positionsdaten innerhalb eines bestimmten geographischen oder geometrischen Bereichs abgefragt werden. Die LBA erhält damit Ausschnitte der auf dem Server befindlichen Fragmente, die sich im angefragten Bereich befinden und kann Berechnungen auf diesen Daten ausführen. Um alle Positionsdaten eines bestimmten Bereichs zu erhalten muss die LBA diesen bei sämtlichen Lokationsservern anfragen.
Es wäre denkbar, dass die LBAs auf Seiten der Lokationsserver authentifiziert werden, damit dieser zwischen LBAs und eventuellen Angreifern unterscheiden kann. Anderenfalls wäre es einem Angreifer möglich über einen Anonymen Zugriff sämtliche Daten eines Lokationsservers abzufragen.

6 Sicherheitsanalyse

Die Fragmentierung soll ein hohes Maß an Sicherheit gewährleisten und es für einen Angreifer möglichst schwierig machen, vollständige Informationen zu erhalten. Das beschriebene Verfahren betrachtet dabei nicht nur die direkt verfügbaren, sondern auch die eventuell herleitbaren Informationen, um damit realistische Angriffsszenarien zu beachten.
Neben einer hohen Sicherheit für die Privatsphäre eines Benutzers sollen berechtigte Anwendungen jedoch weiterhin die Möglichkeit haben, die Trajektorien zu rekonstruieren. Die rekonstruierte Trajektorie soll mit geringem Aufwand möglichst genau der tatsächlichen Trajektorie entsprechen. Wäre das nicht der Fall könnten Fehler bei Berechnungen auf den rekonstruierten Daten auftreten. Wir wollen also versuchen, die Daten möglichst konsistent zu halten.
Im Folgenden wird auf die Möglichkeiten eines Angreifers eingegangen, Teile einer Trajektorie zu rekonstruieren. Die Sicherheit, die durch unser Verfahren erzeugt wird, hängt direkt von der Rekonstruierbarkeit durch Angreifer ab. Als Angreifer werden sämtliche nicht berechtigte Dritte betrachtet, welche die Privatheit eines Benutzers gefährden.

6.1 Rekonstruktion und Identifikation durch einen Angreifer

Ein Angreifer verfügt über die Möglichkeit, einen Lokationsserver zu kompromittieren und dessen Informationen auszulesen. Wir gehen zunächst davon aus, dass ein Angreifer einen einzigen Lokationsserver kompromittiert hat, auf dem sein Zielobjekt Positionsinformationen ablegt. Er gelangt also in den Besitz aller Fragmente, die auf diesem gespeichert wurden und versucht, mithilfe dieser Informationen weitere Informationen herzuleiten. Zunächst wird er dasselbe Vorgehen verfolgen, das auch eine LBA verfolgt. Er wird die Fragmente nach einem bestimmten Pseudonym

filtern und anschließen versuchen, über die Zeit- und Ortsinformation zusammenhängende Fragmente zu finden.
Wir nehmen an, dass einem Angreifer im Vorfeld nicht bekannt ist, welches Pseudonym der Benutzer verwendet, dessen Daten er identifizieren möchte. Um ein Pseudonym zu wählen, nach dem gefiltert werden soll, muss ein Angreifer also bereits zu diesem Zeitpunkt zusätzliches Wissen verwenden. Da durch den gewählten Knoten- und Bewertungsbasierten Fragmentierungsansatz keine Fragmente herausgegeben werden, die sensible Orte wie beispielsweise Wohn- bzw. Arbeitsort enthalten, kann ein Angreifer nicht in deren Besitz gelangen und damit auch keine direkte Verbindung zwischen einer Adresse und einer Person herstellen. Er muss durch Informationen über sein Zielobjekt versuchen, die Menge der infrage kommenden Objekte einzuschränken. Der Angreifer könnte alle Pseudonyme in eine engere Auswahl aufnehmen, die morgens in einem Bereich auftauchen, den das Zielobjekt auf dem Weg zu seiner Arbeitsstelle sicher durchquert. Dies ist nur möglich, wenn der Angreifer bereits Informationen über einen ungefähren Wohn- bzw. Arbeitsort seines Zielobjektes hat. Durch das Verbinden unterschiedlicher Informationen, wie zum Beispiel das Stammlokal des Zielobjektes oder Sportvereinszugehörigkeiten und Trainingszeiten kann die Anzahl der Alternativen weiter verringert werden, wenn sich zu bestimmten Zeiten nur einige pseudonymisierte Objekte im Umkreis der betreffenden Orte aufhalten.

Hat ein Angreifer schließlich ein Pseudonym erfolgreich seinem Zielobjekt zugeordnet, so kann er mit der Rekonstruktion der Trajektorie beginnen. Dabei kann er zwischen direkt zusammenhängenden Fragmenten und Fragmenten, die eventuell zur selben Trajektorie gehören, jedoch nicht zusammenhängen, unterscheiden. Letztere können nur abgeschätzt werden, indem beispielsweise eine Verbindung zwischen beiden Fragmenten gesucht wird, die ungefähr der zeitlichen Distanz der Fragmente entspricht. Dies wäre beispielsweise über eine kürzeste Wege Berechnung möglich, bei der derjenige Pfad angenommen wird, der am ehesten der benötigten Zeit entspricht. Weicht die zeitliche Distanz zwischen zwei Fragmenten stark von denen der kürzesten Wege ab, so kann der Angreifer davon ausgehen, dass es sich entweder um Fragmente unterschiedlicher Trajektorien handelt oder auf einem dieser Pfade ein sensibler Ort liegt, bei dem das Zielobjekt längere Zeit verbracht hat.
Mit den so berechneten Verbindungen und damit der Korrelation der nicht zusammenhängenden Fragmente hat ein Angreifer bereits mehr Informationen hergeleitet, als auf dem Lokationsserver gespeichert waren. Diese Informationen können als

relativ unkritisch betrachtet werden, wenn sie keine für den Benutzer sensiblen Informationen enthalten oder wenn genügend alternative Pfade und damit eventuelle sensible Orte in Betracht gezogen werden müssen.
Durch das *PrivacyLimit* wird gewährleistet, dass Informationen, die über die Grenzen der Fragmente hinaus gehen nur mit geringerer Wahrscheinlichkeit hergeleitet werden können. D.h. sei $F(l) = F_q \cup \ldots \cup F_p$ die geordnete Menge von Fragmenten auf dem kompromittierten Lokationsserver l, so können wir die Knoten $v_{q,first} \in F_q$ und $v_{p,last} \in F_p$, also den Startknoten des ersten Fragments und den Endknoten des letzten Fragments einer Trajektorie auf dem Lokationsserver, als Grenzen betrachten. Hat der Angreifer, wie bisher angenommen, nur einen Lokationsserver kompromittiert, so fällt es ihm schwer, weitere Informationen herzuleiten, ohne im Besitz zusätzlicher Informationen zu sein.

Wollen wir außerdem den Fall betrachten, dass der Angreifer zwei Lokationsserver kompromittiert hat, welche die Daten seines Zielobjektes erhalten haben, so lässt sich das bisher beschrieben Vorgehen übertragen. Aufgrund der unterschiedlichen Pseudonyme pro Lokationsserver ist es dem Angreifer nicht möglich, sein Zielobjekt, das er auf einem Server bereits identifiziert hat, mit einem Pseudonym auf dem anderen Lokationsserver in Verbindung zu bringen. Er muss also beide Pseudonyme zunächst auf das Zielobjekt zurückführen. Anschließend kann er die Fragmente beider Server in Beziehung zueinander stellen und wahrscheinliche Pfade zwischen den Fragmenten suchen, um mehr Informationen zu erhalten. Nach wie vor besteht eine Begrenzung durch den ersten Knoten des ersten Fragments und den letzten Knoten des letzten Fragments einer Trajektorie, ab der nur mit einer durch das *PrivacyLimit* beschränkten Wahrscheinlichkeit weitere Schlüsse gezogen werden können. Allerdings lassen sich durch die zusätzlichen Positionsinformationen alternative sensible Orte ausschließen, die beispielsweise als Ziel oder Start der Trajektorie in Frage kommen könnten. Denn unter der Annahme, dass sich ein mobiles Objekt auf schnellsten Wegen bewegt, müssen alle bekannten Positionen auf einem solchen Weg zu einem sensiblen Ort liegen. Orte auf deren schnellsten Weg von der ersten bekannten Position nicht über alle weitere bekannte Positionen führt, können als Alternativen ausgeschlossen werden. Ein Beispiel für das Ausschließen von Alternativen zeigt Abbildung 6.1. Durch die zusätzlichen Positionsinformationen können sofort 5 mögliche sensible Orte ausgeschlossen werden.

Dasselbe Vorgehen wie bei der Korrelation zweier Lokationsserver lässt sich auch auf mehr als zwei Server erweitern.

Abbildung 6.1: Beispiel für den Ausschluss von Alternativen

6.2 Identifikation sensibler Orte

Es ist davon auszugehen, dass einem Angreifer ein Großteil der POIs bekannt sind, da diese beispielsweise in den Karteninformationen oder in öffentlichen Datenbanken enthalten sind. In unserem Basisverfahren werden Fragmente vor sensiblen Orten nicht veröffentlicht, wenn sie nicht eine durch das *PrivacyLimit* vorgegebene Anzahl an alternativen Orten zulassen. Die Fragmente, die den sensiblen Ort enthalten, werden aus Sicherheitsgründen nicht freigegeben. Möchte der Angreifer also mögliche sensible Orte bestimmen, so muss er zunächst eine Vermutung anstellen, was sensible Orte des Benutzers sein könnten. Hierzu benötigt er Wissen über den Benutzer oder zusätzliche Informationen, die ihn bestimmte sensible Orte ausschließen lassen. Hat er eine Annahme über mögliche sensible Orte getroffen, so ist es ihm möglich, über Kürzeste-Wege-Berechnungen einige Alternativen auszuschließen. Um dabei möglichst effizient vorzugehen, wählt er einen Distanzhorizont, mit dem er abschätzt, wie lange eine Trajektorie ungefähr ist. Damit schränkt der Angreifer die Anzahl der Alternativen weiter ein. Für die Abschätzung eines Distanzhorizonts können beispielsweise Straßenkategorie und Fahrtrichtung verwendet werden. Entfernt sich ein mobiles Objekt auf einer Landstraße von einer Autobahn, so kann ein Angreifer vermuten, dass sich das mobile Objekt einem sensiblen Ort annähert, da für weite Distanzen höhere Straßenkategorien wahrscheinlicher sind. Eine aufwendige Methode, die es einem Angreifer jedoch ermöglicht, sensi-

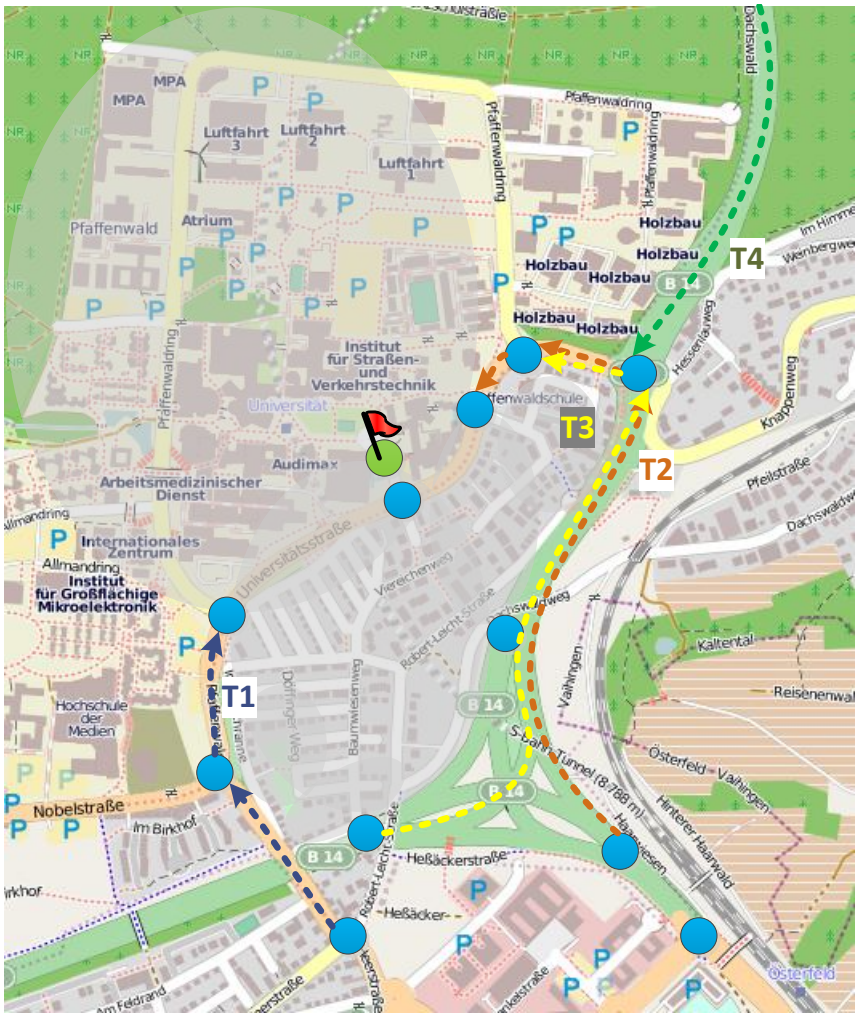

Abbildung 6.2: Bestimmung eines sensiblen Ortes

ble Orte bis zu einem gewissen Grad zu erkennen, wäre, die Positionsinformationen über einen größeren Zeitraum zu sammeln. Er kann dann feststellen, ob Lücken zwischen unterschiedlichen Fragmenten immer an derselben Stelle auftauchen. Nicht zusammenhängende Fragmente, die beispielsweise unterschiedliche Bewegungsrichtungen haben, deuten auf unterschiedliche Trajektorien hin. Enthalten unterschiedliche Trajektorien an derselben Stelle eine Lücke, so ist dies ein Indiz für einen sensiblen Ort. Das Beispiel in Abbildung 6.2 zeigt 4 unterschiedliche Trajektorienfragmente, die, über einen längeren Zeitraum hinweg, in den Besitz eines Angreifers gelangt sind. Unter Zuhilfenahme dieser Fragmente lässt sich ein bestimmter Bereich bestimmen, in dem der sensible Ort liegt. Durch das identifizieren von POIs in

diesen Bereichen und weitere semantische Informationen in diesem Zusammenhang, kann der Bereich weiter eingeschränkt werden.

7 Evaluation

Um die entwickelten Algorithmen zu evaluieren, wurden das vorgestellte System in Form eines Prototypen implementiert. Dabei wurde ein möglichst modulares System mit dem Namen FoPaTh (Abk. Fragment oriented Privacy aware Trajectory handling) entwickelt, dass es ermöglicht verschiedene Fragmentierungs- und Verteilungsarten auf einem mobilen Endgerät zu realisieren und anhand von Testdaten zu evaluieren.
Nachfolgend wollen wir die technische realisierung des Prototypen beschreiben und im Anschluss daran einige Testergebnisse vorstellen.

7.1 Technische Umsetzung

Der Prototyp des Systems besteht aus drei unabhängigen Komponenten, wie sie bereits konzeptionell im Systemmodell (siehe Kapitel 3) beschrieben wurden. Diese sollen im Folgenden näher erläutert werden. Es ist zu beachten, dass der Algorithmus zur Fragmentierung im Mittelpunkt des Systems steht. Alle weiteren eingesetzten Technologien können durch bekannte Mechanismen in puncto Sicherheit noch verbessert werden, wurden aber zunächst so einfach wie möglich gehalten.
Neben den im Systemmodell enthaltenen Komponenten wurde ein Werkzeug entwickelt, welches zur Erstellung von Simulationsdaten und zur Visualisierung der vom mobilen Objekt erstellten Ereignisprotokolle verwendet werden kann.

7.1.1 Lokationsserver

Die Implementierung der Lokationsserver erfolgte als RESTful Service. Dieser Service dient als Schnittstelle, um mit einer Apache Derby Datenbank zu arbeiten, welche die Lokationsupdates speichert und bestimmte Updates zur Verfügung stellt. Die Methoden des Lokationsservers werden über das HTTP Protokoll aufgerufen

und ermöglichen damit eine plattformunabhängige Verwendung.
Neben dem Umgang mit Lokationsupdates stellt der Server außerdem Methoden zur Verfügung, die es erlauben, eine Pseudonymprüfung und -reservierung durchzuführen.

7.1.2 Ortsbasierte Anwendung

Um mit den Daten eines mobilen Objekts zu arbeiten, wurde eine Anwendung implementiert, welche eine Berechtigung vom mobilen Objekt erhält und anschließend in der Lage ist, die jeweiligen Daten von den verwendeten Lokationsservern abzurufen. Die Berechtigung besteht aus der Adresse des betroffenen Lokationsservers und dem zugehörigen Pseudonym des mobilen Objekts. Es ist der LBA außerdem möglich, Bereichsabfragen an den Lokationsserver zu stellen und so als anonyme Anwendung zu agieren.

7.1.3 Mobiles Objekt

Das mobile Objekt wurde für ein handelsübliches Smartphone, auf Basis der momentan am weitest verbreiteten Version 2.3.3 des Google Android Betriebsystems, implementiert. Es nutzt den integrierten GPS Content Provider zur Positionsbestimmung, der mittels eines Callbackmechanismus die Änderungen der Position mitteilt und an unser Verfahren übergibt.
Für die Berechnungen und Visualisierungen wird Kartenmaterial des OpenStreetMap Projekts (vgl. Ope (2012)) verwendet. Die Karteninformationen werden dabei auf mit Kraftfahrzeugen befahrbare Straßen beschränkt, was zu einer Verringerung des Berechnungsaufwands und der zu verwaltenden Datenmenge führt. Es ist jeder Zeit möglich, das Verfahren für den Gebrauch anderer Fortbewegungsmittel anzupassen. Um das laden der Kartendaten zu beschleunigen werden diese in einer SQLite-Datenbank auf dem Gerät zwischengespeichert.
Außerdem erhält die Anwendung eine weitere einfache SQLite-Datenbank, in der die Zugriffsdaten der Lokationsserver gespeichert werden. Diese lässt sich über eine einfache Benutzerschnittstelle verwalten. Eine zweite SQLite-Datenbank hält die POIs des Benutzers persistent, welche ebenfalls über eine einfache Benutzerschnittstelle verwaltet werden können. Außerdem kann ein POI als sensibler Ort markiert werden, um ihn im Verfahren als solchen zu behandeln.

Da unser Verfahren auf dem mobilen Objekt ausgeführt wird, können hier ebenfalls die später definierten Gütekriterien erfasst und mittels des dafür entwickelten Werkzeugs geprüft werden. Als mobiles Objekt diente ein aktuelles Samsung Galaxy Nexus Smartphone.

Vor Beginn des Tests muss der Benutzer ein Ziel in Form eines POIs wählen. Anschließend wird die schnellste Route vom aktuellen Standort zum Ziel berechnet. Hierfür werden bekannte Kürzeste-Wege-Algorithmen verwendet. Aus der so berechneten Route lassen sich notwendige Informationen für die Fragmentierung ableiten.
Für die Evaluation wurden drei Fragmentierungsarten implementiert: Zum einen die zeitbasierte Fragmentierung, die einfache knotenbasierte Fragmentierung und die auf der bewertung von Fragmenten basierte Fragmentierung.

Zeitbasierte Fragmentierung

Bei der Umsetzung einer zeitbasierten Fragmentierung wurden zusätzlich bestimmte Kriterien erfüllen, um diesen Ansatz hinsichtlich Privatheit so gut wie möglich zu optimieren. Die Zeitintervalle, in welche die Trajektorie geteilt wird, werden anhand der geschätzten Dauer der Route und der Anzahl der verfügbaren Lokationsserver berechnet. So soll gewährleistet werden, dass auf jedem Server nur ein einziges Fragment abgelegt wird.
Zur Verteilung der Fragmente wird mittels eines Zufallsgenerators und anschließendem Round-Robin-Verfahren ein Lokationsserver aus der Datenbank gewählt.
Kommt es im Laufe der Trajektorie zu Abweichungen von der berechneten Route, so werden die Zeitintervalle dennoch beibehalten. Überschreitet die Dauer der Trajektorie die vorberechnete Dauer, so werden keine weiteren Lokationsupdates gesendet. Damit wollen wir versuchen, dass Schutzziel der festen Zeitintervalle pro Server und Trajektorie einzuhalten.

Knotenbasierte Fragmentierung

Die knotenbasierte Fragmentierung wird wie in Abschnitt 5.3.4 beschrieben, mit einer Bewertung der Knoten anhand der geltenden Geschwindigkeitslimits durchgeführt. Dabei werden sämtliche Knoten auf der vorberechneten Route bewertet.

Anschließend werden Knoten für einen Serverwechsel bestimmt. Die möglichen Serverwechsel hängen von der Anzahl der zur Verfügung stehenden Lokationsserver $\#L$ ab. Es werden demnach die $\#L$ am höchsten bewerteten Knoten auf der Route für einen Serverwechsel ausgewählt. Zusätzlich wurde der in der fragmentbewertungsbasierten Fragmentierung beschriebene Sicherheitsabstand um diese Knoten realisiert werden. Die Auswahl eines Servers geschieht wie bei der zeitbasierten Fragmentierung mit einem initialen Zufallswert und anschließendem Round-Robin auf der Menge der Server.
Sollte das mobile Objekt von der vorberechneten Route abweichen wird für die neu berechnete Route erneut eine Knotenbewertung durchgeführt. Die Anzahl der für einen Serverwechsel gewählten Knoten wird beläuft sich dann auf die Anzahl der noch nicht für diese Trajektorie verwendeten Lokationsserver. Bei der Serverwahl wird das Round-Robin wie zuvor fortgeführt.

Fragmentbewertungsbasierte Fragmentierung

Bei der Umsetzung, des von uns entwickelten Verfahrens, haben wir zunächst die Funktionalität in den Vordergrund gestellt, um eine Bewertung hinsichtlich der Privatheit durchführen zu können. Von den vorgeschlagenen Vorberechnungen wurde die Routenberechnung umgesetzt, um die dadurch gewonnen Informationen im Verfahren nutzen zu können.

7.2 Gütekriterien

Um die Fragmentierungsarten und unterschiedliche Parameterzusammensetzungen gegenüberstellen zu können, benötigen wir Maßstäbe anhand derer sie sich vergleichen lassen. Als Maß für die Effizienz des Verfahrens wollen wir die Kosten des Verfahrens erfassen. Die Effektivität bestimmen wir durch unsere Bewertungsfunktion, die im Basisverfahren zum Messen des Informationsgehalts verwendet wird.

7.2.1 Kosten

Wir gehen von einem relativ weit gefassten Kostenbegriff aus, wollen allerdings nur für uns unmittelbar relevante Kostenfaktoren betrachten. Diese sind:

- Netzwerkkosten: Die vom Verfahren benötigte Anzahl an Nachrrichten, die zwischen dem mobilen Objekt bzw. der LBA und einem Lokationsserver ausgetauscht wird.
- Rechenaufwand: Der Rechenaufwand, der durch unser Verfahren erzeugt wird, benötigt Zeit und Speicherplatz. Die Zeit kann sich dabei in der Vorberechnung oder im Online-Algorithmus bemerkbar machen. Letztere führt zu einer Verzögerung beim Senden der Lokationsupdates und sollte so gering wie möglich ausfallen. Der Speicherplatz, der bei der Berechnung benötigt wird, wird vom Arbeitsspeicher des mobilen Objekts bestimmt. Da unser Prototyp auf einem Smartphone implementiert wird, hängt der Speicherplatz von den dortigen Gegebenheiten ab. Wir wollen uns bei der Evaluation also auf die benötigte Rechenzeit beschränken.

Netzwerkkosten

Durch jeden der getesteten Ansätze entstehen vonseiten des mobilen Objekts, nach einem initialen Laden der Kartendaten, keine zusätzlichen Netzwerkkosten, da maximal gleich viele Nachrichten an die Lokationsserver gesendet werden, wie wenn überhaupt kein Verfahren zum Schutz der Privatheit genutzt wird. Durch den zusätzlichen Anonymisierungsansatz entsteht pro Lokationsserver ein einmaliger Mehraufwand, um ein Pseudonym auszuhandeln.
Es existieren Ansätze, wie zum Beispiel das Dead Reckoning, um die Netzwerklast weiter zu reduzieren. Es wäre denkbar das entwickelte Verfahren mit einem solchen Ansatz zu erweitern um die Netzwerklast weiter zu reduzieren.
Das laden der Karteninformationen von einem Server des Open Street Map Projekts kann hohe Netzwerklast verursachen, darum wurde eine SQLite-Datenbank für das speichern dieser Daten verwendet. Für eine Umsetzung innerhalb eines Systems existieren Anwendungen, welche die Verwaltung der Kartendaten optimieren und die verwendet werden können. Referenzen zu diesen Projekten finden sich auf den Webseiten des OpenStreetMap Projekts (Ope (2012)).

Im Falle einer identifizierenden LBA erhöht sich der Kommunikationsaufwand proportional zur Anzahl der vom mobilen Objekt verwendeten Lokationsserver. Es genügt nicht mehr nur mit einem Server zu kommunizieren, sondern es müssen in jedem Fall sämtliche durch die Berechtigung angegebenen Lokationsserver angefragt werden. Handelt es sich um eine anonyme LBA, so hängt, je nach Anwendung, die

Qualität ihrer Ergebnisse, ebenso wie ihr Kommunikationsaufwand von der Anzahl der betrachteten Lokationsserver ab. Das heißt je mehr Lokationsserver betrachtet werden, um so größer ist die betrachtete Datenmenge für einen bestimmten Bereich. Die Menge der betrachteten Daten kann wiederum in direkt proportionalem Zusammenhang zur Qualität der Ergebnisse stehen. Die Menge der zur Verfügung stehenden Daten hängt ihrerseits von der Anzahl der Lokationsserver ab, an die Anfragen gestellt werden.

Rechenzeit

Die Rechenzeit während des Online-Algorithmus kann stark verbessert werden, indem möglichst viele Vorberechnungen, wie in Abschnitt 8.1 beschrieben, durchgeführt werden. Diese können allerdings, abhängig von Faktoren wie der Kartengröße und der Länge einer Trajektorie, entsprechend lange Zeit in Anspruch nehmen und müssen bei Änderungen im Kontext, wie einer Abweichung von einer berechneten Route erneut durchgeführt werden.
Viel Rechenzeit und Speicher wird für den Aufbau des Kartengraphen benötigt. An dieser Stelle besteht großes Verbesserungspotential zum Beispiel durch Änderungen an der Datenstruktur des Kartengraphen.
Für die knoten- und zeitbasierte Fragmentierung waren keine Verzögerungen feststellbar, da es sich bei den ausgeführten Operationen hauptsächlich um einfache Vergleiche handelt. Bei der Fragmentbewertung kann es, bedingt durch die Suche nach Alternativen, zu leichten Verzögerungen kommen. Diese bewegen sich allerdings im Rahmen weniger Sekunden, je nach größe des Kartengraphen und der Lage der Alternativen im aktuellen Kontext des mobilen Objekts.

7.2.2 Privatheit

Natürlich steht die Privatheit im Mittelpunkt unseres Verfahrens. Darum wollen wir auch die Güte bezüglich der Privatheit betrachten. Abbildung 7.1 zeigt die Fragmentierung einer Beispieltrajektorie mit den drei unterschiedlichen Fragmentierungsarten. Dabei standen 10 Lokationsserver zur Verfügung, die jedoch nur von der knotenbasierten Fragmentierung alle genutzt wurden. Im Falle der zeitbasierten Fragmentierung wurde die Strecke bei Weitem schneller zurückgelegt als vorherberechnet, was daran lag, dass zur Simulation jede Sekunde ein Lokationsupdate

gesendet wurde und nicht der eigentliche Routenverlauf nachempfunden wurde. Durch eine realistischere Simulation könnte die Ausnutzung der zur Verfügung stehenden Server, bei diesem Ansatz, stark verbessert werden. In der Praxis ist davon auszugehen, dass die tatsächliche Fahrzeit durch die Berechnung eher unterschätzt wird und gegen Ende keine weiteren Server zur Verfügung stehen. Dies führt dazu,

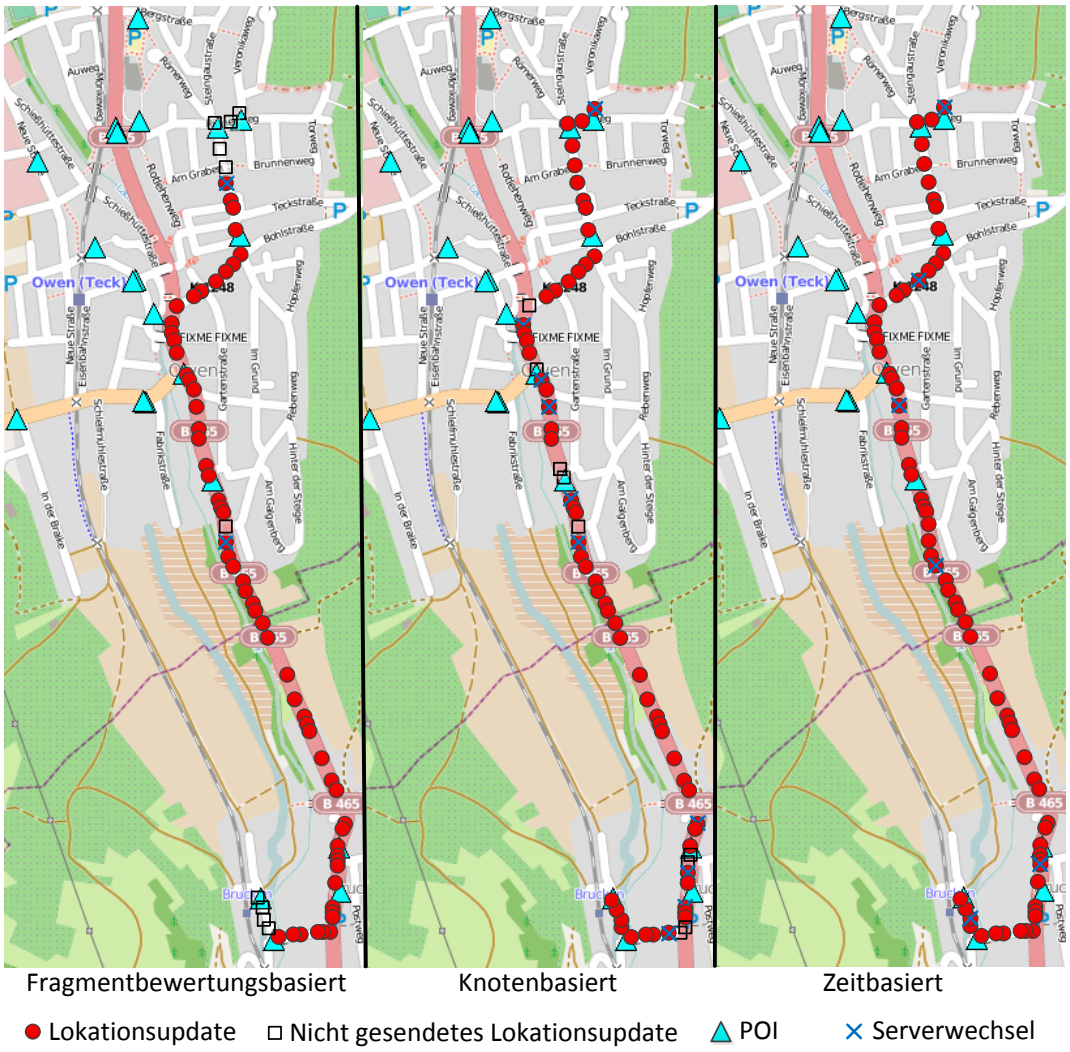

Abbildung 7.1: Vergleich verschiedener Fragmentierungsarten

dass am Ende der Trajektorie keine Lokationsupdates mehr gesendet werden. Die bewertungsbasierte Fragmentierung verwendet nur zwei Lokationsserver. Der einzige Serverwechsel wird durch den *GFC* verursacht, da initial zwei Lokationsserver

in die Menge der zur Verfügung stehenden Server aufgenommen wurden. Werden hier mehr Server aufgenommen, kommt es zu häufigeren Wechseln. Das im Test verwendete *PrivacyLimit* von 0.5 wird nur in der Nähe des Ziels und des Starts überschritten, was dazu führt, dass diese Fragmente nicht freigegeben werden. Bei den anderen beiden Fragmentierungsarten sind Start- und Zielpunkt in Fragmenten enthalten und könnten damit direkt in den Besitz eines Angreifers gelangen.
Kompromittiert ein Angreifer den ersten Lokationsserver bei der bewertungsbasierten Fragmentierung, so muss er beim Start zwei Alternativen betrachten. Da im Startgebiet wenig Alternativen sind, zu denen ein kürzester Pfad führt, kann der Angreifer den Horizont sehr genau abschätzen und den Start mit großer Wahrscheinlichkeit als den weiter entfernteren Punkt identifizieren. Für das Ziel kommen, bei einem richtig gewählten Horizont noch vier Alternativen infrage. Dabei kann der erste POI ausgeschlossen werden, da das Fragment früher beendet worden wäre, wenn es sich bei dieser Alternative um das Ziel handeln würde. Die verbleibenden drei Alternativen sind jedoch gleich wahrscheinlich.
Gelangt ein Angreifer beim knotenbasierten Vorgehen zum Beispiel in den Besitz des zweiten Fragments, so lassen sich relativ wenig Rückschlüsse auf konkrete Orte ziehen. Es wäre einem Angreifer jedoch möglich aufgrund der Knotenbewertungen Rückschlüsse auf die Länge der Trajektorie ziehen. Dazu können auf den an diesem Knoten angrenzenden Straßen nach anderen Knoten gesucht werden. Sind beispielsweise alle auf der Bundesstraße vor dem aktuellen Fragment befindliche Knoten höher bewertet, kann aufgrund des Serverwechsels darauf geschlossen werden, dass zuvor entweder niedrigere Straßenkategorien verwendet wurden und/oder die Trajektorie eine kürzere Strecke umfasst. Würde eine längere Strecke zurücklegen, so würde es Autobahnen verwenden, deren Knoten höher bewertet werden müssten und damit wäre am betrachteten Fragmentknoten kein Serverwechsel durchgeführt worden.
Bei der zeitbasierten Fragmentierung können immer bis zum Vorgänger- beziehungsweise Nachfolgerknoten des Serverwechsels, weitere Positionsinformationen hergeleitet werden. Das Zeitintervall eines Fragments gibt zusätzlich Informationen über die Dauer einer Trajektorie und definiert damit einen Horizont für mögliche Ziele. Hierzu muss ein Angreifer abschätzen, wie viele Fragmente existieren, also wie viele Lokationsserver verwendet werden und an welcher Position in der Trajektorie sich das in seinem Besitz befindliche Fragment befindet. Die Position innerhalb des Fragments lässt sich über die Bewegungsrichtung und die Straßenkategorien bestimmen. Befindet das mobile Objekt sich auf einer niedrigeren Straßenkategorie

und bewegt sich in Richtung einer höheren, wird sich das Fragment am Anfang einer Trajektorie befinden oder sich jedenfalls von einem sensiblen Ort entfernen. Wechselt die Straßenkategorie von einer höheren zu einer niedrigeren, so lässt sich ein Zielbereich vermuten.

Das Wissen, welches ein Angreifer erlangt, hängt bei der knoten- und zeitbasierten Fragmentierung stark vom Fragment ab, das er betrachten kann. Bei der bewertungsbasierten Fragmentierung könnte durch ein strengeres *PrivacyLimit* eine bessere Privatheit erzielt werden. Dies geht allerdings auf Kosten der Datenqualität, da dann weniger Lokationsupdates versendet werden, die eventuell für die Rekonstruktion bei einer ortsbasierten Anwendung wichtig sind.

8 Mögliche Erweiterungen des Verfahrens

Das bisher vorgestellte Basisverfahren kann leicht erweitert werden. Dabei kann z.B. die Sicherheit noch verbessert werden, indem zusätzliche Informationen genutzt werden. Es ist auch möglich, die Zugriffsberechtigungen zu verfeinern und so ein flexibleres Berechtigungskonzept zu erhalten. Außerdem können die verwendeten Schwellwerte und Annahmen verändert werden, um das Verfahren zu beeinflussen und eventuell zu verbessern. Es können ebenfalls alternative Strategien für die Verteilung der Fragmente auf die zur Verfügung stehenden Lokationsserver diskutiert werden.

8.1 Vorberechnung

Um die Berechnungen während des Online-Algorithmus möglichst gering zu halten und damit die Verzögerungszeit von Lokationsupdates zu minimieren, kann eine Vorberechnung durchgeführt werden. Die Vorberechnung kann genutzt werden, um vor Beginn der eigentlichen Trajektorie eine Route zu berechnen. Hierzu muss ein Ziel bekannt sein oder anhand von sensiblen Orten abgeschätzt werden. In einem ersten Schritt kann ein Spannbaum mithilfe eines kürzeste Wege Algorithmus Algorithmus berechnet werden, hierzu kann zum Beispiel der Dijkstra-Algorithmus Dijkstra (1959) verwendet werden. Dieser Baum enthält für jeden Knoten des Graphen den kürzesten Pfad in Form einer Liste von Knoten und der Distanz dieses kürzesten Weges zu ihm. Als Distanzfunktion wird dabei die geschätzte Reisezeit verwendet, welche aus den Distanz- und Geschwindigkeitsinformationen in der Karte berechnet wird. Es werden also die schnellsten Wege zu einem bestimmten Knoten gesucht. Eine Vorberechnung des Baums erhöht die Geschwindigkeit der Berechnungen im Fragmentierungsverfahren, da die schnellste Wege Berechnung nicht für jede

Bewertung von Neuem ausgeführt werden muss, sondern nur entsprechende Unterbäume gewählt werden. Auch die Bewertung und Wahl eines passenden Servers für sämtliche Fragmente der vorberechneten Route ist bereits im Vorberechnungsschritt möglich. Hierbei wird die Vorberechnung zwar aufwändiger und benötigt mehr Zeit, allerdings beschränkt sich der Online-Algorithmus auf das auswählen der bereits berechneten Informationen und das prüfen auf Abweichungen von der berechneten Route. Wird eine Abweichung festgestellt, so muss die Route angepasst und gegebenenfalls neu berechnet werden.

8.2 Servergruppen

Ein denkbares Szenario wäre, dass ein Benutzer einer Anwendung nur bestimmte Teile seiner Trajektorien offenbaren möchte. Beispielsweise nur Fragmente, bei denen sich der Benutzer auf der Autobahn bewegt hat. Um eine solche Funktionalität zu realisieren, könnte man verschiedene Gruppen von Lokationsservern bilden, die ihre Fragmente nach bestimmten Kriterien erhalten. Dieses Kriterium z.B. *Straßenkategorie = Autobahn* wird dann bei der Verteilung geprüft und das Fragment entsprechend einer Gruppe und hier wie im Basisverfahren dem am besten geeigneten Server übermittelt. Der ortsbasierten Anwendung wird dann nur eine Berechtigung für die entsprechende Gruppe von Servern erteilt, indem ihr, wie bereits beschrieben, die entsprechenden Serveradressen und Pseudonymschlüssel mitgeteilt werden. Durch die Einführung dieser Servergruppen erhalten wir also ein feineres Berechtigungskonzept.

8.3 Serverbewertung

Die Lokationsserver können unterschiedliche Vertrauenswerte erhalten, z.B. durch eine Art „Community Voting“, bei dem eine Vielzahl an Benutzern den Server hinsichtlich seiner Vertrauenswürdigkeit bewertet. Anhand dieser Bewertung könnte der Server anschließend einer bestimmten Servergruppe zugewiesen werden.
Es wäre dann denkbar, dass die Zuweisung eines Fragments zu einem bestimmten Lokationsserver mit von diesem Vertrauenswert abhängt und Fragmente, die hohen Informationsgehalt haben, nur auf entsprechend vertrauenswürdigen Servern abgelegt werden.

8.4 Alternative Bewertungsfunktionen und Schwellwerte

Bei der im Basisverfahren verwendeten Bewertungsfunktion gehen wir davon aus, dass der Angreifer einen bestimmten Horizont für die möglichen sensiblen Orte abschätzt. Seine Erfolgswahrscheinlichkeit, dabei den besuchten sensiblen Ort herzuleiten, hängt von dieser Abschätzung ab. Statt wie in unserem Basisverfahren alle Alternativen innerhalb des Horizonts zu betrachten, kann die Menge der Alternativen auch auf andere Weise gewählt werden, um eine exaktere Bewertung zu erhalten. Dies steht jedoch in Verbindung mit der Annahme über das Vorgehen eines Angreifers. Geht man zum Beispiel davon aus, dass ein Angreifer Öffnungszeiten von POIs verwendet, um mögliche Ziele anhand der Zeitinformation auszuschließen, so kann dies ebenfalls in die Bewertungfunktion mit einfließen. Mit der Bewertungsfunktion wird demnach versucht, eine möglichst realistische Abschätzung dessen zu erhalten, was ein Angreifer tatsächlich durch Verwendung bestimmter Positionsinfromationen herleiten kann. Außerdem lässt sich bestimmen mit welcher Genauigkeit diese Herleitung möglich ist.
Es wäre denkbar, dass POIs in Kategorien zusammengefasst werden. So können in der Bewertung für einen sensiblen Ort einer bestimmten Kategorie nur die POIs dieser Kategorie als Alternativen in Betracht gezogen werden.

Um zwischen dem zurückhalten bestimmter Positionsinformationen und dem Wechsel eines Servers im Verfahren unterscheiden zu können, wäre es denkbar zu diesem Zweck unterschiedliche *PrivacyLimits* zu verwenden. Für einen Angreifer wäre der Unterschied zunächst nicht ersichtlich. Allerdings kann so die Häufigkeit eines Serverwechsels und die Genauigkeit mit der eine ortsbasierte Anwendung sensible Orte rekonstruieren kann beeinflusst werden.

8.5 Andere Verteilungsstrategien

Wir gehen davon aus, dass es eine große Menge von Lokationsservern gibt, welche die im Systemmodell (siehe Kapitel 3) definierten Eigenschaften besitzen und damit von unserem Verfahren verwendet werden können. Davon ist unserem Verfahren die Teilmenge L bekannt und wird zum Beispiel in einer Datenbank verwaltet, die entsrpechende für den Zugriff benötigte Daten speichert.

Eine Verteilungsstrategie ordnet dabei einem Fragment einen Lokationsserver zu. Da wir davon ausgehen, dass die Menge der Fragmente einer Trajektorie mächtiger ist als die Menge L, müssen mehrere Fragmente auf einem Server gespeichert werden. Dabei sollen die so zusammengefassten Fragmente einem Angreifer möglichst keine Rückschlüsse erlauben. Intuitiv lässt sich vermuten, dass Fragmente hierfür zusammenhänged sein sollten. Das heißt, dass $F_p, F_q \in T \wedge v_i, v_j, v_k, v_l \in V : F_p = (v_i, v_j) \wedge F_q = (v_k, v_l) \wedge v_j = v_k$. Beide Fragmente grenzen also an demselben Knoten an. Für $v_j \neq v_k$ wäre es einem Angreifer möglich einen Weg zwischen v_j und v_k zu berechnen und mit den ihm bekannten Informationen über die vergangene Zeit zwischen den beiden zugehörigen Fragmenten zu vergleichen.
Außerdem kann man davon ausgehen, dass der Aufwand für einen Angreifer höher wird, je mehr Lokationsserver für eine Trajektorie verwendet werden. Damit würde sich auch die Anzahl der Server erhöhen, die er kompromittieren müsste um die Trajektorie vollständig zu rekonstruieren. Gleichzeitig soll eine Trajektorie jedoch auf möglichst wenig Lokationsservern verteilt werden, um den Rekonstruktionsaufwand für ortsbasierte Anwendungen so gering wie möglich zu halten.

Für eine komplexere Verteilungsstrategie sollte man außerdem in Betracht ziehen, dass auf einem Lokationsserver Fragmente unterschiedlicher Trajektorien eines Benutzers gespeichert werden können. So kann ein Angreifer, der einen Lokationsserver kompromittiert in den Besitz unterschiedlicher Teilsegmente kommen. Sammelt er diese, ist es ihm eventuell möglich mehr über die Bewegungen eines Benutzers herauszufinden. Es wäre also sinnvoll ebenfalls eine Historie der für Fragmente verwendeten Lokationsserver anzulegen, um dann selbe Teilsegmente immer auf dem selben Lokationsserver zu speichern. Gleichzeitig kann das in Abschnitt 6.2 vorgehen eines Angreifers erschwert werden, wenn Teile unterschiedlicher Trajektorien, die zum selben sensiblen Ort führen oder an ihm starten, an verschiedene Lokationsserver gesendet werden.

9 Fazit

Im Rahmen dieser Arbeit wurde ein Verfahren entwickelt, mit dem die Privatheit eines Benutzers, bei der Verwendung ortsbasierter Anwendungen sichergestellt werden soll. Dabei wurden kontinuierliche Positionsinformationen in Form von Benutzertrajektorien betrachtet, wie sie von verschiedenen Anwendungen verwendet werden. Der verfolgte Ansatz versucht, über eine Aufteilung der Benutzertrajektorien, einzelne Fragmente zu bilden, welche für sich abgeschlossene Informationen enthalten und keine sicheren, darüber hinaus gehenden Rückschlüsse ermöglichen. Mittels der entwickelten Bewertungsfunktion ist es möglich den Informationsgehalt eines solchen Fragments abzuschätzen, wobei nicht nur die im Fragment enthaltenen, sondern auch die eventuell herleitbaren Informationen betrachten werden. Damit wird der Wert des Fragments für einen Angreifer bestimmt.
Um die Fragmente sicher zu verwalten, werden sie auf eine Menge von Lokationsservern verteilt. Hierbei wird darauf geachtet, dass ein einzelner Lokationsserver nur eine bestimmte Menge an Informationen erhält, die ebenfalls durch die Bewertungsfunktion bestimmt und einen vorgegebenen Wert - dem *PrivacyLimit* - beschränkt werden kann. Zunächst wurde davon ausgegangen, dass ein Angreifer einen Lokationsserver kompromittiert und sich sein Wissen auf dessen Informationen beschränkt. Um zusätzlich die Korrelation zwischen mehreren Lokationsservern zu erschweren, wurden außerdem Pseudonymen im Verfahren verwendet. Diese verhindern eine direkte Verbindung zwischen Fragmenten eines Benutzers auf unterschiedlichen Lokationsservern.

Um die Funktionalität und die Güte des entwickelten Verfahrens zu zeigen, wurde es in Form eines Prototypen umgesetzt. Dabei wurden drei Ansätze für die Fragmentierung innerhalb des Verfahrens evaluiert: Eine zeit-, ein knotenbasierte und eine auf der bewertung von knotenbasierten Fragmenten basierte. Es konnte gezeigt werden, dass das Verfahren mit vertretbarem Aufwand seitens eines mobilen Objekt umgesetzt werden kann. Darüber hinaus können noch Verbesserungen hinsichtlich Privatheit und Flexibilität durchgeführt werden, die kurz beschrieben wurden.

Neben dem Schutz der Privatheit, ist die einfache Umsetzung auf einem mobilen Objekt, ein Vorteil des Verfahrens. Dadurch ist es unabhängig von vertrauenswürdigen Instanzen, die eventuell Ziel eines Angriffs werden können. Des weiteren bleiben die Daten sowohl für berechtigte, als auch für anonyme Anwendungen nutzbar und können auf den Lokationsservern gefiltert und Berechnungen unterzogen werden.
Durch die Rekonstruktion der Trajektorien entsteht allerdings ein Mehraufwand auf Seiten der Anwendungen. Anbieter einer LBA müssten also eine Rekonstruktion durchführen, was als kritisch für die reale Umsetzbarkeit des Verfahrens betrachtet werden muss. Ohne eine Rekonstruktion seitens der LBAs können diese nicht mehr oder nur noch eingeschränkt genutzt werden.
Mit den Parametern des Verfahrens könnte es einem Benutzer ermöglicht werden, den Grad seiner Privatheit zu steuern. Außerdem erhält er mittels des beschriebenen Berechtigungskonzepts, Kontrolle darüber, welchen LBAs eine einfache Rekonstruktion und Identifikation ermöglicht werden soll.

Wie durch einige Erweiterungen gezeigt, kann das Verfahren gut modifiziert und ausgebaut werden, um zusätzliche Funktionalitäten zu bieten. So kann unter Einbeziehung weiterer Informationen in die Bewertungsfunktion der Schutz der Privatheit noch verbessert werden.

Literaturverzeichnis

[Ope 2012] *Open Street Map*. http://www.openstreetmap.org/. Version: 2012

[Ardagna u. Cremonini 2007] Ardagna, C ; Cremonini, M: Location privacy protection through obfuscation-based techniques. In: *Data and Applications ...* (2007). http://www.springerlink.com/index/A627753563301640.pdf

[Barkhuus 2003] Barkhuus, Louise: Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns Location-Based Services for Mobile Telephony: a study of users' privacy concerns. In: *Intellectual Property* 2003 (2003), Nr. April, 709–712. http://dx.doi.org/10.1.1.10.527. – DOI 10.1.1.10.527

[Beresford u. Stajano 2004] Beresford ; Stajano: Mix zones: user privacy in location-aware services. In: *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second* (2004), 127–131. http://dx.doi.org/10.1109/PERCOMW.2004.1276918. – DOI 10.1109/PERCOMW.2004.1276918. ISBN 0–7695–2106–1

[Chow 2011] Chow, Chi-Yin: Trajectory Privacy in Location-based Services and Data. In: *ACM SIGKDD Explorations Newsletter* 13 (2011), Nr. 1, 19–29. http://sigkdd.org/explorations/issues/13-1-2011-07/v13-01-5-mokbel.pdf

[Dijkstra 1959] Dijkstra, EW: A note on two problems in connexion with graphs. In: *Numerische mathematik* (1959), Nr. 1 959, 269–271. http://www.springerlink.com/index/uu8608u0u27k7256.pdf

[Duckham u. Kulik 2005] Duckham, Matt ; Kulik, Lars: A formal model of obfuscation and negotiation for location privacy. In: *Pervasive Computing* (2005), 152–170. http://www.springerlink.com/index/kwlvm0de5mga8de2.pdf

[Duckham u. Kulik 2006] Duckham, Matt ; Kulik, Lars: Location privacy and location-aware computing. In: *Dynamic & Mobile GIS: Investigating Change in Space and Time* (2006), S. 34 – 51

[Eastlake 2011] Eastlake, D.: US Secure Hash Algorithms. In: *RFC6234* (2011). http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract. ISBN 0011101001001

[Gilbert u. a. 2010] Gilbert, Peter ; Cox, Landon P. ; Wetherall, David: Toward Trustworthy Mobile Sensing. In: *Proceedings of the Eleventh Workshop on Mobile Computing Systems Applications HotMobile 10* (2010). http://dx.doi.org/10.1145/1734583.1734592. – DOI 10.1145/1734583.1734592. ISBN 9781450300056

[Gkoulalas-Divanis u. Kalnis 2010] Gkoulalas-Divanis, A ; Kalnis, P: Providing k-anonymity in location based services. In: *ACM SIGKDD Explorations* 12 (2010), Nr. 1, 3–10. http://dl.acm.org/citation.cfm?id=1882473

[Golle u. Partridge 2009] Golle, Philippe ; Partridge, Kurt: On the anonymity of home/work location pairs. In: *Pervasive Computing* 5538/2009 (2009), 390–397. http://dx.doi.org/10.1007/978-3-642-01516-8_26. – DOI 10.1007/978–3–642–01516–8_26

[Gutscher 2006] Gutscher, a.: Coordinate transformation - a solution for the privacy problem of location based services? In: *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium* (2006), 7 pp. http://dx.doi.org/10.1109/IPDPS.2006.1639681. – DOI 10.1109/IPDPS.2006.1639681. ISBN 1–4244–0054–6

[Hoh u. a. 2008] Hoh, Baik ; Gruteser, Marco ; Herring, Ryan ; Ban, Jeff: Virtual trip lines for distributed privacy-preserving traffic monitoring. In: *Proceeding of the 6th MOBISYS* (2008), 15–28. http://dl.acm.org/citation.cfm?id=1378604. ISBN 9781605581392

[Jorns u. Quirchmayr 2007] Jorns, Oliver ; Quirchmayr, Gerald: A privacy enhancing mechanism based on pseudonyms for identity protection in location-based services. In: *ACSW '07 Proceedings of the fifth Australasian symposium on ACSW frontiers* 68 (2007), 133–142. http://dl.acm.org/citation.cfm?id=1274547

[Nergiz u. Atzori 2008] Nergiz, ME ; Atzori, Maurizio: Towards trajectory anonymization: a generalization-based approach. In: *of the SIGSPATIAL ACM GIS 2008* (2008), 52–61. http://dx.doi.org/10.1145/1503402.1503413. – DOI 10.1145/1503402.1503413. ISBN 9781605583242

[Palanisamy u. Liu 2011] Palanisamy, Balaji ; Liu, Ling: MobiMix: Protecting location privacy with mix-zones over road networks. In: *2011 IEEE 27th International Conference on Data Engineering* (2011), April, 494–505. http://dx.doi.org/10.1109/ICDE.2011.5767898. – DOI 10.1109/ICDE.2011.5767898. ISBN 978–1–4244–8959–6

[Privacy Rights Clearinghouse 2012] Privacy Rights Clearinghouse: Privacy Rights Clearinghouse. In: *https://www.privacyrights.org/data-breach* (2012)

[Shin u. a. 2010] Shin, Heechang ; Vaidya, Jaideep ; Atluri, Vijayalakshmi ; Choi, Sungyong: Ensuring Privacy and Security for LBS through Trajectory

Partitioning. In: *2010 Eleventh International Conference on Mobile Data Management* (2010), 224–226. http://dx.doi.org/10.1109/MDM.2010.29. – DOI 10.1109/MDM.2010.29. ISBN 978–1–4244–7075–4

[Sweeney 2002] Sweeney, Latanya: k-anonymity: A model for protecting privacy. In: *International Journal of Uncertainty Fuzziness and Knowledge Based Systems* 10 (2002), Nr. 5, 1–14. http://arbor.ee.ntu.edu.tw/archive/ppdm/Anonymity/SweeneyKA02.pdf

[Wernke 2012] Wernke, Marius: PShare: Position Sharing for Location Privacy based on Multi-Secret Sharing. In: *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on* (2012), Nr. March, 153–161. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6199862. ISBN 9781467302586

[Wernke u. a. 2012] Wernke, Marius ; Skvortsov, Pavel ; Frank, D ; Rothermel, Kurt: A Classification of Location Privacy Attacks and Approaches. (2012), S. 1–24

[You u. Peng 2007] You, TH ; Peng, WC: Protecting moving trajectories with dummies. In: *Mobile Data Management, 2007* Bd. 1, IEEE, 2007, 278–282

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe bzw. unerlaubte Hilfsmittel angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Stuttgart, den 1. Oktober 2012

Jan Barocka