Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2340

Vertrauliche Verwaltung von Benutzertrajektorien unter Berücksichtigung einer Start- und Endpunktverschleierung

Franz G. Jahn

Studiengang:	Informatik
Prüfer:	Prof. Dr. Kurt Rothermel
Betreuer:	Dipl.-Inf. Marius Wernke
begonnen am:	08. Juni 2011
beendet am:	08. Dezember 2011
CR-Klassifikation:	C.2.0

Inhaltsverzeichnis

Abbildungsverzeichnis

Verzeichnis der Algorithmen

1 Einleitung

In den letzten Jahren nahm die Zahl der ortsbasierten Dienste stetig zu. Verschiedene Anwendungen bieten, abhängig vom aktuellen Standort des Benutzers, Information über nahegelegene Restaurants, zeigen an, welche Freunde sich in der Nähe aufhalten oder liefern aktuelle Stauinformationen, welche mit Hilfe der Positionsdaten der Nutzer ermittelt wurden.
Gleichzeitig wächst im Internet die Anzahl der kostenlosen Dienste, welche sich über Werbung und die Auswertung der Benutzerdaten finanzieren. Unsere heutige Zeit wird daher in den Medien oft als „Post Privacy Zeitalter“ dargestellt und nicht selten fallen Sätze wie „Privatsphäre war gestern“.

Tatsächlich teilt nicht jeder die positive Grundhaltung zum Bild des „Gläsernen Bürgers“ und immer mehr Nutzer suchen nach einer Möglichkeit die Kontrolle über ihre Daten zurückzugewinnen. Möchte man dabei nicht vollständig auf die Nutzung der Dienste verzichten, so gilt es ein geeignetes Maß dafür zu finden, welche Daten man preisgibt und welche man zurückhält, da man sie nicht in den Händen Dritter wissen möchte. Die große Herausforderung liegt dabei darin die Grenze zu ziehen zwischen Daten, welche für die Nutzung des Dienstes preisgegeben werden müssen und Daten, die nicht herausgegeben werden dürfen, da sie zu viele Rückschlüsse ermöglichen.

Für Positionsdaten ergibt sich hierbei eine besondere Problematik, da diese oft mehr Rückschlüsse über eine Person zulassen, als es auf den ersten Blick scheint. Betrachtet man die Aufenthaltsorte einer Person, so lassen sich über die täglich zurückgelegten Wege beispielsweise der Arbeitsplatz, der Wohnort und weitere interessante Bezugspunkte dieser Person bestimmen. Dies erlaubt wiederum Rückschlüsse über das Einkommen, die Hobbys oder das soziale Umfeld der Person.
Besonders interessant sind dabei die Start- und Endpunkte der zurückgelegten Wege. Kann durch alleinige Betrachtung der Positionsdaten nicht mehr das genaue Gebäude ermitteln, zu welchem ein Nutzer jeden morgen fährt, so kann beispielsweise der Arbeitgeber nicht mehr exakt bestimmt werden und damit verbundene Schlüsse werden verhindert.

In dieser Arbeit wird ein Verfahren vorgestellt, welches die Herausgabe von Positionsinformationen bei gleichzeitiger Verschleierung der Start- und Endpunkte der zugehörigen Trajektorien ermöglicht. Die Klassifizierung der Positionsinformationen findet dabei direkt auf dem mobilen Endgerät des Anwenders statt. Insbesondere ist dabei keine vertrauenswürdige Instanz zur Verwaltung der Positionsinformationen notwendig.

Auf diese Weise sollen die Nutzer ortsbasierter Dienste die Möglichkeit erhalten diesen zu nutzen ohne dabei eine unnötige Beeinträchtigung ihrer Privatsphäre in Kauf nehmen zu müssen. Die Nutzer sollen dadurch in ihren Persönlichkeitsrechten, insbesondere ihrem Recht auf informationelle Selbstbestimmung, gestärkt werden.

2 Grundlagen

2.1 Literaturübersicht

In der Literatur werden bereits unterschiedliche Verfahren diskutiert, welche sich mit unterschiedlichen Aspekten des Schutzes der Privatsphäre im Bezug auf Ortsdaten befassen. Es werden dabei verschiedene Definitionen des Begriffs „Location Privacy“ angegeben. Beresford und Stajano definieren Location Privacy beispielsweise als eine spezielle Form der Privatsphäre, welche

> ...die Möglichkeit andere Beteiligte daran zu hindern die eigene aktuelle oder vergangene Position zu erfahren... [BS03]

umfasst. Nach der Definition von Duckham und Kulik befasst sich Location Privacy mit

> ...dem Bedürfnis von Einzelpersonen, Gruppen oder Institutionen für sich selbst zu entscheiden wann, wie und in welchem Maß Ortsinformationen über sie an andere weitergegeben werden. [DK06]

Diese Definition basiert im wesentlichen auf Westins allgemeiner Definition von Privatsphäre [Wes67].
Während die erste dieser beider Definitionen den Fokus eher auf die zu schützenden Informationen legt und betont, dass es wichtig ist neben der aktuellen Position auch die Positionsdaten der Vergangenheit zu schützen, betont die zweite Definition eher die Kontrolle durch den Nutzer. Im Deutschen spricht man auch oft von dem „Recht auf informationelle Selbstbestimmung“, also das Recht jedes einzelnen frei darüber zu entscheiden, welche Daten über ihn von wem erhoben, gespeichert und verarbeitet werden dürfen.

In der Verwaltung von Positionsinformationen gibt es verschiedene Strategien um die Privatsphäre der Nutzer zu schützen. Duckham und Kulik unterscheiden in [DKB06] dabei vier verschiedene Ansätze:

1. Gesetzliche Regelungen: Regeln und Gesetze, die einen verantwortungsvollen Umgang mit personenbezogenen Daten erzwingen.
2. Datenschutzbestimmungen: Freiwillige Absprachen zwischen Vertragspartnern über die Speicherung, Weitergabe und Verwendung von personenbezogenen Daten.

3. Anonymität: Trennung von Personen und den verwendeten Daten, beispielsweise durch Verwendung von Pseudonymen sowie die Schaffung von Ununterscheidbarkeit durch Zusammenfassung mehrerer Individuen zu einer Gruppe, in welcher sie die zugehörigen Daten nicht mehr einer einzelnen Person zugeordnet werden können.
4. Verschleierung: Die Qualität von herausgegebenen Positionsinformationen wird gezielt soweit reduziert, dass die Privatsphäre des Nutzers geschützt bleibt.

Die ersten beiden dieser Lösungsansätze sind politischer bzw. gesellschaftlicher Natur und werden im Folgenden daher nicht weiter diskutiert werden. Die anderen beiden Ansätze, das Anonymisieren und die Verschleierung von Positionsinformationen wollen wir dafür etwas genauer betrachten.

2.1.1 Anonymisierung

Zur Anonymisierung gibt es bereits einige interessante Ansätze. Terrovitis und Mamulis präsentieren in [TM08] eine Möglichkeit Positionsinformationen, welche bei Zahlvorgängen mit Kreditkarte an den Kreditkartenanbieter übermittelt wurden so zu anonymisieren, dass bei einer Veröffentlichung der Datenbank die Privatsphäre der Kunden auch gegenüber den Shopbetreibern gewahrt bleibt, die partielle Informationen von Trajektorien kennen. In [GG03] schlagen Gruteser und Grunwald eine Methode vor, welche die örtliche und zeitliche Genauigkeit von herausgegebenen Positionsinformationen in Abhängigkeit der Anzahl von Personen, die sich im Umkreis befinden, soweit reduziert, dass die eigene Position von mindestens $k - 1$ anderen Personen nicht unterschieden werden kann, für ein vorgegebenes k. Ein ähnliches Verfahren, bei welchem eine vertrauenswürdige Instanz (Trusted Server) die Positionsinformationen der Nutzer verwaltet und anonymisiert an autorisierte Dienstanbieter weitergibt, wird von Bettini, Wang und Jajodia in [BWJ05] vorgestellt. Bei diesen Verfahren werden die Ideen von k-Anonymity [Swe02] und l-Diversity [MKGV07] auf Positionsinformationen angewandt. Für viele Dienste muss neben den Positionsinformationen auch ein Benutzername oder eine Nutzer ID übertragen werden. Damit die Anonymisierung funktionieren kann, sollten diese Dienste nur über verschiedene Pseudonyme genutzt werden. Um die Akkumulation von persönlichen Positionsinformationen zu vermeiden schlagen Beresford und Stajano in [BS03] daher einen häufigen Wechsel des Pseudonyms in sogenannten Mix-Zones vor, in denen sich viele Personen aufhalten. Durch diese Mix-Zones soll gewährleistet werden, dass bei einem Wechsel des Pseudonyms nicht das neue Pseudonym mit dem alten verknüpft werden kann.
Die Anwendung dieser Verfahren ist allerdings an gewisse Voraussetzungen geknüpft. Einige dieser Verfahren setzen eine vertrauenswürdige Instanz voraus, welche die Positionsinformationen der Nutzer verwaltet und nur entsprechend anonymisiert weitergibt. Ein anderer Teil der Verfahren setzt auf einen dezentralen Ansatz, bei dem die Nutzer kooperieren um das notwendige Maß an Anonymität zu erzeugen. Dies impliziert aber, dass es genügend Anwender dieses Verfahrens geben muss, damit Verfahren wie in [GG03] oder [BS03] das gewünschte Maß an Anonymität bieten können.

Eine Alternative zu diesen Verfahren stellt die Erzeugung von Dummy-Datensätzen dar, wie Kido et al. sie in [KYS05] vorschlagen. In [YPL07] zeigen You et al., dass durch die Erzeugung von Positionsinformationen, welche sich schneidende Trajektorien bilden, die Anzahl möglicher Trajektorien erhöht werden kann. Shankar et al. zeigen in [PS09] wie unterschiedliche Verkehrsaufkommen bei der Generierung von realistischen Dummydaten berücksichtigt werden können und Krumm zeigt in [Kru09] wie ein Dummystrecken so generiert werden können, dass sie das Benutzerverhalten realistisch widerspiegeln.
Die Erzeugung von Dummy-Datensätzen bietet allerdings auch den Nachteil, dass sie die Qualität eines Dienstes, welcher basierend auf den Positionen der Nutzer neue Inhalte erzeugt, stark beeinflussen. Betrachtet man beispielsweise einen Dienst zur Stauvorhersage, so könnte dieser eine Straße völlig zu unrecht als Überlastet anzeigen, wenn von den Nutzern entsprechend viele Dummy-Datensätze für diese Route erzeugt werden.

2.1.2 Verschleierung

Wir werden uns in dieser Arbeit vor allem mit der vierten Möglichkeit, der Herausgabe von verschleierten Positionsinformationen, beschäftigen. Auch hier gibt es schon einige Ansätze, welche nur ungenaue oder bewusst verfälschte Positionsinformationen herausgeben. Dadurch soll garantiert werden, dass ein Angreifer die Position des Nutzers nur mit einer gewissen Ungenauigkeit ermitteln kann. Diese Verfahren lassen sich zum einen danach unterscheiden, ob sie für die Herausgabe einzelner Positionen oder für die Verschleierung von Trajektorien entwickelt wurden und zum anderen danach welche grundlegende Idee für die Verschleierung eingesetzt wird.

Die vermutlich naheliegendste Idee zur Verschleierung von Positionsinformationen ist die Adaption klassischer Verschlüsselungs- und Verschleierungsverfahren. So wird in [MDKG05] die Anwendung des Shamir Secret Sharing Protokolls [Sha79] auf Positionsdaten vorgeschlagen. Solche kryptographischen Verfahren fallen allerdings schon eher in den Bereich der Verschlüsselung als der Verschleierung.
Eine einfache Möglichkeit Positionsinformationen zu verschleiern besteht in der Übermittlung einer ungenauen Positionsangabe. So schlagen Dürr et al. in [Dü11] ein Verfahren vor, bei dem die genaue Position durch den Ortsvektor (Master Share) einer ungenauen Position und k zusätzliche Verfeinerungsvektoren (Refinement Shares) dargestellt wird. Durch die Anzahl der zusätzlich zum Ortsvektor der ungenauen Position herausgegebenen Verfeinerungsvektoren, kann damit für jeden genutzten Dienst die Güte der herausgegebenen Positionsinformation bestimmt werden. Dieser Ansatz bietet den Vorteil, dass die Positionsinformationen auf verschiedene Location Server verteilt wird und die exakte Positionsinformation nur dann ermittelt werden kann, wenn alle Refinement Shares bekannt sind. Wird nur einer der Location Server kompromittiert, so lässt sich die tatsächliche Position zwar genauer, aber noch immer nicht vollständig exakt bestimmen. Eine andere Möglichkeit der Verschleierung bietet die Transformation der Koordinaten, welche nur mit Hilfe zusätzlicher Informationen invertiert werden kann [Guto6]. Ardagna et al. schlagen in [ACD+07] ver-

schiedene Verschleierungsoperationen, welche durch Hintereinanderausführung zusätzliche Sicherheit bieten sollen.

2.1.3 Angriffe

Neben dem Schutz der Privatsphäre geben einige Arbeiten auch Aufschluss darüber, welche Informationen tatsächlich aus verfügbaren Positionsinformationen extrahiert werden können.

Eines der naheliegendsten Ziele eines Angreifers ist dabei die Identifikation der Adresse eines Opfers. Dieses Ziel dient dabei nicht nur als Selbstzweck, sondern kann es dem Angreifer auch ermöglichen Trajektorien wieder einzelnen Personen zuzuordnen. Schon bei einer sehr geringen Datenmenge können dabei weitreichende Rückschlüsse gezogen werden. So analysiert Krumm in [Kru07], die GPS Daten von 172 freiwilligen Autofahrern und konnte dabei die Adressen der Fahrer mit einer mittleren Abweichung von 60,6m ermitteln. Zur korrekten Einordnung dieses Ergebnisses sollte noch bedacht werden, dass zwischen der Aufzeichnung zweier aufeinander folgender Punkte im Durchschnitt 6 Sekunden vergingen und 64,4m zurückgelegt wurden.
Ähnliche Ergebnisse erzielen Hoh et al. auch in [HGXA06]. Sie zeichneten eine Woche lang die GPS Daten von 239 Autofahrern im Raum Detroit auf, ohne deren Adresse zu kennen. Sie analysierten eine Untermenge von 65 Fahrern und konnten bei einer Samplerate von einer Minute für 85% der Fahrer eine Adresse finden, von welcher sie annahmen, dass es ich dabei um den Wohnsitz des betreffenden Fahrers handelt. Bei einer Samplerate von vier Minuten konnten vom selben Algorithmus immerhin noch 40% zugeordnet werden. Dieselbe Anzahl konnte selbst bei einer Samplerate von zehn Minuten noch zugeordnet werden, wobei dabei noch weniger Orte fälschlicherweise als Endpunkt erkannt wurden als bei einer Samplerate von vier Minuten.
In [GH05] zeigen Gruteser und Hoh, dass trotz Anonymisierung der Daten durch Anwendung von Multi-Target Tracking Algorithmen Positionsdaten einzelnen Identitäten zugeordnet werden können und entsprechende Analysen dazu genutzt werden können die Anonymität zu brechen.

Auch innerhalb von Gebäuden können entsprechende Informationen gewonnen werden. So konnten Beresford und Stajano in [BS03] durch die Analyse von Indoor-Bewegungsprofilen mit entsprechend genauen Ortsdaten die Personen anhand der Zeit, welche sie sich in verschiedenen Räumen aufgehalten haben, ihren Büros zuordnen.

Krumm analysierte in [KH06] drei verschiedene Maßnahmen zur Verschleierung bezüglich deren Effektivität zur Verschleierung des Wohnsitzes einer Person. Eine Möglichkeit war das Verwerfen von Positionsdaten in einem Kreisförmigen Umfeld, dessen Zentrum zufällig innerhalb eins kleineren kreisförmigen Gebietes um den Wohnsitz gewählt wurde. Für die zuverlässige Verschleierung der Adressen aller 172 Probanden musste hierfür allerdings ein Radius von 2 km gewählt werden. Als zweite Möglichkeit betrachtete er eine Verschleierung

der Positionsinformationen durch additives weißes gaussches Rauschen. Hierbei wurden selbst bei einer Standardabweichung von 5 km nicht alle Probanden geschützt. Für die Dritte betrachtete Möglichkeit, die Angabe von gerundeten Positionsinformationen, durften die Positionsangaben auf maximal 5 km genau gerundet werden, damit keine der Adressen der Probanden ermittelt werden konnte.
Zur Einordnung dieser Ergebnisse sollte zudem erwähnt werden, dass Krumm für diese Analyse keine Algorithmen betrachtete, welche die speziellen Charakteristika der eingesetzten Verschleierungsverfahren berücksichtigten. Zudem wurden hier die Positionsinformationen aus dem Zeitraum von zwei Wochen betrachtet. Für die Betrachtung eines längeren Zeitraums und mit speziell auf die Verschleierungsverfahren zugeschnittene Algorithmen wären gegebenenfalls noch bessere Angriffe möglich.

2.1.4 Messbarkeit

Möchte man die Sicherheit eines Verfahrens evaluieren, so stellt sich natürlich die Frage nach dem geeigneten Maß für Anonymität, nach einer Einheit zur Quantifizierung der Privatsphäre. So unterschiedlich die bisher vorgestellten Verfahren sind, so unterschiedlich sind auch die Metriken, mit welchen sie versuchen ihre Qualität messbar zu machen.
Für die auf k-Anonymity basierenden Ansätze gibt das gewählte k an, wie viele Nutzer voneinander nicht unterscheidbar sind und bietet auf diese Weise ein natürliches Maß für die Anonymität. Gruteser und Grunwald geben in [GG03] den Kehrwert $\frac{1}{k}$ als Reidentifikationsrisiko an. Diese Metrik setzt allerdings voraus, dass eine Zuordnung auch für alle dieser k verschiedenen Identitäten gleich wahrscheinlich ist. Für eine nicht uniforme Wahrscheinlichkeitsverteilung definieren Beresford und Stajano in [BS03] die Entropie einer betrachteten Abfolge von anonymisierten Positionsinformationen. Hierbei berücksichtigen Sie beispielsweise, dass eine Personen selten ein einmal betretenes Gebiet in dieselbe Richtung verlässt, aus der er gekommen war, auch wenn dies natürlich möglich wäre.

Für die Verschleierung von Positionsinformationen wird oft betrachtet, wie weit die für einen Angreifer erwartete Position von der Tatsächlichen Position entfernt liegt [Kru07].

2.2 Allgemeine Betrachtungen

Die meisten der betrachteten Arbeiten versuchen einen allgemeinen Lösungsansatz vorzuschlagen. Es wird meist versucht technische Lösungsansätze aus anderen Problembereichen zu adaptieren und deren Vorteile für die Verwaltung von Positionsinformationen zu nutzen. Die Vorteile dieser Herangehensweise bestehen vor allem in der Erprobtheit der Algorithmen auf anderen Bereichen.
Bei dieser Betrachtungsweise kommt allerdings die Perspektive des Angreifers meist viel zu kurz und die speziellen Charakteristik der Probleme findet oft nicht genügend Beachtung.

Viele Arbeiten beruhen auf einer Betrachtung im Freespace, also auf einer Betrachtung, bei der sich eine Person frei bewegen kann und keine Karteninformationen betrachtet werden. Diese Perspektive erleichtert zwar die Betrachtung, vernachlässigt dabei aber die Tatsache, dass ein Angreifer mit Hilfe von Karteninformationen und der Annahme, dass sich Personen nur auf Straßen oder Fußwegen bewegen die möglichen tatsächlichen Positionen für eine verschleierte Positionsinformation stark einschränken. Die Aussagekraft von Arbeiten, welche auf einem Freespace-Modell basieren ist daher oft nicht besonders hoch. Wir wollen daher in dieser Arbeit insbesondere Karteninformationen nutzen.

Auch die Generierung und Übermittlung von Dummies ist nur begrenzt anwendbar. Während Dummies für die Betrachtung und den Vergleich einzelner Trajektorien durchaus so gestaltet werden können, dass diese von echten Trajektorien durch die aktuell bekannten Algorithmen nicht unterschieden werden können, erreichen diese ihre Grenzen wenn größere Datenmengen und daraus resultierende dynamische Informationen ins Spiel kommen. Nehmen wir an, dass die Positionsinformationen zur Ermittlung von Stauinformationen erhoben werden. Ein Angreifer, welcher auf Seite des Anbieters Zugang zu den Positionsinformationen der Nutzer erlangt, kann selbstverständlich auch ermitteln, wann und wo sich der Verkehr staute. Führt nun eine Trajektorie über eine Straße, welche der Nutzer zu diesem Zeitpunkt nie hätte passieren können, da es dort zur selben Zeit einen Stau gab, so ist klar, dass es sich dabei um einen Dummy handelt. Auch die Berücksichtigung solcher dynamischen Gegebenheiten ist nur begrenzt möglich, da Ereignisse wie Staus teilweise erst zu einem späteren Zeitpunkt durch die Betrachtung vieler Positionsinformationen rekonstruiert werden können. Keine der in Kapitel 2.1.1 betrachteten Arbeiten beschäftigt sich mit dieser Problematik.

Darüber hinaus bieten die in Kapitel 2.1.2 diskutierten Verschleierungsverfahren lediglich die Möglichkeit einzelne Positionen zu verschleiern. Eine Betrachtung mehrerer Positionsinformationen findet nicht statt, weder für wiederkehrende einzelne Positionsinformationen noch für Trajektorien und schon gar nicht für eine Menge von Trajektorien.
Versucht man diese Ansätze auf wiederkehrende Positionsinformationen anzuwenden, so erkennt man verschiedene Probleme. Beispielsweise wird in [Dü11] als verschleiertes Gebiet ein zufällig gewählter Kreis um den zu verschleiernden Punkt gewählt. Wendet man dieses Verfahren mehrfach auf dieselbe Position oder mehrere sehr dich beieinander liegende Positionen an, so werden unterschiedliche Gebiete erzeugt. Da die verschleierte Position aber in jedem dieser Gebiete liegt, liegt die verschleierte Position auch in der Schnittmenge all dieser Gebiete. Einem Angreifer wäre es daher möglich die ursprüngliche Position auf die Schnittmenge dieser Gebiete einzuschränken.
Betrachtet man dieses Verfahren unter der Prämisse, dass eine ganze Trajektorie damit verschleiert werden soll, so ergeben sich zusätzlich noch weitere Probleme. Da sich eine Person nicht beliebig schnell bewegen kann, ergibt sich die Möglichkeit die zurückgelegte Strecke mit Hilfe der zeitliche Abfolge von Positionsinformationen zu rekonstruieren. Durch Anwendung eines Map Matching Verfahrens, welches Karteninformationen und die zeitlichen Beschränkungen berücksichtigt [NK09], könnte die Verschleierung somit gebrochen werden.

Zum einen wird daraus ersichtlich, dass die existierenden Verfahren zur Verschleierung einer einzelnen Position nicht auf eine kontinuierliche Herausgabe von Positionsinformationen übertragen werden können und zum anderen zeigt sich, dass die zufällige Wahl des verschleierten Gebiets für eine kontinuierliche Betrachtung große Risiken mit sich bringt.
Für unsere Betrachtungen ergibt sich damit, dass wir für die Entwicklung eines Verfahrens zur Start- und Endpunktverschleierung nicht auf die bestehenden Verfahren aufbauen können, sondern ein komplett neues Verfahren entwickeln müssen. Was wir jedoch nutzen können, ist die Erkenntnis darüber, wo die bestehenden Verfahren Schwächen besitzen und wodurch diese Schwächen entstehen.

3 Ausgangssituation

3.1 Problemstellung

Als Ergebnis dieser Arbeit soll ein Verschleierungsverfahren entstehen, was es uns ermöglicht die Start- bzw. Endpunkte von Trajektorien auf eine zuverlässige Weise zu verschleiern. Da die Start- bzw. Endpunkte meist die sensibelsten Informationen enthalten, möchten wir uns zunächst darauf beschränken und die speziellen Charakteristika dieses Problems genauer betrachten. Start- und Endpunkte können beispielsweise Rückschlüsse auf den Arbeitgeber oder die private Adresse und damit auch die Identität einer Person zulassen. Unser Ziel muss es sein, dass auch wiederkehrende Start- und Endpunkte selbst bei Vorliegen vieler Trajektorien eines Benutzers nicht zu einer genauen Adresse aufgelöst werden können. Wir wollen hier ein Gebiet angeben, für welches jeder potentielle Endpunkt gleich wahrscheinlich ist. Innerhalb dieses Gebietes soll es dem Angreifer durch die übermittelten Positionsinformationen nicht möglich sein die Menge der potentiellen Start- und Endpunkte weiter einzuschränken.

Als Anwendungsfall betrachten wir hier ein Navigationssystem welches die Auswertung aktueller Positionsdaten seiner Nutzer Stauinformationen generiert und diese den Nutzern für die verkehrsabhängige Routenwahl zur Verfügung stellt. Hier sehen wir den klassischen Interessenkonflikt zwischen dem Schutz der Privatsphäre des Nutzers und der Weitergabe von Informationen, welche notwendig sind, damit der genutzte Dienst dem Nutzer einen Mehrwert bieten kann. Es gilt also den optimalen Mittelweg zwischen der Weitergabe nützlicher Informationen und der Zurückhaltung von Information zum Schutz der Privatsphäre zu finden.

3.2 Systemmodell

In diesem Abschnitt möchten wir die Komponenten des Systems sowie mögliche Angriffsvektoren einführen, welche als Grundlage für die folgenden Betrachtungen dienen sollen. Es soll dabei keiner Komponente vertraut werden müssen, die nicht unserem physikalischen Zugriff unterliegt. Wir greifen dabei auf ein Systemmodell zurück, welches bereits in anderen Arbeiten [Sch11, Dü11] verwendet wurde.

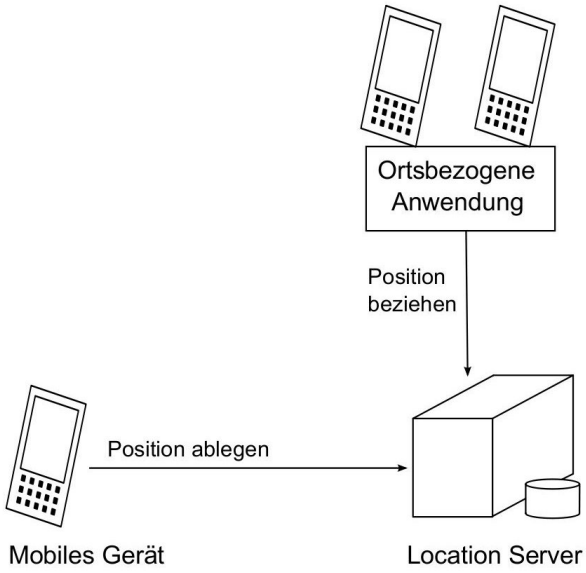

Abbildung 3.1: Angenommenes Systemmodell mit den drei Komponenten: *Mobiles Endgerät*, *Location Server* und *Ortsbezogene Anwendung* - Quelle: [Sch11]

Unser betrachtetes System besteht aus drei verschiedenen Komponententypen. Die Positionsdaten werden zunächst durch das mobilen Endgerät eines Nutzers bestimmt. Diese können beispielsweise mit Hilfe eines Positionierunssystem wie GPS in Form von kartesischen Geo-Koordinaten ermittelt werden. Die Ortsdaten, welche der Benutzer freigeben möchte werden mit einem Zeitstempel an einen oder mehrere Location Server übermittelt. Die Location Server verwalten und speichern diese Positionsdaten. Bei Bedarf werden diese Daten an Ortsbezogene Dienste (Location Based Services) weitergegeben, welche durch den Nutzer für den Zugriff autorisiert wurden. Ein ortsbezogener Dienst könnte beispielsweise eine Anwendung sein, welche den Benutzer abhängig von seiner Position mit aktuellen Stauinformationen versorgt.

Von den drei Komponenten des Systems befindet sich lediglich das mobile Endgerät unter der physikalischen Kontrolle des Benutzers. Wir gehen davon aus, dass dieses Gerät mit der darauf installierten Software vertrauenswürdig ist.
Sowohl die ortsbasierten Dienste als auch der Location Server befinden sich nicht unter Kontrolle des Benutzers und können daher auch nicht als vertrauenswürdig angesehen werden. Wir gehen daher zum einen davon aus, dass sämtliche Daten, welche an einen Location Server übermittelt wurden, dort auch gespeichert werden. Des Weiteren nehmen wir an, dass ein Angreifer ebenfalls Zugriff auf diese Daten erlangen kann. Der Angreifer kann folglich aus den übermittelten Positionsdaten ein ganzes Bewegungsprofil erstellen.
Wir gehen zudem davon aus, dass der Angreifer neben diesen Positionsdaten auch Karteninformationen zur Verfügung hat. Insbesondere gehen wir davon aus, dass sich der

Benutzer nur auf Straßen bewegt und der Angreifer diese Information nutzen kann.

3.3 Karteninformationen

Da wir davon ausgehen müssen, dass ein potentieller Angreifer Zugriff auf Karteninformationen hat, werden wir in unserem Verschleierungsverfahren ebenfalls auf Karteninformationen zurückgreifen. Wir gehen im Folgenden davon aus, dass wir für unseren Verschleierungsalgorithmus eine Datenbank mit den notwendigen Karteninformationen lokal vorliegen haben. Keine der Abfragen an die Datenbank wird protokolliert oder an Dritte übermittelt.
Des Weiteren gehen wir davon aus, dass in den Kartendaten zwischen den verschiedenen Straßentypen unterschieden wird. Als Grundlage für die Kategorisierung soll die "Richtlinie für integrierte Netzgestaltung"[Foro8] der Forschungsgesellschaft für Straßen- und Verkehrswesen dienen. Darin wird unter anderem eine Unterteilung der Straßen in verschiedene Kategorien vorgeschlagen. Die Kategoriestufen, welche wir hierbei voraussetzen wollen umfassen Autobahnen, Landstraßen, Verbindungsstraßen, Hauptverkehrsstraßen und Erschließungsstraßen. Unter die Kategorie Erschließungsstraßen fassen wir diejenigen Straßen zusammen, die in Wohn- und Industriegebiete führen. Ortsdurchfahrten fallen abhängig davon, ob sie bebaut sind in die Kategorie Verbindungsstraßen oder Hauptverkehrsstraßen. Als Landstraßen bezeichnen wir alle Straßen, die nicht nur innerorts verlaufen, sondern der Verbindung mehrerer Orte oder großräumigen Verbindungen dienen.

Des Weiteren gehen wir davon aus, dass eine Hierarchie von Gebieten gibt, welche durch geographische Gegebenheiten bestimmt ist. Jedes Land lässt sich in Regionen unterteilen, welche sich wieder in Landkreise unterteilen lassen, welche wiederum verschiedene Städte und Orte beinhalten. Städte setzen sich wiederum aus verschiedenen Stadtteilen zusammen, Stadtteile und Orte bestehen wiederum aus verschiedenen Wohngebieten. Wir gehen davon aus, dass wir zu einer gegebenen Position aus den Kartendaten zum einen ermitteln können, zu welchem Wohngebiet, zu welcher Stadt und zu welchem Landkreis diese gehört und zum anderen die Grenzen dieser Gebietseinheiten entnehmen können.
Die Einteilung in Landkreis, Stadt und Ort erfolgt entsprechend den behördlich festgelegten Grenzen. Als Wohngebiet betrachten wir ein Gebiet von zusammenhängenden Erschließungsstraßen, welche durch Hauptverkehrsstraßen und natürliche Grenzen umschlossen werden.

Abbildung 3.2: Darstellung eines Wohngebietes. Quelle: OpenStreeMap[Ope]

3.4 Betrachtete Wege

Wir gehen im Folgenden davon aus, dass sich Personen bevorzugt auf optimalen Wegen bewegen. Als optimal betrachten wir jeweils die zeitlich kürzesten Wege. Alternativ könnten hier auch andere Metriken betrachtet werden. Man könnte beispielsweise annehmen, dass Personen immer den Weg wählen, welcher einen minimalen Spritverbrauch verspricht. Wir wollen uns bei unseren Betrachtungen auf zeitlich kürzeste Wege beschränken, alternative Metriken könnten aber durch geringfügige Änderungen ebenfalls berücksichtigt werden. Diese Eigenschaften der Wegwahl können einem Angreifer bei bekannten Teilstrecken Rückschlüsse auf mögliche Ziele ermöglichen, die es zu verhindern gilt.
Tatsächlich weichen reale Wege des öfteren von den als optimal eingestuften Wegen ab, wie wir aus [LKH06] und [Kru09] wissen. Zudem ist der optimale Weg oft Uhrzeit- oder gar Jahreszeit abhängig. Dies wollen wir aber in unseren Betrachtungen nicht berücksichtigen.

Des Weiteren besteht die Möglichkeit, dass bereits die Wahl des Ziels Rückschlüsse auf den Startpunkt eines Weges ermöglicht. Liegen beispielsweise zwei Bäckereien A und B in der Nähe eines verschleierten Gebietes G, so kann die Wahl der Bäckerei bereits Rückschlüsse auf den Startpunkt innerhalb des verschleierten Gebiets ermöglichen, wenn ein Teil der möglichen Startpunkte aus G näher an A liegt und ein Teil der möglichen Startpunkte näher an B liegt. Vor allem kurze Wege können hierüber Rückschlüsse erlauben.
Für lange Wege sind solche Rückschlüsse kaum möglich. Lang bedeutet hierbei, dass die Länge des Weges wesentlich größer ist als die Ausdehnung des verschleierten Gebietes. Wo genau wir zwischen langen und kurzen Wegen die Grenzen ziehen, müssen wir

später genauer betrachten. Wird beispielsweise eine Gebiet mit einer Ausdehnung von 50 m verschleiert und sind alle Bäcker mindestens 10 km entfernt, so lässt die Wahl des angefahrenen Baumarkts keine Rückschlüsse über den Startpunkt zu, da die Entfernungen zu groß sind, als eine Weglänge von 50 m die Auswahl beeinflussen würde.
Für uns folgt daraus, dass wir zwischen kurzen und langen Wegen unterscheiden müssen und über kurze Wege möglicherweise gar keine Informationen herausgeben dürften.

Für lange Wege gehen wir zudem davon aus, dass der größte Teil der Route unabhängig von dem konkreten Startpunkt bzw. Endpunkt innerhalb des verschleierten Start- bzw. Zielgebietes gewählt wird. Anders ausgedrückt soll nur der Beginn und das Ende einer Route vom konkreten Start- bzw. Endpunkt innerhalb des verschleierten Gebietes abhängen.

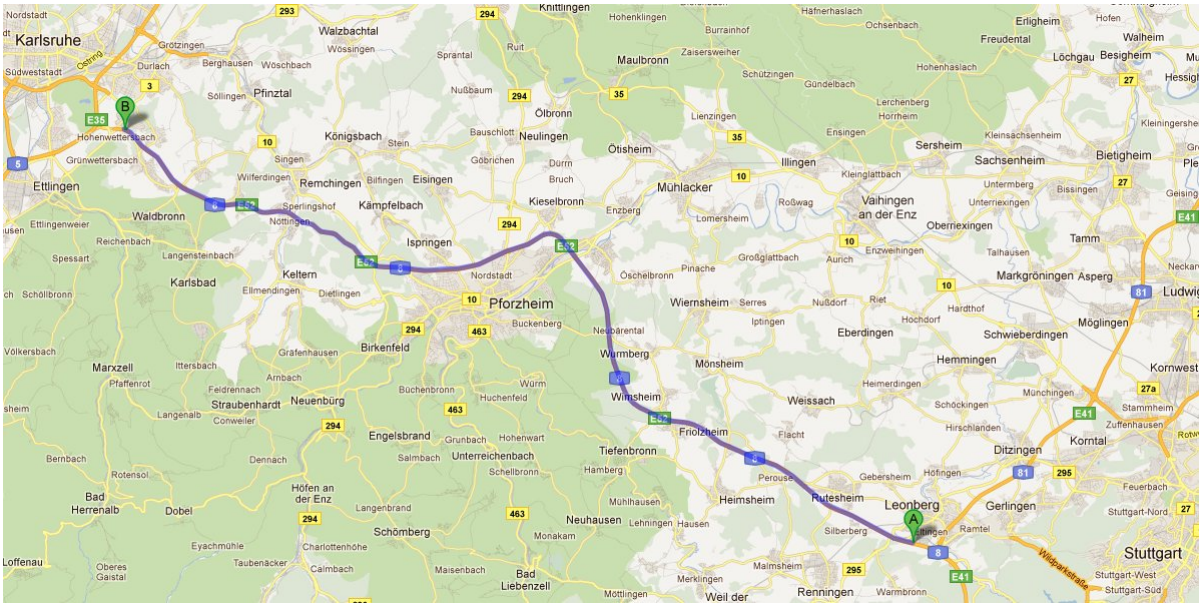

Abbildung 3.3: Karte mit gemeinsamer Teilstrecke aller optimalen Routen von Startpunkten in Stuttgart zu Endpunkten in Karlsruhe. Quelle: GoogleMaps [Goo]

Wird beispielsweise die Strecke Stuttgart-Karlsruhe mit dem Auto zurückgelegt, so hängt zwar die Wahl der Autobahnanschlussstelle von dem konkreten Start- und Endpunkt ab, aber nicht die Wahl des größten Teilstücks über die Autobahn, wie in Abbildung 3.3 dargestellt.

3.5 Langfristige Betrachtung

Während die in Kapitel 2.1.2 vorgestellten Verfahren nur einzelne Positionen betrachten, soll unser Verfahren auch für den dauerhaften Gebrauch geeignet sein. Unser Verfahren muss

folglich auch bei der Anwendung auf mehrere Trajektorien mit teilweise identischen Start- und Endpunkten noch sicher sein. Wir müssen daher auch betrachten, welche möglichen Implikationen sich durch die Betrachtung aufeinander folgender Trajektorien ergeben.

3.5.1 Zeitliche Rahmenbedingungen

Werden für die Nutzung eines ortsbasierten Dienstes Positionsinformationen herausgegeben, so sind diese explizit oder implizit mit einem Zeitpunkt assoziiert. Entweder wird dabei der Zeitpunkt, zu welchem die Position bestimmt wurde, mitübermittelt oder der Zeitpunkt der Anfrage lässt Rückschlüsse über den Zeitpunkt der Ortsbestimmung zu.
Sei nun ein Punkt P sowohl Ziel eines ersten ersten Weges w_1 als auch Startpunkt eines zweiten Weges w_2, wobei der Punkt P innerhalb eines Gebiet G verschleiert werden soll, so besteht die Möglichkeit, dass ein Angreifer anhand der zeitlichen Abfolge der Wege weitere Informationen über diesen Punkt gewinnen kann. Vergehen zwischen dem Zeitpunkt t_1 des Eintritts in das Gebietes G, welcher in den zu w_1 gehörenden Positionsinformationen übermittelt wurde, und dem Zeitpunkt t_2 des darauf folgenden Verlassen des Gebietes, welcher in den zu w_2 gehörenden Positionsinformationen übermittelt wurde nicht genügend viel Zeit, so kann ein Angreifer Punkte ausschließen, welche in dieser Zeit nicht erreichbar gewesen wären. Sei nun t_{max} für das verschleierte Gebiet G das maximale Zeitintervall, welches für zwei in G liegende Punkte p_1 und p_2 benötigt wird um von p_1 nach p_2 zu gelangen. Wir nehmen für die betrachteten Wege an, dass zwischen dem Ereignis des Eintretens in ein Gebiet G und das darauf folgende Verlassen des Gebietes mindestens die Zeit $2 \cdot t_{max}$ vergeht. Damit wird verhindert, dass nicht erreichbare Punkte durch den Angreifer ausgeschlossen werden können.
Sollte diese Annahme verletzt werden, so müsste zusätzlich zu einer örtlichen Verschleierung noch eine zeitliche Verschleierung der Positionsinformationen vorgenommen werden. Diese betrachten wir hier nicht.

3.6 Anforderungen

Für einen gegebenen Start- bzw. Endpunkt soll unser Verfahren ein Gebiet berechnen, innerhalb dessen jede unter Berücksichtigung von Karteninformationen als Start- oder Endpunkt in Frage kommende Position dieselbe Wahrscheinlichkeit hat Start- bzw. Endpunkt der herausgegebenen Strecke zu sein. Als Gebiet bezeichnen wir hierbei einen zusammenhängende Fläche im kartesischen Koordinatensystem. Für einen Angreifer soll sich für die Positionen, die innerhalb dieses Gebietes als Start- bzw. Endpunkt in Frage kommen mit und ohne Kenntnis der herausgegebenen Positionsinformationen dieselbe Wahrscheinlichkeitsverteilung ergeben. Könnte der Angreifer anhand der herausgegebenen Positionsinformationen mögliche Kandidaten ausschließen oder mit einer höheren Wahrscheinlichkeit als Ziel ausmachen, so bezeichnen wir diese Information als kritisch. Durch kritische Informationen kann der Angreifer also mehr Wissen über den Endpunkt erlangen, als wir zulassen wollen. Unser Verfahren darf daher keine kritischen

Positionsinformationen herausgeben. Der Schutz der Privatsphäre, welchen unser Verfahren garantieren soll, bedeutet also, dass der Start- bzw. Endpunkt innerhalb des gewählten Verschleierungsgebietes nicht weiter eingeschränkt werden kann.
Bei der Abwägung zwischen Schutz der Privatsphäre und Herausgabe von Informationen muss beachtet werden, dass einmal herausgegebene Informationen nicht zurückgenommen werden können. Primäres Ziel muss also der Schutz der Privatsphäre sein, die Herausgabe der Information muss im Zweifelsfall hinter diesem Ziel zurückstehen. Informationen dürfen folglich nur dann herausgegeben werden, wenn sie zweifelsfrei als unkritisch eingestuft werden können.

Um Positionsinformationen zu schützen, gibt es in der Literatur bereits verschiedene Ansätze. Wir konzentrieren uns hier darauf die Positionsinformationen zu identifizieren, die es dem Angreifer ermöglichen den Start- bzw. Endpunkt innerhalb des geschützten Gebiets weiter einzuschränken. Diese Informationen dürfen nicht herausgegeben werden.

Wir setzen voraus, dass Start- und Endpunkt eines Weges bekannt sind. Der Endpunkt kann dabei beispielsweise durch Eingabe des Zielorts durch den Benutzer ermittelt werden. Ausgehend davon soll sowohl für den Start- als auch für den Endpunkt deterministisch ein Gebiet ermittelt werden, in welchem der Start- bzw. Endpunkt nicht weiter eingeschränkt werden kann. Wir müssen dabei davon ausgehen, dass die herausgegebenen unkritischen Daten dieses Gebiet charakterisieren und dieses Gebiet damit auch einem Angreifer bekannt ist. Daher darf die Gebietswahl keine weiteren Rückschlüsse über den verschleierten Start- bzw. Endpunkt zulassen. Diese Forderung ist bei einer deterministischen Wahl des Gebietes äquivalent dazu, dass für alle Start- bzw. Endpunkte, die in diesem Gebiet liegen genau dieses Gebiet gewählt wird.

Die resultierenden Anforderungen lassen sich wie folgt zusammenfasse:

1. Zu jedem Start- und Endpunkt ermittelt das Verfahren einen Bereich, innerhalb dessen ein Angreifer mögliche Start- und Endpunkte nicht weiter einschränken kann.
2. Die Wahl des verschleierten Gebiets darf keine Informationen liefern, durch welche die möglichen Start- und Endpunkte weiter eingeschränkt werden können.
3. Sofern Positionsinformationen über das verschleierte Gebiet hinaus keine weiteren Informationen über Start- und Endpunkte liefern, werden diese durch das Verfahren an den Location Server übermittelt.

4 Das Verschleierungsverfahren

Im Folgenden werden wir ein Verschleierungsverfahren angeben, welches die zuvor beschriebenen Anforderungen erfüllt. Das Verfahren umfasst im wesentlichen zwei Phasen. In der Initialisierungsphase wird für einen gegebenen Start- bzw. Endpunkte durch das Verfahren ein Gebiet ermittelt, innerhalb dessen die Kandidaten für Start- bzw Endpunkt durch die herausgegebennen Positionsinformationen nicht weiter eingeschränkt werden können. Wir werden dieses Gebiet daher als verschleiertes Gebiet bezeichnen. Für dieses verschleierte Gebiet wird anschließend ermittelt, welche Positionsinformationen außerhalb dieses Gebietes Rückschlüsse über den Start- bzw. Endpunkt zulassen, mit denen die Kandidaten innerhalb dieses Gebietes weiter eingeschränkt werden können. Wir fassen diese kritischen Positionen zu einem weiteren Gebiet zusammen, welches wir als kritisches Gebiet bezeichnen. Innerhalb eines kritischen Gebiets sollen keine Positionsinformationen herausgegeben werden.
In der zweiten Phase, der Übermittlungsphase, werden durch das Verfahren regelmäßige Positionsupdates an den Location Server übermittelt. Hierbei kontrolliert das Verfahren, dass keine kritischen Positionsinfomationen übermittelt werden.

Wir werden zunächst die Grundidee des Verfahrens diskutieren, anschließend werden wir die beiden Phasen im Detail betrachten und schließlich werden wir das Verfahren algorithmisch fassen und angeben wie wir es mit Karteninformationen, die uns in Form eines Graphen vorliegen, durchführen können.

Für die Diskussion des Verfahrens werden wir den Fall betrachten, dass für gegebene Start- und Endpunkte das zugehörige Wohngebiet verschleiert werden soll. Wie setzen somit ein festes Maß für die Verschleierung fest. Das Verfahren selbst kann auch auf größere oder kleinere Verschleierunggebiete übertragen werden.

4.1 Grundidee

Die Grundidee des Verfahrens basiert auf der Annahme, dass die bezüglich unseres verschleierten Gebietes kritischen Positionsinformationen eines Weges örtlich nah an dem verschleierten Gebiet liegen. Betrachten wir den Fall, dass der Endpunkt innerhalb eines Wohngebietes verschleiert werden soll, so werden zwar die gewählten Zufahrtsstraßen rund um das Wohngebiet Aufschluss über mögliche Endpunkte innerhalb des Wohngebietes

ermöglichen, für Langstrecken ist jedoch die gewählte Autobahnauffahrt unabhängig vom konkreten Start- bzw. Endpunkt innerhalb des Wohngebietes dieselbe.

Die kritischen Positionsinformationen sind also örtlich beschränkt und in der Nähe des verschleierten Gebiets lokalisiert.
Diese Eigenschaft nutzen wir in unserem Verfahren um ein kritisches Gebiet, innerhalb dessen wir keine Positionsinformationen herausgeben dürfen, möglichst exakt angeben zu können. Dazu wählen wir für das Gebiet, welches wir verschleiern möchten, ein umgebendes Gebiet, welches insbesondere auch das Gebiet umfasst, in welchem alle kritischen Positionsinformationen liegen. Wir wählen also das umgebende Gebiet als eine Überapproximation des Gebiets, in welchem kritische Positionsinformationen lokalisiert sind. Anschließend betrachten wir eine geeignete Menge von Wegen, für die der Startpunkt außerhalb des umgebenden Gebietes liegt und der Endpunkt innerhalb des verschleierten Gebiets oder umgekehrt. Wie diese Wege und deren Endpunkte zu wählen sind, werden wir später genauer betrachten. Anhand der betrachteten Wege identifizieren wir anschließend Positionsinformationen, die Rückschlüsse über einen konkreten Start- bzw. Endpunkt innerhalb des verschleierten Gebietes liefern.

Da wir auf diesem Weg, keine Aussagen über Wege treffen, für die sich Start- und Endpunkt innerhalb des selben umgebenden Gebiets befinden, können wir für diese Wege keine Positionsinformationen herausgeben.

4.2 Karteninformationen

Wir gehen davon aus, dass uns die Karteninformationen in Form eines ungerichteten Graphen $G = (V, E)$ vorliegen. Straßenkreuzungen und Endpunkte entsprechen dabei den Knoten, die Straßen werden als Kanten repräsentiert. Ein Knoten $v \in V$ wird dabei durch ein Tripel $v = (vid, long, lat)$ dargestellt, wobei vid ein eindeutiger ganzzahliger Identifier für den Knoten ist und das Tupel $(long, lat)$ die Geographischen Koordinaten des Punktes angeben. Die Kanten $e \in E$ werden durch ein 5-Tupel $e = (eid, v_{start}, v_{end}, type, weight)$ dargestellt, wobei eid ein eindeutiger ganzzahliger Identifier für die Kante ist, v_{start} und v_{end} die Knoten enthält, welche durch v verbunden werden, $type$ die in Kapitel 3.3 beschriebene Kategorie der Straße angibt und $weight$ ein Gewicht für die Kante angibt, welches proportional zur Dauer ist, welche benötigt wird um die Strecke entlang der Kante zurückzulegen.

Des Weiteren liegen Stadt- und Landkreis in Form eines Untergraphen $H \subseteq G$ vor. Zudem können wir effektiv die Menge der Kanten ermitteln, für welche genau einer der beiden zugehörigen Knoten in H liegt.

4.3 Initialisierung

Für gegebene Start- und Endpunkte wird in der Initialisierungsphase zunächst das Gebiet ermittelt, welches verschleiert werden kann. Dieses Gebiet stellen wir durch einen Graphen G_V dar, welcher ein Untergraph der Karteninformationen G ist. Anschließend wird darauf basierend ein Gebiet ermittelt, innerhalb dessen keine Positionsinformationen herausgegeben werden dürfen um die Verschleierung gewährleisten zu können. Auch dieses kritische Gebiet stellen wir ebenfalls durch den Graphen dar, den wir mit G_K bezeichnen.
Die Berechnung dieser beiden Gebiete erfolgt in mehreren Schritten. Bevor wir diese Schritte im einzelnen diskutieren, wollen wir kurz den Ablauf der Initialisierung skizzieren.

Im ersten Schritt berechnen wir basierend auf dem gegebenen Start- bzw. Endpunkt, den Graphen G_V, welcher das verschleierte Gebiet repräsentiert. In diesem Schritt ermitteln wir zudem alle Knoten, die in diesem Gebiet liegen und von welchen eine Kante aus diesem Gebiet heraus führten. Die so ermittelten Knoten bezeichnen wir als inneren Rand des verschleierten Gebietes und werden sie im Folgenden mit R_V bezeichnet. R_V ist eine Teilmenge der zu G_V gehörenden Knotenmenge.
Im zweiten Schritt ermitteln wir ein Gebiet, welches unser verschleiertes Gebiet umgibt und von welchem wir annehmen, dass alle bezüglich unseres verschleierten Gebietes kritische Positionsinformationen innerhalb dieses Gebietes liegen. Dieses Gebiet bezeichnen wir als umgebendes Gebiet und identifizieren es mit dem Graphen G_U. Für ein gegebenes Wohngebiet bestimmen wir beispielsweise die Stadt, in welcher das Wohngebiet liegt, da wir davon ausgehen, dass alle bezüglich eines Endpunkts im Wohngebiet kritischen Positionsinformationen innerhalb der Stadt anfallen. Für den das umgebende Gebiet repräsentierenden Graphen G_U bestimmen wir alle Knoten, welche nicht zu der Knotenmenge von G_U gehören, aber über eine Kante mit einem zu G_U gehörenden Knoten verbunden sind. Diese fassen wir zum äußeren Rand des umgebenden Gebietes zusammen, den wir im Folgenden mit R_U bezeichnen.
Im dritten Schritt berechnen wir alle zu Kanten G_U gehörenden Kanten, für die wir davon ausgehen, dass die Information, dass der Nutzer die Straße befährt, welche durch diese Kante repräsentiert wird, keine kritische Information darstellt. Dies bedeutet insbesondere, dass Positionsinformationen, die wir diesen Kanten zuordnen einem Angreifer keine Möglichkeit geben, den tatsächlichen Endpunkt innerhalb des verschleierten Gebiets weiter einzuschränken. Diese Kanten bezeichnen wir als unkritische Kanten.
Im vierten Schritt ermitteln wir aus den zuvor gewonnen Informationen alle zum Graphen G_U gehörenden Kanten, die wir nicht im vorherigen Schritt als unkritisch eingestuft haben. Da wir nur zweifelsfrei unkritische Positionsinformationen herausgeben dürfen, stufen wir diese Kanten als kritisch ein. Mit den zugehörigen Knoten ergibt sich daraus der Graph G_K. Nach Konstruktion gilt $G_V \subseteq G_K \subseteq G_U$.

Zur Illustration des Verfahrens werden wir die Durchführung am Beispiel des in Abbildung 4.1 dargestellten Ortes Büsnau betrachten.

Abbildung 4.1: Beispiel: Der Ort Büsnau bei Stuttgart
Rot: Endpunkt, der Verschleiert werden soll

Da die Initialisierungsschritte für Start- und Endpunkte analog verlaufen, werden wir im Folgenden zunächst beschreiben, wie der Initialisierungsvorgang für den Endpunkt eines Weges abläuft. Dieses Verfahren lässt sich anschließend analog auf Startpunkte übertragen.

4.3.1 Berechnung des verschleierten Gebietes G_V

Im Folgenden wollen wir den Endpunkt eines Weges in einem Wohngebiet verschleiern. Während wir uns bei der Definition der Gebietsgrößen Stadt- und Landkreis auf anerkannte Konventionen berufen haben und daher auch annehmen können, dass diese Informationen in Kartendaten enthalten sind, müssen wir ein Wohngebiet zunächst berechnen.
Für einen gegebenen Punkt auf der Karte gehen wir davon aus, dass wir die zugehörige Straße kennen, an welcher dieser Punkt liegt. Dafür gehen wir prinzipiell so vor, dass wir wir die angrenzenden Straßen soweit verfolgen, bis wir auf eine Verbindungsstraße treffen und alle dabei besuchten Straßen zum Wohngebiet hinzunehmen.

Für die algorithmische Beschreibung gehen wir davon aus, dass wir als Ausgangspunkt eine Kante haben, zu welcher wir das zu verschleiernde Gebiet ermitteln. Diese Kante kann vorher durch einen Map Matching Algorithmus bestimmt worden sein. Auf Eingabe der Kante berechnen wir wie in Algorithmus 4.1 beschrieben das Wohngebiet, zu welchem diese Kante gehört.

Abbildung 4.2: Rot: Endpunkt, der Verschleiert werden soll
Gelb: Das zum verschleierten Punkt berechnet Wohngebiet

Nachdem wir zu einem gegebenen Punkt das zu verschleiernde Gebiet ermittelt haben, muss noch die Menge der Knoten R_V berechnet werden, die den inneren Rand dieses Gebietes ausmachen. Prinzipiell könnten wir diese Berechnung auch in Algorithmus 4.1 integrieren, wir wollen diese Berechnung hier aber aus zwei Gründen separat diskutieren. Zum einen wird dadurch die Berechnung verständlicher und zum anderen erleichtert es die Anwendung des Verfahrens auf größere oder kleiner Gebiete.
Ausgehend von einem Gebiet, gegeben in Form eines Untergraphen der kompletten Karte, lässt sich wie in Algorithmus 4.2 auf simple Weise die Menge der Knoten berechnen, die eine Verbindung aus diesem Gebiet heraus ermöglichen.

Auf diese Weise erhalten wir durch *calcResidentialArea* das Gebiet G_V, welches wir verschleiern möchten und durch *calcInnerBorder* die zugehörigen Randpunkte R_V.

Das so berechnete G_V beschreibt nun unser Gebiet welches wir verschleiern möchten. R_V beschreibt nach Definition die Menge der Knoten, über welche dieses Gebiet betreten oder verlassen werden kann. Da jeder Weg in dieses Gebiet hinein oder aus diesem Gebiet heraus über einen der Knoten aus R_V führt, wissen wir, dass auch alle kürzesten Wege von einem Punkt innerhalb dieses Gebietes zu einem Punkt außerhalb dieses Gebietes über einen der Knoten aus R_V führt.

Algorithmus 4.1 Berechnung des Wohngebietes

```
1: VertexSet visitedVertices ← ∅;
2:
3: function calcResidentialSubarea(Vertex v)
4:     EdgeSet: E, Graph: K, Vertex: next,
5:     K ← ({v}, ∅)
6:     E ← getIncidentEdges(v)
7:     visitedVertices ← visitedVertices ∪ {v}
8:     for e ∈ E do
9:         if e.type ≠ RESIDENTIALROAD then
10:            return K
11:        end if
12:    end for
13:    for e ∈ E do
14:        next ← e.v_start
15:        if next = v then
16:            next = e.v_end
17:        end if
18:        K ← K ∪ ({v, next}, {e})
19:        if next ∉ visitedVertices then
20:            K ← K ∪ calcResidentialSubarea(next)
21:        end if
22:    end for
23:    return K
24: end function
25:
26: function calcResidentialArea(Edge e)
27:    Vertex: start, end, Graph: K
28:    start ← e.v_start
29:    end ← e.v_end
30:    K ← ({e.v_start, e.v_end}, {e})
31:    K ← K ∪ calcResidentialSubarea(start)
32:    K ← K ∪ calcResidentialSubarea(end)
33:    return K
34: end function
```

Betrachten wir nun einen Knoten k, welcher nicht zu G_V gehört und betrachten für $l \in R_V$ den kürzesten Weg von k nach l. Für eine Kante e welche zu allen diesen kürzesten Wegen gehört wissen wir folglich, dass e zu allen kürzesten Wegen von k zu einem beliebigen Knoten aus G_V gehört. Damit ist die Information, dass sich der Nutzer über die Straße bewegt, welche durch diese Kante repräsentiert wird, unkritisch.

Algorithmus 4.2 Berechnung des inneren Randes eines Gebietes G

```
1:  function calcInnerBorder(Graph G)
2:      VertexSet: R_V, EdgeSet: connections
3:      for v ∈ G.V do
4:          connections ← getIncidentEdges(v)
5:          for e ∈ connections do
6:              if e.v_start ∉ G.E or e.v_end ∉ G.E then
7:                  R_V ← R_V ∪ {v}
8:                  break
9:              end if
10:         end for
11:     end for
12:     return R_V
13: end function
```

4.3.2 Berechnung des umgebenden Gebietes G_U

Gehen wir davon aus, dass wir ein Wohngebiet verschleiern möchten, so wählen wir als umgebendes Gebiet die zugehörige Stadt bzw. den zugehörigen Ort. Da wir bei der Wahl dieser Gebietseinheit an behördlich festgelegten Grenzen orientiert haben, gehen wir davon aus, dass diese Informationen in den Kartendaten enthalten sind. Wir gehen also davon aus, dass wir als umgebendes Gebiet einen Graphen G_U aus den Kartendaten direkt entnehmen können. Auf diesen Daten müssen wir anschließend noch die Knoten ermitteln, die über eine Kante mit einem Knoten aus diesem Gebiet verbunden sind. Diese könne wir durch Algorithmus 4.3 berechnen.

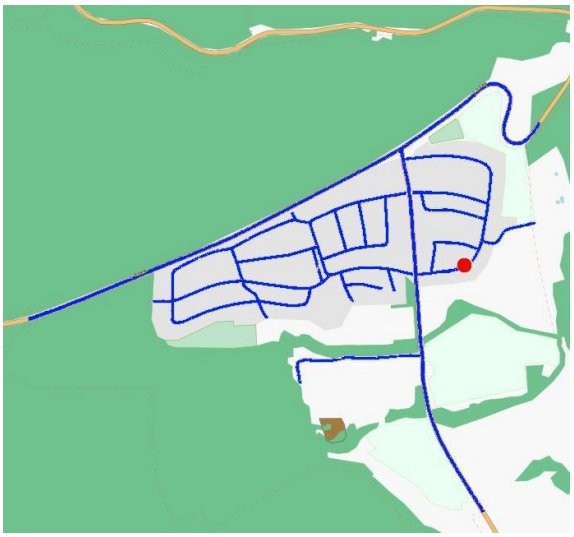

Abbildung 4.3: Rot: Endpunkt, der Verschleiert werden soll
Blau: Das zum verschleierten Punkt berechnet Stadtgebiet

Algorithmus 4.3 Berechnung des äußeren Randes eines Gebietes G

```
1:  function calcOuterBorder(Graph G)
2:      VertexSet: R_U, EdgeSet: connections, Vertex: out
3:      for v ∈ G.V do
4:          connections ← getIncidentEdges(v)
5:          for e ∈ connections do
6:              if e.v_start ∉ G.E and e.v_end ∈ G.E then
7:                  out ← e.v_start
8:                  R_U ← R_U ∪ {out}
9:              else if e.v_start ∈ G.E and e.v_end ∉ G.E then
10:                 out ← e.v_end
11:                 R_U ← R_U ∪ {out}
12:             end if
13:         end for
14:     end for
15:     return R_U
16: end function
```

Nach dieser Berechnung haben wir die folgenden Informationen zur Verfügung:

- G: Die Karteninformationen in Form eines Graphen.
- G_V, R_V: Das zu verschleiernde Gebiet in Form eines Graphen mit den zugehörigen Knoten, welche den inneren Rand dieses Gebietes beschreiben.
- G_U, R_U: Das zu umgebende Gebiet in Form eines Graphen mit den zugehörigen Knoten, welche den äußeren Rand dieses Gebietes beschreiben.

Insbesondere gilt dabei $G_V \subset G_U$.

4.3.3 Berechnung unkritischer Kanten

Ausgehend von den bisher berechneten Informationen ermitteln wir nun für jeden der Knoten $v \in R_U$ des äußeren Randes des umgebenden Gebietes die kürzesten Wege zu allen Knoten $w \in R_V$ des inneren Randes des verschleierten Gebietes. Diese Wege geben uns Aufschluss darüber welche Teilstrecken (Kanten in unserem Graphen) Informationen über mögliche Endpunkte liefern.

Betrachten wir hierzu zunächst einen fixen Knoten $v \in R_U$ und berechnen von dort die kürzesten Pfade zu allen Knoten aus R_V. Für einen Weg, welcher von einem Punkt außerhalb des umschließenden Gebietes G_U zu einem Punkt innerhalb des Gebietes G_V führt, liefern die Positionsinformationen, welche einer Kante e zugeordnet werden genau dann Informationen über diesen Zielpunkt, wenn diese Kante auf mindestens einem, aber nicht auf allen der kürzesten Pfade von v zu einem Knoten aus R_V liegt.

Führen wir diese Betrachtung für alle Knoten $v \in R_U$ durch, so identifizieren wir dadurch alle Kanten, die einem Angreifer bezüglich des verschleierten Gebietes Information liefern können.

Für unseren Algorithmus verwenden wir eine Funktion *getShortestPath*, welche uns für zwei gegebene Knoten v und w den kürzesten Pfad in Form einer Folge von Kanten berechnet. Diese Funktion kann im einfachsten Fall durch einen Dijkstra-Algorithmus realisiert sein, kann aber auch einen effizienteren Routing-Algorithmus verwenden.
Da für uns in diesem Zusammenhang die Reihenfolge, in welcher die Kanten eines Weges besucht werden keine Rolle spielt, behandeln wir die Folge der Kanten als Kantenmenge.
Die Berechnung der unkritischer Kanten erfolgt wie in Algorithmus 4.4 angegeben.

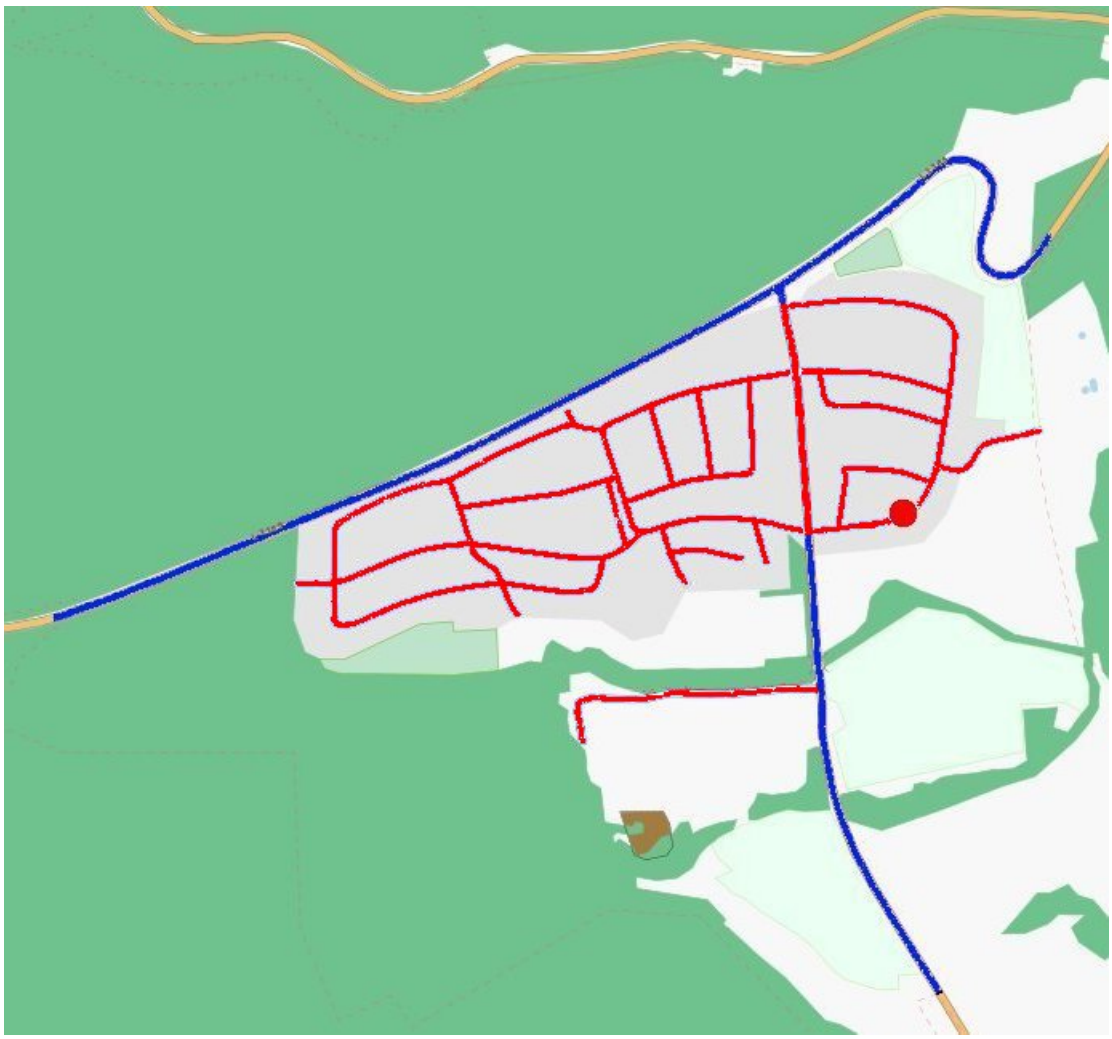

Abbildung 4.4: Blau: Unkritische Kanten
Rot: Kritisches Gebiet

Algorithmus 4.4 Berechnung unkritischer Kanten

```
1: function calcUncriticalEdges(VertexSet: R_U, VertexSet: R_V, Graph: G_U, Graph: G_V)
2:     EdgeSet: critical, uncritical
3:     EdgeSet: localCritical, localUncritical
4:     EdgeSet: path
5:     critical ← ∅
6:     uncritical ← ∅
7:     for out ∈ R_U do
8:         localUsed ← ∅
9:         localUncritical ← G_U.E
10:        for in ∈ R_V do
11:            path ← getShortestPath(out, in)        // we only need the set of used edges
12:            localUsed ← localUsed ∪ path
13:            localUncritical ← localUncritical ∩ path
14:        end for
15:        uncritical ← uncritical ∪ localUncritical
16:        critical ← critical ∪ (localUsed \ localUncritical)
17:    end for
18:    uncritical ← uncritical \ critical
19:    return uncritical
20: end function
```

4.3.4 Berechnung des kritischer Gebiets G_K

Nachdem wir nun die unkritischen Kanten berechnet haben, könnten wir die zu G_U gehörenden Kanten in drei verschiedene Kategorien einordnen. Zum einen gibt es natürlich Kanten, die wir entsprechend dem vorherigen Berechnungsschritt als unkritisch eingestuft haben. Als zweites haben wir Kanten, die wir zweifellos als kritisch einstufen können und als drittes gibt es Kanten, über die wir in unseren bisherigen Betrachtungen keine Aussage getroffen haben. Da wir jedoch nur dann Positionsinformationen herausgeben, wenn wir diese zweifelsfrei als unkritisch einstufen können, werden wir für unseren Algorithmus keine Unterscheidung zwischen der zweiten und der dritten Kategorie machen. Wir betrachten alle Kanten, die wir entsprechend Algorithmus 4.4 nicht als unkritisch eingestuft haben als kritische Kanten.
Die Berechnung der kritischen Kanten kann folglich entsprechend Algorithmus 4.5 erfolgen.

Algorithmus 4.5 Berechnung kritischer Kanten

```
1: function calcCriticalEdges(VertexSet: R_U, VertexSet: R_V, Graph: G_U, Graph: G_V)
2:     EdgeSet: critical, uncritical
3:     uncritical ← calcUncriticalEdges(R_U, R_V, G_U, G_V)
4:     critical ← G_U.V \ uncritical
5:     return critical
6: end function
```

Der Graph $G_K = (V_K, E_K)$, welcher das kritische Gebiet beschreibt, ergibt sich aus den kritischen Kanten und den zugehörigen Knoten in natürlicher Weise. Da wir für die folgenden Berechnungen nur die Menge der zu G_K gehörenden Kanten E_K benötigen, beschränken wir uns auf die Berechnung dieser Kantenmenge.

4.3.5 Zusammenfassung der Initialisierungsphase

Nachdem wir nun die einzelnen Schritte der Initialisierung beschrieben haben, können wir daraus den gesamten Ablauf der Initialisierung beschreiben.
Für die vollständige Initialisierung müssen die beschrieben Schritte natürlich zweimal durchgeführt werden, einmal für den Startpunkt und einmal für den Endpunkt. Da wir unsere Berechnungen auf einem ungerichteten Graphen ausführen, spielt es für die Berechnungen keinen Rolle, ob wir den Start- oder den Endpunkt betrachten. Wie sich unser Modell erweitern lässt um auch Einbahnstraßen berücksichtigen zu können, werden wir später in Kapitel 6.3 betrachten.

Algorithmus 4.6 Initialisierung

```
1: EdgeSet: E_K_start, E_K_end
2: function init(Edge start, Edge end)
3:     Graph: obfuscated, surrounding
4:     EdgeSet: innerBorder, outerBoder
5:     obfuscated ← calcResidentialArea(start)
6:     surrounding ← calcCityArea(start)
7:     innerBorder ← calcInnerBorder(obfuscated)
8:     outerBorder ← calcOuterBorder(surrounding)
9:     E_K_start ← calcCriticalEdges(outerBorder, innerBorder, surrounding, obfuscated)
10:
11:    obfuscated ← calcResidentialArea(end)
12:    surrounding ← calcCityArea(end)
13:    innerBorder ← calcInnerBorder(obfuscated)
14:    outerBorder ← calcOuterBorder(surrounding)
15:    E_K_end ← calcCriticalEdges(outerBorder, innerBorder, surrounding, obfuscated)
16: end function
```

Für die weiteren Berechnungen benötigen wir die beiden Mengen der bezüglich Startpunkts kritischen Kanten E_K_start und die bezüglich Endpunkts kritischen Kanten E_K_end. Diese werden entsprechend Algorithmus 4.6 berechnet und stehen anschließend für weitere Berechnungen zur Verfügung.

4.4 Positionsupdate

Nachdem wir in der Initialisierungsphase bereits bestimmt haben, welche Straßenabschnitte, repräsentiert durch die Kanten eines Graphen, kritische Positionsinformationen liefern können, müssen wir bei der Herausgabe unserer Positionsinformationen nur noch prüfen, ob die aktuelle Position, an der wir uns befinden als kritisch eingestuft wird.

In Algorithmus 4.7 betrachten wir die Übermittlung von Positionsinformationen. Wir gehen davon aus, dass die Funktion *positionUpdate* aufgerufen wird, wenn das Gerät eine Veränderung der Position feststellt. Als Parameter verwendet nimmt diese Funktion den Längen- und Breitengrade der neuen Position entgegen und überprüft daraufhin, ob die neue Positionsinformation weitergegeben werden darf oder nicht. Darf die Position weitergegeben werden, so sorgt die aufgerufene Funktion *sendPositionUpdate* dafür, dass die Position an den Location Server übertragen wird.
Die Funktion *mapMatch* gibt dabei für eine durch Längen- und Breitengrad gegeben Position die Kante zurück, welcher die Positionsinformation durch einen Map Matching Algorithmus zugeordnet wird. Dieser Algorithmus kann im einfachsten Fall diejenige Kante ermitteln, welche die geringste örtliche Distanz zum angegebenen Punkt besitzt. Da die Positionsbestimmung im Normalfall einer gewissen Toleranz unterliegt, können hier auch fortgeschrittenere Verfahren Einsatz finden, welche die vorhergehenden Positionen mit einbeziehen um eine möglichst exakte Angabe zu ermöglichen.

Algorithmus 4.7 Positionsupdate

```
1:  EdgeSet: E_K_start, E_K_end                    // calculated during initialization
2:  function positionUpdate(Float latitude, Float longitude)
3:      Edge: currentEdge
4:      currentEdge ← mapMatch(latitude, longitude)
5:      if currentEdge ∉ (E_K_start ∪ E_K_end) then
6:          sendPositionUpdate(latitude, longitude)
7:          return TRUE
8:      else
9:          return FALSE
10:     end if
11: end function
```

Die Funktion *positionUpdate* gibt über einen boolschen Rückgabewert an, ob die aktuelle Position an den Location Server übermittelt wurde, oder nicht. Diese Information könnte beispielsweise dazu genutzt werden um zu entscheiden

5 Analyse des Verfahrens

5.1 Ermittelte Daten

Zur Analyse des Verfahrens wurde exemplarisch für sämtliche Erschließungsstraßen der Stadt Berlin das zugehörige verschleierte Gebiet (Wohngebiet) berechnet. Als Grundlage der Analyse wurden die OpenStreetMap Kartendaten für Berlin [Geo] verwendet.

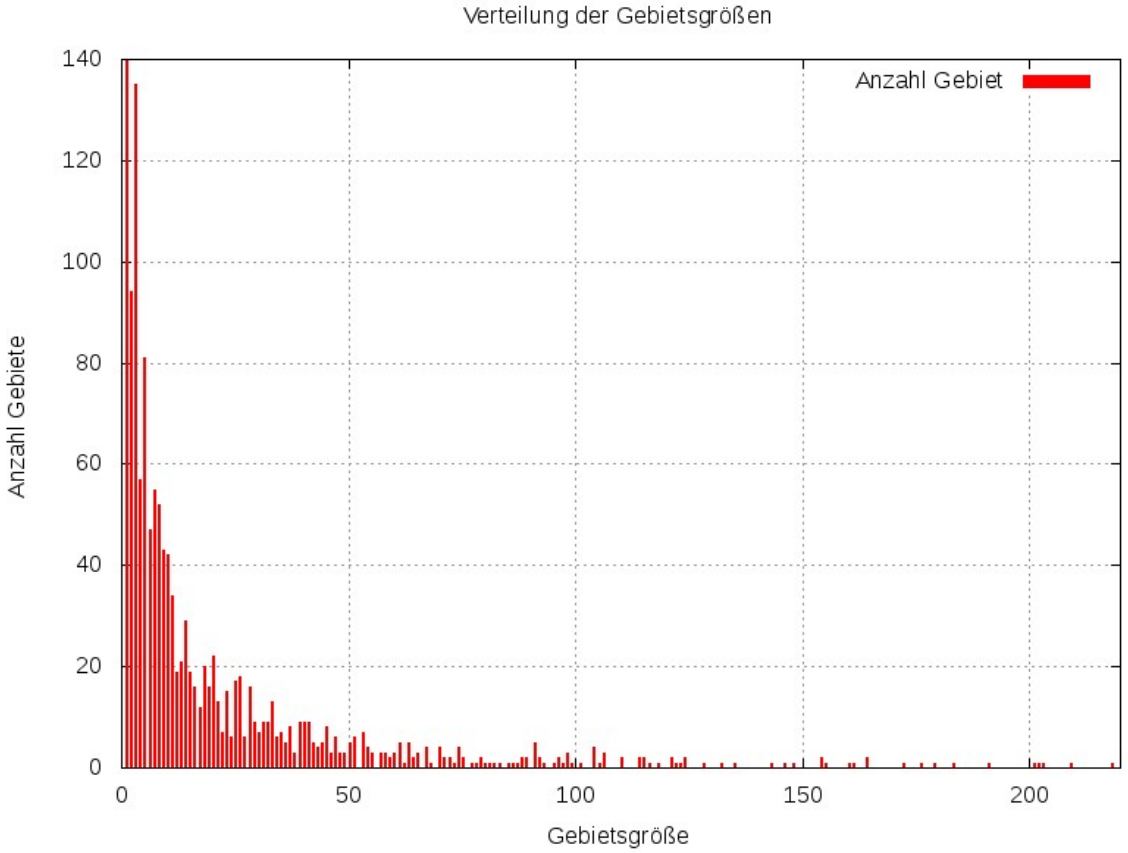

Abbildung 5.1: Größenverteilung der berechneten Wohngebiete. Es ergaben sich insgesamt 1438 Gebiete mit genau einer Kanten. Für eine bessere Darstellung wurde die Anzahl der Kanten auf bei 140 abgeschnitten.

Analysiert wurden dabei 27260 Kanten, welche die Erschließungsstraßen repräsentierten. Es wurden hierfür insgesamt 2639 disjunkte Gebiete berechnet . Hierbei ergab sich die in Abbildung 5.1 dargestellte Verteilung der ermittelten Gebietsgrößen.
Die größe der berechneten Gebiete bewegte sich zwischen einer und 218 Kanten. Für insgesamt 1438 Kanten (ca 5,3%) enthält das zugehörige Wohngebiet lediglich diese eine Kante.

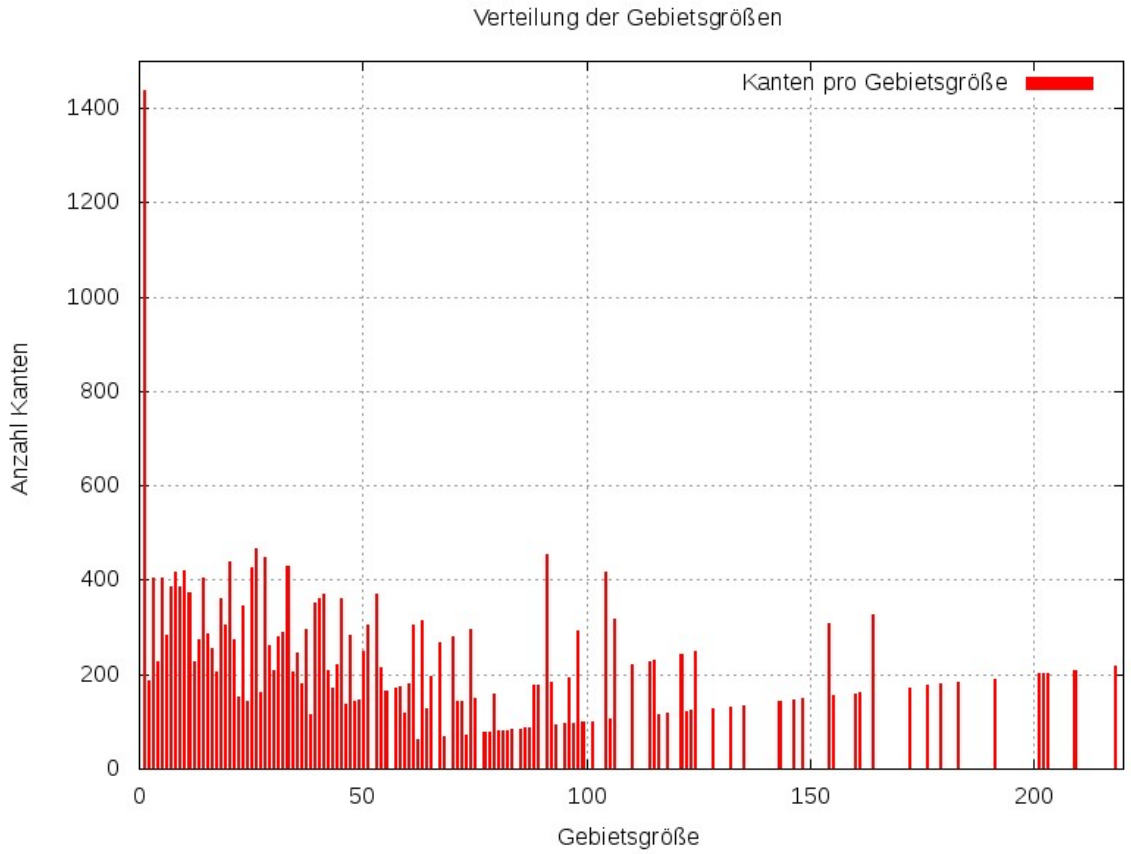

Abbildung 5.2: Anzahl der Kanten pro Gebietsgröße

Die Anzahl der Kanten, die auf die jeweiligen Gebietsgrößen abgebildet werden, ist in Abbildung 5.2 dargestellt. Die durschnittliche Größe eines Wohngebietes beträgt 10,3 Kanten.

Die Analyse der Daten ergab zudem, dass die in Kapitel 3.6 gestellten Anforderungen für das analysierte Gebiet erfüllt werden konnten. Insbesondere ergab sich sich bei den Berechnungen für alle zu einem Wohngebiet gehörenden Kanten dasselbe Wohngebiet.

5.2 Sicherheitsanalyse

Um die Sicherheit unseres Verfahrens bewerten zu können, müssen wir zunächst mehrere Fragen beantworten. Wir werden zunächst ermitteln, wie wir die Sicherheit messen können und welches Maß an Sicherheit unser Verfahren grundsätzlich bieten kann. Darauf aufbauend werden wir uns damit auseinandersetzen, welche Vor- und Nachteile unser Verfahren mitbringt und an welcher Stellen ein Angreifer möglicherweise Schwächen des Verfahrens ausnutzen kann.
Abschließend werden wir versuchen die gewonnenen Erkenntnisse richtig einzuordnen um eine Bewertung des Verfahrens vornehmen zu können und auf mögliche Lösungsansätze für erkannte Probleme hinzuweisen.

5.2.1 Messbarkeit

Grundsätzlich stellt sich natürlich zunächst die Frage, welches Maß an Sicherheit unser Verfahren dem Nutzer gewähren kann. Da unser Verfahren auf Karteninformationen in Form eines Graphen arbeitet, bietet es sich an dafür die Anzahl der Kanten zu zählen, auf welchen sich ein Start- oder Endpunkt aus Sicht eines Angreifers befinden kann.
Sei n die Anzahl der Kanten, auf welchen ein Endpunkt aus Sicht eines Angreifers potentiell liegen könnte, so können wir das Risiko, dass ein Angreifer die richtige Kante errät mit $\frac{1}{n}$ angeben. Diese Metrik erinnert dabei stark an das von Gruteser und Grunwald in [GG03] unter der Bezeichnung Reidentifikationsrisiko angegebene Maß der Anonymität bei der Verwendung von k-Anonymity.

Wenn wir davon ausgehen, dass die Verschleierung zuverlässig funktioniert, so entspricht die Anzahl der Kanten, auf welche ein Angreifer die Möglichen Endpunkte einschränken kann genau der Anzahl der Kanten, welche zu einem verschleierten Gebiet gehören. Für ein verschleiertes Gebiet, welches wir durch einen Graphen $G_V = (V, E)$ identifizieren, können wir folglich das Risiko, dass ein Angreifer den Straßenabschnitt (repräsentiert durch eine Kante), auf welchem sich der Endpunkt tatsächlich befindet mit $\frac{1}{|E|}$ angeben.

5.2.2 Vorteile

Das hier vorgestellte Verfahren bietet einige grundlegend neue Ansätze in der Betrachtung und Verschleierung von Positionsinformationen. Im Gegensatz zu bisherigen Ansätzen betrachtet dieses Verfahren zur Verschleierung der kritischen Information die Sicht eines Angreifers auf das Problem und integriert diese bei der Bewertung der Positionsinformationen. Auf diese Weise wird es möglich zwischen kritischen und unkritischen Informationen zu unterscheiden und mögliche Angriffe vorwegzunehmen.

Zudem werden bei der Berechnung und Verwendung von Gebieten natürliche und von Menschenhand geschaffene Strukturen berücksichtigt. Auf diese Weise werden im Wesentlichen zwei positive Effekte erzielt. Zum einen wird durch diese Art der Betrachtung die Anzahl der betrachteten Wege minimiert, da es möglich ist die Betrachtungen auf Zufahrtswege zu beschränken. Viel wichtiger ist jedoch, dass dadurch die Verschleierung auch auf einem Gebiet stattfindet, welches sich durch eine minimale Anzahl von Zufahrtswegen beschreiben lässt. Das Verfahren zielt darauf ab, dass ein Angreifer nicht ermitteln kann, über welchen der Zufahrtwege das verschleierte Gebiet betreten oder verlassen wird. Auf diese Weise werden mögliche Rückschlüsse über den tatsächlichen Start- oder Endpunkt verhindert.

5.2.3 Nachteile und Unzulänglichkeiten

Die in Kapitel 5.2.2 beschriebenen Vorteile der Ausnutzung örtlicher Gegebenheiten stellt leider auch einen der größten Nachteile des Verfahrens dar. Die Anzahl der Kanten, die zum Graph eines verschleierten Gebiets gehören hängt demnach maßgeblich von den örtlichen Gegebenheiten ab, die somit zu einem beschränkenden Faktor für die Sicherheit werden. Durch alleinige Betrachtung des Verfahrens kann kein $n > 1$ angegeben werden , so dass das Risiko, dass ein Angreifer die korrekte Kante identifiziert, zu welcher ein Start- oder Endpunkt gehört mit $\frac{1}{n}$ nach oben abgeschätzt werden kann.
Die örtlichen Gegebenheiten können jedoch noch viel eher zu einem Sicherheitsrisiko werden, wenn sie im Verfahren nicht berücksichtigt werden. Die in Kapitel 2.1.2 vorgestellten Verfahren zur Verschleierung einzelner Positionsinformationen betrachten diese Gegebenheiten tatsächlich nicht.
Wie wir in Abschnitt 5.1 festgestellt haben, ergibt sich bei den analysierten Daten für etwas mehr als 5% der zu Erschließungsstraßen gehörenden Kanten ein Wohngebiet, welches nur eine Kante umfasst. Des Weiteren werden mit diesem Verfahren auch alle Verbindungsstraßen auf ein Wohngebiet mit genau einer Kante abgebildet. Für diese Fälle kann das Verfahren somit nur ein minimales Maß an Sicherheit garantieren.

Insgesamt können wir anhand der ermittelten Daten zwar feststellen, dass das Verfahren im Mittel durchaus beachtliche Ergebnisse erzielen kann, allerdings ist diese Betrachtung für eine sicherheitskritische Anwendung noch nicht besonders aussagekräftig. Im schlechtesten Fall kann das Verfahren, wie es bisher betrachtet wurde, zu einen Start- oder Endpunkt nur eine einzelne Kante verschleiern. Für einen praktischen Einsatz des Verfahrens muss es ermöglicht werden ein Mindestmaß an Verschleierung zu gewährleisten. Wie dies beispielsweise geschehen könnte, werden wir in Kapitel 6.1 diskutieren.

Das vorgestellte Verfahren basiert zudem auf einigen Annahmen über Kartendaten und Benutzerverhalten, welche für eine realistische Bewertung der Sicherheit dieses Verfahrens durchaus kritisch betrachtet werden müssen.

Zunächst haben wir in Kapitel 3.4 die Annahme getroffen, dass sich Personen auf kürzesten Wegen bewegen. Diese Annahme wurde in Kapitel 4.3.3 als Grundlage für die Bewertung von Positionsinformationen verwendet. Reale Wege weichen aber häufig von den algorithmisch als optimal eingestuften Wegen ab. Hierdurch bietet sich gegebenenfalls die Möglichkeit für einen Angreifer durch die Information, wann ein optimaler Weg verlassen wurde Rückschlüsse über mögliche Endpunkte zu ziehen. Wie wir diesem Risiko entgegen wirken können, werden wir in Kapitel 6.2 diskutieren.

Die Beschreibung des grundlegenden Verfahrens basiert zudem auf der Annahme, dass die Karteninformationen in Form eines ungerichteten Graphen vorliegen. Tatsächlich lassen sich damit aber viele Möglichkeiten der Verkehrsführung nicht abbilden. Einbahnstraßen und Abbiegebeschränkungen führen folglich dazu, dass die in unserem Verfahren angenommenen Wege nicht unbedingt mit der Realität übereinstimmen. Das Verfahren muss also auch auf gerichteten Graphen funktionieren. Wie es dahingehend angepasst werden kann, werden wir in Kapitel 6.3 betrachten.

Daneben muss ebenfalls betrachtet werden, dass es bei der Positionsbestimmung durch verschiedene Einflüsse zu Messungenauigkeiten kommen kann. Die Ermittlung der Position muss nicht immer korrekt sein und Positionsinformationen, welche auf Grund von Messfehlern falsch bewertet wurden können gegebenenfalls auch Rückschlüss auf den Start- oder Endpunkt zulassen. Wie diesem Problem begegnet werden kann, werden wir in Kapitel 6.4 betrachten.

5.2.4 Zusammenfassung

Insgesamt zeigt sich, dass das Verfahren neben einigen wichtigen Fortschritten gegenüber bestehenden Verfahren die Möglichkeit bietet das Angreiferwissen in die Bewertung von Positionsinformationen zu integrieren. Es gibt allerdings noch einige Stellen, an denen es noch zu kurz greift und verbessert werden muss. Wir werden in Kapitel 6 einige Verbesserungsmöglichkeiten betrachten und damit auch sehen, dass sich das Verfahren an vielen Stellen anpassen lässt um weiteres Angreiferwissen, abweichendes Benutzerverhalten und zusätzliche Umgebungsinformationen zu berücksichtigen.
Das Verfahren bietet damit zwei wesentliche Vorteile gegenüber bestehenden Verfahren. Es bestimmt zum einen die Grenze zwischen kritischen und unkritischen Informationen und ermöglicht damit eine optimale Nutzung von nützlichen Informationen bei gleichzeitiger Garantie der Privatsphäre des Nutzers. Zum anderen kann es an weitere Gegebenheiten angepasst werden.

5.3 Performance

Für Performance Betrachtungen muss bei dem vorgestellten zwischen den zwei verschiedenen Phasen der Anwendung unterschieden werden. Zum einen muss die in Kapitel 4.3 vorgestellte Initialisierung zu Beginn einer Fahrt beachtet werden und zum anderen sollte das Positionsupdate, welches in Kapitel 4.4 beschrieben wurde, analysiert werden. Da diese beiden Schritte nicht nur unterschiedliche Anforderungen erfüllen, sondern auch unterschiedlich oft ausgeführt werden, müssen auch die Performance Aspekte entsprechend unterschiedlich betrachtet werden.
Da die Initialisierung nur einmalig zu Fahrtbeginn ausgeführt werden muss, darf dieser Schritt durchaus einige Zeit in Anspruch nehmen. Einige Sekunden Berechnungszeit sind für den beschriebenen Anwendungsfall durchaus vertretbar. Bei einem Positionsupdate sind hingegen nur minimale Verzögerungen hinnehmbar. Diese Updates können bei heutigen Geräten durchaus im Sekundentakt oder häufiger durchgeführt werden, aufwendige Berechnungen können in diesem Zeitrahmen nicht vorgenommen werden.

5.3.1 Initialisierung

Bei der Initialisierung handelt es sich um den berechnungsaufwendigsten Teil des Verfahrens. Gleichzeitig handelt es sich aber auch um einen Teil des Verfahrens, welcher nicht unbedingt vollständig auf dem mobilen Endgerät des Benutzers ausgeführt werden muss. Die Berechnung der zu einem Wohngebiet gehörenden Menge kritischer Kanten könnte bereits vorher berechnet und bei Bedarf nurnoch abgefragt werden.

Als aufwendigsten Schritt der Initialisierung zeigt sich die Berechnung der unkritischen Kanten. Hierzu müssen $|R_V| \cdot |R_U|$ kürzeste Wege berechnet werden. Die Aufwand für die Berechnung der kürzesten Wege selbst hängt stark vom eingesetzten Routingalgorithmus ab. Für die Berechnung der kürzesten Wege besteht hierbei durchaus einiges Optimierungspotential. Da die berechneten Wege oft große Teilabschnitte gemeinsam haben könnte durch die Wiederverwendung von Zwischenergebnissen die Komplexität stark reduziert werden.

5.3.2 Positionsupdate

Bei Einsatz dieses Verschleierungsfahrens muss für jede Positionsinformation geprüft werden, ob diese herausgegeben werden darf. Dies geschieht wie in Alorithmus 4.7 beschrieben. Da diese Berechnung für jede einzelne Positionsupdate durchgeführt werden muss, darf diese nur wenig Zeit in Anspruch nehmen um einerseits keine große Verzögerung beim Positionsupdate zu verursachen und zum anderen nicht zur Beschränkung für die Frequenz der herausgegen Positionsinformationen wird.

Bei genauerer Betrachtung von Algorithmus 4.7 stellen wir fest, dass für die Überprüfung der Positionsinformationen im wesentlichen zwei notwendig sind. In Zeile 3 wird die

Positionsinformation auf eine Kante des Graphen gematcht und in Zeile 4 wird für die dabei ermittelte Kante geprüft, ob sie in der Menge der kritischen Kanten enthalten ist.
Beim Map Matching handelt es sich um einen Berechnungsschritt, welcher bei Einsatz eines Navigationssystems ohnehin notwendig ist. Wir wissen daher zum einen, dass es sich um eine Berechnung handelt, die effizient durchgeführt werden kann und zum anderen, dass wir hier die Ergebnisse des Navigationssystems ohne weiteren Berechnungsaufwand nutzen könnten.
Die Anschließende Prüfung, ob die ermittelte Kante in der Menge der kritischen Kanten enthalten ist, lässt sich durch Einsatz einer Hashtabellen mit konstantem Aufwand in wenigen Berechnungsschritten realisieren.

Für das Positionsupdate können wir damit feststellen, dass die Prüfung, ob Positionsinformationen herausgegeben werden dürfen, mit vertretbarem Berechnungsaufwand realisierbar ist.

6 Mögliche Verbesserungen

6.1 Auswahl des Verschleierungslevels

Die Stärke des Verfahrens hängt, wie bereits in Kapitel 5.2.3 beschrieben von der Umgebung ab, in welcher sich Start- und Endpunkt eines Weges befinden. In der Praxis ist dieser Effekt natürlich nicht wünschenswert, für den Benutzer sollte ein einstellbares Mindestmaß an Sicherheit gewährleistet werden können. In unserem Verfahren wäre dies die Anzahl der Kanten, auf welchen Start- und Endpunkte potentiell liegen können.
Zur Lösung dieses Problems können verschiedene Ideen in Betracht gezogen werden. Zum einen wäre es möglich das Verfahren nicht nur ausgehend von Wohngebieten, sondern von größeren Gebietseinheiten anzuwenden. Zum anderen besteht durchaus die Möglichkeit mehrere Wohngebiete zu einem größeren Gebiet zusammenzufassen und als dieses Gebiet zu verschleiern. Vor- und Nachteile sowie Grenzen dieser Möglichkeiten werden wir im Folgenden betrachten.

6.1.1 Betrachtung größerer Gebiete

In den bisherigen Betrachtungen sind wir immer davon ausgegangen, dass das zu einem Start- oder Endpunkt gehörende Wohngebiet verschleiert wird. Unser Verfahren kann aber grundsätzlich auch auf andere Gebietsgrößen angewendet werden. Eine alternative Möglichkeit besteht darin, dass die zugehörige Stadt bzw. der zugehörige Ort verschleiert wird. Hierzu würden wir den Untergraph, welcher das Stadtgebiet beschreibt als G_V wählen, den Untergraph, welcher den Landkreis beschreibt würden wir als G_U wählen und mit diesen beiden Graphen die weiteren Schritte des Verfahrens durchführen. Auf diese Weise lässt sich unser Konzept auf größere Gebiete übertragen.

Bei einer Anwendung unseres Verfahrens auf größere Gebiete ist dabei zu beachten, dass die für den sicheren Einsatz des Verfahrens notwendigen Rahmenbedingungen nicht verletzt werden. Hier muss auf jeden Fall darauf geachtet werden, dass die bezüglich des verschleierten Gebiets kritischen Positionsinformationen tatsächlich innerhalb des umgebenden Gebiets angenommen werden können. Des Weiteren wird es bei der Betrachtung größerer Gebiete auch zunehmend wahrscheinlicher, dass die in Kapitel 3.5.1 angenommenen zeitlichen Rahmenbedingungen für aufeinanderfolgende Wege verletzt werden.

6.1.2 Zusammenfassung mehrerer Gebietseinheiten

Eine andere Möglichkeit eine Skalierung des verschleierten Gebiets zu ermöglichen, besteht in der Zusammenfassung von Gebietseinheiten. Hierzu könnten benachbarte Wohngebiete zu einem Gebiet zusammengefasst werden. Die anschließenden Berechungen könnten auf dieselbe Weise wie bisher erfolgen. Hierbei müssten eindeutige Regeln zur Zusammenfassung der jeweiligen Gebiete festgelegt werden.

6.1.3 Allgemeine Betrachtungen

Unabhängig von der gewählten Methode zur Übertragung des Ansatzes auf größere Gebiete muss darauf geachtet werden, dass die getroffenen Annahmen noch gelten und die gestellten Anforderungen noch erfüllt werden können. Insbesondere sollte hierbei darauf geachtet werden, dass die in Kapitel 3.6 beschrieben Anforderunge zur Wahl des Gebietes nicht verletzt werden. Die Wahl des verschleierten Gebietes darf dabei keine darüber hinausgehenden Rückschlüsse über den verschleierten Start- oder Endpunkt zulassen.

Darüber hinaus wächst mit zunehmender Größe des verschleierten Gebiets die Gefahr, dass zeitliche Betrachtungen eine größere Rolle spielen und möglicherweise ungewünschte Rückschlüsse über Start- und Endpukte ermöglichen. Insbesondere sollte hierbei überprüft werden, ob die in Kapitel 3.5.1 formulierten zeitlichen Rahmenbedingungen für die betrachtete Gebietsgröße und die betrachteten Wege noch zutreffen.

6.2 Berücksichtigung nicht-optimaler Wege

Für die Verschleierung der Start- und Endpunkte sind wir bisher davon ausgegangen, dass sich Personen auf kürzesten Wegen bewegen. In der Realität weichen die tatsächlich gewählten Wege häufig von den berechneten optimalen Wegen ab. Wie Letchner in [LKH06] zeigt, sind für die Wahl einer Route auch oft persönliche Präferenzen und Gewohnheiten ausschlaggebend. In dieser Studie wird gezeigt, dass ein durchschnittlicher Fahrer nur in 35% der Fälle den schnellsten Weg wählt. Welcher Weg tatsächlich bevorzugt wird, kann zudem auch von Tageszeit und Wochentag abhängen, wie es in [LKH06] ebenfalls festgestellt wird.

Eine naheliegende Möglichkeit diese Eigenschaft der Routenwahl in unserem Algorithmus zu berücksichtigen bestünde darin die Algorithmen zur Routenwahl entsprechend zu personalisieren und die Ansätze zur individualisierten, zeitabhängigen Routenwahl auch für unseren Algorithmus zu nutzen. Allerdings verkleinern wir damit die Probleme nur, können sie allerdings nicht vollständig beseitigen, da diese Algorithmen zum einen trainiert werden müssen, wir aber von Anfang an keine kritischen Informationen preisgeben dürfen und zum anderen auch diese Vorhersagen noch von der tatsächlich gewählten Strecke abweichen

können.

Wir müssen folglich alle in Frage kommenden Strecken abdecken. Eine geeignete Möglichkeit besteht darin für die Strecke zwischen zwei Punkten nicht nur den schnellsten Weg betrachten, sondern eine Menge von plausiblen Wegen. Für eine dahingehende Verbesserung unseres Verfahrens muss die Funktion zur Berechnung der kritischen Kanten (Algorithmus 4.4) entsprechend angepasst werden. Dies kann wie in Algorithmus 6.1 beschrieben geschehen.

Algorithmus 6.1 Verbesserte Berechnung unkritischer Kanten

```
1:  function calcUncriticalEdges(VertexSet: R_U, VertexSet: R_V, Graph: G_U, Graph: G_V)
2:      EdgeSet: critical, uncritical
3:      EdgeSet: localCritical, localUncritical
4:      Set of EdgeSets: paths
5:      critical ← ∅
6:      uncritical ← ∅
7:      for out ∈ R_U do
8:          localUsed ← ∅
9:          localUncritical ← G_U.E
10:         for in ∈ R_V do
11:             paths ← getPlausiblePaths(out, in)
12:             for paht ∈ paths do
13:                 localUsed ← localUsed ∪ path
14:                 localUncritical ← localUncritical ∩ path
15:             end for
16:         end for
17:         used ← used ∪ localUsed
18:         uncritical ← uncritical ∪ localUncritical
19:         critical ← critical ∪ (localUsed \ localUncritical)
20:     end for
21:     uncritical ← uncritical \ critical
22:     return uncritical
23: end function
```

Die wesentlichen Verbesserung in Algorithmus 6.1 befinden sich zwischen Zeile 10 und 14. Statt eines einzelnen Weges wird eine Menge von plausiblen Wegen betrachtet. Kanten werden nur noch dann als unkritisch eingestuft, wenn sie zu all diesen plausiblen Wegen gerechnet werden.

Da die realen Wege zwar nicht dem schnellsten Weg entsprechen, aber dennoch durch den Nutzer als optimal eingestuft werden, können wir davon ausgehen, dass sie nur bis zu einem gewissen Maß vom schnellsten Weg abweichen. Wie stark diese Abweichung sein kann, müsste noch genauer ermittelt werden. Nehmen wir an, dass ein Nutzer nur Wege wählt, für

welche er maximal das 1,5-fache der Zeit braucht, die er für den schnellsten Weg benötigen würde, so können wir die Menge der plausiblen Wege dadurch ermitteln, dass wir alle Wege berechnen, für welche maximal diese Zeit benötigt würde. Weitere Einschränkungen dieser Wege wären für eine Verbesserung dieses Verfahrens durchaus denkbar.

6.3 Berücksichtigung von Einbahnstraßen und Abbiegebeschränkungen

In den bisherigen Betrachtungen sind wir davon ausgegangen, dass uns die Karteninformationen in Form eines ungerichteten Graphen vorliegen. Die Verkehrsführung in der Realität lässt sich damit leider noch nicht vollständig abbilden. Während es die Betrachtungen erleichtert nur Straßen zu betrachten, welche in beide Richtungen befahrbar sind, gibt es in der Realität auch viele Einbahnstraßen. Darüber hinaus gibt es auch an vielen Kreuzungen Abbiegebeschränkungen, welche ebenfalls nicht auf einen ungerichteten Graphen abgebildet werden können.

Um diese realen Verkehrssituationen berücksichtigen zu können, müssen wir folglich unser Verfahren auf einen gerichteten Graphen erweitern. Diese Erweiterung kann bereits durch geringfügige Änderungen vorgenommen werden. Im wesentlichen müssen wir hierzu bei der Ermittlung kritischer Kanten zwischen Start- und Endpunkten unterscheiden, da die Richtung des kürzesten Weges zwischen zwei Punkten nun eine Rolle spielt.
Algorithmus 4.4 wäre damit nur für die bezüglich eines Endpunktes kritischen Kanten geeignet, da in Zeile 10 der kürzeste Pfad zu diesem Gebiet hin berechnet wird. Für einen Startpunkt müssten folglich die beiden Argumente der Funktion *getShortestPath* vertauscht werden. Alle anderen Schritte des Verfahrens bleiben identisch.

6.4 Berücksichtigung von Ungenauigkeiten bei der Positionsbestimmung

Bei einem Positionsupdate sind wir davon ausgegangen, dass ein Map Matching Algorithmus uns für die aktuelle Position die exakte Straße liefert, repräsentiert durch eine Kante im Graphen, zu welcher diese Position zugeordnet wurde. Oft kann die Positionsbestimmung jedoch nicht exakt vorgenommen werden. Diese Ungenauigkeiten können durch technische Beschränkungen, temporäre Fehler oder einen schlechten Empfang von Satelittensignalen, beispielsweise in einem Tunnel, hervorgerufen werden. In diesen Fällen kann durch Map Matching Algorithmen zwar eine relativ gute Fehlerkorrektur vorgenommen werden, allerdings ist diese immer noch mit einer nicht vernachlässigbaren Fehlerwahrscheinlichkeit verbunden.

Da wir Positionsinformationen nur dann herausgeben möchten, wenn wir diese zweifelsfrei als unkritisch einstufen können, sollte unser Verfahren auch diese Ungenauigkeiten berücksichtigen können. Hierzu könnte statt der bei Map Matching bestimmten Kante, welche mit der höchsten Wahrscheinlichkeit die Straße angibt, auf welcher sich der Nutzer aktuell befindet, alle Kanten bestimmt werden, auf welchen sich der Nutzer aktuell aufhalten könnte. Wir nutzen für unser Verfahren folglich nicht mehr das Endergebnis des Map Matchings, sondern greifen auf ein Zwischenergebnis das Map Matchings zurück und verwenden die in Frage kommenden Kanten.

Die Anpassungen unseres Verfahrens müssten natürlich beim Positionsupdate vorgenommen werden. Die in Algorithmus 4.7 beschriebene Funktion *positionUpdate* müsste statt nur einer gematchten Kante nun eine Menge von möglichen Kanten berücksichtigen. Wie dies aussehen kann, wird in Algorithmus 6.2 beschrieben, hierbei bestimmt die Funktion *mapMatchCandidates* die Menge möglicher Kanten, welchen die aktuelle Positionsinformation zugeordnet werden könnte.

Algorithmus 6.2 Positionsupdate mit Berücksichtigung ungenauer Positionsbestimmung

```
1:  EdgeSet: E_K_start, E_K_end                    // calculated during initialization
2:  function POSITIONUPDATE(Float latitude, Float longitude)
3:      EdgeSet: possibleEdges
4:      possibleEdges ← MAPMATCHCANDIDATES(latitude, longitude)
5:      if (possibleEdges ∩ (E_K_start ∪ E_K_end)) = ∅ then
6:          SENDPOSITIONUPDATE(latitude, longitude)
7:          return TRUE
8:      else
9:          return FALSE
10:     end if
11: end function
```

7 Offene Probleme

7.1 Schutz bei vorheriger Nutzung eines Dienstes

Wurden an einen Anbieter vor Einsatz des Verschleierungsverfahrens bereits Positionsdaten übermittelt, so ist ein Schutz der Privatsphäre nur sehr eingeschränkt möglich. Wurden Trajektorien übermittelt, welche vom eigenen Wohnsitz weg oder zum eigenen Wohnsitz hin führten, so wird dieser auch für die verschleierten Trajektorien erkennbar sein, sofern die zuvor ungeschützt herausgegebenen Positionsdaten damit in Verbindung gebracht werden können.

Es ist daher wichtig zu garantieren, dass die bei Nutzung dieses Verfahrens übermittelten Positionsinformationen nicht mit anderen Bewegungsprofilen verknüpft werden können. Eine notwendige Maßnahme wäre beispielsweise die Nutzung eines zuvor ungenutzten Accounts oder Pseudonyms. Je nach Anwendungsfall können aber auch weitere Maßnahmen notwendig sein.

7.2 Zusammenführung von Nutzerdaten

Neben der Verschleierung der eigenen Positionsinformationen muss aus Nutzersicht auch berücksichtigt werden, ob eventuelle Begleitpersonen ebenfalls eine geeignete Verschleierung einsetzen. Erzeugen beispielsweise zwei Nutzer nahezu identische Trajektorien, so ist es naheliegend, dass Start- und Endpunkt für beide Trajektorien identisch sind. Nutzt nur einer der beiden Nutzer eine Verschleierung, so kann diese durch Verknüpfung dieser beiden Trajektorien gebrochen werden.

Die Herausforderung, welche sich durch diese Möglichkeit der Informationsgewinnung ergibt, ist allerdings eher gesellschaftlicher als technischer Natur. Auf technischer Ebene wird es leider kaum möglich sein dieses Problem zu lösen.

7.3 Logisches Schließen aus Umgebungsinformationen

Die Betrachtungen, welche wir für die Verschleierung durchgeführt haben, finden im Wesentlichen auf der Ebene von Graphen statt, welche wiederum Straßeninformationen

repräsentieren. Hierbei wird der Fokus vor allem darauf gerichtet, wie die Straßen untereinander verbunden sind und welche Funktion diese einnehmen. Es wir dabei allerdings vernachlässigt, wie die Straßen bebaut sind, wie die Besiedelungsdichte im verschleierten Gebiet aussieht und wie die Gebäude, in welchen sich meist die eigentlichen Ziele eines Weges befinden, genutzt werden.

Tatsächlich können Informationen über die Besiedelung und die Nutzung der innerhalb eines Gebiets befindlichen Gebäude durchaus interessante Rückschlüsse über mögliche Start- und Endpunkte betrachteter Wege liefern. Diese Rückschlüsse können zum einen durch alleinige Betrachtung des Gebietes gewonnen werden (statische Betrachtung) oder durch den Zeitpunkt, zu welchem Wege mit einem Start- oder Endpunkt im jeweiligen Gebiet getätigt wurden noch verbessert werden (zeitliche Betrachtung). Welche Informationen man daraus gewinnen kann, werden wir im Folgenden betrachten.

7.3.1 Statische Betrachtung

Für unser Verschleierungsverfahren werden die verschleierten Gebiete über die zugehörigen Straßen definiert. Wir haben somit durch die Anzahl der Straßen, welche als mögliche Ziele in Frage kommen, ein Maß dafür, wie stark der Schutz ist, welchen unser Verfahren bietet. Tatsächlich ist eine Straße aber nicht das eigentlich Ziel eines Weges, viel eher ist das Ziel eines der anliegenden Gebäude. Nun hängt die Anzahl möglicher Ziele natürlich von der Anzahl der Gebäude ab, welche zu einer betrachteten Straße gehören. Stehen an einer Straße beispielsweise 50 Häuser, so gibt es mehr potentielle Ziele, als es bei einer Straße mit zwei Häusern der Fall wäre.

Des Weiteren hängt die Wahrscheinlichkeitsverteilung für potentielle Endpunkte natürlich auch von der Art der Bebauung ab. Stehen an einer Straße beispielsweise zehn Einfamilienhäuser und zusätzlich ein großer Wohnkomplex mit zwanzig Wohnungen, so ist die Wahrscheinlichkeit, dass dieser Wohnkomplex der Endpunkt einer betrachteten Trajektorie ist a priori deutlich höher als es die Wahrscheinlichkeit für eines der Einfamilienhäuser ist.

Grundsätzlich stellt sich hierbei natürlich die Frage, was als Ziel eines Weges betrachtet werden soll und was tatsächlich durch Positionsinformationen beschrieben wird. Durch Positionsinformationen in Form von Geo-Koordinaten lässt sich sicherlich das Gebäude, nicht aber die Wohnung oder das Büro identifizieren, welches das eigentliche Ziel darstellt. Unter dieser Betrachtungsweise genießen Bewohner eines Wohnkomplexes bereits ein gewisses Maß an Anonymität. Die auf Gebäude bezogene Wahrscheinlichkeitsverteilung wäre zudem irrelevant, wenn man Wohn- oder Büroeinheiten als potentielle Endpunkte auffasst. Es wäre daher eher denkbar, dass man die zu einer Straße gehörenden Wohn- und Büroeinheiten zählt und diese Anzahl potentieller Endpunkte für die gewünschte Verschleierung berücksichtigt. Wie genau sich diese Eigenschaften einer realen Umgebung in unserem

Modell berücksichtigen lassen könnnen, kann allerdings im Rahmen dieser Arbeit nicht mehr diskutiert werden.

7.3.2 Zeitliche Betrachtung

Kommen zusätzlich zu den örtlichen Gegebenheiten auch noch zeitliche Betrachtungen ins Spiel, so kann es je nach Umgebung möglich sein weitere Rückschlüsse über möglich Start- und Endpunkte zu gewinnen. Einige potentielle Ziele eines Weges sind nicht zeitlich unbeschränkt zugänglich, haben Öffnungszeiten oder dienen nur als Veranstaltungsort für zeitlich begrenzte Veranstaltungen.

Betrachten wir beispielsweise ein Gewerbegebiet, in welchem sich neben einigen Supermärkten, Autohäusern und Gewerbebetrieben auch eine Diskothek befindet. Führt nun ein Weg Samstag abends nach 22 Uhr in dieses Gewerbegebiet, so ist die Wahrscheinlichkeit sehr hoch, dass der Endpunkt des Weges die Diskothek ist. Umgekehrt kann man mit einiger Sicherheit davon ausgehen, dass ein Weg, welcher vormittags einem Werktag dort hin führt einen anderen Endpunkt als diese Diskothek hat. Abhängig von den Öffnungszeiten ansässiger Betriebe könnte wären je nach Tageszeit noch weitere Einschränkungen denkbar.

Nun mag die Information, dass eine Diskothek besucht wurde keine besondere Brisanz mit sich bringen, ganz anders sieht es jedoch aus, wenn es sich bei dem fraglichen Gebäude nicht um eine Diskothek, sondern um ein Freudenhaus handelt.

Es sollte hierbei bedacht werden, dass diese Rückschlüsse auf Grund von zeitlichen Betrachtungen stattfinden und unabhängig vom eingesetzten Verschleierungsverfahren sind. Auch das für diese Rückschlüsse notwendige Angreiferwissen ist mittlerweile durchaus einfach zu erlangen. Verschiedene Kartendienste zeigen in ihren Karten sehr detailierte Informationen über ansässige Gewerbebetriebe, Einkaufsmöglichkeiten und Freizeitangebote. Wie detailiert die angebotenen Informationen bereits bei kostenlosen Diensten sind, zeigt Abbildung 7.1.

7.4 Logisches Schließen aus Verhaltensmustern

Neben den reinen Positionsinformationen muss vor allem bei wiederkehrenden Strecken beachtet werden, dass bereits der Zeitpunkt zu welchem diese Wege zurückgelegt werden mögliche Rückschlüsse über das Ziel ermöglichen. Sobald zeitlich begrenzt stattfindende Veranstaltungen oder wiederkehrende Verhaltensmuster ins Spiel kommen, werden die Ziele eines Weges oft schon allein durch den Zeitpunkt, an welchem er zurückgelegt wurde offensichtlich.

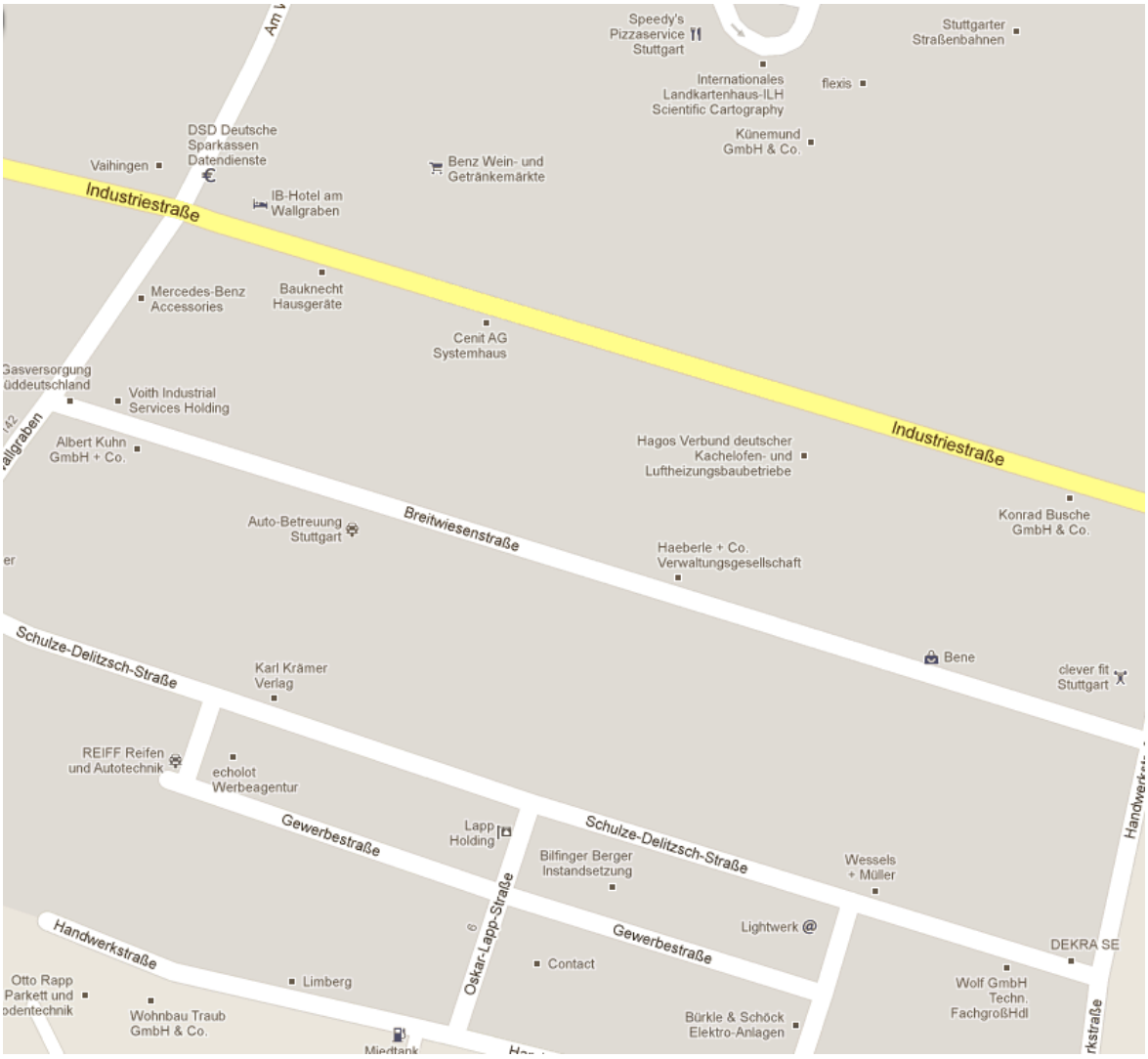

Abbildung 7.1: Darstellung eines Gewerbegebiets mit ansässigen Firmen, Händlern und Gastronomiebetrieben - Quelle: GoogleMaps [Goo]

Fährt beispielsweise eine Person am jeden Sonntag morgen zur selben Uhrzeit in die nächste Stadt und tritt etwa eine Stunde später den Rückweg an, so ist es naheliegend, dass sie zur Kirche fährt. Eine Verschleierung mag hier zwar den Wohnsitz der Person verschleiern, das genaue Ziel jedoch ergibt sich allein schon durch den Zeitpunkt und das ungefähre Ziel.

Begibt sich eine Person unregelmäßig samstags in die nächste Großstadt und lassen sich die Zeitpunkte mit den Spielen des dort ansässigen Fußballvereins in Einklang bringen, so liegt es nahe, dass es sich bei der Person um einen Anhänger dieses Vereins handelt und das Stadion das Ziel der Fahrt ist.

Betrachtet man die täglichen Fahrten, so sind mit großer Wahrscheinlichkeit die beiden dabei am häufigsten vorkommenden Start- und Endpunkte der Arbeitsplatz und der eigene Wohnsitz. In den meisten Fällen führt morgens der Weg zur Arbeit und abends wieder nach Hause. Führt der allmorgendliche Weg aber in einen Ort, in dem es nur einen einzigen potentiellen Arbeitgeber gibt, so kann die Verschleierung durch die einfache Beobachtung von menschlichen Verhaltensmustern gebrochen werden.

Leider lassen sich die Möglichkeiten, die das Wissen über die Eigenschaften von menschlichen Verhaltensmustern auszunutzen, kaum eingrenzen. Man könnte die Liste der Beispiele beinahe beliebig fortsetzen und jedes dieser Beispiele zeigt einen Fall in dem kein Verfahren Schutz bieten kann, sofern nicht auch diese Verhaltensmuster mit verschleiert werden.

8 Zusammenfassung / Fazit

In dieser Arbeit wurde ein Verfahren vorgestellt, welches die Verschleierung der Start- und Endpunkten von Trajektorien ermöglicht. Dabei wurden gegenüber den bestehenden Verfahren einige grundlegende Fortschritte erzielt.

Zum einen wurde damit der Übergang von der Betrachtung einzelner Punkte zur Betrachtung ganzer Trajektorien, mit dem speziellen Blick auf deren Start- und Endpunkte gewagt. Zum anderen wurde aber auch ein Verfahren entwickelt, welches nicht nur für die Betrachtung einer einzelne Trajektorie, sondern auch für den Einsatz bei einer große Menge von Trajektorien geeignet ist ist.
Darüber hinaus werden für diese Verfahren auch Karteninformationen genutzt. Während viele Verschleierungsverfahren für einzelne Positionsinformationen nur im Freespace funktionieren, stellt die Kenntnis von Karteninformationen bei Einsatz unseres Verfahrens keine Bedrohung dar.
Insgesamt wurden damit sowohl bei den betrachteten Datenmengen und Anwendungsfällen, als auch bei Berücksichtigung des Angreiferwissens große Fortschritte erzielt.

Des Weiteren wurde in dieser Arbeit die Sicht eines möglichen Angreifers nicht nur bei der Bewertung des Verfahrens berücksichtigt, sondern direkt in die Verschleierung integriert. Es wurde hierbei zunächst davon ausgegangen, dass der Angreifer durch Betrachtung kürzester Wege Rückschlüsse über mögliche Start- und Endpunkte ziehen kann, aber auch weitere Angriffstechniken könnten prinzipiell in das Verfahren integriert werden.

Trotz dieser großen Fortschritte muss aber auch festgestellt werden, dass für das vorgestellte Verfahren noch an einigen Stellen Optimierungsbedarf besteht. Zum einen kann die Laufzeit der Initialisierung noch deutlich verbessert werden, zum anderen sollte aber auch die Möglichkeit geschaffen werden das Verfahren für die persönlichen Sicherheitsanforderungen konfigurierbar zu machen.

Betrachtet man geographische Gegebenheiten, so zeigt sich aber auch, dass die Möglichkeiten der Verschleierung oft von der Umgebung einer zu verschleiernden Position abhängig sind. Zudem kann in vielen Fällen auch der mit Positionsinformationen assoziierte Zeitpunkt für weitere Schlussfolgerungen genutzt werden.
Wie in Kapitel 7 beschrieben ergeben sich dadurch einige Probleme, für die es bisher noch keine Lösungsansätze gibt.

Bei näherer Betrachtung dieser Probleme muss möglicherweise auch erneut die Frage gestellt werden, welche Informationen überhaupt geschützt werden sollen und können. In den vielen Fällen geht es ja weniger darum die genaue Position eines Nutzers zu verbergen, sondern vielmehr die Schlüsse zu verhindern, die daraus gezogen werden können. Besonders langfristige Betrachtungen können für einen Angreifer jedoch ganz neue Möglichkeiten zur Informationsgewinnung bieten.

Möglicherweise sollte für einen geeigneten Schutz der Privatsphäre auch ein ganz anderes Vorgehen etabliert werden. Möglicherweise sollte an dieser Stelle der Grundsatz der Datensparsamkeit zur Anwendung kommen und zunächst die Frage gestellt werden, welche Daten für den Anwendungszweck überhaupt erhoben werden müssen. Hierdurch könnte für viele Zwecke die Menge der herausgegebenen Informationen deutlich reduziert werden. Kombiniert man anschließend noch Datensparsamkeit mit einer Verschleierung herausgegebener Daten, so könnte sicherlich ein deutlich besserer Schutz der Privatsphäre möglich werden.

Literaturverzeichnis

[ACD+07] C. A. Ardagna, M. Cremonini, E. Damiani, S. De di Vimercati, P. Samarati. Location Privacy Protection Through Obfuscation-based Techniques. In *Proceedings of the 21st annual IFIP WG 11.3 working conference on Data and applications security*. 2007. (Zitiert auf Seite 11)

[BS03] A. R. Beresford, F. Stajano. Location privacy in pervasive computing. *IEEE_M_PVC*, 2(1):46–55, 2003. (Zitiert auf den Seiten 9, 10, 12 und 13)

[BWJ05] C. Bettini, X. Wang, S. Jajodia. Protecting Privacy Against Location-Based Personal Identification. In W. Jonker, M. Petkovic, editors, *Secure Data Management*, volume 3674 of *Lecture Notes in Computer Science*, pp. 185–199. Springer Berlin / Heidelberg, 2005. (Zitiert auf Seite 10)

[DK06] M. Duckham, L. Kulik. Location privacy and location-aware computing, 2006. (Zitiert auf Seite 9)

[DKB06] M. Duckham, L. Kulik, A. Birtley. A Spatiotemporal Model of Strategies and Counter Strategies for Location Privacy Protection. In *GIScience*, pp. 47–64. 2006. (Zitiert auf Seite 9)

[Dü11] Dürr, F. and Skvortsov, P. and Rothermel, K. Position sharing for location privacy in non-trusted systems. In *Proc. IEEE Int Pervasive Computing and Communications (PerCom) Conf*, pp. 189–196. 2011. (Zitiert auf den Seiten 11, 14 und 17)

[For08] Forschungsgesellschaft für Straßen- und Verkehrswesen e. V. Richtlinien für integrierte Netzgestaltung, 2008. (Zitiert auf Seite 19)

[Geo] Geofabrik. OpenStreetMap Kartendaten für Berlin. URL http://download.geofabrik.de/osm/europe/germany/berlin.osm.bz2. (Zitiert auf Seite 39)

[GG03] M. Gruteser, D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys '03, pp. 31–42. ACM, New York, NY, USA, 2003. (Zitiert auf den Seiten 10, 13 und 41)

[GH05] M. Gruteser, B. Hoh. On the Anonymity of Periodic Location Samples. In *In Proceedings of the Second International Conference on Security in Pervasive Computing*, pp. 179–192. Springer, 2005. (Zitiert auf Seite 12)

[Goo] GoogleMaps. URL http://maps.google.de. (Zitiert auf den Seiten 21 und 56)

[Gut06] A. Gutscher. Coordinate transformation - a solution for the privacy problem of location based services? In *Proc. 20th Int. Parallel and Distributed Processing Symp. IPDPS 2006*. 2006. (Zitiert auf Seite 11)

[HGXA06] B. Hoh, M. Gruteser, H. Xiong, A. Alrabady. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE_M_PVC*, 5(4):38–46, 2006. (Zitiert auf Seite 12)

[KH06] J. Krumm, E. Horvitz. Predestination: Inferring Destinations from Partial Trajectories. In P. Dourish, A. Friday, editors, *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pp. 243–260. Springer Berlin / Heidelberg, 2006. (Zitiert auf Seite 12)

[Kru07] J. Krumm. Inference Attacks on Location Tracks. In A. LaMarca, M. Langheinrich, K. Truong, editors, *Pervasive Computing*, volume 4480 of *Lecture Notes in Computer Science*, pp. 127–143. Springer Berlin / Heidelberg, 2007. (Zitiert auf den Seiten 12 und 13)

[Kru09] J. Krumm. Realistic Driving Trips For Location Privacy. In H. Tokuda, M. Beigl, A. Friday, A. Brush, Y. Tobe, editors, *Pervasive Computing*, volume 5538 of *Lecture Notes in Computer Science*, pp. 25–41. Springer Berlin / Heidelberg, 2009. (Zitiert auf den Seiten 11 und 20)

[KYS05] H. Kido, Y. Yanagisawa, T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proc. Int. Conf. Pervasive Services ICPS '05*, pp. 88–97. 2005. (Zitiert auf Seite 11)

[LKH06] J. Letchner, J. Krumm, E. Horvitz. Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning. In *in Eighteenth Conference on Innovative Applications of Artificial Intelligence*. The AAAI Press, 2006. (Zitiert auf den Seiten 20 und 48)

[MDKG05] G. F. Marias, C. Delakouridis, L. Kazatzopoulos, P. Georgiadis. Location privacy through secret sharing techniques. In *Proc. Sixth IEEE Int. Symp. a World of Wireless Mobile and Multimedia Networks WoWMoM 2005*, pp. 614–620. 2005. (Zitiert auf Seite 11)

[MKGV07] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam. *l*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007. (Zitiert auf Seite 10)

[NK09] P. Newson, J. Krumm. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pp. 336–343. ACM, New York, NY, USA, 2009. (Zitiert auf Seite 14)

[Ope] OpenStreetMap. URL http://www.openstreetmap.org. (Zitiert auf Seite 20)

[PS09] L. I. Pravin Shankar, Vinod Ganapathy. Privately Querying Location-based Services with SybilQuery. In *UbiComp'09: Proceedings of the 11th International Conference on Ubiquitous Computing*, pp. 31–40. 2009. doi:10.1145/1620545.1620550. (Zitiert auf Seite 11)

[Sch11] B. Schembera. *Platzierungsoptimierung für vertrauliche Verwaltung der verteilten Positionsinformationen*. Master's thesis, Universität Stuttgart, Holzgartenstr. 16, 70174 Stuttgart, 2011. (Zitiert auf den Seiten 17 und 18)

[Sha79] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, 1979. (Zitiert auf Seite 11)

[Swe02] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002. (Zitiert auf Seite 10)

[TM08] M. Terrovitis, N. Mamoulis. Privacy Preservation in the Publication of Trajectories. In *Proc. 9th Int. Conf. Mobile Data Management MDM '08*, pp. 65–72. 2008. (Zitiert auf Seite 10)

[Wes67] A. F. Westin. *Privacy and freedom*. Atheneum, New York, 1967. (Zitiert auf Seite 9)

[YPL07] T.-H. You, W.-C. Peng, W.-C. Lee. Protecting Moving Trajectories with Dummies. In *Proc. Int Mobile Data Management Conf*, pp. 278–282. 2007. (Zitiert auf Seite 11)

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Franz G. Jahn)