Schutz der Privatsphäre in Ad-hoc-Fahrzeugnetzen

Dissertation

zur Erlangung des akademischen Grades
Dr. rer. nat.
an der Fakultät für Mathematik, Informatik und Naturwissenschaften der Universität Hamburg

eingereicht beim Fach-Promotionsausschuss Informatik von

Florian Scheuer
aus München

Januar 2013

Gutachter
Prof. Dr.-Ing. Hannes Federrath
Prof. Dr. Günther Pernul

Angenommen am
14. Dezember 2012

Danksagung

Mein erster Dank gilt meinem Erstgutachter Prof. Dr.-Ing. Hannes Federrath, der für alle Fragen stets ein offenes Ohr hatte und und den Entstehungsprozess dieser Arbeit mit Ideen, Hinweisen und Anregungen begleitet hat.

Herzlichen Dank auch Prof. Dr. Günther Pernul, meinem Zweitgutachter, der mir viele gute Ratschläge für meine Forschungsarbeit gegeben hat.

Weiterhin möchte ich mich bei meinen Kollegen bedanken, die mich oft dabei unterstützt haben, Ideen zu entwickeln und mit ihren kritischen Fragen dazu beigetragen haben, Schwachstellen in Gedankengängen aufzudecken. Ein besonderer Dank geht dabei an Karl-Peter Fuchs, der sich darauf eingelassen hat, die Ideen zur *ProMix-Zone* mit mir zusammen zu evaluieren und publizieren.

Dank auch an alle Studenten, die ihre Seminar- oder Qualifikationsarbeiten in meinem Themenfeld absolviert haben. Ihre Ideen und Gedanken haben immer wieder Denkanstöße gegeben und somit neue Impulse für das Entstehen dieser Arbeit gesetzt. Hervorheben möchte ich dabei Bernhard Gruber und Andreas Tomandl, deren Arbeiten und Einsatz dazu beigetragen haben, dass der *VANET-Simulator* umgesetzt werden konnte.

Ich möchte mich auch bei meiner Familie bedanken, die mir das Studium ermöglicht und mich auch während der Arbeit an der Dissertation stets in jeder Hinsicht unterstützt hat.

Mein ganz besonderer Dank gilt meiner lieben Frau Nadja, die mich jederzeit ermutigt und unterstützt hat. Ohne die Kraft und die Freude, die sie mir jeden Tag schenkt, wäre diese Arbeit kaum zu schaffen gewesen. Zusammen haben wir gelernt, die wichtigen von den unwichtigen Dingen zu unterscheiden. Darüber hinaus hat ihre unermüdliche Ausdauer bei der Suche nach Tippfehlern und ungünstigen Formulierungen mit Sicherheit zur Verständlichkeit dieser Arbeit beigetragen.

Oktober 2012 Florian Scheuer

Zusammenfassung

Fahrzeugnetze können eine Vielzahl unterschiedlicher Dienste ermöglichen, die dazu beitragen, Unfälle zu verhindern oder ihre Folgen abzumildern, Reisen oder Flottenmanagement effizienter zu gestalten und den Insassen eines Fahrzeugs Zugang zum Internet bereitzustellen. In den kommenden Jahren sollen die ersten Fahrzeuge mit einer entsprechenden Ausstattung in die Serienfertigung gehen, was den Beginn der Vernetzung von Fahrzeugen untereinander markiert.

Allerdings ist ein Schutz der Kommunikationswege dringend erforderlich, damit ein Missbrauch des Netzes unterbunden werden kann oder zumindest erkennbar ist, um den Störer sanktionieren zu können. Die zu diesem Zweck eingesetzten Schutzmaßnahmen führen in der Regel zu einer Identifizierbarkeit der Nutzer und einer Verkettbarkeit ihrer Handlungen. Dies hat zur Folge, dass die Privatsphäre der Teilnehmer eines Fahrzeugnetzes empfindlich beeinträchtigt wird, da jede ihrer Bewegungen aufgezeichnet und ausgewertet werden kann.

Aus diesem Spannungsfeld ergeben sich zwei Teilprobleme, die in dieser Arbeit identifiziert werden: die Sicherstellung einer Erreichbarkeit von Teilnehmern über Positionsdienste ohne ihre Privatsphäre einzuschränken und der Schutz vor Beobachtung und Verfolgung der Bewegungen von Nutzern des Netzes. Um eine datenschutzfreundliche Nutzung von Fahrzeugnetzen zu gewährleisten, gilt es, beide Problemstellungen zu lösen. Bereits existente Lösungsansätze aus der Literatur werden analysiert und auf ihre Eignung hin untersucht.

Es zeigt sich, dass alle Konzepte, die einen datenschutzfreundlichen Positionsdienst vorschlagen, keinerlei Schutz vor dem Betreiber des Dienstes bieten können. Aus diesem Grund wird ein eigener Ansatz vorgestellt, der dieses Problem mit Hilfe von über Chaum'sche Mixe transportierten verschlüsselten Nachrichten und einer hierarchischen Struktur beseitigen kann. Eine Evaluation des Konzepts zeigt, dass ein Einsatz in der Praxis mit dem aktuellen Stand der Technik möglich ist.

Die Ansätze zur Verhinderung der Verfolgbarkeit von Teilnehmern in Ad-hoc-Fahrzeugnetzen bewegen sich in einem Spannungsfeld zwischen dem Schutz der Privatsphäre durch Pseudonymwechsel unter Funkstille auf der einen und der Aufrecht-

erhaltung der Funktionalität von Fahrassistenzsystemen auf der anderen Seite. Zur effektiven Unterbindung einer Verfolgung ist es unerlässlich, dass eine Weile Funkstille gewahrt und das Pseudonym gewechselt wird. Dies führt jedoch zu einer vorübergehenden Unverfügbarkeit der durch das Fahrzeugnetz bereitgestellten Dienste und Assistenzsysteme. Da es keinem der in der Literatur bekannten Ansätze gelingt, an dieser Stelle einen geeigneten Kompromiss zu finden, wird mit der ProMix-Zone ein eigenes Konzept erarbeitet. Dieses zeigt unter Berücksichtigung der Angreifermodelle und Anforderungen einen Weg auf, den Schutz der Privatsphäre deutlich zu verbessern und dennoch eine ausreichende Funktionalität des Fahrzeugnetzes für Fahrassistenzsysteme zu bieten. Die Balance gelingt dabei auf eine flexible Art und Weise, da das Schutzniveau und die Verfügbarkeit von Fahrassistenzsystemen an die jeweilige Verkehrssituation des Einsatzgebietes angepasst werden können.

Um die Konzepte zum Schutz der Privatsphäre auch empirisch untersuchen zu können, wurde ein *VANET-Simulator* entwickelt, der die Simulation von Straßen- und Datenverkehr auf realem Kartenmaterial von OpenStreetMap ermöglicht und zudem neben der ProMix-Zone drei weitere bekannte Konzepte – die Mix-Zone, Silent Periods und SLOW – implementiert. Diese werden in einer Simulationsreihe aus 450 Durchläufen in drei städtischen Szenarien auf ihre Effektivität bezüglich des Schutzes vor Angreifern in Abhängigkeit von ihrer Positionierung, der Verkehrsdichte und individuellen Konfigurationsparametern hin untersucht. Dabei zeigt sich, dass das SLOW-Konzept in einem urbanen Szenario ein relativ hohes Schutzniveau für Teilnehmer eines Ad-hoc-Fahrzeugnetzes bieten kann.

Insgesamt zeigen die Ergebnisse dieser Arbeit, dass es in der Praxis gelingen kann, den Schutz der Privatsphäre in Ad-hoc-Fahrzeugnetzen sicherzustellen, sofern die gewonnenen Erkenntnisse bereits bei der Konzeption Berücksichtigung finden.

Abstract

Inter-vehicle communication can provide several services that have the potential to prevent accidents or reduce their consequences, to improve the efficiency of traveling or fleet management and supply passengers with internet access. In the next few years, the first serial production of communication-enabled vehicles will commence which marks the dawn of inter-vehicle networks.

The protection of the communication links is crucial to prevent misuses of the network or at least provide necessary information to sanction adversaries. All techniques for this purpose generally lead to identification of users and therefore to linkability of their actions. As a result, the privacy of users is compromised since each movement might be recorded and exploited.

Concerning this interplay, two subproblems emerge: The participants' reachability has to be ensured by location services without reducing their privacy and tracking of users has to be prevented. These two subproblems have to be addressed in order to provide a privacy-friendly usage of inter-vehicle networks. Several approaches are provided in the literature for both problems. They are investigated to determine their applicability as a solution in this work.

Surprisingly, all concepts that propose a privacy-aware location service are not able to provide protection against the service provider. For this reason, a new approach is presented that resolves this issue with the usage of Chaum's mix network, encrypted storage of data and a hierarchical structure of the network. An evaluation demonstrates that a practical application is possible with today's hardware.

The concepts to prevent tracking of users in inter-vehicle networks aim to strike a balance between privacy protection (ensured by pseudonym changes during radio silence) and sustaining of the driver assistance systems' functionality. An effective prevention of tracking requires pseudonym changes during radio silence which causes temporary unavailability of services and assistance systems provided by the network. None of the published approaches is able to compromise, so a new concept is presented – the ProMix Zone. This concept offers a balance between privacy protection and availability of driver assistance systems while considering threat models and

(functional) requirements. This is possible since the level of protection might be adapted according to traffic situations in the field of application.

To facilitate empirical analysis of the concepts for privacy protection, a *VANET Simulator* is developed that allows the simulation of road and network traffic on real road networks provided by the OpenStreetMap project. Additionally, the ProMix Zone and three further well-known concepts – the Mix Zone, Silent Periods and SLOW – are implemented. These approaches are examined in a series of 450 simulations in three urban scenarios to work out the effectivity of their privacy protection against adversaries according to their positioning, the traffic density and individual configuration parameters. It appears that the SLOW concept offers a relatively high level of privacy protection for participants in vehicular ad hoc networks in an urban scenario.

Overall the results of this work show that privacy protection in vehicular ad hoc networks is possible in practice as long as the gained insights are taken into account during the concept phase of a *VANET*.

Inhaltsverzeichnis

Abbildungsverzeichnis **xi**

Tabellenverzeichnis **xvii**

Algorithmenverzeichnis **xix**

Abkürzungsverzeichnis **xxi**

1. Einführung **1**
1.1. Ad-hoc-Fahrzeugnetze 3
1.2. Bedrohungen bei mangelnder IT-Sicherheit in VANETs 5
1.3. Aufbau der Arbeit 6
1.4. Beitrag und Erkenntnisse dieser Arbeit 7

2. Begriffe und Grundlagen **9**
2.1. Grundlagen der Informationssicherheit 9
2.1.1. Sicherheit 9
2.1.2. Schutzziele 10
2.1.3. Kryptographie 16
2.1.4. Kommunikationsmixe 20
2.2. Grundlagen der mobilen Netze 22
2.2.1. Drahtlose Kommunikation 22
2.2.2. Positions- und Zeitdienste 27
2.2.3. Routing 28
2.3. Besonderheiten von Ad-hoc-Fahrzeugnetzen 34
2.3.1. Identitäten in Ad-hoc-Fahrzeugnetzen 34
2.3.2. Speicherung von kryptographischem Material 36
2.3.3. Nachrichten in VANETs 37

3. Datenschutzfreundliche Positionsdienste **45**
3.1. Prinzip der Positionsdienste und Angriffe 45

3.2. Schutz durch Unschärfe . . . 48
3.2.1. Spatial and Temporal Cloaking . . . 48
3.2.2. CliqueCloak . . . 52
3.2.3. New Casper . . . 55
3.3. Weitere Ansätze . . . 59
3.3.1. Token-based Abuser Identification . . . 60
3.3.2. Anonymous Location Service . . . 63
3.4. Fazit . . . 65
3.5. Positionsdienst mit minimalen Vertrauensanforderungen . . . 65
3.5.1. Angreifermodell . . . 66
3.5.2. Notation . . . 66
3.5.3. Überblick und Architektur . . . 67
3.5.4. Sicheres Positionsupdate . . . 69
3.5.5. Sichere Positionsanfrage . . . 71
3.5.6. Effizientere Positionsanfrage ohne Verschleierung des Absenders 74
3.5.7. Evaluation . . . 75
3.6. Fazit . . . 78

4. Sicherer Pseudonymwechsel 79
4.1. Bedrohung der Privatsphäre . . . 79
4.1.1. Sicht des Angreifers . . . 79
4.1.2. Pseudonyme in VANETs . . . 83
4.1.3. Herausforderung beim Pseudonymwechsel . . . 84
4.2. Räumliche Ansätze . . . 87
4.2.1. Mix-Zone nach Beresford und Stajano . . . 87
4.2.2. Die Mix-Zone für Ad-hoc-Fahrzeugnetze . . . 90
4.2.3. Die CMIX-Zone . . . 92
4.2.4. Pseudonym-on-demand . . . 93
4.2.5. Density Zones . . . 95
4.2.6. Social Spots . . . 98
4.3. Zeitliche Ansätze . . . 100
4.3.1. Silent Periods . . . 100
4.3.2. Silent Cascade . . . 102
4.3.3. CARAVAN und AMOEBA . . . 104
4.4. Benutzerzentrierte Ansätze . . . 107
4.4.1. Swing & Swap . . . 107
4.4.2. SLOW . . . 113
4.4.3. Random Encryption Periods . . . 115
4.5. Fazit . . . 117
4.6. Die ProMix-Zone . . . 122

4.6.1. Angreifermodell und Vorüberlegungen 122
4.6.2. Architektur und Konzept 123
4.6.3. Auswahl relevanter Beacons 126
4.6.4. Evaluation 128
4.6.5. Fazit 134

5. Simulation von Ad-hoc-Fahrzeugnetzen 137
5.1. Motivation und Problemstellung 137
5.2. Vorhandene Simulationslösungen 138
5.3. Eigene Entwicklung und Implementierung 140
5.3.1. Anforderungen 140
5.3.2. Implementierungsdetails und Geschwindigkeitsoptimierung . . 142
5.3.3. Modell und Implementierung der Karte 145
5.3.4. Szenarien als Sammlung der dynamischen Komponenten . . . 153
5.3.5. Modell und Implementierung der Fahrzeuge 154
5.3.6. Road-Side-Units 173
5.3.7. Ereignisse 173
5.3.8. Konzepte zum Schutz der Privatsphäre 174
5.3.9. Graphische Benutzeroberfläche 178
5.3.10. Simulation 184
5.3.11. Fazit 188
5.4. Untersuchung von Konzepten zum Schutz der Privatsphäre 189
5.4.1. Vorüberlegungen 189
5.4.2. Angriffe 193
5.4.3. Metriken 202
5.4.4. Ergebnisse 206
5.4.5. Fazit 236

6. Fazit 239
6.1. Ergebnisse der Arbeit 239
6.2. Ausblick 241

Literaturverzeichnis 243

A. Statistische Verkehrsdaten 265

B. VANETsim XML-Schemata 267
B.1. XML-Schema der VANETsim-Karte 267
B.2. XML-Schema des VANETsim-Szenarios 270

C. Simulationsergebnisse 275

C.1. Verfolgung über mehrere Mix-Zonen hinweg 275
C.2. Mix-Zonen bei unterschiedlichen Konfigurationen 280
C.3. Verfolgung über mehrere Silent Periods hinweg 285
C.4. Effektivität der Silent Periods an unterschiedlichen Orten 290
C.5. Verfolgung über mehrere SLOW-induzierte Funkstillen hinweg 295
C.6. Einfluss des Straßennetzes auf die Effektivität von SLOW 298

Abbildungsverzeichnis

1.1. Todesfälle im Straßenverkehr der Bundesrepublik Deutschland (Quelle: [Sta11]) 2
1.2. Beispielhafte Kommunikation in Ad-hoc-Fahrzeugnetzen (in Anlehnung an [Plö09]) 4

2.1. Symmetrische Verschlüsselung (Abbildung nach [FP97]) 16
2.2. Symmetrische Authentikation (Abbildung nach [FP97]) 17
2.3. Asymmetrische Verschlüsselung (Abbildung nach [FP97]) 18
2.4. Digitale Signatur (Abbildung nach [FP97]) 19
2.5. Prinzip von Kommunikationsmixen (1) 20
2.6. Prinzip von Kommunikationsmixen (2) 21
2.7. Empfänger bei Single- und Multihop-Broadcast-Kommunikation . . . 23
2.8. Position (a) vs. spatial (b) awareness nach [TC03] 28
2.9. Einordnung von Routingverfahren in *MANET*s nach [MWH01] 29
2.10. Klassen von Positionsdiensten mit Beispielen 31
2.11. Location Based Multicast mit rechteckiger Forwarding Zone (nach [Mai04]) 33
2.12. Greedy Forwarding mit lokalem Maximum und Caching-Mechanismus (nach [MES04]) 34
2.13. Lokales Abbild der Verkehrssituation durch Beacons 38
2.14. Fortschrittliche Fahrassistenzsysteme 39
2.15. Sendeintervalle der *Beacons* in Abhängigkeit von der Geschwindigkeit (nach [Sch08]) 40

3.1. Minimales Prinzip eines Positionsdienstes 46
3.2. Aufhebung der Anonymität in ungünstiger Situation (nach [GG03]) . 51
3.3. Beispiel einer Situation in CliqueCloak (nach [GL05]) 54
3.4. Pyramidenstrukturen für Casper (nach [MCA06]) 56
3.5. Architektur des Backbones 68
3.6. Sicheres Positionsupdate (*LLS* und *CLS*) 70

3.7. Sicheres Positionsupdate (nur *LLS*) 72
3.8. Sichere Positionsanfrage 72
3.9. Effizentere Positionsanfrage ohne Verschleierung des Absenders 74

4.1. Verfolgbarkeit durch Beacons (nach [Sch08, Plö09]) 85
4.2. Pseudonymwechsel bei Funkstille (nach [Sch08, Plö09]) 86
4.3. Überblick über die untersuchten Ansätze 87
4.4. Mix-Zone nach Beresford und Stajano (nach [BS03]) 88
4.5. Untersuchung zur Bewegung in einer Mix-Zone [BS03] (nach [Sch08, Plö09]) 89
4.6. Aufbau einer Mix-Zone nach [Sch08, Plö09] 90
4.7. Prinzip der Mix-Zone 91
4.8. Simulationsaufbau nach [SWL10] 96
4.9. *Small Social Spot* nach [LLL+11]: Der gelb hinterlegte Bereich umreißt die Anonymitätsgruppe. 98
4.10. *Large Social Spot* nach [LLL+11] 99
4.11. Prinzip der *Silent Period* (nach [HMYS05]) 101
4.12. *Silent Cascade* (nach [HYMS06]) 103
4.13. Nutzung von *LBS* in CARAVAN (nach [SHL+05]) 106
4.14. *Swing* und *Swap* nach [LSHP06]. Kästen stellen Pseudonyme dar, unterbrochene Linien bedeuten Funkstillen. 109
4.15. Beschränkte Aufenthaltsmöglichkeiten von Teilnehmern in *VANET*s . 112
4.16. Aufbau der Nachrichten 124
4.17. Die drei Schritte der ProMix-Zone 125
4.18. Relevanz von Fahrzeugen aus Sicht eines Teilnehmers 128
4.19. Auswirkungen von n auf *Schritt 1* 131
4.20. Auswirkung von n auf *Schritt 3* 132
4.21. Auswirkung der Größe der Beacons auf *Schritt 1* 133
4.22. Auswirkung von k auf *Schritt 3* 133

5.1. Funktionsweise von TraNS (vgl. [PRL+08]) 139
5.2. *VANET-Simulator*: Anlegen einer neuen Karte (hier mit 100 Regionen) 146
5.3. *VANET-Simulator*: Modell des Straßennetzes 147
5.4. *VANET-Simulator*: Prinzip des *LaneContainers* (nach [Gru08]) . . . 149
5.5. Zonenaufteilung nach dem *UTM*-System (Quelle: Wikimedia Commons, CC-BY-SA 3.0) 151
5.6. Ablauf eines Simulationsschritts des Nagel-Schreckenberg-Modells . . 157
5.7. Konzept des IDM nach [TH02] 160
5.8. Arbeitsweise von Dijkstra, Best-first-search und A^* (Quelle: [Pat11]). Untersuchte Knoten sind eingefärbt, die Linie stellt die gefundene Route dar. 167

5.9. *VANET-Simulator*: Logischer Aufbau der Datenstruktur für Prioritäten einer Kreuzung am Beispiel mit vier Eingängen und drei Ausgängen 168
5.10. *VANET-Simulator*: Beispiel der Ermittlung einer Vorfahrtsstraße . . 169
5.11. *VANET-Simulator*: Stauwarnung 172
5.12. *VANET-Simulator*: Relevante Fahrzeuge der implementierten ProMix-Zone 176
5.13. *VANET-Simulator*: Graphische Oberfläche 179
5.14. *VANET-Simulator*: Erstellen von Karten 181
5.15. *VANET-Simulator*: Erzeugen von Fahrzeugen 182
5.16. *VANET-Simulator*: Weitere Konfigurationsmöglichkeiten 185
5.17. *VANET-Simulator*: Ablauf der Simulation 186
5.18. *VANET-Simulator*: Beispielhafte Aufteilung der 25 Regionen einer Karte auf *WorkerThreads* bei zwei Prozessorkernen (nach [Gru08]) . . 186
5.19. *VANET-Simulator*: Mögliche Inkonsistenz bei der Simulationsdurchführung ohne Barrieren (nach [Gru08]) 187
5.20. Kartenausschnitt Berlin 190
5.21. Kartenausschnitt New York City 191
5.22. Kartenausschnitt Puebla 192
5.23. Angriffserfolge und Metriken bei Mix-Zonen in Berlin und New York . 208
5.24. Angriffserfolge und Metriken bei Mix-Zonen in Puebla 209
5.25. Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in Berlin 211
5.26. Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in Berlin 212
5.27. Überblick über die Mix-Zonen in Puebla in unterschiedlichen Größen 214
5.28. Überblick über die Mix-Zonen in Berlin bei unterschiedlichen Verkehrsdichten 215
5.29. Detailansicht der Mix-Zone an der Otto-Braun-Straße in Berlin . . . 216
5.30. Angriffserfolge und Metriken bei Silent Periods in Berlin und New York 218
5.31. Angriffserfolge und Metriken bei Silent Periods in Puebla 219
5.32. Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in Berlin 221
5.33. Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in Berlin 222
5.34. Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in New York bei unterschiedlichen Verkehrsdichten 225
5.35. Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in Puebla bei unterschiedlichen Längen der Silent Periods 227
5.36. Angriffserfolge bei SLOW in Berlin und New York 228
5.37. Angriffserfolge bei SLOW in Puebla 229

5.38. Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in Berlin . 231
5.39. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Berlin bei unterschiedlichen Schwellengeschwindigkeiten 234
5.40. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Puebla bei unterschiedlichen Verkehrsdichten 235

A.1. Personenverkehrsaufwand nach Verkehrsträgern in Deutschland [Bun10] 265
A.2. Fahrzeugbestand in Deutschland [KB11] (seit 2008 nur noch Fahrzeuge ohne vorübergehende Stilllegung) 266
A.3. Polizeilich gemeldete Verkehrsunfälle in Deutschland [Sta11] 266

C.1. Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in New York 276
C.2. Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in New York 277
C.3. Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in Puebla 278
C.4. Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in Puebla 279
C.5. Überblick über die Mix-Zonen in New York in unterschiedlichen Größen 281
C.6. Überblick über die Mix-Zonen in New York bei unterschiedlichen Verkehrsdichten 282
C.7. Überblick über die Mix-Zonen in Berlin in unterschiedlichen Größen . 283
C.8. Überblick über die Mix-Zonen in Puebla bei unterschiedlichen Verkehrsdichten 284
C.9. Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in New York 286
C.10. Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in New York 287
C.11. Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in Puebla 288
C.12. Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in Puebla 289
C.13. Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in New York bei unterschiedlichen Längen der Silent Periods . . 291
C.14. Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in Berlin bei unterschiedlichen Verkehrsdichten 292
C.15. Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in Berlin bei unterschiedlichen Längen der Silent Periods 293
C.16. Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in Puebla bei unterschiedlichen Verkehrsdichten 294

C.17. Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in New York 296
C.18. Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in Puebla 297
C.19. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in New York bei unterschiedlichen Schwellengeschwindigkeiten 299
C.20. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in New York bei unterschiedlichen Verkehrsdichten 300
C.21. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Berlin bei unterschiedlichen Verkehrsdichten 301
C.22. Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Puebla bei unterschiedlichen Schwellengeschwindigkeiten 302

Tabellenverzeichnis

2.1. Schutzziele nach [FP97] 11
2.2. Berechtigungen privilegierter Teilnehmer im VANET (nach [Sch08, Plö09]) 43

4.1. Übersicht über Ansätze zum sicheren Pseudonymwechsel 121

5.1. Straßentypen aus OpenStreetMap, ihre geschätzte Maximalgeschwindigkeit im Simulator sowie ihre Färbung 152
5.2. Ereignisse eines durchschnittlichen Fahrers mit einer durchschnittlichen Geschwindigkeit von 60 km/h und einer Jahresfahrleistung von 20.000 km (Untersuchung von [HL91] nach [AK00]) 155
5.3. Typische Parameter des IDM nach [TH02] 161
5.4. Übersicht über die untersuchten Verkehrsmodelle 162
5.5. Überblick über die Simulationsergebnisse 237

Algorithmenverzeichnis

3.1. Adaptive-interval Cloaking-Algorithmus (vgl. [GG03]) 49
3.2. Bottom-up Cloaking-Algorithmus (nach [MCA06]) 58

5.1. Angriff auf Mix-Zone (einfach) 196
5.2. Angriff auf Mix-Zone (komplex) 197
5.3. Angriff auf SilentPeriod (einfach) 199
5.4. Angriff auf SilentPeriod (komplex) 200
5.5. Angriff auf SLOW 202

Abkürzungsverzeichnis

802.11p	Standard der IEEE für die drahtlose Kommunikation zwischen Fahrzeugen
a	Beschleunigung
ABS	Antiblockiersystem
ADAS	Advanced Driver Assistance System
AES	Advanced Encryption Standard
ALS	Anonymous Location Service
AODV	Ad-hoc On-Demand Distance Vector Routing
ARSU	Attacker RSU
B	Beacon
BAG	Bundesamt für Güterverkehr
BDSG	Bundesdatenschutzgesetz
c	öffentlicher Schlüssel eines asymmetrischen Kryptosystems
C2C	Car-to-car
C2I	Car-to-infrastructure
CA	Cellular Automata
CA	Certificate Authority
cert	Zertifikat einer CA über den öffentlichen Schlüssel eines asymmetrischen Kryptosystems

CGGC	Cached Greedy Geocast
CI	Contact Information
cid	Cell Identifier
CLS	Central Location Service
CMIX-Zone	Cryptographic Mix-Zone
d	privater (geheimer) Schlüssel eines asymmetrischen Kryptosystems
DES	Data Encryption Standard
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DoS	Denial of Service
DREAM	Distance Routing Effect Algorithm for Mobility
DSDV	Destination-Sequence Distance Vector Routing
DSPS	Distributed Secure Position Service
DSR	Dynamic Source Routing
DSRC	Dedicated Short Range Communication
ECC	Elliptic Curve Cryptography
ESP	Elektronisches Stabilitätsprogramm
ETRS89	Europäisches Terrestrisches Referenzsystem 1989
et al.	et alii, und andere
FIPS	Federal Information Processing Standard
GL	Group Leader
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing

HLR	Home Location Register
HMAC	Hashed Message Authentication Code
ID	Identifier, identifizierendes Merkmal
IDM	Intelligent Driver Model
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv6	Internet Protocol in der Version 6
ITS	Intelligent Transport System
IVC	Inter-vehicle communication
JavaVM	Java Virtual Machine
k	symmetrischer Schlüssel eines Kryptosystems
l	Position
LBM	Location-based Multicast
LBS	Location-based Service
LKW	Lastkraftwagen
LLS	Local Location Service
loc	Location, Aufenthaltsort
LREQ	Location Request
LRES	Location Response
LUP	Location Update
M	Message, Nachricht
MAC	Message Authentication Code
MAC-Adresse	Media Access Control-Adresse
MANET	Mobile Ad Hoc Network

MOBIL	Minimizing Overall Braking Induced by Lane-Changes
MP	Mapping-Pseudonym
NIST	National Institute of Standards and Technology
NOW	Network on Wheels
ns-2	Network Simulator Version 2
OSI-Modell	Open Systems Interconnection Reference Model
P	Pseudonym
PKI	Public Key Infrastructure
POI	Point of Interest
ProMix-Zone	Proxy-based Mix-Zone
QoS	Quality of Service
RA	Registration Authority
REP	Random Encryption Periods
RFC	Request for comment
RSA	Asymmetrisches Kryptosystem, benannt nach den Entwicklern Rivest, Shamir and Adleman
RSU	Road-Side Unit
SA	Service Area
SLOW	Silence at low Speeds
SLS	Simple Location Service
SP	Service Provider
StVG	Straßenverkehrsgesetz
StVO	Straßenverkehrsordnung
StVZO	Straßenverkehrszulassungsordnung

SUMO	Simulation of Urban Mobility
t	Zeit
TA	Trusted Authority
THW	Technisches Hilfswerk
TN	Teilnehmer
TPD	Tampe-Proof Device
TPM	Tamper-Proof Module bzw. Trusted Plattform Module
TraCI	Traffic Control Interface
TRH	Tamper-Resistant Hardware
TTL	Time-to-live
TTP	Trusted Third Party
UID	Unique Identifier, eindeutiges identifizierendes Merkmal
UTM	Universal Transverse Mercator
v	Geschwindigkeit
V2I	Vehicle-to-infrastructure
V2V	Vehicle-to-vehicle
VANET	Vehicular Ad Hoc Network, Ad-hoc-Fahrzeugnetz
VANET-ID	Eindeutige Kennung (ID) eines Teilnehmers im Ad-hoc-Fahrzeugnetz
Veins	Vehicles in Network Simulation
VHZ	Virtual Homezones
VLR	Visitor Location Register
WGS84	World Geodetic System 1984
XML	Extensible Markup Language
ZRP	Zone Routing Protocol

Kapitel 1

Einführung

Seit der Entwicklung des Automobils im 19. Jahrhundert nahm seine Bedeutung als Fortbewegungsmittel stetig zu. In Deutschland werden inzwischen 80 % der Gesamtfahrleistung durch Automobile erbracht (vgl. Anhang A). Die enorm gestiegene Verbreitung seit der Markteinführung führte schnell zu einer hohen Verkehrsdichte, was insbesondere zu Beginn der Entwicklung, auch mangels Verkehrs- und Sicherheitsregelungen, zu einer großen Anzahl an Unfällen mit Personenschaden führte (vgl. [Sta11], siehe auch Anhang A). Aus diesem Grund waren Automobilhersteller früh bestrebt, den Straßenverkehr sicherer zu gestalten; im Rahmen der Werbung war die Sicherheit von Fahrzeugen bereits kurz nach der Markteinführung des Automobils ein zentraler Aspekt (vgl. [Zim01]). Auch der Gesetzgeber reagierte auf die stetig wachsende Zahl an Unfällen. Im Jahr 1909 wurde das erste Kraftfahrgesetz für das Deutsche Reich erlassen, das Vorläufer der Straßenverkehrsordnung (StVO[1]) und Straßenverkehrszulassungsordnung (StVZO[2]) war und Regelungen wie beispielsweise Tempolimits enthielt. In den 1920er Jahren wurden die ersten Lichtzeichenanlagen an Kreuzungen installiert [NH99]. Mit der Straßenverkehrsordnung (vgl. Eingangsformel StVO) wurde 1952 in Deutschland ein umfassendes Regelwerk verabschiedet, das bis heute – in überarbeiteter Form – Gültigkeit besitzt.

Trotz dieser sicherheitsrelevanten Maßnahmen wurde auch in der Folgezeit eine Zunahme an Unfällen im Straßenverkehr verzeichnet, was insbesondere auf die weiterhin steigende Anzahl an Automobilen zurückzuführen war. Ein trauriger Höhepunkt wurde schließlich im Jahr 1970 mit annähernd 20 000 Verkehrstoten ([Sta11]) in der Bundesrepublik Deutschland erreicht.

Als Konsequenz daraus wurden einerseits zahlreiche weitere Maßnahmen zur Erhö-

[1] abrufbar unter http://www.gesetze-im-internet.de/stvo/index.html
[2] abrufbar unter http://www.gesetze-im-internet.de/stvzo_2012/index.html

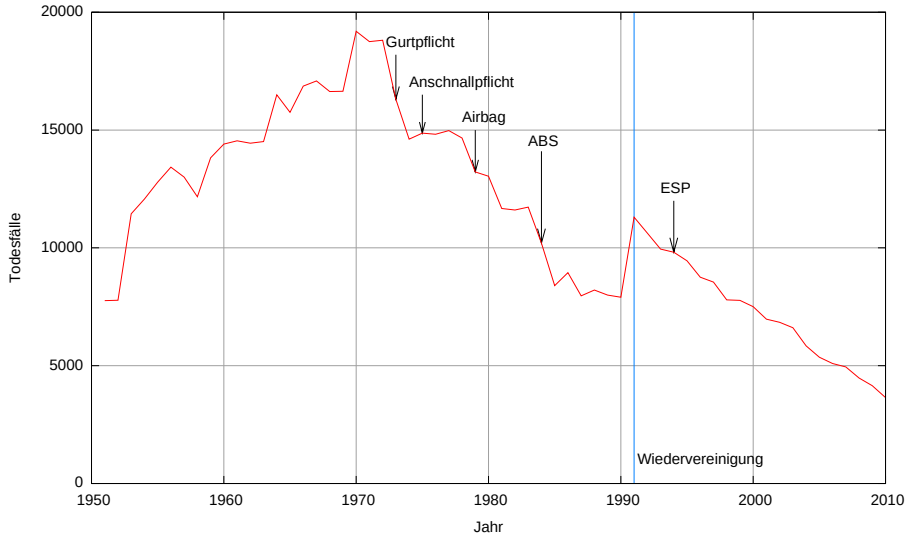


Abbildung 1.1.: Todesfälle im Straßenverkehr der Bundesrepublik Deutschland (Quelle: [Sta11])

hung der Sicherheit im Straßenverkehr getroffen (z. B. die Einführung einer Promille-Grenze für Fahrer und die Inbetriebnahme des ersten Rettungshubschraubers *Christoph 1* [ADA11]). Andererseits wurde nun verstärkt auch auf Sicherheitssysteme im Fahrzeug Wert gelegt. In den folgenden Jahrzehnten wurden etwa der Dreipunktgurt und eine Gurt- sowie Anschnallpflicht, der Airbag, ABS, ESP und weitere moderne aktive und passive Sicherheitssysteme[3] eingeführt. Durch das Zusammenspiel von organisatorischen Maßnahmen und Sicherheitssystemen konnte endlich eine Trendwende erreicht werden. Seit damals nimmt die Zahl der Verkehrstoten – trotz einer weiterhin steigenden Anzahl an Fahrzeugen – stetig ab (vgl. Abbildung 1.1); für das Jahr 2010 meldet das Statistische Bundesamt 3648 Todesfälle ([Sta11]).

Trotz der positiven Entwicklung bestehen keine Zweifel daran, dass es eine zentrale Herausforderung ist, jegliche Todesfälle im Straßenverkehr zu vermeiden, was durch entsprechende Forderungen von Automobil- und Verkehrsclubs (z. B. „Vision Zero – Null Verkehrstote“ des *Verkehrsclub Deutschland* [Waß04]) zum Ausdruck gebracht wird.

Inzwischen ist es gelungen, Fahrzeuge und Verkehrswege so sicher zu bauen, dass Unfälle (insbesondere mit schweren Folgen) vermieden werden könnten – wenn nicht

[3] In dieser klassischen Einteilung versteht man unter aktiven Systemen diejenigen, die zur Unfallvermeidung in einer Gefahrensituation aktiv werden oder zur Gefahrvermeidung dienen (z. B. ABS, ESP, Toter-Winkel-Warnung), unter passiven Systemen dagegen jene, welche im Falle eines unvermeidbaren Unfalls dessen Folgen abschwächen sollen (z. B. Gurtstraffer, verstärke Fahrgastzelle, Airbag).

menschliches Fehlverhalten weiterhin zu Katastrophen führen würde:

> „Der Mensch macht Fehler. Das System Verkehr muss mit diesen Fehlern rechnen und sie verzeihen.” [Ver09]

Damit dieses „Verzeihen“ im Straßenverkehr gelingen kann, ist es notwendig, den Fahrer optimal dabei zu unterstützen, ein Fahrzeug zu führen. Fahrassistenzsysteme können ihm Aufgaben abnehmen und auf diese Weise entlasten oder ihn frühzeitig vor Gefahren warnen:

> „Technische Verbesserungen und Aufklärungskampagnen können ihren Beitrag zur Unfallvermeidung leisten und die Schwere von Unfällen verringern. Gurtpflicht, Airbags, ABS und bessere Rettungs- und Intensivmedizin haben die Unfallfolgen gemildert. Um den nächsten Schritt zu einer deutlichen Reduzierung der Unfälle zu schaffen, braucht es aber ein umfassenderes Konzept.“ [Ver09]

Die Einführung eines Ad-hoc-Fahrzeugnetzes bietet das Potential, als ein solch umfassendes Konzept zu fungieren und die Zahl der Unfälle sowie der Toten und Verletzten im Straßenverkehr durch menschliches Versagen deutlich zu reduzieren. Der Einsatz dieser Netze ist allerdings nicht allein auf die Erhöhung der Verkehrssicherheit beschränkt (wobei dies ein wesentlicher Bestandteil sein wird), sondern bietet zudem die Möglichkeit, den Verkehrsfluss zu optimieren und die Fortbewegung mit Fahrzeugen effizienter und angenehmer zu gestalten.

1.1. Ad-hoc-Fahrzeugnetze

Unter einem Ad-hoc-Fahrzeugnetz (*vehicular ad hoc network*, *VANET*) versteht man ein flexibles Funknetzwerk zwischen Fahrzeugen (und möglicherweise weiteren Verkehrsteilnehmern), wobei straßennahe Infrastruktureinheiten (*road-side units*, *RSU*s) sowie zentrale Koordinierungsstellen und Systeme von Diensteanbietern eingebunden werden können. Darüberhinaus ist es denkbar, ein Gateway zum Internet zu integrieren.

Systematisch werden *VANET*s häufig als Untergruppe der sogenannten mobilen Ad-hoc-Netze (*mobile ad hoc network*s, *MANET*s) [CM99, MC98] verstanden. Die wesentlichen Unterschiede zu einem *MANET* bestehen darin, dass sich die mobilen Endgeräte (häufig als *Knoten* oder *Teilnehmer* bezeichnet) in Fahrzeugen (PKWs, LKWs, Motorrädern, Bussen, etc.) befinden und sich somit nicht komplett frei, sondern nur entlang von Straßen bewegen können. Zudem sind im Zusammenhang mit *VANET*s Verkehrsregeln wie Einbahnstraßen zu beachten. Ein weiterer Unterschied

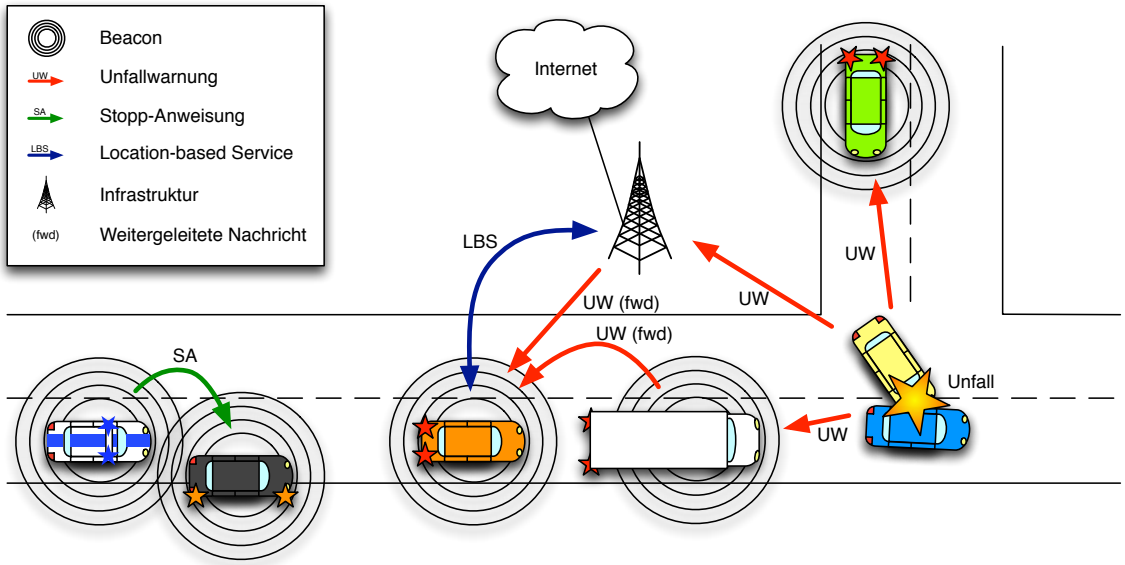


Abbildung 1.2.: Beispielhafte Kommunikation in Ad-hoc-Fahrzeugnetzen (in Anlehnung an [Plö09])

besteht in der zumeist deutlich größeren Relativgeschwindigkeit zwischen den Knoten eines *VANET*s, die bis zu 500 km/h[4] betragen kann.

Zwischen den Teilnehmern des *VANET*s – also in der Regel Fahrzeugen – können, sofern sie sich in Funkreichweite befinden, Nachrichten und Informationen ausgetauscht werden. Technisch wird dies auf Basis einer WLAN-Variante, dem Standard IEEE 802.11p [IEE10] – oft auch als *DSRC*, *dedicated short range communication* bezeichnet – realisiert, dessen Reichweite auf zwischen 300 m und 1000 m – je nach Abschattung des Signals – geschätzt wird. Die Kommunikation zwischen Fahrzeugen wird oft auch *C2C-* oder *V2V-Communication* (*Car-to-car* bzw. *vehicle-to-vehicle*), sowie gelegentlich *IVC* (*inter-vehicle communication)* oder – insbesondere im Ingenieursumfeld – *ITS* (*Intelligent Transport System*) genannt. Diese Unterschiede in der Bezeichnung sind darin begründet, dass viele Ideen und Konzepte für *VANET*s aus diversen Fachbereichen (z. B. Netzwerktechnik, Verkehrstechnik, Straßenbau, Navigationsdienstleister, Automobilbranche, etc.) eingebracht wurden, die alle ein unterschiedliches Vokabular benutzen.

Die zwischen den Fahrzeugen ausgetauschten Informationen können Telematikdaten der einzelnen Teilnehmer (wie beispielsweise ihre Position, Geschwindigkeit, etc. in sog. *Beacons*, vgl. Abschnitt 2.3.3.1) oder Hinweise auf gefährliche Straßenverhält-

[4] Seit den 1980er Jahren riegeln die meisten Automobilhersteller auf Grund einer Selbstverpflichtung ihre Fahrzeuge bei 250 km/h ab.

nisse (z. B. Aquaplaning, Blitzeis, etc.) oder Verkehrssituationen (Unfälle, Stauenden, etc.) enthalten, die genutzt werden, um dem Fahrer Hilfestellungen in Form von Fahrassistenzsystemen zu geben oder ihn frühzeitig auf akute Gefahren hinzuweisen, wodurch möglicherweise Unfälle vermieden oder zumindest ihre Folgen abgemildert werden können (vgl. Abbildung 1.2). Weiterhin kann ein *VANET* die Arbeit von Einsatzkräften erleichtern, da diese mit speziellen Nachrichten anderen Fahrzeugen Anweisungen geben und sogar ganze Straßenabschnitte für den Verkehr sperren können. Wird in ein *VANET* – wie eingangs erwähnt – Infrastruktur integriert, die eine Verbindung zu Diensteanbietern von Location-based Services (LBS) oder dem Internet ermöglicht, so können weitere Komfortdienste wie beispielsweise Tourismusinformationen oder E-Mail-Kommunikation im Fahrzeug angeboten werden.

Der Informationsaustausch zwischen Fahrzeugen und Infrastruktur – häufig als *C2I-* oder *V2I-Communication* (car-to-infrastructure bzw. vehicle-to-infrastructure) bezeichnet – kann auf Basis der gleichen Netzwerkprotokolle stattfinden und unterscheidet sich daher – mit Ausnahme der Unbeweglichkeit der RSUs – nicht von der Kommunikation zwischen Fahrzeugen.

Plößl fasst in seiner Dissertation die Eigenschaften von automobilen Ad-hoc-Netzen wie folgt zusammen:

> „Ein VANET (Vehicular Ad-Hoc-Network) ist ein Ad-hoc-Netz aus Automobilen, die selbstständig Funkverbindungen untereinander etablieren. Automobile bewegen sich i. d. R. entsprechend ihrer Rahmenbedingungen (Straßennetz, Verkehr und Verkehrsregeln) und sind in der Lage, sich an daraus resultierende Veränderungen der Netztopologie, die häufig und abrupt auftreten können, eigenständig anzupassen. Daten können über mehrere Automobile hinweg weitergeleitet werden, um ihr Ziel zu erreichen. Ein VANET kann sowohl autonom von anderen Netzen oder Infrastruktur existieren als auch stationäre Transmitter nutzen und mit weiteren Netzen verbunden werden (hybrides VANET).“ [Plö09]

Diese Definition findet auch in der vorliegenden Arbeit Anwendung, wobei meist ein *hybrides VANET* vorausgesetzt wird. Auf Ausnahmen wird explizit hingewiesen.

1.2. Bedrohungen bei mangelnder IT-Sicherheit in VANETs

Die Kommunikation zwischen Fahrzeugen bietet ein großes Potential an Möglichkeiten, die Reise in einem Straßenfahrzeug sicherer und angenehmer zu gestalten. Vor-

aussetzung dafür ist allerdings, dass das entstehende Ad-hoc-Netz vor Missbrauch geschützt ist.

Wenn sich Fahrzeuge darauf verlassen, dass die Informationen, die sie von anderen Verkehrsteilnehmern oder der Infrastruktur erhalten, der Realität entsprechen, kann es sich negativ auf den Verkehrsfluss auswirken oder sogar lebensbedrohliche Folgen haben, wenn falsche Informationen übermittelt werden. Doch nicht nur das Senden gefälschter Informationen birgt große Gefahren in sich. So könnte auch die Privatsphäre von Teilnehmern, die – wie bereits erwähnt – möglicherweise in sehr kurzen Abständen sämtliche Informationen über ihre Fortbewegung preisgeben, stark bedroht werden. Mit den Daten, die somit zur Verfügung stehen, ist eine automatisierte Überwachung und Verfolgung von Fahrzeugen sehr leicht möglich. Die Sicherstellung der Echtheit von Nachrichten und der Privatsphäre der Verkehrsteilnehmer ist unerlässlich und stellt einen entscheidenden Faktor für die sichere, erfolgreiche und auf Akzeptanz stoßende Einführung von Ad-hoc-Fahrzeugnetzen dar.

1.3. Aufbau der Arbeit

In der vorliegenden Arbeit sollen zu Beginn in Kapitel 2 die relevanten Begriffe herausgearbeitet und anschließend in den Kontext der Ad-hoc-Fahrzeugnetze eingeordnet werden. Weiterhin werden Grundlagen der relevanten Forschungsbereiche kurz dargestellt, um ein gemeinsames Grundverständnis zu schaffen. Diese Betrachtungen schließen die Felder der IT-Sicherheit, drahtlose Netzwerke im Allgemeinen und *VANET*s im Speziellen ein.

Das anschließende Kapitel 3 ist der datenschutzfreundlichen Kommunikation zwischen zwei Teilnehmern eines Ad-hoc-Fahrzeugnetzes über größere Distanzen hinweg gewidmet. Einer detaillierten Betrachtung der Problemstellung und der in der Literatur beschriebenen Lösungsansätze folgt – aufbauend auf den gewonnen Erkenntnissen – ein eigener Lösungsansatz, der deutlich stärkeren Angreifermodellen genügt als in der Literatur Anwendung finden.

In Kapitel 4 wird eine weitere bekannte Bedrohung der Privatsphäre in *VANET*s untersucht – die Verfolgbarkeit von Teilnehmern. An eine eingehende Betrachtung der Problematik schließt sich eine Untersuchung und Klassifikation der großen Anzahl an Vorschlägen unterschiedlicher Autoren an, dieser Gefahr zu begegnen. Da auch in diesem Zusammenhang bisher keine optimale Lösung bekannt ist, wird ein Ansatz vorgestellt, der versucht, einen Kompromiss zwischen den widerstreitenden Interessen der beteiligten Parteien zu finden.

Kapitel 5 widmet sich dem größten Schwerpunkt dieser Arbeit. In der ersten Hälfte werden der selbst entwickelte *VANET-Simulator* und die zu seiner Entstehung füh-

renden wissenschaftlichen Arbeiten und Designentscheidungen beleuchtet und umfassend dargelegt. Der Simulator wird in der zweiten Hälfte des Kapitels genutzt, um drei nach noch detailliert zu spezifizierenden Kriterien ausgewählte Ansätze zur Verhinderung einer Verfolgbarkeit in Ad-hoc-Fahrzeugnetzen (siehe Kapitel 4) eingehend auf ihre Einflussgrößen und ihre Effektivität hin zu untersuchen. Die Konfigurationen und die Ergebnisse der dazu durchgeführten 450 Simulationsläufe werden dabei ausführlich dargestellt.

Die Arbeit schließt mit einem Fazit über die gewonnen Erkenntnisse und einem Ausblick auf die noch offenen Fragestellungen dieses Themengebiets.

1.4. Beitrag und Erkenntnisse dieser Arbeit

Wie in der oben dargestellten Struktur ersichtlich, widmet sich die Arbeit drei zentralen Themenbereichen:

Datenschutzfreundliche Kommunikation zwischen Teilnehmern In dieser Arbeit wird neben einer detaillierten Betrachtung der (wenigen) bereits vorhandenen relevanten Lösungsansätze für eine datenschutzfreundliche Kommunikation zwischen Teilnehmern eines Ad-hoc-Netzes über größere Distanzen hinweg ein eigenes Konzept auf Basis von Chaum'schen Mixen vorgestellt. Dieses zeichnet sich dadurch aus, dass es Schwachstellen der bekannten Ansätze vermeidet und durch die Hinzunahme der in dem Forschungsbereich des technischen Datenschutzes verbreiteten Kommunikationsmixe ein deutlich höheres Schutzniveau als die existierenden Ansätze ermöglicht. Eine Evaluation des Konzepts zeigt, dass einem Einsatz in der Praxis keine technischen Hindernisse im Weg stehen.

Sicherer Pseudonymwechsel zur Verhinderung der Verfolgung von Teilnehmern Dieser Abschnitt betrachtet eingangs umfassend die Bedrohung, der sich Teilnehmer eines Ad-hoc-Netzes ausgesetzt sehen, wenn keine Maßnahmen zum Schutz ihrer Privatsphäre ergriffen werden. In einem nächsten Schritt werden die in der Literatur bekannten Lösungsansätze analysiert, kategorisiert und bewertet. Dabei wird nicht nur der Schutz der Privatsphäre berücksichtigt, sondern auch die Einschränkungen, die mit dem Einsatz der jeweiligen Ansätze einhergehen. Aus den gewonnen Erkenntnissen dieser Betrachtungen wird anschließend ein eigener Vorschlag eingebracht und detailliert ausgearbeitet, der zwar weiterhin auf den Schutz der Privatsphäre der Teilnehmer großen Wert legt, jedoch gleichzeitig die negativen Konsequenzen auf die Straßenverkehrssicherheit, die durch viele andere effektive Ansätze ausgelöst werden, minimiert.

Entwicklung eines *VANET-Simulators* und simulative Untersuchung von Konzepten zum sicheren Pseudonymwechsel Mangels geeigneter umfassender Alternativen wird ein *VANET-Simulator* entwickelt, der zur Untersuchung einer Vielzahl von Aspekten in Ad-hoc-Fahrzeugnetzen geeignet ist und neben einer guten Erweiterbarkeit auch eine einfache und intuitive Bedienung ermöglicht. Dazu wird systematisch die relevante Literatur – unter anderem auf dem Gebiet der Verkehrssimulation, der Wegsuche, der Kartenprojektion oder der performanten Programmierung – herangezogen und berücksichtigt. Die Struktur, die Modellierung und getroffene Designentscheidungen werden aufbereitet dargestellt und können zur Einarbeitung in den Simulator oder für ähnliche Projekte herangezogen werden.

Mit Hilfe des *VANET-Simulators* wird anschließend eine detaillierte Analyse der drei Konzepte (Mix-Zonen, Silent Periods und SLOW) zum sicheren Pseudonymwechsel vorgenommen, die in der vorangegangen Betrachtung als am effektivsten in Hinblick auf den Schutz der Privatsphäre eingeschätzt werden. Zu diesem Zweck werden 450 Simulationsläufe mit unterschiedlichen Parametern und Konfigurationen durchgeführt, die die Einflussgrößen auf die Effektivität der Konzepte aufdecken und somit dazu beitragen, die bisher entweder analytisch oder in sehr kleinem Rahmen durchgeführten empirischen Untersuchungen um eine einheitliche Betrachtung zu ergänzen und Vergleichbarkeit herzustellen. Die gewonnenen Erkenntnisse schließlich zeigen auf, welches Konzept unter welchen Rahmenbedingungen und Gegebenheiten Erfolg versprechend ist.

Kapitel 2

Begriffe und Grundlagen

Dieses Kapitel dient dazu, Grundbegriffe aus dem Umfeld sowohl der Informationssicherheit als auch der Ad-hoc-Fahrzeugnetze, die kontextabhängig aufgefasst und verstanden werden können, in Bezug zu dieser Arbeit zu setzen. Zudem sollen die Grundlagen der Kryptographie und mobiler Netzwerkkommunikation sowie die Bausteine, die für die Gestaltung komplexer Systeme notwendig sind, kurz dargestellt werden, um ein gemeinsames Verständnis aufzubauen.

2.1. Grundlagen der Informationssicherheit

2.1.1. Sicherheit

Der Begriff *Sicherheit* kann in der deutschen Sprache mit unterschiedlichen Konnotationen in Verbindung gebracht werden. In diesem Abschnitt sollen zur Vorbeugung von Missverständnissen die beiden für diese Arbeit relevanten Aspekte von Sicherheit – Verkehrssicherheit und Informationssicherheit – kurz herausgearbeitet werden. Sollte es nicht ausdrücklich anders erwähnt sein, so wird der Begriff *Sicherheit* in dieser Arbeit mit *Informationssicherheit* gleichgesetzt.

2.1.1.1. Verkehrssicherheit

Verkehrssicherheit bezieht sich in dieser Arbeit grundsätzlich auf den Straßenverkehr. Unter dem Begriff ist der Schutz vor Unfällen oder anderen bedrohlichen Situationen für die Unversehrtheit von Verkehrsteilnehmern und Fahrzeugen zu verstehen.

Sicherheit im Straßenverkehr hängt natürlich sehr stark von der Aufmerksamkeit und den Fähigkeiten der Verkehrsteilnehmer ab. Diese können durch aktive Fahrassistenzsysteme dabei unterstützt werden, Fehler zu vermeiden oder die Folgen von Fehlern abzumildern. Denkbar sind hierbei beispielsweise Systeme wie Bremsassistenten, Spurwechselassistenten, Überwachung des toten Winkels und Abstandswarnungen [Lan03, FGF06, Rei10]. Eine weitere Rolle spielen heute verbreitete passive Schutzmechanismen für Insassen bei Unfällen wie Gurtstraffer, Airbags und verstärkte Fahrgastzellen [Lan03, FGF06]. Diese werden in dieser Arbeit jedoch keine Betrachtung finden.

2.1.1.2. Informationssicherheit und mehrseitige Sicherheit

Im Gegensatz zur *Verkehrssicherheit* bezeichnet die *Informationssicherheit* die Sicherheit der innerhalb von oder zwischen Fahrzeugen übertragenen Daten sowie Informationen über Kommunikationsbeziehungen und -teilnehmer. Diese Daten können Angriffen ausgesetzt sein oder auf Grund von Übertragungsfehlern verloren gehen oder verfälscht werden. Die Gefährdung von kritischen Informationen oder Daten kann ihrerseits zu einer geringeren Verkehrssicherheit führen, wenn beispielsweise Fahrzeugführer mit falschen Informationen versorgt werden.

Während bei der Verkehrssicherheit ein grundsätzlicher Konsens über das Schutzbedürfnis aller Teilnehmer besteht, sieht dies bei der Informationssicherheit etwas anders aus. Es gibt durchaus Schutzinteressen von Beteiligten, die in einer konfliktären Beziehung stehen. Beispielsweise hat der Betreiber des *VANET*s ein Interesse daran, die Absender aller Nachrichten zur Ahndung von Missbrauch genau identifizieren zu können. Dies offenbart allerdings möglicherweise sehr viel über die Gewohnheiten und Bewegungsprofile der Teilnehmer, die ihre Privatsphäre schützen möchten.

Der Begriff *mehrseitige Sicherheit* beschreibt genau diese Problematik. Nach [FP97] „umfasst [mehrseitige Sicherheit] die Einbeziehung der Schutzinteressen aller Beteiligten sowie das Austragen der daraus resultierenden Schutzkonflikte beim Entstehen einer Kommunikationsverbindung“. Ziel ist es also nicht, optimalen Schutz für die eine oder andere Partei zu bieten, sondern einen akzeptablen Kompromiss für alle Beteiligten auszuarbeiten. Auch im Rahmen dieser Arbeit sollen stets die Interessen aller Beteiligten im Auge behalten und ein bestmöglicher Schutz unter diesen Rahmenbedingungen erarbeitet werden.

2.1.2. Schutzziele

Der relativ grobe Begriff der Informationssicherheit lässt sich in verschiedene Schutzziele unterteilen, je nachdem, vor welchem Ereignis Daten oder Systeme geschützt

	Schutz gegen beabsichtigte Angriffe		… gegen unbeabsichtigte Ereignisse
	Inhalte	Umstände	
Vertraulichkeit	Schutz der Inhalte und Daten Sicherheit gegen unbefugten Gerätezugriff	Anonymität Unbeobachtbarkeit Unverkettbarkeit Pseudonymität	
Integrität	Übertragungs-integrität Unverfälschtheit	Unabstreitbarkeit / Zurechenbarkeit Abrechnungs-sicherheit	
Verfügbarkeit	Ermöglichen von Kommunikation Verfügbarkeit von Inhalten und Daten Nutzbarkeit von Geräten und Diensten		Funktionssicherheit Technische Sicherheit

Tabelle 2.1.: Schutzziele nach [FP97]

werden sollen. Zum einen existiert eine grundsätzliche Unterscheidung zwischen dem Schutz vor beabsichtigten Angriffen durch Dritte (*Security*) und unbeabsichtigt eintretenden Schadensereignissen wie höhere Gewalt (*Safety*) (vgl. [FP97]). Zum anderen lässt sich jedes Schutzziel in den Schutz von Daten oder Nachrichteninhalten sowie in den Schutz von Informationen über Daten oder Kommunikationsumstände unterteilen (vgl. [FP00]). Bevor die drei Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit im Detail erläutert werden, bietet Tabelle 2.1 nach [FP97] eine Übersicht über die Unterteilung.

2.1.2.1. Vertraulichkeit

Das Schutzziel Vertraulichkeit beschreibt den Schutz von Daten und Informationen sowie Kommunikationsumständen vor unbefugter Kenntnisnahme durch Dritte.

Schutz von Inhaltsdaten Inhalte von Kommunikationen oder gespeicherte Informationen lassen sich in der Regel relativ gut mit Verschlüsselungstechniken schützen. Dabei besitzen nur zur Kenntnisnahme der Daten berechtigte Personen oder Systeme einen kryptographischen Schlüssel, mit dessen Hilfe die Informationen aus einem Schlüsseltext rekonstruiert werden können. Ohne Kenntnis eines passenden Schlüssels ist es bei Verwendung eines ausreichend starken Verschlüsselungssystems nicht möglich, die Daten einzusehen.

Schutz der Kommunikationsumstände Das Schützen von Kommunikationsumständen, das heißt wer, wann, wo und wie lange mit wem kommuniziert, gestaltet sich im Gegensatz zum Schutz der Inhalte meist schwieriger. Bei vielen Nachrichtenübermittlungen müssen zur Zustellung zumindest Informationen über den Empfänger für den oder die Übermittler einsehbar sein. Allerdings können diese Daten auch missbraucht werden, um Teilnehmer zu orten und zu beobachten (Erstellung von Bewegungsprofilen). Hier ist es somit notwendig, Konzepte zu entwickeln, die Kommunikationspartner nicht zu stark exponieren, eine Nachrichtenübermittlung aber dennoch nicht unmöglich machen.

Privatsphäre Unter der Privatsphäre versteht man einen nicht-öffentlichen Bereich, in dem sich ein Mensch frei entfalten kann. Sehen sich Menschen einer ständigen Beobachtung ausgesetzt, so werden sie ihr Verhalten möglicherweise ändern, um bestimmte Erwartungen eines Beobachters zu erfüllen.

Der Gedanke des Konzepts der Privatsphäre gilt als erstmals 1890 von Warren und Brandeis in ihrem Aufsatz im Harvard Law Journal [WB90] explizit formuliert. Der Auslöser für ihre Publikation war nach Bendrath (vgl. [Ben07]) wahrscheinlich der technische Fortschritt, durch den sich Warren in seiner Ungestörtheit beeinträchtigt fühlte: Die Einführung der Kodak-Kamera ermöglichte plötzlich schnelle und einfache Aufnahmen. Dadurch fand der recht bekannte Warren eines Tages Fotos der Hochzeit seiner Tochter in einer Tageszeitung – damals ebenfalls eine Neuerung – wieder.

Dieser Artikel eröffnete den Beginn eines Diskurses, der auch heute noch in allen Bereichen sehr aktuell ist: Wo ist die Grenze zwischen der Überwachung aus einem legitimen öffentlichen Interesse heraus oder zur Sicherstellung des Schutzes der Menschen vor Bedrohungen und dem berechtigten und notwendigen Bedürfnis des Einzelnen, unbeobachtet und ungestört zu handeln? Diese Überlegungen finden vor dem Hintergrund einer zunehmend technisierten, digitalen und vernetzten Gesellschaft statt, in der die Erhebung, Speicherung, Verarbeitung und Weitergabe von Daten einfacher und kostengünstiger wird.

Juristisch wird in §1 (1) des BDSG[1] sowie in der EU-Richtlinie 95/46/EG[2] klargestellt, dass Maßnahmen zum Schutz der Privatsphäre des Einzelnen ergriffen werden müssen. Eine totale Überwachung aller Lebensbereiche ist somit unabhängig von moralischen oder sozialen Aspekten auch juristisch nicht zulässig. Dieser Tatsache muss in modernen Informationssystemen und Netzen Rechnung getragen werden. Auch abseits von gesetzlichen Regelungen ist es jedoch notwendig, die Privatsphäre zu schützen. So wird der Einsatz von neuen Technologien, die den Datenschutz nicht oder nicht ausreichend berücksichtigen, möglicherweise auf Widerstand stoßen, wie in der letzten Zeit an den Beispielen von elektronischen Ausweisdokumenten (vgl. [Hei05]), der elektronischen Gesundheitskarte (vgl. [Hei10c, Opp12]) oder Google Streetview (vgl. [Hei10a, Hei10b]) zu beobachten war.

Personenbezogene Daten, die zur Verwendung in verschiedenen Prozessen (beispielsweise Rechnungslegung) erhoben werden, sollten einerseits in ihrem Umfang möglichst gering gehalten werden (Prinzip der Datensparsamkeit) und andererseits durch technische Maßen einen besonderen Schutz erfahren. Ist der zweite Aspekt nicht gegeben, kann dies zu dramatischen Missbrauchsfällen führen[3]. Vor diesen Gefahren sind die Benutzer eines IT-Systems zu schützen.

Bei der Ausgestaltung eines Ad-hoc-Fahrzeugnetzes sollen genau diese Bedrohungen berücksichtigt werden. Dötzer führt in [Döt05] eine Reihe von auf *VANET*s bezogene beispielhafte Situationen an, die anschaulich zeigen, wie vielschichtig diese Problematik ist:

- Die Polizei überwacht die Kommunikation von Fahrzeugen. Aus ihren Nachrichten kann sie die Geschwindigkeit extrahieren und vollständig automatisch überall selbst die geringfügigste Geschwindigkeitsübertretung ahnden.
- Ein Arbeitgeber beobachtet Fahrzeuge auf dem betriebseigenen Parkplatz und kann nach kurzer Zeit eine Zuordnung zu den Angestellten herstellen. Von diesem Moment an kann er vollständig überwachen, wer sich wann und wo aufhält.
- Ein Privatdetektiv kann ein anderes Fahrzeug anhand von ausgesendeten Nachrichten außerhalb des Sichtbereichs verfolgen.
- Versicherungen können ihre Versicherungsnehmer überwachen und bei den kleinsten Verfehlungen Haftungsausschlüsse durchsetzen.

[1] Bundesdatenschutzgesetz, abrufbar unter http://www.gesetze-im-internet.de/bdsg_1990/index.html

[2] vgl. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:DE:HTML, abgerufen am 10.11.2011

[3] So wurde im Mai 2012 bekannt, dass mehr als 1.000 britische Staatsbedienstete unrechtmäßig aus privater Motivation in staatlich erhobene Daten (u. a. U.K. medial records und social security records) Einsicht genommen haben (siehe [Whi12]).

- Kriminelle Organisationen können die Bewegung von Polizeistreifen verfolgen. Das Gleiche ist ausländischen Geheimdiensten in Bezug auf prominente Persönlichkeiten möglich.

Diese Beispiele sollen aufzeigen, dass es unerlässlich ist, Maßnahmen zu ergreifen, die den Schutz der Privatsphäre sicherstellen.

2.1.2.2. Integrität

Die Integrität umfasst die Unverfälschtheit sowohl von Kommunikationsinhalten und Daten als auch von Kommunikationsumständen. Integritätssichernd wirken also Maßnahmen, die entweder sicherstellen, dass Daten nicht geändert werden können (weder vorsätzlich noch durch Übertragungsfehler) oder es zumindest ermöglichen, dass mindestens eine an der Übertragung beteiligte Partei in der Lage ist, Veränderungen zweifelsfrei als solche zu erkennen.

Unverfälschtheit der Inhaltsdaten Bei der Übertragung von Informationen kann es vorkommen, dass Nachrichteninhalte durch technische Fehler oder aktive Manipulation einer beteiligten Partei verfälscht werden. In vielen Fällen – insbesondere bei verkehrssicherheitskritischen Daten – ist es jedoch unerlässlich, dass sich der Empfänger einer Nachricht auf die Korrektheit der Daten verlassen kann. Es ist somit notwendig, dass diese Informationen beispielsweise mit Hilfe von *Message Authentication Codes* (*MAC*s) geschützt werden.

Zurechenbarkeit/Unabstreitbarkeit Bei der Zurechenbarkeit (oder Unabstreitbarkeit) ist es das Ziel, eine Nachricht einem Sender oder Empfänger zweifelsfrei zuordnen zu können. Dazu müssen die zur Übermittlung der Nachricht genutzten Daten (Informationen über Absender, Empfänger, etc.) korrekt sein. Der entsprechende Kommunikationsteilnehmer kann also nicht abstreiten, der Sender oder Empfänger einer Nachricht gewesen zu sein. Somit bedarf es einer Beweisbarkeit der Integrität der Kommunikationsumstände. Technisch kann dies beispielsweise durch eine digitale Signatur erreicht werden.

2.1.2.3. Verfügbarkeit

Die Verfügbarkeit eines Systems ist gegeben, wenn es in vollem Umfang und fehlerfrei genutzt werden kann, wann immer es benötigt wird. Die Verfügbarkeit kann durch technische Defekte (Software und Hardware), Umweltereignisse (z. B. Stromausfall, Überflutung, ...), Fehlkonfigurationen oder gezielte Angriffe eingeschränkt

werden. Der Schutz vor Angriffen wird im Englischen mit *Security* bezeichnet, die Maßnahmen gegen die anderen Ursachen für Ausfälle dagegen mit *Safety*. Mit Hilfe von Redundanztechniken und Diversität auf der einen und durch fehlertolerante Protokolle und Implementierungen auf der anderen Seite muss eine möglichst hohe Fehlersicherheit geschaffen werden.

2.1.2.4. Abgeleitete Schutzziele

Von den drei beschriebenen Schutzzielen – Vertraulichkeit, Integrität und Verfügbarkeit – lassen sich eine Reihe weiterer Schutzziele ableiten, die ebenfalls in der Literatur häufig Anwendung finden.

Anonymität Die Anonymität bezeichnet die Übertragung von Nachrichten, ohne dass die Kommunikationspartner oder Außenstehende erkennen können, wer kommuniziert. Dazu sind meist aufwändigere technische Lösungen notwendig, da die Zustellung der Nachrichten ja stattfinden, aber keinerlei Rückschlüsse auf die Kommunikationsteilnehmer ermöglichen soll.

Pseudonymität Bei der Pseudonymität treten die Kommunikationspartner unter einer Kennung auf, die zwar als solche wiedererkennbar ist, jedoch keine Rückschlüsse auf die Identität ermöglicht. Es ist somit möglich, einen Teilnehmer wiederzuerkennen, bis er sein Pseudonym gewechselt hat. Da eine echte Anonymität technisch sehr schwierig zu realisieren ist, wird in vielen Systemen Pseudonymität eingesetzt und teilweise fälschlicherweise als Anonymität bezeichnet.

Unverkettbarkeit Die Unverkettbarkeit fordert, dass mehrere Aktionen eines Subjekts nicht miteinander in Verbindung gebracht werden können. Es soll also beispielsweise unmöglich sein, zwei unterschiedliche Nachrichten ein- und demselben Sender oder Empfänger zuordnen zu können.

Rechtssicherheit Die Rechtssicherheit geht noch über die Zurechenbarkeit hinaus: Neben der zweifelsfreien Zuordnung einer Nachricht zu einem Sender oder Empfänger ist auch beweisbar sichergestellt, dass der Nachrichteninhalt nicht verfälscht wurde. Somit wird die Integrität sowohl von Inhalten als auch Kommunikationsumständen gefordert. Auch dies lässt sich technisch mit digitalen Signaturen bewerkstelligen.

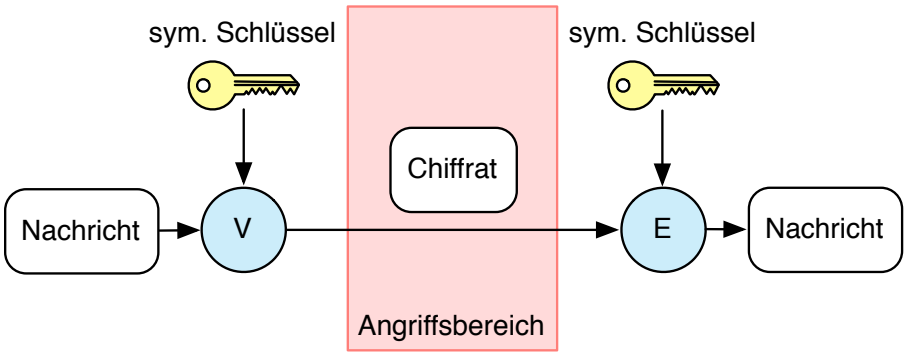


Abbildung 2.1.: Symmetrische Verschlüsselung (Abbildung nach [FP97])

2.1.3. Kryptographie

Beutelspacher et al. beschreiben in [BNS05] die Kryptographie folgendermaßen: „*Kryptographie ist eine öffentliche mathematische Wissenschaft, in der Vertrauen geschaffen, übertragen und erhalten wird.*“ Diese Definition umfasst bereits die beiden Kernbereiche der Kryptographie: Verschlüsselung und Authentikation. Mit Hilfe von Verschlüsselung werden Informationen so verändert, dass sie von jemandem, der nicht über den zur Entschlüsselung nötigen Schlüssel verfügt, nicht verstanden werden können. Authentikation ermöglicht es, nachzuvollziehen, ob Informationen tatsächlich von einem bestimmten Absender (oder einer Gruppe von Absendern) stammen, der Kenntnis über den Schlüssel zur Erzeugung der Authentizitätsinformationen besitzt. Grundsätzlich lassen sich sowohl Verschlüsselungs- als auch Authentikationssysteme mit Hilfe symmetrischer und asymmetrischer Kryptographie realisieren.

2.1.3.1. Symmetrische Kryptographie

Bei symmetrischer Kryptographie besitzen alle Kommunikationspartner eine Kopie des Schlüssels, der für alle Operationen verwendet wird. Dieser muss vor der Kommunikation auf eine sichere Weise ausgetauscht werden. In Frage kommen dafür Medienbrüche oder sichere Schlüsselaustausch- bzw. -vereinbarungsprotokolle wie beispielsweise der Diffie-Hellman-Merkle-Schlüsselaustausch [DH76, Res99].

Bei der symmetrischen Verschlüsselung (vgl. Abbildung 2.1) wird aus einer Nachricht mit Hilfe einer Verschlüsselungsfunktion (V) und unter Einbeziehung des Schlüssels ein Schlüsseltext (Chiffrat) erzeugt, der anschließend über einen unsicheren Kanal gesendet werden kann. Der Empfänger kann nun in seinem Vertrauensbereich mit Hilfe einer Entschlüsselungsfunktion (E) und dem ihm ebenfalls bekannten symmetrischen Schlüssel die Nachricht aus dem Schlüsseltext wiederherstellen.

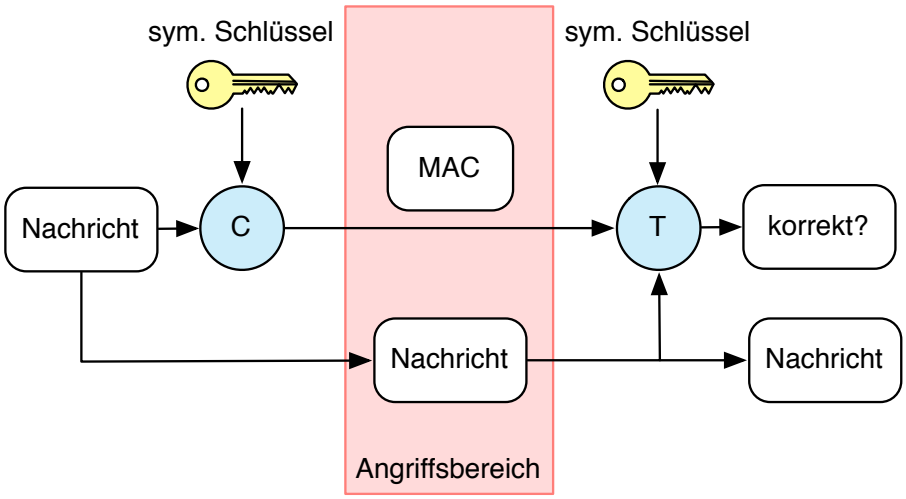


Abbildung 2.2.: Symmetrische Authentikation (Abbildung nach [FP97])

Die Nachricht selbst wird bei modernen Verfahren zur Verschlüsselung meist in Blöcke gleicher Größe aufgeteilt (*Blockchiffren*) und anschließend abgearbeitet. Diese Verfahren sind in der Regel längentreu und basieren meist auf Permutationen und Substitutionen.

Symmetrische Authentikation (vgl. Abbildung 2.2) wird relativ ähnlich realisiert. Anstelle der Verschlüsselung wird jedoch mit Hilfe einer Codierungsfunktion (C) auf Basis der Nachricht und unter Einbeziehung des symmetrischen Schlüssels ein *Message Authentication Code* (*MAC*) erzeugt, der nun zusammen mit der Nachricht über einen unsicheren Kanal übertragen werden kann. Auf der Gegenseite kann der Empfänger in der Testfunktion (T) ebenfalls den *MAC* mit Hilfe der Nachricht und des Schlüssels bilden und vergleichen. Somit erhält er die Information, ob die Nachricht (und der *MAC*) korrekt übermittelt wurden.

Bekannte Vertreter von symmetrischen Kryptosystemen sind der *Data Encryption Standard* (DES) [Nat77], der *Advanced Encryption Standard* (AES) [Fed01], Twofish [SKW+98] und Serpent [BAK98].

2.1.3.2. Asymmetrische Kryptographie

Im Gegensatz zur symmetrischen wird bei der asymmetrischen Kryptographie mit Schlüsselpaaren gearbeitet, die von ihrem Eigentümer erzeugt werden. Der öffentliche Schlüssel (*Public Key*) wird veröffentlicht, der private Schlüssel (*Private Key*, *Secret Key*) muss dagegen geheim gehalten werden. Da der öffentliche Schlüssel (wie es der Name bereits vermuten lässt) keinerlei Geheimhaltung bedarf, ist es somit

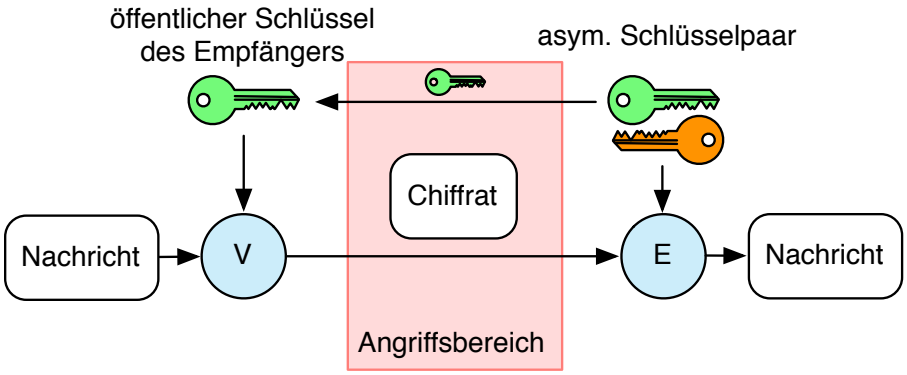


Abbildung 2.3.: Asymmetrische Verschlüsselung (Abbildung nach [FP97])

ein Leichtes, ihn auch über unsichere Kanäle auszutauschen und zur Verfügung zu stellen. Allerdings ist es dann unerlässlich, Maßnahmen zur Authentizitätssicherung des Schlüssels selbst zu treffen, da ansonsten eine Beziehung zwischen einer Person oder einem System und einem Schlüssel nicht zweifelsfrei festgestellt werden kann. Zu diesem Zweck werden *PKI*s (*Public Key Infrastructures*) eingesetzt.

Eine Certification Authority (*CA*) einer solchen PKI-Infrastruktur bestätigt mit einer kryptographischen Signatur die Echtheit des Schlüssels und damit dessen Zugehörigkeit zu einer Person oder einem System. Nun muss ein Einzelner nur noch dieser *CA* vertrauen und kann in der Folge sicher sein, dass Schlüssel, die die Unterschrift dieser *CA* tragen, authentisch sind.

Für die Verschlüsselung einer Nachricht in einem asymmetrischen Kryptosystem (vgl. Abbildung 2.3) benötigt der Sender den öffentlichen Schlüssel des Empfängers. Nach Anwendung der Verschlüsselungsoperation (V) ist auch der Absender nicht mehr in der Lage, die Nachricht zu entschlüsseln – dies kann nur noch der Empfänger mit Hilfe des geheimen Schlüssels und der Entschlüsselungsoperation (E). Die Nachricht kann somit über ein unsicheres Medium übertragen werden.

Bei asymmetrischen Authentikationssystemen spricht man in der Regel von digitalen Signatursystemen (vgl. Abbildung 2.4), da sie dem Prinzip einer Unterschrift bzw. dem mittelalterlichen Siegeln von Dokumenten relativ nahe kommen. Der Absender einer Nachricht erzeugt mit der Signaturfunktion (S) und seinem privaten Schlüssel eine Signatur über der Nachricht. Diese kann er nun zusammen mit der Nachricht (und evtl. seinem öffentlichen Schlüssel) über einen unsicheren Kanal senden. Der Empfänger kann mit Hilfe des öffentlichen Schlüssels die Signatur in einer Testfunktion (T) überprüfen und erfährt so, ob diese gültig und die Nachricht somit unverfälscht ist und tatsächlich von dem Absender stammt.

Bekannte Vertreter der asymmetrischen Kryptographie sind das RSA-Kryptosystem

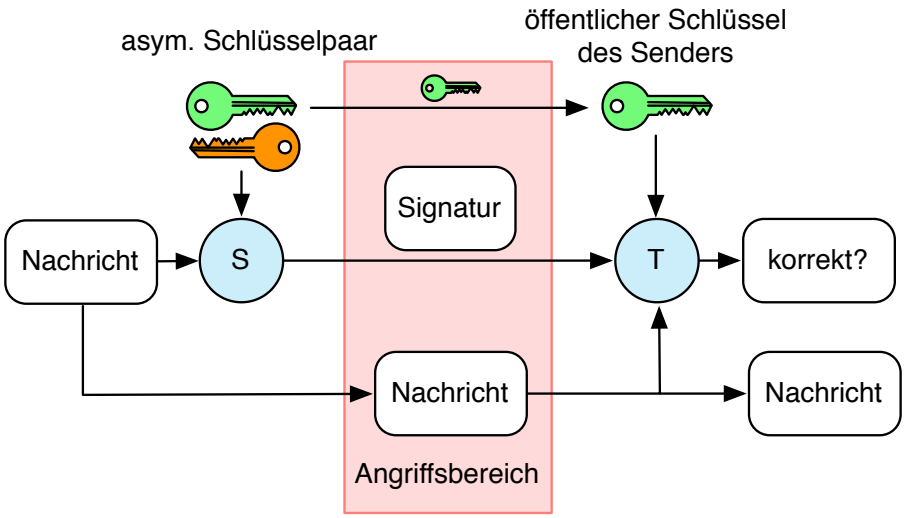


Abbildung 2.4.: Digitale Signatur (Abbildung nach [FP97])

(nach den Erfindern Rivest, Shamir und Adleman) [RSA78], das ElGamal-Kryptosystem (benannt nach seinem Entwickler Taher Elgamal) [Elg85] und die Elliptische-Kurven-Kryptographie (Elliptic Curve Cryptography, ECC) [Mil85].

2.1.3.3. Performance kryptographischer Systeme

Asymmetrische Kryptosysteme bieten den großen Vorteil des einfacheren Schlüsselaustauschs und der einfacheren Schlüsselverwaltung gegenüber symmetrischen Systemen. Dennoch besitzen asymmetrische Systeme einen gravierenden Nachteil: Auf Grund ihrer komplexen Mathematik benötigen alle Operationen wesentlich mehr Zeit als vergleichbar sichere symmetrischen Verfahren. Federrath und Pfitzmann schätzen in [FP97] die 100- bis 10 000-fache Dauer für vergleichbare Operationen. Plößl kann einen deutlichen Performanceunterschied zwischen symmetrischen und asymmetrischen Verfahren in [Plö09] bestätigen, wobei sich zeigt, dass es selbst zwischen unterschiedlichen Bibliotheken, die den gleichen Algorithmus implementieren, deutliche Unterschiede in der Verarbeitungsgeschwindigkeit gibt.

Im Vergleich asymmetrischer Verfahren untereinander zeigt sich bereits bei den Schlüssellängen von ECC ein deutlicher Vorteil gegenüber anderen Verfahren: 160 Bit gelten heute als ausreichend und bieten eine vergleichbare Sicherheit wie 1024 Bit bei RSA oder Elgamal. Damit liegen die Schlüssellängen in der Größenordnung der symmetrischen Verfahren, bei denen 128 Bit (z. B. bei AES) derzeit als ausreichend betrachtet werden. Alle mathematischen Operationen können damit auf kleineren Zahlen durchgeführt werden und benötigen somit weniger Speicher und

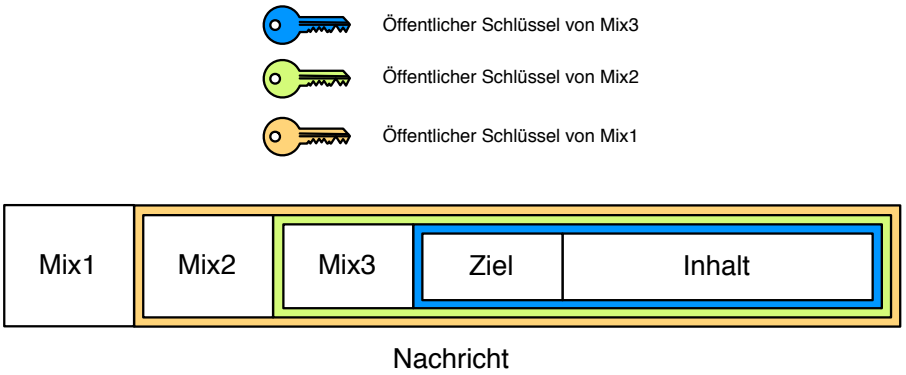


Abbildung 2.5.: Prinzip von Kommunikationsmixen (1)

Energie. So kristallisiert sich ECC als das performanteste asymmetrische Verfahren heraus (vgl. [KGP+09]). Dennoch liegt auch hier die Performanz weiterhin in einer deutlich schlechteren Größenordnung als symmetrische Verfahren.

Da die Rechenkapazität der kryptographischen Einheiten der Fahrzeuge in *VANET*s sehr beschränkt ist und zentrale Instanzen zwar über deutlich mehr Rechenleistung verfügen, möglicherweise aber eine sehr große Anzahl an Nachrichten bearbeiten müssen, ist zu berücksichtigen, dass symmetrische Kryptographie asymmetrischen Verfahren vorgezogen werden sollte, wo immer dies möglich ist.

2.1.4. Kommunikationsmixe

David Chaum hat mit seinem Verfahren der umkodierenden Mixe [Cha81] den Grundstein für eine heute in der Forschung weit verbreitete Möglichkeit zur anonymen Kommunikation gelegt. Die von ihm ursprünglich vorgeschlagene Technik beinhaltete schon viele wesentliche Aspekte, wurde aber dennoch über die Jahrzehnte von vielen Forschern analysiert (vgl. u. a. [PP89, BDD+04]), verfeinert (vgl. u. a. [FJP96]), erweitert (vgl. u. a. [DS03, KEB98]) und neuen Anwendungen (Chaum schlägt sie für E-Mail-Kommunikation vor) zugeführt (vgl. u. a. [PPW91, Fed99b, BFK00]).

Das auf asymmetrischer Kryptographie basierende Konzept ermöglicht einen anonymen Nachrichtenaustausch über unsichere Netze. Dazu wird eine Nachricht mehrfach mit den öffentlichen Schlüsseln der speziellen Knoten des Netzes (*Mixe*) verschlüsselt (vgl. Abbildung 2.5). Der gesamte verschlüsselte Block der Nachricht für einen Mix wird als Datenfeld der Nachricht des nächsten Verschlüsselungsschrittes betrachtet. Somit entsteht ein zwiebelschalenähnliches Prinzip einer mehrschichtigen (im Beispiel dreischichtigen) Verschlüsselung.

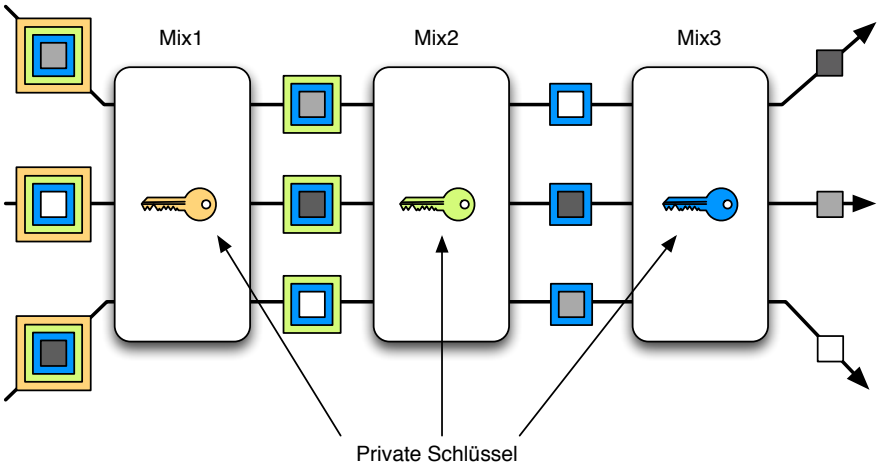


Abbildung 2.6.: Prinzip von Kommunikationsmixen (2)

Die mehrfach verschlüsselte Nachricht wird an den ersten Mix übermittelt. Dieser ist in der Lage – da er den passenden privaten Schlüssel zu dem zur Verschlüsselung verwendeten öffentlichen Schlüssel besitzt – die äußerste Verschlüsselungsschicht zu entfernen und die darunterliegende Nachricht zu erhalten (vgl. Abbildung 2.6). Diese kann an den nächsten Mix (für den die nun äußerste Verschlüsselungsschicht erzeugt wurde) weitergeleitet werden, der wiederum einen Entschlüsselungsvorgang durchführen kann. Nachdem der letzte Mix die letzte Verschlüsselung entfernt hat, kann er – da die Nachricht nun im Klartext vorliegt – diese an den eigentlichen Empfänger zustellen.

Es ist jedoch nicht die einzige Aufgabe der Mixe, eine Schicht der eingehenden Nachrichten zu entschlüsseln und sie anschließend weiterzuleiten. Damit die Verfolgung einer bestimmten Nachricht für einen Beobachter schwierig oder gar unmöglich wird, sind einige weitere Handlungen nötig.

So verarbeiten Mixe grundsätzlich nicht nur eine Nachricht, sondern warten (mit unterschiedlichen Strategien) darauf, dass eine Reihe von Nachrichten gleichzeitig vorliegt. Diese Nachrichten werden neben den Entschlüsselungsvorgängen auch durchmischt – gemixt –, bevor sie ausgegeben werden (vgl. Abbildung 2.6). Ein Beobachter kann also weder anhand der Reihenfolge der Nachrichten (Eingangs- und Ausgangsreihenfolge unterscheiden sich), noch an dem Aussehen der Nachrichten (dieses hat sich durch das einmalige Entschlüsseln verändert) erkennen, welche die von ihm gesuchte ist. Lediglich der einzelne Mix weiß, welche Ausgangs- einer bestimmten Eingangsnachricht entspricht.

Nur, wenn alle Mixe, die eine Nachricht bearbeitet haben, zusammenarbeiten, ist es möglich, die Bewegung dieser aufzudecken. Verhält sich nur ein einziger Mix vertrau-

enswürdig (er gibt also die bei ihm stattgefundenen Zuordnungen nicht preis), so bietet dieses System Schutz. Daraus resultiert auch die übliche Vertrauensannahme bei Mix-Systemen: Wenigstens ein beteiligter Mix arbeitet nicht mit den anderen zusammen. Ist dies erfüllt, so kann ein Beobachter, der alle Kommunikationsverbindungen überwachen kann, selbst mit Hilfe aller anderen Mixe keine Zuordnung herstellen und die Kommunikationsverbindung zwischen Sender und Empfänger nicht aufdecken.

2.2. Grundlagen der mobilen Netze

2.2.1. Drahtlose Kommunikation

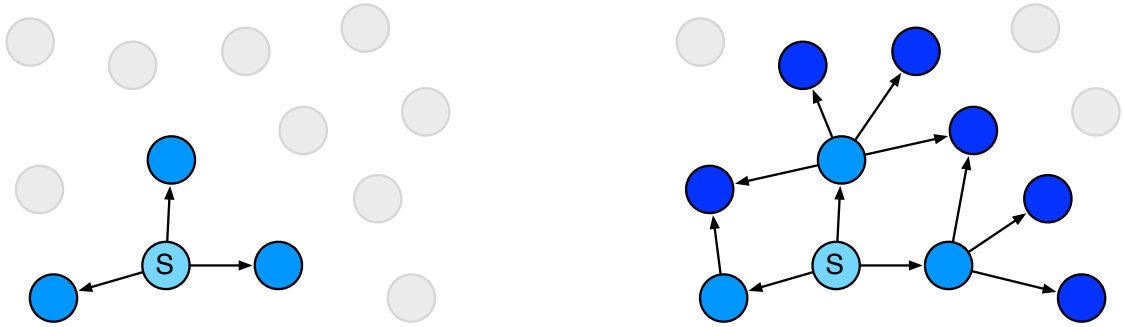
Kabelgebundene Kommunikation ist in vielerlei Hinsicht relativ einfach zu überblicken: Signale fließen durch einen Leiter und somit klar definiert von einem Ende des Kabels zu dem anderen. Alle an diesen Leiter angeschlossenen Geräte können die Informationen zur Kenntnis nehmen und gegebenenfalls an über andere Kabel verbundene Geräte weiterleiten. Die Ausbreitung von Nachrichten ist somit klar festgelegt; die leichte elektromagnetische Abstrahlung, die stromdurchflossene Kabel aufweisen, sei hier unberücksichtigt.

Anders stellt sich der Sachverhalt bei drahtloser Kommunikation dar: Nachrichten werden grundsätzlich auf ein gemeinsames Medium, den Raum zwischen den Geräten, gesendet. Die Signalstärke der elektromagnetischen Wellen nimmt dabei in Abhängigkeit der Ausbreitungsdistanz ab. Damit ist es grundsätzlich jedem Gerät, das sich in Sendereichweite befindet, möglich, jede über die Luftschnittstelle übertragene Nachricht zu empfangen. Soll eine größere Distanz als die Sendereichweite der Datenquelle überbrückt werden, so ist eine erneute Aussendung derselben Nachricht durch näher am Ziel gelegene Geräte, die die ursprüngliche Nachricht aber empfangen konnten, notwendig. Auf diese Weise können in einem Netz aus kooperierenden Knoten auch größere Distanzen überbrückt werden, solange die Knotendichte ausreichend ist.

Die Begrifflichkeiten dieser drahtlosen Kommunikation sollen in diesem Abschnitt geklärt werden.

2.2.1.1. Übertragungsweite

Wird eine Nachricht über die Luftschnittstelle übertragen, hat – wie bereits dargelegt – jeder Knoten in Sendereichweite die Möglichkeit, die Nachricht zur Kenntnis zu nehmen, was ja bereits einer Übertragung entspricht; Kommunikation hat somit



(a) Singlehop **(b)** 2-Hop

Abbildung 2.7.: Empfänger bei Single- und Multihop-Broadcast-Kommunikation

stattgefunden. Die Nachricht wurde einen Schritt, einen *Hop*, weit übertragen und man spricht von einer *Singlehop*-Kommunikation (vgl. Abbildung 2.7a).

Soll eine Nachricht jedoch größere Strecken überwinden, ist es notwendig, dass sie durch günstig positionierte Knoten erneut ausgesendet wird. Jede erneute Aussendung wird dabei als *Hop* betrachtet und eine Übertragung über mehrere Knoten somit als *Multihop*-Kommunikation bezeichnet. Stehen zwei Knoten auf diese Weise über n Schritte miteinander in Verbindung, so spricht man von einer n-*Hop*-Nachbarschaft (vgl. Abbildung 2.7b).

Um Kontrolle über die Ausbreitungsdistanz einer Nachricht zu bewahren, kommt die sogenannte *Time-to-live* (TTL) zum Einsatz. Dies ist ein Wert, der in der Regel im Nachrichtenkopf mitübertragen wird und anzeigt, über wie viele *Hops* die entsprechende Nachricht übermittelt werden soll oder darf. Jeder Knoten, der diese Nachricht erneut aussendet, reduziert dabei die TTL um 1; erreicht dieser den Wert 0, so wird die Nachricht nicht weiter gesendet.

2.2.1.2. Adressaten

Da die Kommunikation – wie soeben beschrieben – über die Luftschnittstelle erfolgt und weitergeleitet wird, ist es jedem Knoten in Reichweite möglich, Nachrichten zur Kenntnis zu nehmen. Kommunikation findet also auf einem Broadcastmedium statt. Dennoch haben alle Knoten ein Interesse daran, eben nicht alle empfangbaren Nachrichten zur Kenntnis nehmen zu müssen und erst anhand des Inhalts zu erkennen, ob sie zu Adressaten dieser Nachricht bestimmt wurden. Auf einer logischen Ebene wird somit die Broadcast-Eigenschaft des Mediums aufgebrochen. Sollte diese Filterung nicht stattfinden, könnten eine Vielzahl an irrelevanten Nachrichten die Leistung des empfangenden Knotens beeinträchtigen. Daher werden Unterscheidungsmerkmale –

wie Adressen oder Kennungen der Empfänger – im Kopfbereich der Nachrichten verwendet. Die unterschiedlichen Möglichkeiten, Nachrichten zu adressieren sollen im Folgenden ebenso untersucht werden wie die dabei verwendeten Begrifflichkeiten. Dabei handelt es sich stets um logische Beziehungen, da die physikalischen Eigenschaften des Mediums immer eine faktische Broadcast-Kommunikation erzwingen. Diese Betrachtungen sind unabhängig von der Übermittlungsweite der Nachrichten; diese kann von 1 (*Singlehop*) bis n reichen. Die Beschreibungen basieren auf den Quellen [Tan03, KR08, NS10, Mai04].

Broadcast Es gibt Situationen, in denen gerade kein expliziter Empfänger gewünscht ist, da eine Nachricht für alle Knoten, die diese empfangen können, relevant ist bzw. von ihnen zur Kenntnis genommen werden soll. Nach Tanenbaum ist somit das Broadcasting „das gleichzeitige Senden eines Pakets an alle Ziele“ [Tan03]. Als Beispiel der analogen Welt kann der mittelalterliche Marktschreier herangezogen werden: Dieser rief Informationen laut aus und konnte (je nach Kräftigkeit seiner Stimme) von allen Menschen in seinem Umfeld gehört werden. In der Informationstechnik funktioniert dies in der Regel mit einer speziellen Adresse, die als Broadcastadresse bezeichnet wird. Auf diese Weise adressierte Pakete werden von allen empfangenden Knoten verarbeitet.

Unicast Die vermutlich intuitivste Form der Nachrichtenübertragung ist die Kommunikation zwischen zwei Teilnehmern – einem Sender und einem Empfänger. Dies ist in der analogen Welt zum Beispiel bei klassischen Briefsendungen oder Telefongesprächen üblich. Bei der Kommunikation über die Luftschnittstelle wird zu diesem Zweck die zu sendende Nachricht an den Empfänger adressiert und anschließend über Routingmechanismen (vgl. Abschnitt 2.2.3) zugestellt. In Bezug auf die Kommunikationssicherheit ist nun zu berücksichtigen, dass jeder Knoten in Reichweite des Senders oder eines weiterleitenden Knotens diese Nachricht zur Kenntnis nehmen kann.

Multicast Soll eine Nachricht an mehrere Teilnehmer übermittelt werden, so ließe sich das natürlich mit entsprechend vielen Unicasts realisieren. Effizienter ist es jedoch, wenn man ein und dieselbe Nachricht an die gewünschten Empfänger adressiert und anschließend sendet, da man auf diese Weise eventuell Übertragungskapazität einsparen kann. Die Multicast-Adressierung ist nicht mehr trivial möglich: Entweder benötigt man eine definierte Liste von Empfängern (bzw. ihrer Adressen), die zusammengestellt und möglicherweise aktuell gehalten werden muss, oder die Empfänger schließen sich zu einer Gruppe zusammen und erhalten eine gemeinsame Gruppenadresse, an die eine Nachricht versendet werden kann.

Anycast Die Kommunikationsform Anycast stellt eine Besonderheit dar: Einerseits kontaktiert man ähnlich dem Multicast eine Reihe von Knoten, andererseits erwartet man nur eine einzige Rückantwort von einem beliebigen Mitglied dieser Anycast-Gruppe. Im RFC4291 (IPv6-Architektur) [HD06] wird der Anycast ebenfalls definiert; dabei ist das Ziel, den – je nach Metrik des Routingalgorithmus (vgl. Abschnitt 2.2.3) – nächst gelegenen Knoten mit der gegebenen Anycast-Adresse zu kontaktieren. Anycast kann also dann sinnvoll genutzt werden, wenn man einen beliebigen Knoten, der jedoch bestimmte definierte Eigenschaften aufweist oder Dienste erbringt und dadurch Teil der jeweiligen Anycast-Gruppe werden konnte, möglichst effizient (z. B. in möglichst kurzer Zeit, mit möglichst wenigen Hops, etc.) kontaktieren möchte. Anwendungsbeispiele sind Erbringer einer bestimmten Dienstleistung, wie beispielsweise Aktualisierungen von Informationen wie Staumeldungen oder Kartenmaterial. Wenn diese Informationen auf mehreren Knoten gleichartig vorgehalten werden, ist es irrelevant, von welchem dieser Knoten man die Daten bezieht. Die Adressierung erfolgt, wie bereits angedeutet, ähnlich dem Multicast über eine bestimmte Adresse, die eine Gruppe von Knoten beschreibt. Jeder der adressieren Knoten, der eine Anycast-Anfrage erhält, kann darauf antworten. In der Regel wird nur die erste Antwort, die bei dem Absender eintrifft, berücksichtigt.

Geocast: GeoBroadcast bzw. GeoMulticast In mobilen Netzen ist es in verschiedenen Situationen nützlich, Nachrichten an alle Knoten an einem bestimmten Ort beziehungsweise in einer bestimmten Region zu übermitteln. Dabei ist unerheblich, *wer* der Empfänger ist; entscheidend ist nur, *wo* sich dieser aufhält. Diese Form der Nachrichtenübertragung wird als Geocast bezeichnet, was laut [CAR07] eine Kurzform für *geographical broadcast*, also ein auf eine bestimmte geographische Region beschränkter Broadcast, ist. Andere Autoren (z. B. [KV99]) sehen den Geocast als eine Form des Multicast, dessen Gruppenzugehörigkeit sich durch ihren Aufenthaltsort bestimmt. Diese beiden Auffassungen stehen nicht wirklich in Konkurrenz, da sie das inhaltliche Verständnis teilen und lediglich eine andere Begrifflichkeit verwenden. Adressiert wird jedenfalls in beiden Fällen über eine geographische Referenz. Dies können beispielsweise GPS- oder GALILEO-Koordinaten sein oder jede andere vorab im Konsens definierte Referenzierung.

GeoAnycast Analog zu Geocast als GeoMulicast wird bei GeoAnycast (vgl. [CAR07]) eine Region adressiert, in der sich beliebige, nicht bekannte Knoten aufhalten können. Der Absender einer GeoAnycast-Nachricht in eine solche Region erwartet die Rückantwort von lediglich einem beliebigen Knoten im adressierten Bereich.

2.2.1.3. Adressierung

Damit eine Nachricht an den intendierten Empfänger übermittelt werden kann, muss sie mit einem Merkmal versehen werden, mit dessen Hilfe der Adressat erkennt, dass eine Nachricht für ihn bestimmt ist. Dieses Merkmal – im Folgenden als *Adresse* bezeichnet – kann nach Pfitzmann et al. [PPW88, Pfi93] unterschiedlich ausgestaltet werden:

Explizite Adressierung Hierbei wird auf ein Merkmal zurückgegriffen, das nicht nur durch den Empfänger genutzt wird, um an ihn adressierte Nachrichten zu erkennen. Vielmehr beschreibt das Merkmal den Empfänger auch für andere Knoten erkennbar. Somit kann die explizite Adresse auch verwendet werden, um Nachrichten in einem Netz über mehrere Knoten hinweg zuzustellen. Ein Beispiel für eine explizite Adresse wäre die IP-Adresse, die neben ihrer eindeutigen Zuordenbarkeit zu einem Knoten auch dessen Position im Netz abbildet.

Implizite Adressierung Im Gegensatz zu der expliziten Adressierung ermöglicht eine implizite Adresse nur dem empfangenden Knoten zu erkennen, dass er der Adressat ist. Grundsätzlich lassen sich *offene implizite Adressen* und *verdeckte implizite Adressen* unterscheiden. Offene implizite Adressen können über eine beliebige Folge von Bits realisiert werden, die zwischen Sender und Empfänger vereinbart wurde. Diese lassen Außenstehenden grundsätzlich keine Rückschlüsse auf den Adressaten zu. Allerdings ist es möglich, durch Beobachtung mehrerer Nachrichten mit der selben Adresse eine Verkettung herzustellen. Daher muss eine offene implizite Adresse regelmäßig gewechselt werden.

Die verdeckte implizite Adresse lässt dagegen für Außenstehende keine Rückschlüsse auf den Empfänger zu. Realisiert wird dies meist mit Hilfe von asymmetrischer Kryptographie (vgl. Abschnitt 2.1.3.2)[4]. Die Nachricht wird dabei um eine Redundanz, also eine zwischen dem Absender und Empfänger (oder global) vereinbarte Bitfolge, ergänzt und asymmetrisch mit dem öffentlichen Verschlüsselungsschlüssel des Empfängers verschlüsselt und gesendet. Jeder Knoten, der diese Nachricht empfängt, entschlüsselt diese probeweise mit dem eigenen geheimen Schlüssel und überprüft, ob die Redundanz korrekt entschlüsselt wurde. Dies ist nur der Fall, wenn der zu dem zur Verschlüsselung verwendeten öffentliche Schlüssel passende private Schlüssel zur Entschlüsselung genutzt wurde, also der legitime Empfänger der

[4] Der Einsatz eines symmetrischen Verfahrens ist ebenfalls denkbar. In diesem Fall wird der gemeinsame symmetrische Schlüssel der Kommunikationspartner zur Verschlüsselung verwendet. Jeder Empfänger der Nachricht muss nun probeweise mit allen Schlüsseln der potentiellen Sender entschlüsseln.

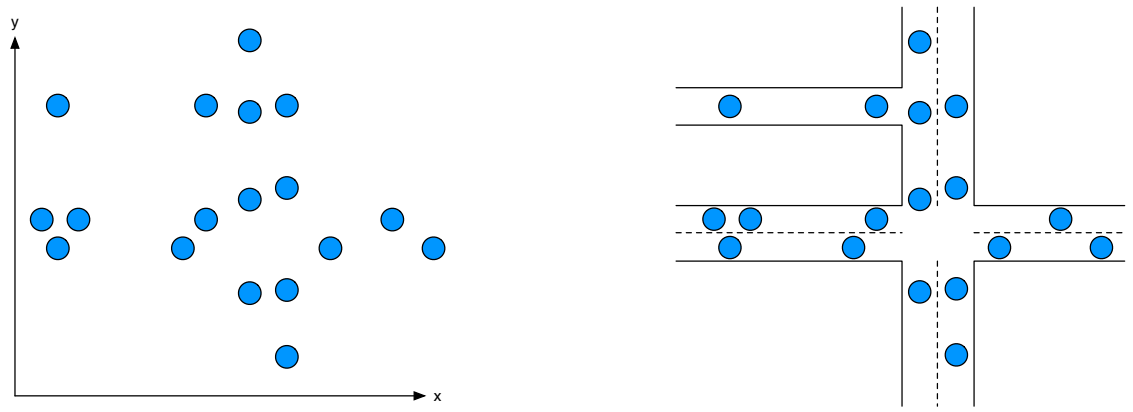
Nachricht die Operation ausgeführt hat. Diese kann die Nachricht nun zur Kenntnis nehmen; alle anderen Knoten, die nicht erfolgreich entschlüsseln konnten, verwerfen die Nachricht.

2.2.2. Positions- und Zeitdienste

Für viele Dienste in mobilen Netzen sowie für einige Verfahren für den Nachrichtentransport (vgl. Abschnitt 2.2.3) ist es erforderlich, dass sich die Knoten ihrer globalen Position bewusst sind (*position* bzw. *spatial awareness*). Mao et al. liefern in [MFA07] einen grundsätzlichen Überblick über die Techniken der Positionsbestimmung. Dabei kann neben Basisstationen und anderen Teilnehmern (u. a. diskutiert und vorgestellt in [CH06, LP06, RCC07, CRCS08]) auch ein GNSS (*Global Navigation Satellite System*) wie das US-amerikanische *NAVSTAR GPS* oder in naher Zukunft das europäische *GALILEO* genutzt werden. Der Argumentation von Plößl in [Plö09] folgend bietet sich die Verwendung des letzteren an (soweit es die Verfügbarkeit erlaubt), da es als neuestes System eine Reihe von Sicherheitsproblemen der älteren Systeme vermeidet (vgl. [Gal03]). Zu den wesentlichen Neuerungen zählen zudem die Einführung von Notfalldiensten, die eine sehr hohe Präzision für sicherheitsrelevante Anwendungen bieten, und die Authentikation der Positionssignale, die durch digitale Signaturen eine Zurechenbarkeit der Signale zu den legitimen Satelliten ermöglichen. Dies soll dazu führen, dass der Einsatz von Sendern mit gefälschten Signalen zur Irreführung von Empfängergeräten (beispielsweise beschrieben in [Vol01]) nicht mehr möglich ist. Problematisch ist jedoch der Einsatz von GNSS im Zusammenhang mit Unterführungen, Einhausungen und Tunnels, da hier das Signal in der Regel abgeschottet ist und eine Positionsbestimmung unmöglich wird. In diesen Zonen sollten zusätzliche Basisstationen zum Einsatz kommen, die die Aufgabe der nicht erreichbaren Satelliten übernehmen.

Die Untersuchung der Sicherheitsfunktionalität von und der Angriffe auf GNSS würde den Rahmen dieser Arbeit sprengen. Daher wird im Folgenden von der Annahme ausgegangen, dass jeder Knoten eines *VANET*s mit Hilfe eines GNSS seine Position jederzeit hinreichend genau bestimmen kann und dass die von Satelliten empfangenen Positionssignale authentisch sind. Die Sicherheitssysteme von *GPS* und *GALILEO* sowie Angriffe auf GNSS können unter anderen in folgenden Quellen nachgelesen werden: [Vol01, LG03, CKQ03, PWK05, LB06].

Nach Tian und Coletti [TC03] kann mit Positionsdiensten alleine lediglich *position awareness* erreicht werden, da *spatial awareness* zusätzlich zur Kenntnis der eigenen Position Wissen über die Umwelt (konkret im *VANET*: den Straßenverlauf) mit einschließt (vgl. Abbildung 2.8). Dieses umfassendere Wissen kann wiederum genutzt werden, um die Übermittlung von Nachrichten optimaler zu gestalten, allerdings



(a) Position awareness **(b)** Spatial awareness

Abbildung 2.8.: Position (a) vs. spatial (b) awareness nach [TC03]

benötigt es dazu geeignetes Kartenmaterial. Da bereits seit einigen Jahren ausreichend genaues Kartenmaterial in Navigationssystemen vertrieben wird, kann diese Voraussetzung ebenfalls als erfüllt betrachtet werden.

Einige Anwendungen im *VANET* erfordern den Einsatz von synchronen Uhren bei den Knoten und gegebenenfalls in der Infrastruktur (*temporal awareness*). Der Einbau von Atomuhren in Fahrzeuge wäre zu teuer und ist beim Einsatz eines GNSS unnötig, da Satelliten selbst als Zeitgeber arbeiten. Dieses Signal wird mit hochgenauen Atomuhren erzeugt; die Abweichung von einer idealen Uhr liegt pro Tag im Nanosekundenbereich [Eur05]. Mit Hilfe eines Zeitsignals eines GNSS kann also davon ausgegangen werden, dass alle Knoten des *VANET*s über hinreichend genaue und synchrone Uhren verfügen.

2.2.3. Routing

Routing, also das Finden eines Weges, kann im Zusammenhang mit *VANET*s zwei ganz unterschiedliche Bedeutungen haben: Der Begriff kann sich auf das Finden einer Strecke für Fahrzeuge von einem Ausgangs- zu ihrem Zielort beziehen oder den Transport von Nachrichten von einem Sender zu einem Empfänger innerhalb des *VANET*s beschreiben. Die erstgenannte Bedeutung ist allerdings nicht Gegenstand dieser Arbeit, und somit wird im Folgenden der Begriff *Routing* ausschließlich im Kontext des digitalen Nachrichtentransports verwendet.

In Computernetzen mit statischen oder sich langsam ändernden Topologien (wie beispielsweise dem Internet) müssen sich Routingalgorithmen – also das Vorgehen, mit dem ein Nachrichtentransport stattfindet – nur bis zu einem gewissen Grad

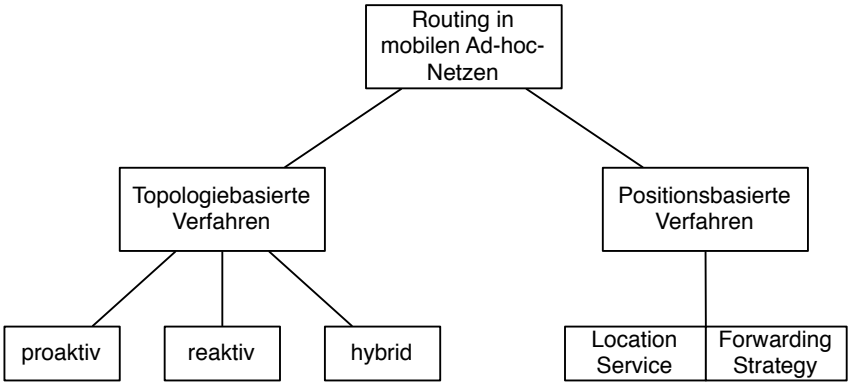


Abbildung 2.9.: Einordnung von Routingverfahren in *MANET*s nach [MWH01]

an einen veränderten Aufbau des Netzes anpassen können. In hochvolatilen Netzen wie *MANET*s oder *VANET*s können diese Mechanismen jedoch auf Grund ihrer langsamen Adaption auf neue Gegebenheiten nicht eingesetzt werden, so dass neue Techniken zur Ermittlung eines Übertragungsweges gefunden werden müssen.

Mauve et al. unterscheiden in [MWH01] wie in Abbildung 2.9 dargestellt zwei grundsätzliche Techniken für Routing in mobilen Netzen – topologiebasiertes und positionsbasiertes Routing. Erstere lassen sich wiederum in *proaktive*, *reaktive* und *hybride* Verfahren unterteilen; letztere setzen sich dagegen aus einem *Location Service* und einer *Forwarding Strategy* zusammen.

Topologiebasierte Verfahren Bei topologiebasierten Routingmechanismen in *MANET*s erfolgt die Entscheidung, über welchen Weg eine Nachricht von ihrem Sender an den Empfänger transportiert wird, auf Basis der Struktur – der Topologie – des Netzes. Da sich diese schnell verändert, werden bei *proaktiven Verfahren* (z. B. *Destination-Sequence Distance Vector Routing* (*DSDV*) [PB94]) ständig Nachrichten über die aktuellen Positionen aller Teilnehmer ausgetauscht. Somit verfügt jeder Knoten über die Kenntnis der Struktur des Netzes und wählt eine – nach den Kriterien des jeweiligen Verfahrens – günstige Route aus. Dieses Konzept erzeugt allerdings durch die ständigen Informationen über die Positionen aller Teilnehmer einen enormen Overhead und erscheint in großen Anwendungen nicht praktikabel. *Reaktive Verfahren* (z. B. *Dynamic Source Routing* (*DSR*) [JM96], *Ad-hoc On-Demand Distance Vector Routing* (*AODV*) [PR99]) versuchen dieses Problem damit zu umgehen, dass ein Knoten vor dem Senden einer Nachricht mit Hilfe einer sogenannten *Route Discovery*-Nachricht versucht, eine intakte Route zu dem Empfänger der eigentlichen Nachricht zu etablieren. Der Overhead kann somit stark reduziert werden, jedoch kann das Suchen einer Route relativ viel Zeit in Anspruch nehmen. Daher möchten *hybride Verfahren* (z. B. *Zone Routing Protocol* (*ZRP*) [HP01]) die Vorteile

beider Konzepte verbinden, wobei die Annahme zu Grunde liegt, dass der überwiegende Teil der Kommunikation nur über kurze Distanzen stattfindet. Im Nahbereich werden die Routen stets durch proaktive Verfahren aktuell gehalten, für Kontakte zu entfernten Teilnehmern werden mit Hilfe reaktiver Verfahren Routen lediglich bei Bedarf gesucht.

Insgesamt zeigen sich jedoch topologiebasierte Verfahren als zu schlecht skalierbar für die große Anzahl an erwarteten Knoten in *VANET*s und als zu träge, um auf die schnellen Veränderungen der Topologie in mobilen Ad-hoc-Netzen reagieren zu können (vgl. [MWH01, FMH+02]). Dies führte zu der Entwicklung von positionsbasierten Routingverfahren.

Positionsbasierte Verfahren Positionsbasierte Routingverfahren (teilweise auch als *Geographical Routing* [FFBA07] oder *Location Aware Routing* [TWLC01] bezeichnet) arbeiten nach einem grundsätzlich anderen Prinzip als topologiebasierte Mechanismen. Eine Kenntnis über die Topologie des Netzgraphen ist dabei nicht notwendig, jedoch muss jedes Fahrzeug über zumindest *position awareness* (vgl. Abschnitt 2.2.2) verfügen. Jeder positionsbasierte Mechanismus besteht aus zwei Komponenten: einem Positionsdienst (*location service*) und einer Weiterleitungsstrategie (*forwarding strategy*) [MWH01].

Jeder Knoten teilt dem Positionsdienst regelmäßig seine aktuelle Position mit und kann von diesem Moment an kontaktiert werden. Ein Knoten A, der an einen anderen Knoten B eine Nachricht senden möchte, kontaktiert zuerst den Positionsdienst und erfragt den aktuellen Aufenthaltsort von Knoten B. Anschließend wird die eigentliche Nachricht mit Hilfe der Weiterleitungsstrategie an die erhaltene Position übermittelt und dort an B zugestellt.

2.2.3.1. Positionsdienste

In der Literatur findet sich eine große Spannbreite an Konzepten für Positionsdienste. Diese lassen sich in infrastrukturabhängige und infrastrukturlose Dienste unterteilen. Bei infrastrukturabhängigen Positionsdiensten sind die Positionsinformationen in einer zentralen oder mehreren verteilten Datenbanken außerhalb des Ad-hoc-Bereichs in einem statischen Teil des Netzes hinterlegt. Fahrzeuge kommunizieren mit diesem Backbone über RSUs. Infrastrukturlose Positionsdienste kommen dagegen – wie der Name bereits vermuten lässt – ohne Infrastruktur aus und verlegen die Funktionalität des Dienstes komplett in das eigentliche Ad-hoc-Netz. Dies führt dazu, dass Teilnehmer die Positionsinformationen für andere vorhalten.

Die Ausgestaltungsmöglichkeiten dieser infrastrukturunabhängigen Positionsdienste bieten eine weite Spanne und wurden von Mauve et al. in [MWH01] klassifiziert.

Erbringer \ Nutzer	some	all
some	DSPS some-for-some	(Infrastuktur) some-for-all
all	VHZ / SLS all-for-some	DREAM all-for-all

Abbildung 2.10.: Klassen von Positionsdiensten mit Beispielen

Die Autoren unterscheiden dabei jeweils auf der Seite der Dienstleister und der Dienstnutzer anhand der beiden Kategorien *some* und *all*. Halten alle Teilnehmer eines Ad-hoc-Netzes Positionsinformationen als Dienstleister für andere vor, so fallen sie in die Klasse *all-for-x*. Andererseits ist es ebenfalls möglich, dass nur eine (wie auch immer definierte) begrenzte Anzahl an Teilnehmern als Positionsdienste fungiert und Aufenthaltsinformationen bereitstellt (*some-for-x*). Auf der Seite der Dienstnutzer – darunter verstehen Mauve et al. nicht die anfragenden Teilnehmer, sondern diejenigen, deren Ortsinformationen bei einem bestimmten Dienst gespeichert und vorgehalten werden – kann die gleiche Unterteilung stattfinden: Hält der Positionsdienst (aus wie vielen Teilnehmern er auch bestehen mag) die Daten sämtlicher im Netz vorhandener Teilnehmer vor oder beschränkt er sich auf einen (zu definierenden) Teil der Benutzer und andere Dienstleister sind für die verbleibenden Teilnehmer zuständig, so ergibt sich auch hier die Unterteilung in *some* und *all*. In der Kombination bilden sich somit die folgenden vier Klassen von Positionsdiensten heraus (vgl. Abbildung 2.10): *some-for-some* (z. B. *Distributed Secure Position Services* (*DSPS*) [WNR05]), *some-for-all*, *all-for-some* (z. B. *Virtual Homezones* (*VHZ*) [Sto99, GH00] oder der *Simple Location Service* (*SLS*) [CBW02]) und *all-for-all* (z. B. *DREAM* [BCSW98]).

Für die Klasse *some-for-all* gibt es keine Beispiele, da der Overhead, den die als Dienstleister fungierenden Teilnehmer zu bewältigen hätten, nicht praktikabel wäre. Allerdings ließen sich die infrastrukturbasierten Ansätze in diese Klasse einordnen, da nur eine kleine Anzahl an Beteiligten (in einem hybriden Ad-hoc-Netz sind auch Infrastrukturkomponenten Teil des Netzes) als Dienstleister für alle anderen fungieren.

Auf die Thematik der Positionsdienste in Ad-hoc-Fahrzeugnetzen wird – insbesondere unter dem Aspekt des Schutzes der Privatsphäre der Teilnehmer – noch detaillierter in Kapitel 3 eingegangen.

2.2.3.2. Weiterleitungsstrategien

Nachdem der Positionsdienst einem anfragenden Teilnehmer die Position seines Kommunikationspartners mitgeteilt hat, kann die eigentliche Nachricht erzeugt und auf den Weg gebracht werden. Diese wird über eine beliebige Anzahl an Knoten zum Empfänger übermittelt. Jede der Zwischenstationen entscheidet anhand der Weiterleitungsstrategie, ob und an wen eine empfangene Nachricht weitergeleitet wird. Im Folgenden sollen exemplarisch zwei weit verbreitete Strategien kurz umrissen werden.

Flooding Die einfachste Methode der Nachrichtenweiterleitung ist das Flooding. In der einfachsten Variante wird jede empfangene Nachricht von jedem Teilnehmer erneut gebroadcastet. Die Nachricht verteilt sich somit kreisförmig von ihrem Sender an alle Teilnehmer des Netzes; dieses wird somit *geflutet*. Da jeder Teilnehmer des Netzes die Nachricht erhält, kann man sicher sein, dass sie auch den intendierten Empfänger erreicht. Somit ist offensichtlich in diesem Fall auch kein Location Service notwendig. Da jedoch der Overhead dieser Art der Übermittlung einer Nachricht enorm ist (immerhin muss jeder Teilnehmer jede im Netz gesendete Nachricht empfangen und weiterleiten), kann die Verteilungs- oder Weiterleitungsstrategie des Flutens – *Flooding* – lediglich bei kleinen Netzen mit wenigen Knoten eingesetzt werden.

Um die Performance zu verbessern, wurde das Prinzip des *Directed Floodings* entwickelt und beispielsweise unter dem Namen *Location Based Multicast (LBM)* veröffentlicht ([KV99], vgl. [Mai04]). Dabei kann zwischen dem Sender und dem Empfänger ein Bereich als sogenannte *Forwarding Zone* definiert werden. Nur noch innerhalb dieser *Forwarding Zone* findet ein *Flooding* statt; Teilnehmer außerhalb der Zone, die die Nachricht empfangen, senden sie nicht erneut aus (vgl. Abbildung 2.11). Der geflutete Bereich des Netzes wird also auf eine klar definierte Zone beschränkt. Somit kann der Overhead deutlich reduziert werden. Natürlich ist es in diesem Fall für den Sender nötig, die Position des Empfängers zu kennen, um die *Forwarding Zone*, deren Ausdehnung in der Nachricht mittransportiert wird, festzulegen.

Greedy Forwarding *Greedy Forwarding* (von Takagi und Kleinrock in [TK84] als „*most forward within r*“ bezeichnet) stellt eine Alternative zum *Directed Flooding* dar und versucht, die Anzahl an Knoten, die eine Nachricht erhalten und weiterleiten, zu minimieren. Mit Hilfe eines Beaconing-Verfahrens – also einem ständigen Austausch von kurzen Nachrichten, die andere Teilnehmer in der Nähe über die eigene Position und Identität informieren – wird stets eine Liste aller Nachbarn in jedem Knoten aufrechterhalten. Möchte nun ein Teilnehmer eine Nachricht senden

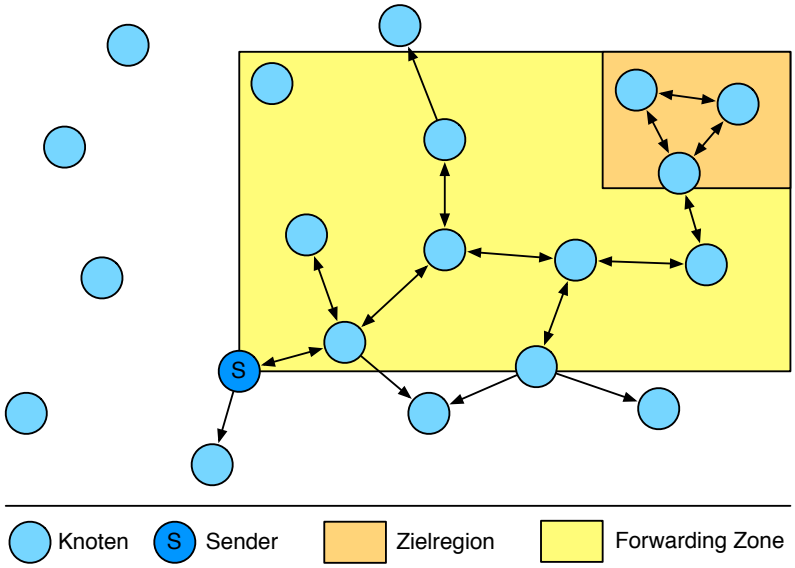


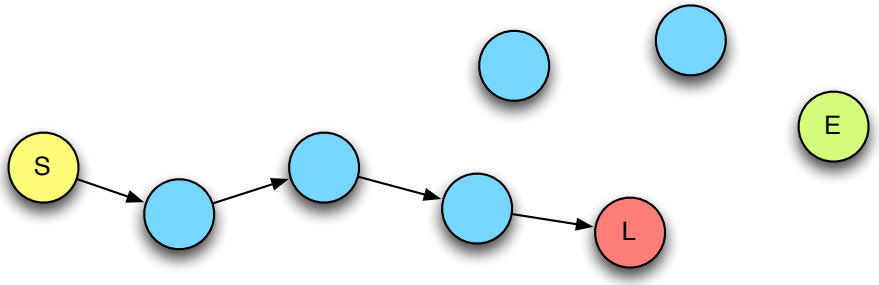
Abbildung 2.11.: Location Based Multicast mit rechteckiger Forwarding Zone (nach [Mai04])

oder weiterleiten, so wählt er denjenigen seiner Nachbarn aus, der dem Ziel der Nachricht am nächsten liegt, und sendet sie an diesen adressiert. Auf diese Weise wird die Nachricht von Knoten zu Knoten weitergegeben, bis sie ihr Ziel erreicht hat.

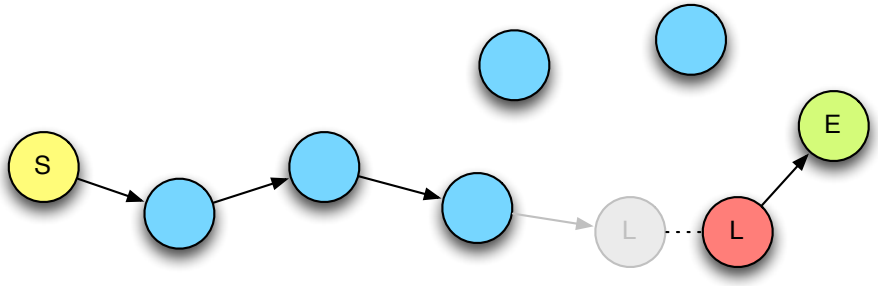
Offensichtlich ist auch für diesen Ansatz ein Positionsdienst erforderlich, da der Sender einer Nachricht dieser die Position des Empfängers auf den Weg mitgeben muss, damit die weiterleitenden Knoten eine Entscheidungsgrundlage für die Auswahl der nächsten Station erhalten.

Als problematisch kann sich diese Technik dann erweisen, wenn beim Transport einer Nachricht vom Sender (S) zum Empfänger (E) ein lokales Maximum (L) erreicht wird (vgl. Abbildung 2.12a). Darunter versteht man einen Knoten, der keinen erreichbaren Knoten hat, der näher als er selbst am Ziel der Nachricht ist. In diesem Fall kann er dem Algorithmus folgend die Nachricht nicht mehr weiterleiten, obwohl eine Route von Sender zu Empfänger existieren würde.

Um mit diesem Problem umzugehen, gibt es eine Reihe von Vorschlägen. Beim *Greedy Perimeter Stateless Routing (GPSR)* ([KK00]) versucht beispielsweise ein Knoten, der erkennt, dass er sich in einem lokalen Maximum befindet, die Nachricht an einen nahegelegenen anderen Knoten weiterzuleiten, auch wenn dieser weiter von dem Ziel entfernt sein sollte. Beim *Cached Greedy Geocast (CGGC)* ([MES04]) dagegen werden Nachrichten, die auf Grund eines lokalen Maximums nicht weitergeleitet werden können, zwischengespeichert. Da sich die Knoten gerade in Ad-hoc



(a) Greedy Forwarding mit lokalem Maximum (L)



(b) Erfolgreiche Nachrichtenübermittlung durch Caching

Abbildung 2.12.: Greedy Forwarding mit lokalem Maximum und Caching-Mechanismus (nach [MES04])

Fahrzeugnetzen relativ schnell zueinander bewegen, kann die Nachricht vermutlich zu einem späteren Zeitpunkt weitergesendet werden (vgl. Abbildung 2.12b).

Durch den sehr geringen Overhead bei einer dennoch sehr hohen Rate an erfolgreich zugestellten Nachrichten bietet das *Greedy Forwarding* recht vielversprechende Möglichkeiten für Ad-hoc-Fahrzeugnetze.

2.3. Besonderheiten von Ad-hoc-Fahrzeugnetzen

2.3.1. Identitäten in Ad-hoc-Fahrzeugnetzen

Bei der Betrachtung von digitalen Identitäten geht es in der Regel um Wiedererkennung und Identifizierung von Objekten. Aus mathematischer Sicht (für einen Überblick über andere Betrachtungswinkel sei auf [Kun03] verwiesen) bedeutet eine Identität zweier Objekte, dass sämtliche ihrer Attribute die gleiche Ausprägung besitzen. Andersherum lässt sich somit ableiten, dass zur eindeutigen Identifizierung

eines Objekts dieses sich von allen anderen Objekten in mindestens einem Attribut unterscheidet, dem identifizierenden Merkmal (*Identifikator*, *Identifier*, *ID*, vgl. [KSW05]). Ist eine solche Identifizierung notwendig, so kann – wenn keine eindeutige Kennung vorliegt – das Objekt um einen expliziten Identifikator erweitert werden.

Generell lassen sich nach Kargl et al. drei unterschiedliche Typen expliziter IDs unterscheiden [KSW05]:

- geräteabhängige Identifikatoren, wie beispielsweise die MAC-Adresse eines Interfaces,
- generische Identifikatoren, die als Zeichenfolgen zusätzlich vergeben werden, wie beispielsweise Seriennummern oder
- Public-Key-Identifikatoren, die durch den öffentlichen Schlüssel (bzw. das Zertifikat) eines asymmetrischen Schlüsselpaares repräsentiert werden.

Kargl et al. machen deutlich, dass zur Verhinderung des Identitätswechsels eines Objekts eine vertrauenswürdige dritte Partei notwendig ist, die die Identifikatoren vergibt oder zumindest an ihrer Erzeugung beteiligt sein muss.

Von den drei beschriebenen Möglichkeiten erscheinen für Ad-hoc-Fahrzeugnetze Public-Key-Identifikatoren am geeignetsten, da damit eine zweifelsfreie Identifizierung und Authentifizierung möglich ist und Identitätsdiebstahl somit nicht allein durch das Ausspähen des zur Identifikation herangezogenen Merkmals (des öffentlichen Schlüssels) möglich ist.

Als nächstes stellt sich die Frage, wer oder was in einem Ad-hoc-Fahrzeugnetz eindeutig identifizierbar sein muss. Dies kann grundsätzlich der Fahrer eines Fahrzeugs (personenbezogene Identität) oder das Fahrzeug selbst (fahrzeugbezogene Identität) sein. Plößl diskutiert diese Fragestellung ausführlich in [Plö09].

Für eine *personen-* und *fahrerbezogene Identität* spricht, dass der Fahrer eines Fahrzeugs bei Unfällen oder Verkehrsordnungswidrigkeiten zweifelsfrei bestimmt werden kann und nicht zunächst generell der Halter haftbar gemacht wird (vgl. StVG[5] §7). Zudem könnte leichter sichergestellt werden, dass Fahren ohne gültige Fahrerlaubnis nicht möglich ist oder zumindest sofort offenkundig gemacht wird.

Es ist jedoch die gängige Praxis, dass eine *fahrzeugbezogene Identität* verwendet wird. Das Fahrzeug wird heute zu diesem Zweck mit einem eindeutigen Kennzeichen ausgestattet, das letzten Endes den Halter pseudonymisiert. Die aktuelle Gesetzeslage (vgl. StVG §7, siehe oben) ist darauf ausgerichtet; daher empfiehlt es sich – auch in Hinblick auf die Akzeptanz der Nutzer – dieses Prinzip beizubehalten und *fahrzeugbezogene Identitäten* einzusetzen. Zu diesem Schluss kommt auch Plößl nach

[5] Straßenverkehrsgesetz, abrufbar unter http://www.gesetze-im-internet.de/stvg/index.html

einer detaillierteren Betrachtung [Plö09]. Ein weiterer Aspekt, der ebenfalls gegen eine *personenbezogene Identität* spricht, ist die Wahrung der Privatsphäre der Fahrzeugführer. Es ist nun nicht mehr einfach zu erkennen, wer gerade ein bestimmtes Fahrzeug steuert.

Der Einsatz von *fahrzeugbezogenen Identitäten* wird auch von anderen Autoren vorgeschlagen. Hubaux et al. sprechen beispielsweise in [HCL04] von einem *Electronic License Plate*, also einem elektronischen Nummernschild.

Kommt jedoch solch eine *fahrzeugbezogene Identität* zum Einsatz, muss sichergestellt werden, dass diese auch tatsächlich untrennbar mit dem Fahrzeug verbunden ist und nicht mit anderen Fahrzeugen ausgetauscht werden kann. Diese Anforderung können sichere Hardwarebausteine erledigen, die im nächsten Abschnitt untersucht werden.

2.3.2. Speicherung von kryptographischem Material

Ein Knoten in einem Fahrzeugnetz – also ein Automobil – muss über eine ganze Reihe von kryptographischen Schlüsseln verfügen. Diese können zum verschlüsselten Nachrichtenaustausch, zur Integritätssicherung von Nachrichten oder zur Authentikation und Identifikation eines Teilnehmers verwendet werden. Die Speicherung im Fahrzeug ist jedoch nicht unkritisch, da es sich dabei um keine sichere Umgebung handelt. Ein Auto beispielsweise kann zu leicht unbefugt geöffnet werden. Zudem befindet es sich immer wieder außerhalb der Kontrolle des Halters, beispielsweise in einer Werkstatt oder bei einer Überprüfungsstelle. Ebenfalls problematisch stellt sich die Sachlage bei Mietwägen dar, da sich diese üblicherweise unter der physischen Kontrolle von Fremden befinden. In all diesen Fällen darf das Schlüsselmaterial nicht einfach zugänglich für jemanden sein, der Zugang zum Fahrzeug hat.

In der Literatur (u. a. [LBH+06, GFL+07, RJP+06, PBH+07, RPA+07, Plö09]) wird die Verwendung von manipulationssicherer Hardware (*Tamper-Proof Module*, *TPM*, auch: *Trusted Plattform Module* oder *Tamper-Resistant Hardware*, *TRH* bzw. *Tamper-Proof Device*, *TPD*) vorgeschlagen. Darunter versteht man ein gekapseltes Modul, das über einen Speicher für Schlüsselmaterial und einen Kryptoprozessor zur Durchführung von kryptographischen Operationen verfügt. Das Modul ver- (nur bei symmetrischen Verfahren) und entschlüsselt und erzeugt digitale Signaturen ohne dass kryptographisches Material nach außen gelangt. Lediglich die Ergebnisse der Operationen werden über eine Schnittstelle an andere Komponenten gemeldet. Zudem muss manipulationssichere Hardware über Mechanismen verfügen, Manipulationen zu erkennen und adäquat darauf zu reagieren. Sollte also ein unautorisierter Versuch, physischen Zugriff auf Teile des Moduls zu erhalten, erkannt werden, müssen Gegenmaßnahmen wie beispielsweise das Löschen des Speichers eingeleitet werden.

Das US-amerikanische *National Institute of Standards and Technology (NIST)* hat 2001 eine Aktualisierung des Standards *FIPS 140* [Nat01] herausgegeben. Nach ihm können manipulationssichere Hardwaremodule auf Basis ihrer Sicherheit in vier Stufen klassifiziert werden. Die beiden höchsten Stufen (3 und 4) fordern dort beispielsweise, dass physische Öffnungsversuche mit hoher Wahrscheinlichkeit erkannt werden und zu einer Löschung des gespeicherten kryptographischen Materials führen. Module, die derart eingestuft sind, sollten für einen Einsatz in Ad-hoc-Fahrzeugnetzen geeignet sein. Allerdings ist zu bedenken – wovor auch Anderson und Kuhn in [AK96] warnen –, dass ein konkretes TPM nicht dauerhaft als sicher einzuschätzen ist, sondern früher oder später seine Schutzwirkung verliert. Dennoch ist die Verwendung eines TPM allgemein anerkannt und die wohl beste Möglichkeit, einen gewissen Schutz für kryptographische Schlüssel zu bieten.

2.3.3. Nachrichten in VANETs

In Ad-hoc-Fahrzeugnetzen werden eine Reihe von Nachrichtentypen unterschieden. Die Unterscheidung basiert meist auf dem Anwendungszweck; dieser führt aber auch in der Regel zu unterschiedlichen Sicherheits- und Performanceanforderungen. Die Einteilung in dieser Arbeit orientiert sich im Wesentlichen an [Plö09], jedoch wird an entsprechender Stelle auch auf andere Publikationen verwiesen. Eine alternative Einteilung, die sich nicht wie bei Plößl auf den Zeck der Kommunikation bezieht, sondern anhand der Kommunikationspartner vorgenommen wird, geht aus dem durch das BMBF[6] geförderte Projekt *Network on Wheels*[7] (*NOW*) hervor und wird von Aijaz et al. in [ABD+06] diskutiert. Da sich jedoch der Schutzbedarf von Nachrichten in der Regel aus ihrem Zweck ableitet, erscheint Plößls Einteilung sinnvoller und wird im Folgenden Anwendung finden.

2.3.3.1. Beacons

Den größten Anteil aller gesendeten Nachrichten in einem *VANET* dürften die *Beacons* (vgl. [Plö09, CPHL07], auch *heartbeat messages* [BHV07, BHWW09], *general safety messages* [RH05, FRF+07] oder *floating car data* [RFSK08] genannt) ausmachen. Jedes Fahrzeug broadcastet in sehr kurzen und regelmäßigen Abständen (in der o. g. Literatur wird von einem Intervall von 150 bis 300 ms ausgegangen) eine solche Nachricht. Diese enthält die wesentlichen Telematikdaten des Fahrzeugs (wie beispielsweise Position, Geschwindigkeit, Richtung und Beschleunigung) und wird in der Regel nicht von anderen Teilnehmern weitergeleitet (Single-Hop-Nachricht).

[6] Bundesministerium für Bildung und Forschung, http://www.bmbf.de/, abgerufen am 10.02.2012
[7] http://www.network-on-wheels.de/, abgerufen am 10.02.2012

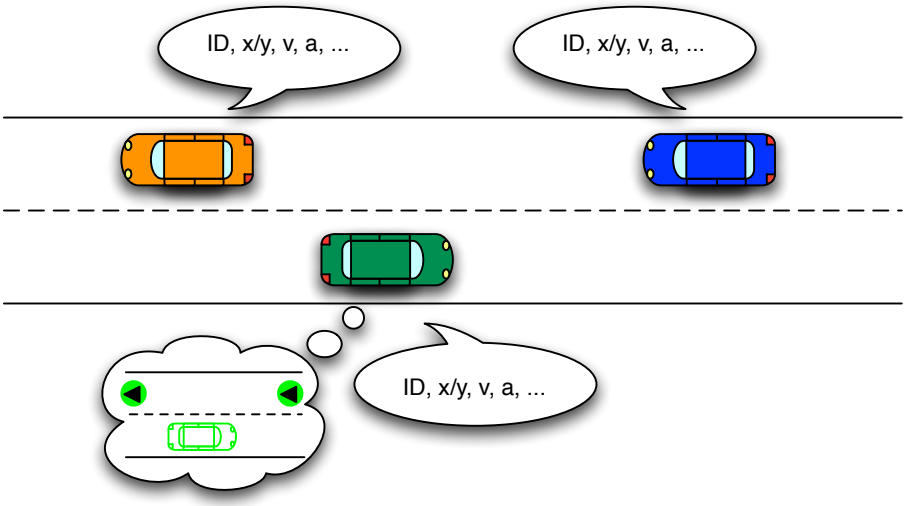


Abbildung 2.13.: Lokales Abbild der Verkehrssituation durch Beacons

Die Reichweite eines *Beacons* beträgt je nach Gegebenheiten zwischen 300 und 1000 Metern.

Da jeder Teilnehmer in sehr kurzen Abständen die Beacons aller Fahrzeuge in seiner Umgebung erhält, entsteht somit lokal ein Abbild der Verkehrssituation (vgl. Abbildung 2.13). Diese Informationen können somit zur Implementierung vieler *ADAS* (*Advanced Driver Assistance Systems*, fortschrittliche Fahrassistenzsysteme) dienen. Dazu können folgende Anwendungen gehören (vgl. Abbildung 2.14):

- *Spurwechsel-/Toter-Winkel-Assistent (1):* Ein Fahrer kann gewarnt werden, wenn sich andere Fahrzeuge in seinem sog. toten Winkel befinden oder sich im Falle eines Spurwechsels schnell von hinten nähern. Diese können anhand der von ihnen ausgesendeten *Beacons* Identifiziert und als Gefahrenquelle erkannt werden.
- *Bremsassistent (2):* Im Falle eines starken Bremsvorgangs von Fahrzeugen hinter einer Kurve oder durch von großen Verkehrsteilnehmern wie LKWs verdeckte Fahrzeuge kann ein nachfolgender Fahrer frühzeitig gewarnt werden und die Geschwindigkeit seines Fahrzeugs gefahrlos verringern.
- *Abstandsassistent (Cooperative Adaptive Cruise Control) (3):* Fahrzeuge können voranfahrenden Verkehrsteilnehmern gefahrlos folgen, da sie frühzeitig über Geschwindigkeitsveränderungen oder Lenkbewegungen informiert werden und diese ebenfalls durchführen können.
- *Cooperative Driving:* Einen Schritt weiter geht das auch als *Platooning* bezeichnete *Cooperative Driving*. Hierbei nutzen Fahrzeuge (insbesondere LKWs)

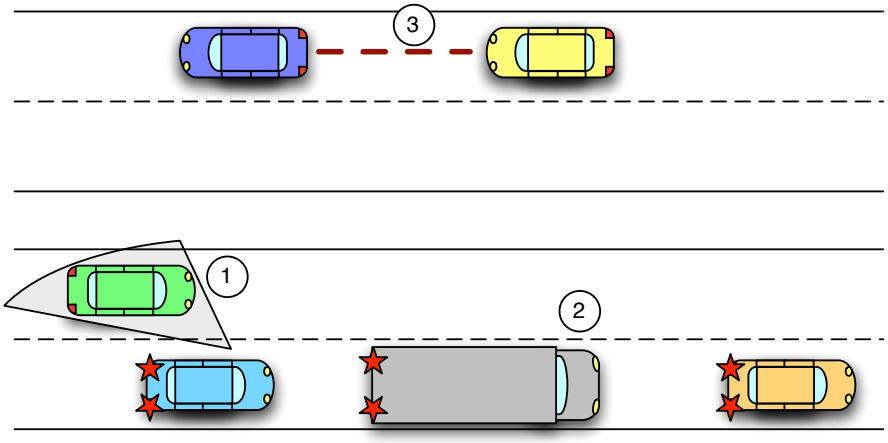


Abbildung 2.14.: Fortschrittliche Fahrassistenzsysteme

die *Beacons* ihrer Vorderfahrzeuge um den Abstand zu verringern und somit durch Ausnutzung des Windschattens Treibstoff einzusparen. Brems- und Beschleunigungsvorgänge werden dabei automatisch simultan durchgeführt um ein Auffahren zu vermeiden.

- *Überholassistent:* Bei Überholvorgängen kann ein vergrößerter Überblick durch *Beacons* anderer Verkehrsteilnehmer vor herannahendem Gegenverkehr oder anderen kritischen Situationen warnen.
- *Kreuzungsassistent:* Beim Annähern an Kreuzungen entnehmen Teilnehmer den *Beacons* Informationen über andere sich nähernde Fahrzeuge und sind somit auf die Kreuzungssituation besser vorbereitet. Auch Ampelanlagen können die Beacons zu einer effizienteren Steuerung der Ampelschaltung verwenden.
- *Precrash-Systeme:* Ist ein Unfall unvermeidbar geworden, so kann mit Hilfe von *Beacons* die Richtung und die Wucht des Aufpralls geschätzt und entsprechend optimiert können Notsysteme ausgelöst werden. Zu diesen Notsystemen gehören beispielsweise Airbags, Gurtstraffer, Aufrichtung von Sitzen oder Entriegelungen von Fahrzeugtüren.

Beacons können in konstanten zeitlichen Intervallen gesendet werden (wie bereits erwähnt wird von rund 200 ms ausgegangen). Alternativ dazu kann auch das Sendeintervall geschwindigkeitsabhängig angepasst werden. In [Sch08] wurde eine Verteilung vorgeschlagen, die einen unterproportionalen Zusammenhang zwischen Geschwindigkeit und Sendehäufigkeit nach der Gleichung 2.1 liefert:

$$t(v) = \frac{2}{\frac{v}{25} + 1} \tag{2.1}$$

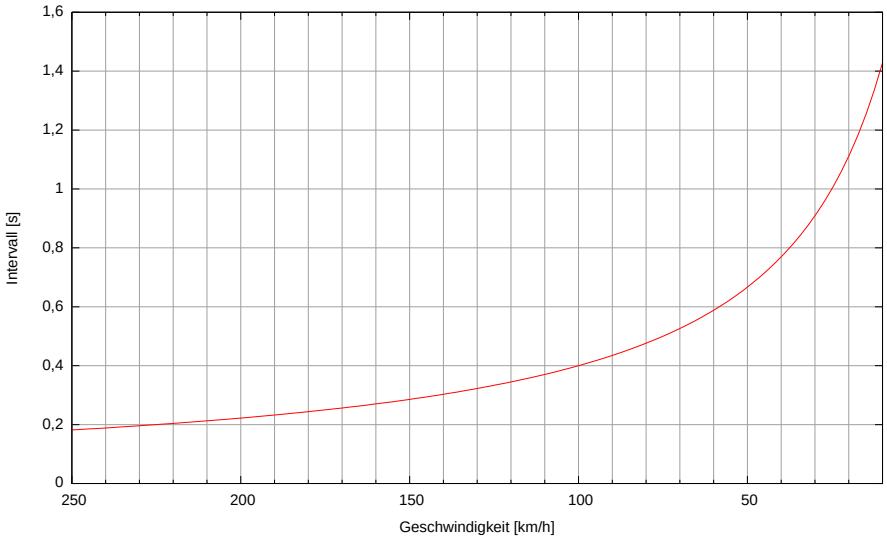


Abbildung 2.15.: Sendeintervalle der *Beacons* in Abhängigkeit von der Geschwindigkeit (nach [Sch08])

Das geschwindigkeitsabhängige Sendeintervall $t(v)$ wird dabei in Sekunden bemessen, die Geschwindigkeit v wird in $^{\mathrm{km}}/_{\mathrm{h}}$ einbezogen. Abbildung 2.15 zeigt die Sendeintervalle für den relevanten Bereich von $250\,^{\mathrm{km}}/_{\mathrm{h}}$ bis $10\,^{\mathrm{km}}/_{\mathrm{h}}$.

Die Kalkulation geht aus der Überlegung hervor, dass bei geringeren Geschwindigkeiten Fahrassistenzsysteme nicht so häufig aktualisierte Daten benötigen, da sich die Situation nicht wesentlich verändert hat.

2.3.3.2. Warnungen

Im Gegensatz zu *Beacons* werden *Warnungen* (in [RH05] als *liability-related safety messages* bezeichnet) nur relativ selten versendet. Sie kommen immer dann zum Einsatz, wenn von einem Teilnehmer eine gefährliche Verkehrssituation erkannt wird und er diese Information an möglicherweise ebenfalls betroffene Fahrzeuge weitergibt. Nach [Plö09] können diese Situationen sowohl mit Hilfe fahrzeugeigener Sensoren als auch durch Aggregation von Telematiknachrichten (vgl. Abschnitt 2.3.3.1) mit Hilfe von Algorithmen ermittelt werden.

Für fahrzeuginterne Sensoren, die Daten für *Warnungen* und *Beacons* zur Verfügung stellen können, nennt [Plö09] unter anderem die folgenden Beispiele:

- Das *Abstandsradar* wird aktuell hauptsächlich für *Adaptive Cruise Control* und *Precreash*-Systeme genutzt, könnte aber eine plötzliche und schnelle An-

näherung an ein Hindernis erkennen und somit einen drohenden Unfall identifizieren.

- Der *Lenkwinkelsensor* (heutiges Hauptanwendungsgebiet: *ESP*) kann Aufschluss darüber geben, in welche Richtung sich ein Fahrzeug fortbewegt.
- Mit Hilfe des *Regensensors*, der meist nur zur Einstellung des Scheibenwischers dient, lassen sich plötzliche Unwetter erkennen. Diese Information kann dann an andere Verkehrsteilnehmer weitergegeben werden.
- Der Drehzahlsensor kann ein Durchdrehen der Reifen auf Grund von glatter Fahrbahn identifizieren, was heute in der *ABS*-Steuerung genutzt wird.

Darüber hinaus wäre es möglich, weitere Sensoren zu etablieren, die beispielsweise den Zustand der Airbags überwachen und daran Unfälle erkennen könnten. Die Beispiele sind offensichtlich vielfältig, und eine gezielte Nutzung der Sensoren kann wertvolle Informationen über den Zustand und die Fortbewegung des Fahrzeugs liefern.

Dennoch besteht keine Beschränkung auf die fahrzeuginternen Sensoren und es ist möglich, die ohnehin stetig empfangenen *Beacons* ebenfalls zu analysieren und im Falle einer erkannten Gefahr eine *Warnung* an andere Teilnehmer zu senden. So weist beispielsweise eine große Anzahl sich langsam fortbewegender Fahrzeuge auf einer Autobahn vermutlich auf einen Stau hin; sich nähernde Fahrzeuge können gewarnt und zum Bremsen aufgefordert werden.

Ausgesendete *Warnungen* werden nicht an bestimmte Fahrzeuge adressiert, sondern sind für alle Teilnehmer in der relevanten Region bestimmt (Geocast). Diese Region muss zumindest die Fahrzeuge umfassen, die sich mit sehr hoher Wahrscheinlichkeit der Gefahrenstelle nähern. Um diese Fahrzeuge, für die diese Informationen relevant sind, zu erreichen, müssen diese Nachrichten häufig über mehrere Schritte von einem Teilnehmer zum nächsten weitergeleitet (Multihop-Nachricht) werden.

2.3.3.3. Alarmsignale und Anweisungen

Die Anwendungskategorie *Alarmsignale und Anweisungen* umfasst Nachrichten, die hoch priorisiert sind und von den Empfängern befolgt werden müssen. Zum Senden berechtigte Teilnehmer sind beispielsweise Einsatzfahrzeuge der Polizei, der Feuerwehr, des Technischen Hilfswerks (THW) und des Bundesamts für Güterverkehr (BAG), Rettungssanitäter und Notärzte sowie Baustellenfahrzeuge von Autobahn- und Straßenmeistereien oder städtischen Bauhöfen. In Notfällen und Gefahrensituationen werden bereits seit langer Zeit optische und akustische Signale eingesetzt – auch um Einsatzfahrzeugen eine freie Fahrt zu Einsatzstellen zu ermöglichen (vgl. §38 StVO). Dieses Prinzip kann mit Hilfe des *VANET*s ergänzt werden.

Polizeibeamte – und seit Einführung der Autobahnmaut auch Kontrolleure des BAG – sind berechtigt, Verkehrsteilnehmer dazu aufzufordern, anzuhalten oder einem Fahrzeug zu folgen (vgl. §36 Abs. 5 StVO). Dies geschieht meist durch das Winken eines Beamten mit einer Kelle oder durch eine Anzeige auf einem Matrix-Feld. Da im ersten Fall ein Beamter seinen Arm aus dem Fenster eines fahrenden Autos halten oder vom Straßenrand aus auf sich aufmerksam machen muss, kann dies zu gefährlichen Situationen führen. Zudem müssen die Signale auf mehrspurigen Straßen nicht unbedingt eindeutig zu interpretieren sein oder können von Verkehrsteilnehmern übersehen werden. In *VANET*s können Einsatzfahrzeuge zusätzlich zu optischen und akustischen Signalen *Alarmsignale oder Anweisungen* an Verkehrsteilnehmer richten, die den Empfängern in ihrem Fahrzeug unmissverständlich mitgeteilt werden können.

In dieser Anwendungskategorie sind zwei grundsätzliche Arten von Nachrichten zu unterscheiden (vgl. [Sch08, Plö09]): *Allgemeine* und *spezielle Anweisungen*. Die erste Gruppe setzt sich aus *Notfallanweisungen* und *verkehrssteuernde Anweisungen* zusammen. *Allgemeine Anweisungen* werden nach dieser Unterteilung grundsätzlich an alle Fahrzeuge in einer bestimmten Region übermittelt, *spezielle Anweisungen* dagegen sind stets für einen oder mehrere gezielt ausgewählte Empfänger bestimmt.

Notfallanweisungen werden insbesondere dann gesendet, wenn Verkehrsteilnehmer in einer kritischen Situation schnellstmöglich einer Anweisung nachkommen müssen. Dies könnte etwa die Bildung einer Rettungsgasse oder das Räumen einer Kreuzung oder Einfahrt für Einsatzfahrzeuge sein (vgl. u. a. §11 Abs. 2 StVO: Bildung einer Rettungsgasse). Zu diesen Notfallanweisungen lassen sich auch Kommunikationen mit Infrastrukturen zählen, die dazu dienen, Einsatzfahrzeugen freie Fahrt zu ermöglichen. Ein Beispiel hierfür wäre die Manipulation der Ampelschaltung, um dem Einsatzfahrzeug in der gewünschten Fahrtrichtung „grün“ anzuzeigen. Dies ist beispielsweise in [DKKS05] beschrieben. *Vehrkehrssteuernde Anweisungen* dagegen sind in der Regel dort zu verwenden, wo Fahrzeuge an Unfällen, Gefahren- und Baustellen, Mäh- und Markierungsarbeiten vorbei- oder von Straßen oder Autobahnen auf Umleitungen abgeleitet werden sollen.

Wie bereits erwähnt, sind zum Senden dieser Alarmsignale und Anweisungen spezielle Rechte nötig. Dennoch benötigen nicht alle Fahrzeuge mit höheren Privilegien die Rechte, alle Nachrichten dieser Kategorie zu senden, sondern oft nur eine Teilmenge daraus (vgl. [Sch08, Plö09]). Alle Nachrichten dürften demnach von Polizeifahrzeugen gesendet werden, die ein vielfältiges Aufgabengebiet haben. So ist es manchmal erforderlich, schnell einen bestimmten Ort zu erreichen, Verkehr umzuleiten oder einzelne Verkehrsteilnehmer zum Anhalten aufzufordern. Einsatzkräfte der Feuerwehr, Notärzte und Rettungssanitäter oder auch das THW benötigen nur Rechte, allgemeine Anweisungen zu senden, da es nicht zu ihren Aufgaben gehört, einzelne

	Allgemeine Anweisungen		Spezielle Anweisungen
	Notfall-anweisungen	Verkehrssteuernde Anweisungen	
Polizei	+	+	+
Feuerwehr, THW, Notarzt, Rettungssanitäter	+	+	-
BAG	-	-	+
Fahrzeuge von Autobahn- oder Straßenmeistereien sowie städtischen Bauhöfen	-	+	-
Stationäre Infrastruktur (Lichtzeichenanlagen, Verkehrszeichen)	+	+	+

Tabelle 2.2.: Berechtigungen privilegierter Teilnehmer im VANET (nach [Sch08, Plö09])

Verkehrsteilnehmer zu kontrollieren. Anders verhält es sich beim BAG, das lediglich spezielle Anweisungen senden darf, da von ihm ausschließlich Fahrzeugkontrollen zur Durchsetzung der Autobahnmaut durchzuführen sind. Baufahrzeuge benötigen lediglich verkehrssteuernde Anweisungen, da sie nicht in Notfällen agieren, aber dennoch in der Lage sein müssen, bei Bau- oder Sanierungsmaßnahmen den Verkehr zu warnen und umzuleiten. Nach [DKKS05] steht ebenfalls zur Diskussion, ob Ampeln ihren Status und Verkehrszeichen ihre Bedeutung noch einmal per Nachricht an Teilnehmer übertragen und sogar Warnungen an Fahrzeuge übermittelt werden sollten, die beispielsweise eine Einbahnstraße nicht beachtet haben. Somit ist die Infrastruktur auch (je nach Verkehrszeichen oder Lichtzeichenanlage) berechtigt, alle Nachrichten zu senden. Auch die Automobilhersteller befassen sich inzwischen intensiv mit diesem Thema, so hat beispielsweise VW[8] angekündigt, einen Kreuzungsassistenten zu entwickeln, der durch Kommunikation zwischen Fahrzeugen und Infrastruktur Gefahrensituationen identifiziert, Fahrer warnt und so zur Vermeidung von Unfällen beitragen soll.

Tabelle 2.2 fasst diese oben beschriebenen Berechtigungen noch einmal zusammen. Zu beachten ist dabei, dass diese Zuordnung nach Möglichkeit die aktuelle Gesetzeslage in Deutschland abbildet. Diese kann sich natürlich stets verändern, und auch in anderen Ländern könnten die Zuordnungen unterschiedlich ausgestaltet sein.

[8] vgl. `http://www.volkswagenag.com/content/vwcorp/content/de/innovation/driver_assistance/intersection_assistant.html`, abgerufen am 22.08.2012

2.3.3.4. Komfortdienste

Die *Komfortdienste* bilden eine sehr heterogene Gruppe von Diensten. Im Rahmen eines *Komfortdienstes* kommunizieren in der Regel zwei Partner, von denen einer der Nutzer und ein anderer der Diensterbringer ist. Dies bedeutet auch, dass normalerweise Unicast-Nachrichten versendet werden und die Kommunikation häufig zwischen einem mobilen Teilnehmer und einer Infrastrukturkomponente stattfindet. Plößl ([Plö09]) und Dikaiakos et al. ([DFNI07]) beschreiben eine Reihe von Diensten, die sich in einem *VANET* realisieren ließen. Diese reichen von aktuell bereits in ähnlicher Form vorhandenen Implementierungen wie beispielsweise Straßenmauterhebung oder Navigation mit Hilfe der Erkenntnisse anderer Teilnehmer über bereits testweise durchgeführte Verfahren wie fahrstilabhängige Kfz-Versicherungskosten wie das *Pay as you drive*-Produkt des britischen Versicherers *Norwich Union* (heute *Aviva*) [Hei08] bis hin zu mobile Internetverbindungen, mobilen Bezahlverfahren (vgl. [Hei06]) oder gar veränderbaren Motorleistungen ([RG05]).

Kapitel 3

Datenschutzfreundliche Positionsdienste

Dieses Kapitel beleuchtet die Bedrohung der Privatsphäre der Teilnehmer eines Ad-hoc-Netzes, in dem positionsbasiertes Routing zum Einsatz kommt. Zuerst wird untersucht, wodurch diese Problematik entstehen kann und warum es Angreifern meist einfach möglich ist, Bewegungsprofile von Nutzern zu erzeugen. Anschließend folgt eine eingehende Betrachtung publizierter Konzepte, die dieser Gefahr begegnen sollen, auf ihre Wirksamkeit und Praktikabilität hin. Schließlich wird ein eigener Vorschlag unterbreitet. Dieser selbstentwickelte Ansatz bietet ein besseres Schutzniveau als die analysierten Konzepte aus der Literatur und bedient sich dazu Methoden aus dem Bereich der datenschutzfreundlichen Techniken wie Kommunikationsmixen.

3.1. Prinzip der Positionsdienste und Angriffe

In mobilen Netzen, insbesondere in sehr dynamischen wie einem VANET, bietet sich – wie bereits in Abschnitt 2.2.3 dargelegt – der Einsatz von positionsbasiertem Routing an, da die Übermittlung der Nachrichten bei diesem Konzept im Wesentlichen von der Position des Empfängers abhängt und nicht, wie bei topologiebasiertem Routing, von der Position aller Knoten zwischen dem Sender und dem Empfänger. Die aktuellen Positionen der potentiellen Empfänger von Nachrichten werden durch einen Positionsdienst vorgehalten, wo sie von allen potentiellen Sendern abgefragt werden können. Ist dem Sender die Position des Empfängers bekannt, kann er die Nachricht dorthin adressieren und absenden. Die eingesetzte Forwardingstrategie entscheidet anschließend darüber, auf welche Art und Weise die Nachricht übermittelt wird.

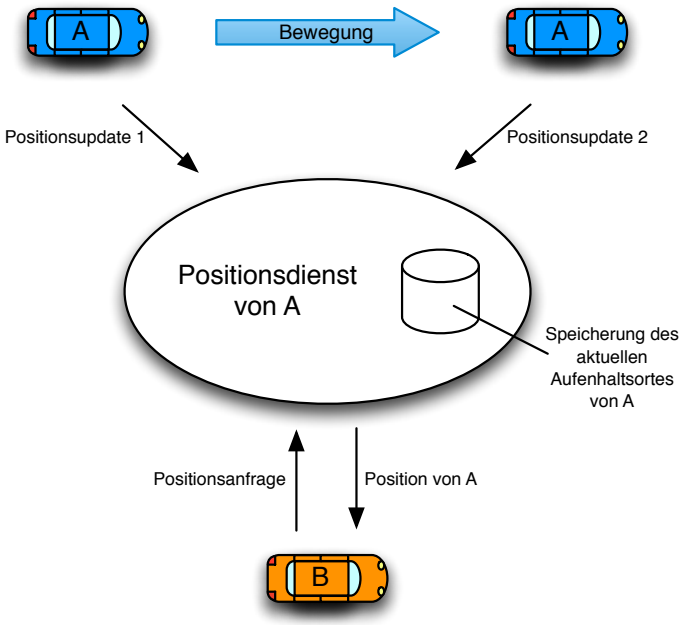


Abbildung 3.1.: Minimales Prinzip eines Positionsdienstes

Positionsdienste können sehr unterschiedlich ausgestaltet sein. So können die Informationen über den Aufenthaltsort der potentiellen Empfänger von einer zentralen oder mehreren verteilten Instanzen der Infrastruktur vorgehalten werden, die von Fahrzeugen über RSUs kontaktiert werden können. Dieses Prinzip wird jedoch häufig im Bereich der mobilen Netze vermieden, da diese in der Auffassung vieler Autoren vollständig autonom und ohne jegliche Infrastruktur funktionieren sollen. Dennoch erscheint es in einigen Bereichen sinnvoll, auf eine Infrastruktur zu setzen, da nur mit ihr tatsächlich Dienstequalitäten (*Quality of Service, QoS*) garantiert und eingefordert werden können.

Alternativ dazu können andere Fahrzeuge als Positionsdienst dienen und diese Informationen bereitstellen; der Einsatz einer Infrastruktur ist damit nicht erforderlich. Die Auswahl derjenigen Teilnehmer, die verantwortlich für die Vorhaltung der Positionsdaten eines bestimmten potentiellen Empfängers sind, kann auf sehr unterschiedliche Arten getroffen werden. Sehr häufig ist jedoch der Aufenthaltsort eines Teilnehmers dafür entscheidend, für welche Teilnehmer er Positionsinformationen vorhalten muss (siehe auch [BCSW98, Sto99, GH00, LJDC+00, MJK+00]).

Unabhängig davon, wie der Positionsdienst konkret ausgestaltet wird, arbeitet er stets nach dem in Abbildung 3.1 skizzierten Schema: Jeder Teilnehmer, der in der Lage sein möchte, Nachrichten zu empfangen, sendet in regelmäßigen Abständen seine aktuelle Position an die für ihn zuständige Instanz des Positionsdienstes (*Lo-*

cation Update, *LUP*). Die zuständige Instanz kann beispielsweise dadurch festgelegt sein, dass es nur einen zentralen Dienst gibt oder dass auf Grund der Kennung des Fahrzeugs eine Instanz eines verteilten Dienstes ermittelt wird. Ein Update der Position ist immer dann notwendig, wenn

- der Teilnehmer sich so weit von der bisher gespeicherten Position entfernt hat, dass eine Zustellung einer dorthin adressierten Nachricht nicht mehr möglich ist,
- die Gültigkeit des alten Eintrags in der Datenbank des Positionsdiensts abgelaufen ist,
- die Kennung des Teilnehmers sich geändert hat oder
- kein älterer Eintrag bei dem zuständigen Positionsdienst existiert (Initialisierung).

Ist der aktuelle Aufenthaltsort eines Teilnehmers bei der zuständigen Instanz des Positionsdienstes hinterlegt, so kann ein anderer Teilnehmer, der ihm eine Nachricht schicken möchte, jederzeit eine Anfrage an den Positionsdienst (*Location Request*, *LREQ*) stellen und somit die gespeicherte Position erfahren. Die Updates, die Anfragen sowie die Rückantworten (*Location Response*, *LRES*) müssen natürlich verschlüsselt erfolgen – ansonsten würden alle Teilnehmer allein durch die Kommunikation mit dem Positionsdienst ihre Bewegung einem Lauscher offenlegen. Damit ist die Aufgabe des Positionsdienstes auch schon erfüllt; alle weiteren Vorgänge betreffen nun ausschließlich den Sender und Empfänger der Nachricht sowie eine Reihe weiterer Teilnehmer, die mit Hilfe der jeweiligen Weiterleitungsstrategie den Nachrichtentransport übernehmen.

Angreifer, die als Teilnehmer eines *VANET*s zu der Gruppe der potentiellen Sender von Nachrichten gehören, können jederzeit die Position eines anderen Teilnehmers bei einem Positionsdienst erfragen. Dies kann dazu missbraucht werden, ein Bewegungsprofil zu erstellen, also die komplette Bewegung mit allen Aufenthaltsorten und den jeweiligen Zeitpunkten eines Teilnehmers zu speichern und auszuwerten. Die Privatsphäre des Opfers kann somit auf das Äußerste beeinträchtigt werden.

Weiterhin ist natürlich der Positionsdienst selbst (bzw. sein Betreiber) in der Lage, jederzeit die bei ihm gespeicherten Positionen der Teilnehmer zur Kenntnis zu nehmen und Bewegungsprofile zu erstellen. Selbst wenn ein Positionsdienst nicht böswillig agiert, kann es sein, dass durch einen Angriff auf die Datenbank des Dienstes Informationen an Außenstehende gelangen, die diese missbrauchen. Es ist somit unerlässlich, dass die bei einem Positionsdienst hinterlegten Daten geschützt werden, ohne die Funktionalität des Dienstes zu stark einzuschränken.

Leider wird diese Anforderung sehr häufig bei der Gestaltung von Positionsdiensten nicht berücksichtigt; in der Regel steht die Performance und die Skalierbarkeit

– ebenfalls wichtige Eigenschaften – im Vordergrund. Daher werden in der Folge bei der Untersuchung bekannter Dienste nicht nur für positionsbasiertes Routing entwickelte Konzepte untersucht, sondern auch verwandte Ansätze aus dem Bereich der Location-based Services (*LBS*), die ebenfalls eine Instanz benötigen, die Diensterbringern (also den Sendern der Nachrichten) die Position der Dienstnutzer (der Teilnehmer) bereitstellen.

3.2. Schutz durch Unschärfe

Eine vielfach beschriebene Möglichkeit, um Benutzer von Positionsdiensten und Location-based Services vor Beobachtung zu schützen, ist, ihre Position nicht exakt, sondern lediglich vage anzugeben. Samarati und Sweeney haben unterschiedliche Möglichkeiten dafür in einem umfassenden Artikel (siehe [SS98]) zusammengefasst, Mascetti hat in seiner Dissertation [Mas07] eine informationstheoretische Analyse verschiedener Ansätze durchgeführt.

Unschärfe kann allgemein dadurch erreicht werden, dass der Aufenthaltsort eines Teilnehmers nicht mehr exakt als Tupel (x, y, t), also den x- und y-Koordinaten zu einem bestimmten Zeitpunkt t, sondern vielmehr als ein Tupel bestehend aus drei Intervallen $([x_1, x_2], [y_1, y_2], [t_1, t_2])$ angegeben wird (vgl. [GG03]). $[x_1, x_2]$ und $[y_1, y_2]$ spannen dabei eine zweidimensionale Region auf, in der sich der entsprechende Teilnehmer irgendwann während des Zeitintervalls $[t_1, t_2]$ aufhält. Diese Unschärfe kann jedoch nur dann Schutz bieten, wenn nicht nur ein, sondern mehrere Teilnehmer mit dem Tupel beschrieben werden können.

Alle k Teilnehmer, auf die eine solche Beschreibung zutrifft, bezeichnet man zusammenfassend als Anonymitätsgruppe (*k-anonymity*). Die Kennzahl k dient dabei als Maß für den Schutz eines Einzelnen unter der Annahme, dass ein Außenstehender nicht in der Lage ist, die Gruppenmitglieder voneinander zu unterscheiden. Die folgenden Varianten für datenschutzfreundliche Positionsdienste basieren auf diesen Vorüberlegungen.

3.2.1. Spatial and Temporal Cloaking

Gruteser und Grunwald stellen in ihrer Arbeit [GG03] einen Positionsdienst vor, der die Aufenthaltsorte – und damit die Privatsphäre – der Teilnehmer schützen soll. Dieser ist aus ihrer Perspektive zwar vorrangig für *LBS* vorgesehen und dient also dazu, Diensteanbietern die Kontaktaufnahme zu mobilen Teilnehmern zu ermöglichen. Damit geht jedoch keine Einschränkung der Allgemeingültigkeit einher, da es grundsätzlich keine Rolle spielt, welche nicht per se vertrauenswürdige Gruppe auf

Algorithmus 3.1 Adaptive-interval Cloaking-Algorithmus (vgl. [GG03])

```
q ← total area covered by anonymizer
q_prev ← total area covered by anonymizer
vehicles ← all known vehicles
p ← position of requested vehicle
loop
    if number of vehicles in vehicles < k_min then
        return q_prev
    end if
    q_prev ← q
    Divide q to quadrants of equal size
    q ← quadrant that includes p
    remove all vehicles outside q from vehicles
end loop
```

Positionsdaten zugreifen und diese möglicherweise legitim zu einer Kontaktaufnahme oder illegitim zur Verfolgung nutzen möchte.

Die Autoren gehen von einem vertrauenswürdigen Positionsdienst aus, an den alle mobilen Teilnehmer in regelmäßigen Abständen ihre exakte Position in der ihnen größtmöglichen Auflösung melden, damit diese dort für Abfragen vorgehalten werden kann. Der Positionsdienst kennt somit die exakten Positionen aller Teilnehmer und macht sich dieses Wissen zu Nutze, um ihren Aufenthaltsort bei Anfragen zu verschleiern.

Zum Schutz der Teilnehmer wird bei Positionsanfragen stets auf eine Anonymitätsgruppe von wenigstens k_{min} geachtet; das heißt, die zurückgelieferte Region, in der sich der angefragte Teilnehmer aufhält, wird durch den Positionsdienst so groß gewählt, dass sich mindestens $k_{min} - 1$ andere Teilnehmer ebenfalls in diesem Bereich im gegebenen Zeitraum aufhalten. Dies wird mit Hilfe eines in Anlehnung an den Quadtree-Algorithmus (vgl. [Sam90]) entwickelten Verfahrens realisiert (vgl. Algorithmus 3.1):

Zu Beginn wird der gesamte Bereich betrachtet, für den der Positionsdienst zuständig ist und sämtliche Teilnehmer darin werden gezählt. Kann bereits jetzt keine ausreichend große Anonymitätsgruppe gebildet werden (da sich weniger als k_{min} Teilnehmer im gesamten Gebiet aufhalten), muss dennoch die Gesamtregion als Antwort zurückgeliefert werden, damit die Funktionalität sichergestellt ist. Befinden sich mehr als k_{min} Teilnehmer in der Region, so wird diese in mehrere (in der Regel vier) gleich große Teile untergliedert, und nun wird lediglich der Teil, der die Position des angefragten Teilnehmers enthält, betrachtet. Befinden sich nun weniger als k_{min} Teilnehmer in dieser Unterregion, so ist diese zu klein um die Privatsphäre

des Angefragten zu schützen und die Region aus der vorherigen Granularitätsstufe muss zurückgeliefert werden (dort war die Bedingung Anzahl Teilnehmer $\geq k_{min}$ erfüllt). Bietet jedoch auch der aktuell betrachtete kleinere Bereich eine ausreichende Anonymitätsgruppe, so wird die Genauigkeit durch einen weiteren Teilvorgang erhöht und es findet eine weitere Überprüfung statt. Dieses Verfahren aus Teilen und Überprüfen wird so lange durchgeführt, bis die geforderte Anonymitätsgruppe nicht mehr erreicht wird.

Gruteser und Grunwald schlagen neben dieser Unschärfe in der Position (als *spatial cloaking* bezeichnet) auch als Alternative ein Konzept vor, das mit zeitlicher Unschärfe (*temporal cloaking*) arbeitet. Dabei wird eine Positionsanfrage durch den Positionsdienst verzögert beantwortet und erst dann eine Region zurückgeliefert, wenn eine Mindestanzahl an Fahrzeugen k_{min} diese Region durchfahren haben. Auf diese Weise wird das Zeitfenster vergrößert, wenn die räumliche Auflösung zu unscharf würde oder ansonsten die Anonymitätsgruppe (und damit der Schutz der Privatsphäre) für angefragte Fahrzeuge zu klein wäre.

Die Autoren untersuchen ihre Konzepte mit Hilfe auf realen Verkehrsdichten basierenden Simulationen in unterschiedlichen Gebieten zu unterschiedlichen Tageszeiten. Der Algorithmus arbeitet dabei mit einem Schwellwert von $k_{min} = 5$ und unterteilt in jedem Durchlauf die aktuelle (quadratische) Region wiederum in vier quadratische Teile.

Wie erwartet zeigt sich, dass dicht befahrene Highway-Abschnitte zu einer höheren Auflösung der Position führen als schwächer befahrene Nebenstraßen. Der Median dieser Auflösung variiert in den Untersuchungen zwischen 30 und 250 Metern, wobei die durchschnittliche Anonymitätsgruppe bei allen Szenarien um 10 herum schwankt. Weiterhin können Gruteser und Grunwald zeigen, dass die Genauigkeit der Auflösung negativ mit der geforderten Mindestgröße der Anonymitätsgruppe k_{ming} korreliert, was ebenfalls der intuitiven Erwartung entspricht.

Bei dem Einsatz der zeitlichen Unschärfe kann mit einer Auflösung von 15 m bei gleichbleibendem k_{min} von 5 eine Auflösung von unter 30 s in Highway-Szenarien erreicht werden; im Fall der weniger stark befahrenen Straßenabschnitte steigt sie auf etwa 70 s an.

In ihren abschließenden Untersuchungen zeigen die Autoren noch einen möglichen Angriff auf, der sich auf Grund ungünstiger Konstellationen für einzelne Teilnehmer ergeben kann (vgl. Abbildung 3.2). In diesem Beispiel wird der oben beschriebene Algorithmus zur Bestimmung der zurückzumeldenden unscharfen Position angewendet ($k_{min} = 3$). Dabei werden die folgenden Tupel als Antwort auf die Anfragen nach den Positionen der vier Fahrzeuge A bis D innerhalb so kurzer Zeit zurückgeliefert, dass eine signifikante Positionsänderung nicht möglich ist:

- A: $([0,1], [0,1], [t_1, t_2])$

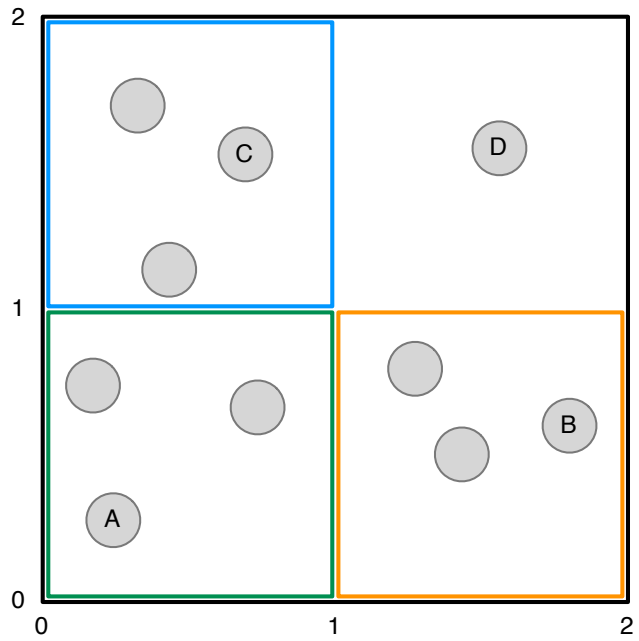


Abbildung 3.2.: Aufhebung der Anonymität in ungünstiger Situation (nach [GG03])

- B: $([1,2],[0,1],[t_1,t_2])$
- C: $([0,1],[1,2],[t_1,t_2])$
- D: $([0,2],[0,2],[t_1,t_2])$

Die Antworten auf die Anfragen nach den Teilnehmern A bis C liefern die kleinen in Abbildung 3.2 farbig hervorgehobenen einander benachbarten Quadrate zurück, wohingegen die vierte Abfrage das große schwarze Quadrat zurückliefert, das die drei anderen enthält. Beobachtet nun ein Angreifer die dargestellte Verteilung der Teilnehmer, so kann er schließen, dass Anfrage D aus dem Quadranten $([1,2],[1,2])$ stammen muss und kann somit das Fahrzeug D eindeutig zuordnen. An dieser Stelle ist somit eine Verbesserung des Algorithmus erforderlich.

Bewertung Mit ihrem Konzept zeigen Gruteser und Grunwald erstmals auf, wie sich ein datenschutzfreundlicher Positionsdienst ausgestalten lässt. Ihre vorgestellten Methoden ermöglichen es den Teilnehmern, ihr Schutzniveau festzulegen und sichergestellt zu bekommen, dass dieses auch eingehalten wird. Auch wenn die Autoren selbst darlegen, dass der eingesetzte Algorithmus in seiner Grundform noch Schwächen aufweist, die in bestimmten Konstellationen einzelne Teilnehmer exponieren können, kann der vorgeschlagene Dienst dennoch in den meisten Fällen bereits ein

sehr gutes Schutzniveau bieten.

Etwas nachteilig ist, dass die Autoren in ihrem Angreifermodell den Positionsdienst als vertrauenswürdig annehmen und ihm alle Informationen in höchstmöglicher Auflösung zur Verfügung stellen. Wünschenswert wäre es, wenn ein Teilnehmer nur Vertrauen in Komponenten setzen müsste, die er technisch kontrollieren kann – auch aus dem Grund, dass der Positionsdienst zwar möglicherweise tatsächlich vertrauenswürdig ist, jedoch selbst Opfer eines Angriffs werden kann und die vorgehaltenen Daten somit durch eine dritte Partei missbraucht werden könnten.

Ein Problem der Übertragbarkeit des Ansatzes liegt darin begründet, dass Gruteser und Grunwald den Dienst nicht direkt für den Nachrichtenversand an einzelne Teilnehmer, sondern vielmehr die Einbettung der von ihnen gesendeten Daten in ein geographisches Umfeld vorschlagen. Als Beispiele nennen die Autoren Unwetterwarnungen oder Informationen über Straßenzustände, die allesamt entweder ein größeres Gebiet betreffen (Unwetter) oder aber eine durchaus längere Gültigkeit besitzen (Straßenschäden). Damit ist für sie stets eine ausreichende Auflösung – räumlich oder zeitlich – gegeben.

Möchte man jedoch gezielt ein Fahrzeug kontaktieren, so muss sichergestellt sein, dass es sich bei der Übermittlung einer Nachricht noch in Funkreichweite der übermittelten Position befindet. Bei einer geschätzten Funkreichweite zwischen 300 und 1000 m darf die Auflösung daher bei einem sehr kleinen Zeitfenster nicht gröber sein. Arbeitet man mit zeitlicher Unschärfe, muss das Zeitfenster ausreichend klein gewählt werden, sodass sichergestellt ist, dass sich das gesuchte Fahrzeug noch in Reichweite befindet. Diese Nebenbedingung kann unter Umständen mit der Forderung nach einer ausreichend großen Anonymitätsgruppe kollidieren; in den Simulationsergebnissen der betrachteten Straßenabschnitte konnte jedoch meist eine ausreichend hohe Auflösung bei gleichzeitiger Gewährleistung der Anonymität erzielt werden. Trotzdem kann insbesondere in ländlichen Regionen, die häufig eine geringe Verkehrsdichte aufweisen, dieses Konzept an seine Grenzen stoßen.

3.2.2. CliqueCloak

Gedik und Liu wählen mit ihrem in [GL05] beschriebenen Ansatz einen ähnlichen Weg und möchten ebenfalls mit Hilfe von *temporal* und *spatial cloaking* den Aufenthaltsort von Teilnehmern schützen, die, wie auch in dem Konzept von Gruteser und Grunwald, Nachrichten an einen *Location-based Service* schicken. Sie arbeiten dabei direkt mit den Nachrichten, die innerhalb eines Zeitfensters übermittelt und mit Hilfe eines Proxys, der sich um die Anonymisierung kümmert, zugestellt werden. Zudem kann der Sender die gewünschte Anonymitätsgruppe sowie die maximale

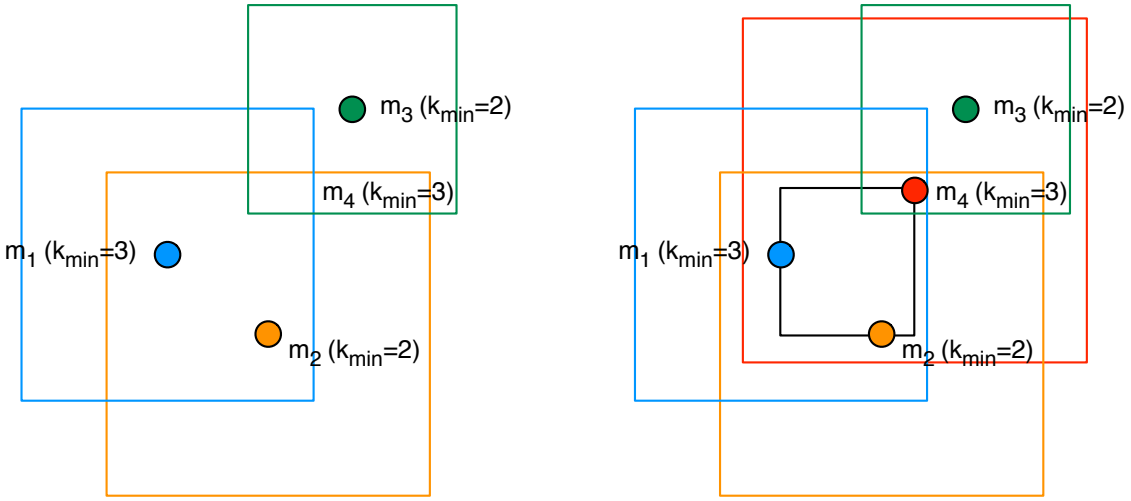
räumliche und zeitliche Unschärfe für jede Nachricht individuell festlegen und dem Anonymisierungsdienst übermitteln.

Jede Nachricht enthält daher neben dem exakten Ort und der exakten Zeit des Sendens auch noch Werte, die die maximale Abweichung von den tatsächlichen Daten sowie die minimale Anonymitätsgruppe (k_{min}, von den Autoren mit k bezeichnet) bestimmen. Die bei dem Anonymisierungsdienst eingehenden Nachrichten werden analysiert und in einem Puffer vorgehalten, bis es gelungen ist, andere Nachrichten zu finden, die zu einer Anonymitätsgruppe unter Wahrung der Rahmenbedingungen zusammengefasst und transformiert werden können. Das vorgegebene Zeitfenster dient dabei dazu, den Zeitpunkt zu bestimmen, zu dem eine Nachricht – sofern sie noch nicht transformiert werden konnte, da sich keine geeignete Anonymitätsgruppe gefunden hatte – verworfen und somit aus dem Speicher entfernt wird. In diesem Fall ist die Zustellung fehlgeschlagen.

Für jede eingegangene Nachricht wird untersucht, ob sich andere Nachrichten finden lassen, deren Sendepositionen sich gegenseitig innerhalb der jeweils vorgegebenen räumlichen Unschärfe befinden und dies auf mindestens so viele Nachrichten zutrifft, wie das größte unter ihnen vorhandene k_{min} vorgibt. Dies ist notwendig, damit sämtliche Anforderungen (sowohl die maximale räumliche und zeitliche Unschärfe als auch die Mindestgröße der Anonymitätsgruppe) der betreffenden Nachrichten erfüllt ist. Lässt sich eine Gruppe – eine *Clique* – bilden, so können die beteiligten Nachrichten transformiert und zugestellt werden. Bei dieser Transformation werden die Anonymitätsanforderungen entfernt sowie die exakte Position durch eine Region, die alle Sendepositionen der Nachrichten der *Clique* enthält, und der Sendezeitpunkt durch die aktuelle Zeit ersetzt.

Um den Ablauf zu veranschaulichen, bietet sich das in Abbildung 3.3 illustrierte Beispiel nach [GL05] an: In der ersten Situation sind drei Nachrichten ($m_1 \ldots m_3$) am Anonymisierungsproxy eingetroffen. Der Ort ihres Sendens ist dabei mit einem Punkt markiert, die Umrahmung gibt den (rechteckigen) maximalen Toleranzbereich für die räumliche Abweichung an. Weiterhin besitzt jede Nachricht ein von dem Sender gefordertes k_{min}, das die minimale Anonymitätsgruppe bestimmt. Die zeitlichen Komponenten sind aus Gründen der Übersichtlichkeit in der Abbildung nicht dargestellt. Die drei eingetroffenen Nachrichten können nicht transformiert und weitergeleitet werden, da sich nur die Nachrichten m_1 und m_2 gegenseitig innerhalb der maximal zulässigen räumlichen Unschärfe befinden und zudem Nachricht m_1 eine Anonymitätsgruppe von mindestens 3 einfordert. Daher werden die Nachrichten während ihres zeitlichen Gültigkeitsbereichs im Proxy vorgehalten.

Innerhalb dieser Zeit trifft nun, wie im zweiten Schritt ersichtlich, eine weitere Nachricht, m_4, ein. Diese befindet sich nun sowohl im gemeinsamen Gültigkeitsbereich von m_1 und m_2 sowie im Bereich von m_3. Mit m_3 zusammen ist keine Gruppe



(a) Schritt 1 **(b)** Schritt 2

Abbildung 3.3.: Beispiel einer Situation in CliqueCloak (nach [GL05])

möglich, da m_4 ein $k_{min} = 3$ besitzt. Allerdings kann mit m_1 und m_2 zusammen eine Anonymitätsgruppe gebildet werden, da alle drei Nachrichten gegenseitig in ihren Toleranzbereichen liegen und zudem keiner eine Anonymitätsanforderung von größer als 3 besitzt. Diese drei Nachrichten können nun transformiert und weitergeleitet werden, wobei ihre Sendeposition durch ein möglichst kleines Rechteck ersetzt wird, das alle drei Nachrichten der Anonymitätsgruppe enthält (schwarzes Rechteck in Abbildung 3.3b), und als Sendezeitpunkt der Zeitpunkt der Transformation verwendet wird. Die Nachricht m_3 konnte nicht verarbeitet werden und wird daher weiterhin vorgehalten, bis entweder eine Transformation möglich ist oder ihr zeitlicher Toleranzbereich überschritten wurde. In diesem Fall würde m_3 als unzustellbar verworfen.

In Simulationen zeigen Gedik und Liu, dass ihr Ansatz funktioniert und ein großer Anteil an Nachrichten trotz der beschriebenen Einschränkungen erfolgreich zugestellt werden kann. Sie nutzen dabei Anonymitätsanforderungen aus dem Intervall $[1 \ldots 5]$ (Zipf-Verteilung mit einem Parameter von $0,6$), eine mittlere zeitliche Toleranz von $30\,\mathrm{s}$ (Varianz: $12\,\mathrm{s}^2$) sowie eine mittlere räumliche Toleranz von $100\,\mathrm{m}$ (Varianz: $40\,\mathrm{m}^2$).

Bewertung Dem Konzept von Gedik und Liu merkt man relativ deutlich an, dass sie ein Konzept für anonymisierte Kommunikation mit *Location-based Services* beschrieben haben und der Einsatz für einen Positionsdienst nicht ohne Weiteres möglich ist. Dennoch ließe sich ein Dienst konstruieren, der Nachrichten von den Teil-

nehmern seines Zuständigkeitsbereichs vorhält und bei Bedarf – also im Falle einer Anfrage nach der Position eines Teilnehmers – eine entsprechende Unschärfe anhand der durch die Teilnehmer selbst bestimmten Rahmenbedingungen ermittelt.

CliqueCloak weist jedoch – ebenso wie der von Gruteser und Grunwald beschriebene Ansatz – die Schwäche auf, dass eine dritte Instanz verwendet wird, die für die Anonymisierung alleine zuständig ist und ihrerseits sämtliche Details kennt. So ein Dienst ist zwar denkbar, jedoch wäre es wünschenswert, wenn nicht das gesamte Vertrauen in eine einzelne Entität gesetzt werden müsste.

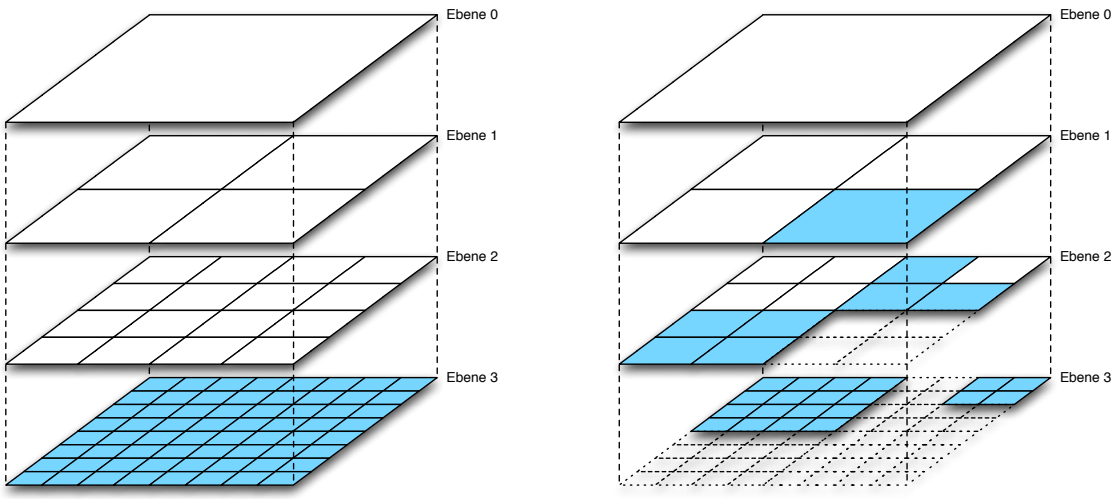
Weiterhin argumentieren die Autoren mit einer höheren Flexibilität gegenüber dem Ansatz von Gruteser und Grunwald, da jeder Sender einer Nachricht (auf einen Positionsdienst übertragen: jeder potentielle Empfänger von Nachrichten, der seine Position hinterlegt) die gewünschte minimale Größe der Anonymitätsgruppe selbst festlegen kann. In ihren Untersuchungen berücksichtigen Gedik und Liu jedoch lediglich Anonymitätsgruppen von ≤ 5, was von Gruteser und Grunwald als Mindestwert in den von ihnen durchgeführten Evaluationen festgelegt und auf Grund des Algorithmus in der Regel deutlich überschritten wurde. Gedik und Liu dagegen bieten dies als Maximalwert und ihre Ergebnisse zeigen, dass Anonymitätsgruppen aus fünf Teilnehmern nie überschritten wurden. An dieser Stelle bleibt somit die Frage offen, ob nicht größere Anonymitätsgruppen wünschenswert wären und ob das vorgestellte Konzept dies leisten kann oder ob dies zu einer sehr schlechten Zustellrate von Nachrichten führen würde.

Der Rechenaufwand zur Ermittlung der Anonymitätsgruppe ist zudem – nach Aussagen der Autoren – beträchtlich und die Skalierung dieses Konzepts ist somit stark beeinträchtigt. Dies ist vermutlich ebenfalls ein Grund dafür, dass Gedik und Liu nur kleine Anonymitätsgruppen zulassen, die einfacher zu kalkulieren sind.

3.2.3. New Casper

Aufbauend auf den vorangegangenen Arbeiten möchten Mokbel et al. mit ihrem in [MCA06] beschriebenen Ansatz weitere Optimierungen und Verbesserungen vornehmen. Der Name ihres Konzepts leitet sich nach eigenen Aussagen von dem gleichnamigen Cartoon-Charakter aus den 1960er Jahren ab, der sich selbst unsichtbar machen kann und als freundlicher Geist Menschen hilft. Ebenso wie sein Namensgeber soll der vorgeschlagene Dienst im Hintergrund arbeiten und dort sein gutes Werk verrichten.

Konkret schlagen die Autoren einen *Location Anonymizer* vor, der als Proxy zwischen den Teilnehmern eines mobilen Netzes und einer Datenbank – der *Location-based Database* – vermittelt. Dabei werden die von Seiten der Teilnehmer ankommenden Positionsupdates auf Grundlage individueller Profile mit einer räumlichen



(a) Basiskonzept **(b)** Optimierung

Abbildung 3.4.: Pyramidenstrukturen für Casper (nach [MCA06])

Unschärfe versehen und anschließend zur Speicherung an den Datenbankserver weitergeleitet. An diesen Server können anschließend von verschiedenen Entitäten (andere Teilnehmer oder Dienstleister von *Location-based Services*) Anfragen über die Position einzelner Teilnehmer oder aggregierte Daten („Wie viele Teilnehmer befinden sich in Bereich b?“) gestellt werden. Ein wichtiger Gestaltungsgrundsatz war der Einsatz von effizienten Berechnungs- und Speichermöglichkeiten für diese Informationen.

Mokbel et al. entschieden sich für eine hierarchische Datenstruktur, die von Tanimoto und Pavlidis in [TP75] als *Pyramid Data Structure* vorgestellt wurde. Sie besteht aus einer Reihe von Ebenen, die jeweils eine andere Granularitätsstufe abbilden (vgl. Abbildung 3.4a). Die von den Erfindern der Struktur gewählte Bezeichnung der Pyramide passt in diesem Fall nicht optimal, da hier jede Ebene das gesamte Versorgungsgebiet in jeweils einer anderen Granularitätsstufe abbildet; ursprünglich hatte jede Ebene auch eine andere Fläche repräsentiert, was zu einer Zuspitzung am oberen Ende führt, das auch als Wurzel bezeichnet wird.

In dieser Datenstruktur wird bei einem Übergang von einer Ebene in die nächste die Anzahl an Zellen vervierfacht, was zu einer Zellzahl von 4^h auf der Ebene h führt. Eine Zelle auf der *Ebene* h wird somit auf der Ebene $h + 1$ in vier Zellen unterteilt; das führt dazu, dass eine eindeutige Beziehung zwischen Elternzelle (auf der höheren Ebene) und Kindzellen (auf der darunterliegenden Ebene) besteht.

Bei *Casper* kann jeder Teilnehmer individuell festlegen, wie streng seine Anforderungen an die Privatsphäre sind, indem er eine Mindestgröße der Anonymitätsgruppe

k sowie der zurückzuliefernden Region A_{min} festlegt. Der *Location Anonymizer* soll diesen Anforderungen genügen und eine geeignete Zielregion R zurückliefern, so dass gilt: $k_R \gtrsim k$ und $A_R \gtrsim A_{min}$. Dazu wird die oben beschriebene Pyramidenstruktur eingesetzt, bei der jede Zelle über das Tupel (cid, N) repräsentiert wird, wobei *cid* die Kennung der Zelle darstellt und N die Anzahl an Teilnehmern, die sich aktuell innerhalb der Zelle befinden, angibt. Weiterhin wird eine Hash-Tabelle verwendet, in der jeder Teilnehmer mit dem Tupel $(uid, profile, cid)$ hinterlegt ist, das sich aus der eindeutigen Kennung des Teilnehmers *uid*, seinen Anforderungen an die Privatsphäre ($profile$) und der Kennung der aktuell besuchten Zelle zusammensetzt. Die Kennung der Zellen, *cid*, bezieht sich dabei stets auf Zellen in der untersten Ebene der Datenstruktur (eingefärbt in Abbildung 3.4a).

Wann immer ein Teilnehmer ein *Location Update* an den Dienst schickt, wird anhand seiner übermittelten Position überprüft, ob er sich nun in einer anderen Zelle als der hinterlegten befindet. Nur wenn dies der Fall ist, muss der Datensatz des Teilnehmers aktualisiert werden. Weiterhin wird in diesem Fall der Wert N der alten Zelle um eins reduziert und in der neuen Zelle um eins erhöht. Dies muss rekursiv so lange auf den Elternzellen durchgeführt werden, bis eine gemeinsame Elternzelle erreicht wird. Dieser Aufwand ermöglicht in der Folge sehr performante Abfragen auf der Datenstruktur.

Geht eine Positionsabfrage nach einem Teilnehmer bei dem Dienst ein, so wird der *Cloaking Algorithm* herangezogen, um den Datenschutzanforderungen des angefragten Teilnehmers genüge zu leisten (vgl. Algorithmus 3.2). Dabei wird jeweils die aktuelle Zelle daraufhin untersucht, ob die Anforderungen hinsichtlich Anonymitätsgruppe (k) sowie der Mindestgröße der zurückzuliefernden Region (A_{min}) erfüllt sind. Ist dies nicht der Fall, wird die aktuelle Zelle *cid* jeweils probehalber um ihre horizontale und vertikale Nachbarzelle erweitert und erneut eine Überprüfung durchgeführt. Eine Nachbarzelle ist dabei so definiert, dass sie direkt (horizontal oder vertikal) neben der betrachteten Zelle liegt und zudem die gleiche Elternzelle besitzt. Somit ist eine Eindeutigkeit der Nachbarzellen sichergestellt.

Können auch in Kombination mit den Nachbarzellen die Anforderungen des Teilnehmers nicht erfüllt werden, so wird in einem rekursiven Ablauf die Elternzelle herangezogen und die Untersuchungen laufen erneut ab. Dieser Algorithmus terminiert spätestens auf *Ebene 0*, da vorausgesetzt wird, dass A_{min} kleiner als das Versorgungsgebiet (entspricht *Ebene 0*) und k kleiner als die Gesamtteilnehmerzahl sein muss.

Mokbel et al. stellen in der Folge eine Optimierung der Datenstruktur vor, die Abfragen mit einer geringeren Anzahl an rekursiven Aufrufen ermöglichen soll, dabei jedoch höheren Wartungsaufwand bei Änderungen der Positionen von Teilnehmern erfordert. In der optimierten Struktur können Zellen sämtlicher Ebenen – nicht nur

Algorithmus 3.2 Bottom-up Cloaking-Algorithmus (nach [MCA06])

```
function BottomUpCloaking(k, A_min, cid)
    if cid.N ≥ k and cid.Area ≥ A_min then
        return cid.Area
    end if
    cid_V ← The vertical neighbor cell of cid
    cid_H ← The horizontal neighbor cell of cid
    N_V ← cid.N + cid_V.N
    N_H ← cid.N + cid_H.N
    if (N_V ≥ k ∨ N_H ≥ k) ∧ 2cid.Area ≥ A_min then
        if (N_H ≥ k ∧ N_V ≥ k ∧ N_H ≤ N_V) ∨ N_V < k then
            return cid.Area ∪ cid_H.Area
        else
            return cid.Area ∪ cid_V.Area
        end if
    else
        BottomUpCloaking(k, A_min, cid.Parent)
    end if
end function
```

die der untersten – als Aufenthaltsorte der Teilnehmer herangezogen werden. Bei der Speicherung der aktuellen Zelle im Datensatz des Teilnehmers wird die entsprechende Zelle derjenigen Ebene gewählt, die bereits die Anforderungen bezüglich der räumlichen Auflösung (A_{min}) des Teilnehmers erfüllen. Wenn nun ein Bereich der untersten Ebene von keinem einzigen Teilnehmer referenziert wird, muss dieser auch nicht in der Datenstruktur gespeichert werden. In dem Beispiel in Abbildung 3.4b wird auf *Ebene 2* die Zelle rechts unten nicht benötigt, da alle Teilnehmer dieses Bereichs so strenge Anforderungen bezüglich der Mindestgröße der Region besitzen, dass sie auf *Ebene 1* oder *Ebene 0* referenziert werden. Die entsprechenden vier Kindzellen auf *Ebene 2* und die 16 Enkelzellen auf *Ebene 3* müssen nicht abgebildet werden. In der Abbildung ist jeweils die unterste berücksichtigte Ebene der Bereiche farblich hervorgehoben.

Die Autoren weisen allerdings zu Recht darauf hin, dass sich diese Optimierung nur in Netzen mit moderater Bewegung der Knoten günstig auf die Performance auswirkt. In sehr dynamischen Netzen entstehen sehr hohe Kosten bei dem Neuaufbau der Datenstruktur und der Ermittlung der benötigten Ebenen. Somit lässt sich feststellen, dass die Wahl des Algorithmus und der Datenstruktur sehr stark von dem Verhältnis der Anzahl an Positionsanfragen zu der Anzahl an Positionsupdates abhängt.

Eine Reihe durchgeführter Simulationen des vorgestellten Konzepts zeigen einige interessante Aspekte. Auf dem Straßennetz von Hennepin County, Minnesota, USA wurde die Bewegung von Teilnehmern simuliert, die sich gleichverteilt in dem Gebiet aufhielten. Bei den Untersuchungen wurden Wertebereiche ausgewählt, die teilweise deutlich restriktiver waren als die der vorangegangenen Konzepte. Beispielsweise wurde k gleichverteilt aus dem Intervall $[1 \ldots 50]$ gewählt und in manchen Untersuchungen auf bis zu 200 erhöht.

Es zeigt sich bei *Casper* eine sehr gute Skalierbarkeit hinsichtlich der Teilnehmerzahl, da bei einer dichteren Besetzung der Region die Anzahl an rekursiven Aufrufen beim Durchlaufen des *Cloaking*-Algorithmus geringer ist, was sich positiv auf die benötigte Rechenzeit auswirkt. Die Höhe der Pyramide selbst dagegen zeigt einen quadratischen Zusammenhang zu der Rechenzeit, die geforderte Mindestgröße der Anonymitätsgruppe (k) führt im einfachen Fall zu einem unterproportionalen Anstieg der Rechenlast; im optimierten Szenario dagegen zeigt sich ein konstanter Trend.

Bewertung Mokbel et al. stellen mit *Casper* einen weiteren Ansatz dar, der einige Schwächen (insbesondere in der Performance) der vorangegangen Konzepte beheben kann. Die Simulationen zeigen, dass das System in einem bestimmten Rahmen skaliert und deutlich besseren Schutz bieten kann als beispielsweise *CliqueCloak*. Allerdings sind die tatsächlichen Rechenzeiten der Abfragen schwer einzuschätzen, da die Autoren keine Informationen über die Maschine, auf der die Operationen durchgeführt wurden, liefern. Insgesamt konnten jedoch bei ihnen alle Abfragen in einem einstelligen Millisekundenbereich durchgeführt werden.

Trotz der Weiterentwicklung mit Hilfe optimierter Datenstrukturen bleibt das Problem bestehen, dass der Positionsdienst sämtliche Aufenthaltsinformationen vorhält und somit über vollständiges Wissen verfügt. Weiterhin besitzt der Dienst auch noch das Wissen über die Anforderungen von Teilnehmern an die Privatsphäre und kann somit theoretisch eine Liste von zum jeweiligen Zeitpunkt besonders paranoiden Knoten identifizieren, deren aktuelle Position möglicherweise von besonderem Interesse sein könnte.

3.3. Weitere Ansätze

Auf Verschleierung des exakten Aufenthaltsortes eines Teilnehmers basierende Konzepte bringen den Nachteil mit sich, dass unter Umständen eine notwendige Maximalgröße des mitgeteilten Gebiets nicht unterschritten werden kann, um die Anonymitätsgruppe ausreichend groß zu halten. Diese Maximalgröße kann auf Grund

technischer Restriktionen gefordert werden, da sich ein Empfänger in hinreichender räumlicher Nähe zu dem zurückgelieferten Ort aufhalten muss, damit ihm eine Nachricht erfolgreich zugestellt werden kann. Ein Kriterium könnte somit die technisch bedingte Funkreichweite der einzelnen Teilnehmer darstellen, aber auch andere Kriterien sind – basierend auf den eingesetzten Routingmechanismen und -algorithmen – denkbar.

Um diesen Nachteilen zu begegnen, wurden in der Literatur alternative Ansätze vorgestellt, die ebenfalls die Privatsphäre von Teilnehmern in einem mobilen Ad-hoc-Netz schützen sollen, dabei jedoch andere Herangehensweisen wählen, um diesen Schutz zu realisieren.

3.3.1. Token-based Abuser Identification

Wu und Nita-Rotaru beschreiben in ihrer Publikation [WNR05] einige Angriffe auf verschiedene Aspekte von Positionsdiensten und entwickeln Gegenmaßen. Dabei betrachten sie unter anderem auch Aspekte wie falsche Positionsangaben durch Teilnehmer oder das Abhören von *Location Update*-Nachrichten. Diese Untersuchungen sollen jedoch an dieser Stelle keine Berücksichtigung finden und es sei lediglich auf [WNR05] verwiesen.

Interessant ist jedoch der von den Autoren beschriebene Mechanismus, um die Erstellung von Bewegungsprofilen von Teilnehmern durch häufiges Abfragen ihres Aufenthaltsortes am Positionsdienst zu verhindern. Wu und Nita-Rotaru gehen dabei von einem durch andere Netzteilnehmer bereitgestellten Positionsdienst in einem mobilen Ad-hoc-Netzwerk aus; ihr Konzept lässt sich jedoch ohne Schwierigkeiten auf jeden beliebigen Dienst übertragen.

Grundsätzlich arbeitet der vorgeschlagene Positionsdienst wie jeder andere auch: Jeder Teilnehmer schickt an ihn in regelmäßigen Abständen seine Aufenthaltsinformationen, die anderen Teilnehmern zur Verfügung gestellt werden können, wenn diese Nachrichten übermitteln möchten. Die Kommunikation mit dem Positionsdienst erfolgt dabei stets verschlüsselt und authentifiziert. Wu und Nita-Rotaru machen konkrete Vorschläge für den Einsatz kryptographischer Systeme, die jedoch ebenfalls hier nicht umfassend ausgeführt werden sollen, da auch Alternativen ohne Einschränkungen in der Funktionalität denkbar wären.

Jeder Teilnehmer erhält in dem beschriebenen System ein Konto an *Tokens*, mit denen jeweils eine Anfrage an den Positionsdienst (bzw. die entsprechende Antwort) erkauft werden kann. Der Positionsdienst führt über die *Token*-Guthaben sämtlicher Teilnehmer Buch und verwaltet diese auch. Stellt also ein Teilnehmer eine Positionsanfrage, so erhält er von dem Positionsdienst die gewünschte Antwort, und

dieser reduziert gleichzeitig die Anzahl der verfügbaren *Tokens* um eins. Verfügt ein Teilnehmer über keinerlei *Tokens*, so erhält er von dem Positionsdienst auch keine Antworten mehr auf seine Anfragen.

Das System soll jedoch nicht dazu dienen, generell nur eine geringe Anzahl an Anfragen zu ermöglichen, sondern nur die Anzahl an unerwünschten Anfragen stark beschränken. Handelt ein anfragender Teilnehmer also entsprechend des Protokolls und initiiert eine Kommunikation mit dem angefragten Teilnehmer, so kann letzterer dem Positionsdienst dies bestätigen und somit eine Rückerstattung des eingesetzten *Tokens* veranlassen. Um den Kommunikationsaufwand gering zu halten, wird die Bestätigung der Kommunikation einfach zusammen mit dem nächsten Positionsupdate übertragen.

Damit ein Angreifer nun nicht lediglich eine belanglose Nachricht an den von ihm zum Zweck der Erstellung eines Bewegungsprofils angefragten Teilnehmer schicken muss, um wieder über *Tokens* zu verfügen, kann der angefragte Teilnehmer stets entscheiden, ob er das Kommunikationsinteresse des Anfragenden für legitim hält – und somit eine Rückerstattung auslöst – oder nicht. Ein Angreifer könnte somit nur einige wenige Anfragen stellen, die von den angefragten Teilnehmern als unangemessen betrachtet werden.

Der Positionsdienst selbst überwacht ebenfalls die Anfragen und die möglichen Rückerstattungen und kann somit auch feststellen, ob ein Angreifer versucht, das Bewegungsprofil eines Teilnehmers zu erstellen. Da die Anfragen authentifiziert sind, kann eine illegitime Nutzung geahndet werden. Zudem ist es möglich, den betroffenen Teilnehmer über die Bedrohung zu informieren.

Ein Teilnehmer erhält für seine Anfragen initial nicht nur einen, sondern mehrere *Tokens*. Dies hat den Hintergrund, dass es stets vorkommen kann, dass ein legitim angefragter Benutzer noch kein Positionsupdate gesendet hat, um das verwendete *Token* rückzuerstatten, aber dennoch bereits eine weitere Anfrage an den Positionsdienst gestellt werden soll. Weiterhin können Nachrichten auf dem Weg ihrer Zustellung verloren gehen (dies passiert je nach Dichte der Teilnehmer und dem eingesetzten Routingverfahren mit einer gewissen Wahrscheinlichkeit) und somit kann es vorkommen, dass auch der Steller einer legitimen Positionsanfrage keine Rückerstattung erhalten kann.

Wu und Nita-Rotaru führen eine Reihe von Simulationen durch, um die minimale Anzahl an initial benötigten *Tokens* zu bestimmen. Sie arbeiten dabei mit einem Gebiet aus $1000\,\mathrm{m} \times 1000\,\mathrm{m}$, in dem sich zwischen 50 und 200 Teilnehmer gleichverteilt bewegen. Ihre Untersuchungen ergeben, dass mit einer Mindestanzahl von drei *Tokens* die Wahrscheinlichkeit, dass keine Kommunikation zustande kommt, verschwindend gering wird. Insgesamt hängt die benötigte Anzahl jedoch sehr stark

von der Dichte der Netzteilnehmer ab. So ist für sehr schwach besetzte Regionen eine deutlich größere Anzahl an *Tokens* erforderlich.

Bewertung Mit ihrem *Token*-basierten Ansatz gelingt es Wu und Nita-Rotaru einen Dienst zu schaffen, der die Privatsphäre von Teilnehmern effektiv schützen kann, ohne dass dabei Informationen unpräzise wiedergegeben werden müssen und somit die damit einhergehenden (oben beschriebenen) Probleme umgangen werden können. Der Einsatz der *Tokens* verhindert das Erstellen von Bewegungsprofilen und bietet dennoch ausreichend Flexibilität, so dass auch fehlgeschlagene Kommunikationen nicht sofort zu einem *fail stop*, also einem bei Auffälligkeiten aus Sicherheitsgründen vorgenommenen Stopp des Dienstes, führen.

Die Verwaltung der *Tokens* direkt bei dem Positionsdienst ermöglicht eine relativ effiziente Gestaltung dieses Berechtigungskonzepts, da in vielen vorgeschlagenen *VANET*-Sicherheitsinfrastrukturen ohnehin Entitäten existieren, die eine Verwaltung der Teilnehmer im Netz und beispielsweise ihrer Pseudonyme (vgl. [Plö09]) realisieren. Ein weiteres Datenfeld dieser Datenbank (für die Verwaltung der *Tokens*) dürfte somit kaum ins Gewicht fallen. Alternativ dazu könnten blinde Signaturen[1] zum Einsatz kommen, um eine anonyme Vergabe von *Tokens* zu ermöglichen, die auch bei verteilten Positionsdiensten unproblematisch vorgezeigt werden können. Diese sind jedoch mit aufwändigen kryptographischen Operationen und damit verbundenen organisatorischen Regelungen (Einsatz einer Public-Key-Infrastruktur) zu bewerkstelligen.

Obwohl sich dieser Ansatz als recht vielversprechend darstellt, löst er weiterhin nicht das Problem, dass dem Positionsdienst (unabhängig von der konkreten Ausgestaltung) uneingeschränktes Vertrauen entgegengebracht werden muss. Da er vollständige Informationen besitzt, ist er stets in der Lage, alle Teilnehmer zu überwachen oder mit Angreifern zusammenzuarbeiten. Außerdem ist der Ablauf der Schutzmechanismen für den einzelnen Teilnehmer nicht zu erkennen, und somit kann keine Überprüfung auf eine korrekte Funktionsweise durchgeführt beziehungsweise Fehlverhalten kann nicht ohne Weiteres erkannt werden.

[1] Blinde Signaturen sind ein von David Chaum in [Cha82, Cha83] vorgestelltes Verfahren, mit dem der Aussteller einer Signatur Daten digital mit einem asymmetrischen Kryptosystem unterschreiben kann, ohne den Inhalt zu kennen. Der Empfänger der Signatur hat anschließend die Möglichkeit, die *geblendeten* – also unkenntlich gemachten Daten – wiederherzustellen, ohne dabei die Signatur zu beschädigen. In der analogen Welt könnte dies als „Blankounterschrift“ bezeichnet werden.

3.3.2. Anonymous Location Service

Der von Zhou und Yow in [ZY05] vorgestellte *Anonymous Location Service* (*ALS*) ist Teil ihres umfassenden Konzeptes zur Realisierung datenschutzfreundlicher Kommunikation. Sie stellen darin ebenfalls eine *Anonymous Neighbor Table* vor, die der Verwaltung benachbarter Teilnehmer für das eingesetzte *Anonymous Greedy Forwarding* dient. Diese Weiterleitungsstrategie (vgl. Abschnitt 2.2.3.2) ermöglicht den Transport von Nachrichten, ohne die Identitäten der weiterleitenden Knoten, des Absenders oder des Empfängers zu kennen. Diese Aspekte der Arbeit sollen an dieser Stelle jedoch nicht vertieft werden und es sei auf die oben genannte Publikation verwiesen.

Mit dem von ihnen beschriebenen *ALS* möchten die Autoren jedoch explizit eine Lösung für das eingangs beschriebene Problem des Erstellens von Bewegungsprofilen durch häufiges Abfragen der Positionen der Opfer vorschlagen. Um den entsprechenden Schutz bieten zu können, basiert dieser Vorschlag auf dem Einsatz von asymmetrischer Kryptographie (vgl. Abschnitt 2.1.3.2).

Das Konzept von Zhou und Yow setzt eine existierende Public-Key-Infrastruktur voraus sowie daraus resultierend die Verfügbarkeit von authentischen öffentlichen Schlüsseln aller Teilnehmer des Ad-hoc-Netzes. Weiterhin bringen die Autoren einen dezentralen Positionsdienst zum Einsatz; allerdings funktioniert ihr Konzept ebensogut mit einem zentral organisierten Dienst.

Der entscheidende Unterschied zu einem gewöhnlichen Positionsdienst besteht darin, dass die Teilnehmer bei Positionsupdates ihre Identität und ihren Aufenthaltsort nicht im Klartext, sondern verschlüsselt für einen potentiell interessierten Teilnehmer bereitstellen. Damit Teilnehmer A seine Position loc_A für Teilnehmer B bei dem Positionsdienst S zur Verfügung stellen kann, verschlüsselt er diese mit dem öffentlichen Schlüssel c_B von B und übermittelt das Chiffrat $c_B\,(loc_A)$ an S. Da A jedoch nicht möchte, dass diese Information unter seiner Identität A beim Positionsdienst abgelegt ist, ergänzt er die Nachricht vor dem Senden um einen ebenfalls mit c_B verschlüsselten Block, der aus den Identitäten A und B besteht. Der Positionsdienst verwendet also $c_B\,(A, B)$ als Schlüssel seiner Datenbank, unter dem der Eintrag $c_B\,(loc_A)$ abgelegt ist.

Möchte B nun die Position von A erfragen, so verschlüsselt er die Identität von A und die eigene Identität – B – mit dem eigenen öffentlichen Schlüssel c_B, sendet den Schlüsseltext $c_B\,(A, B)$ in der Positionsanfrage an S und erhält von diesem den zugehörigen Eintrag $c_B\,(loc_A)$, den er mit seinem zu c_B gehörenden privaten Schlüssel d_B entschlüsseln kann. Damit steht ihm die Position von A zur Verfügung und B kann eine Nachricht absenden. Durch die Verschlüsselung wird sichergestellt, dass sowohl auf den Übertragungswegen von und zu S sowie in dessen Datenbank

die Informationen niemals ungeschützt sind. Zudem stellt A die eigenen Positionsinformationen gezielt für B zur Verfügung (sie sind mit dem öffentlichen Schlüssel von B verschlüsselt) und kann somit sicher sein, dass kein unbefugter Dritter ein Bewegungsprofil mit Hilfe der bei S hinterlegten Daten generiert.

Bewertung Der *Anonymous Location Service* von Zhou und Yow berücksichtigt durch den Einsatz verschlüsselter Datensätze erstmals die Möglichkeit, dass der Positionsdienst entweder selbst nicht vertrauenswürdig sein muss oder aber Opfer eines Angriffs wird und eine dritte Partei in den Besitz der hinterlegten Informationen kommt. Bei *ALS* sind diese Bedrohungen durch den Einsatz kryptographischer Systeme ausgeschaltet.

Allerdings wird dieser Vorteil im Schutz der Privatsphäre der Teilnehmer sehr teuer erkauft: Neben dem rechenintensiven Einsatz von asymmetrischer Kryptographie könnten die hinterlegten Positionsinformationen nicht spontan beliebigen Teilnehmern zur Verfügung gestellt werden, sondern es werden sogar individuelle Datensätze für jeden einzelnen potentiellen Sender einer Nachricht benötigt. Möchte also ein Teilnehmer zehn anderen die Möglichkeit einräumen, ihn zu kontaktieren, muss bei jedem einzelnen Positionsupdate die Postion zehnmal mit jeweils unterschiedlichen öffentlichen Schlüsseln verschlüsselt und an den Positionsdienst übermittelt werden. Das bedeutet einen enormen Overhead, der vermutlich stärker als linear mit der Anzahl an Teilnehmern wächst und somit schlecht skaliert.

Zhou und Yow sind sich dieses Problems bewusst und argumentieren, dass die Position eines Teilnehmers möglicherweise nur in bestimmten Zeitfenstern geschützt werden müsse und dieser ansonsten seine Informationen im Klartext bereitstellen könne; der Positionsdienst also auch eine herkömmliche Dienstqualität anbieten könnte. Diese Aussage ist sehr problematisch, da sie dazu führt, dass ein Teilnehmer selbst die Information preisgibt, wann er sich in Regionen aufhält, die er geheimhalten möchte und sich somit unter Umständen verdächtig macht. Außerdem sollte ein Schutz der Privatsphäre für alle Teilnehmer zu jedem Zeitpunkt aktiv sein, da es andernfalls über große Zeiträume für einen Angreifer leicht möglich wäre, Bewegungsprofile zu erstellen.

Somit ist dieser Ansatz, der auf den ersten Blick sehr vielversprechend wirkt, weil er erstmals auch einen Schutz vor dem Positionsdienst bietet, für die Praxis wohl untauglich, da er zu viele Einschränkungen mit sich bringt. Neben dem großen Overhead muss ein Empfänger alle potentiellen Sender von Nachrichten kennen und für sie gezielt Informationen bereitstellen oder auf einen Schutz seiner Privatsphäre vollständig verzichten.

3.4. Fazit

Die vorgestellten Positionsdienste (oder Dienste, die für Location-based Services eine ähnliche Funktion erfüllen) zeigen, wie schwierig es ist, eine ausreichende Performance des Systems bereitzustellen, dabei jedoch auch den Schutz der Aufenthaltsinformationen (Ort und Zeitpunkt) der einzelnen Teilnehmer effektiv zu schützen. Insbesondere ein Schutz vor dem Positionsdienst selbst stellt sich dabei als besondere Schwierigkeit heraus, weshalb in den meisten Fällen der Positionsdienst als vertrauenswürdig angenommen wird.

Diese Annahme ist – wie bereits mehrfach dargelegt wurde – als problematisch zu betrachten, da ein Missbrauch von Daten, die vorhanden sind, kaum ausgeschlossen werden kann (vgl. Abschnitt 2.1.2.1). Selbstverständlich kann zudem natürlich auch ein erfolgreicher Angriff auf eine derart vielversprechende Datenbank niemals vollständig ausgeschlossen werden. Somit muss es eine zentrale Aufgabe bei der Konzeption eines Positionsdienstes sein, dass nicht nur das auf häufigen Abfragen basierende Erstellen von Bewegungsprofilen durch potentielle Sender von Nachrichten effektiv verhindert wird, sondern dass auch ein Vertrauen in den Positionsdienst selbst nicht erforderlich ist oder Missbrauch zumindest nachgewiesen werden kann.

In diesem letzten Punkt stellen sich die bisherigen Lösungen als völlig unzureichend dar. Lediglich das Konzept von Zhou und Yow möchte an dieser Stelle Schutz bieten; dabei gelingt es den Autoren jedoch nicht, den vorgestellten *Anonymous Location Service* so zu gestalten, dass der gebotene Schutz nicht zu einer drastischen Einschränkung der Funktionalität führt und damit möglicherweise bewirkt, dass viele Nutzer über große Zeiträume überhaupt keinen Schutz ihrer Privatsphäre erlangen können.

3.5. Positionsdienst mit minimalen Vertrauensanforderungen

Als Lösungsansatz für die beschriebene Problematik wurde ein Konzept für einen datenschutzfreundlichen Positionsdienst, der das nötige Vertrauen in die beteiligten Instanzen minimiert, erarbeitet und bereits in [SBF10] veröffentlicht. Ein wesentlicher Bestandteil dieses Ansatzes ist der Einsatz von Kommunikationsmixen (vgl. Abschnitt 2.1.4) auf Übertragungsstrecken zwischen den Teilen eines hierarchisch verteilten Positionsdienstes und dem daraus resultierenden Schutz vor Verkehrsanalysen. Weiterhin werden alle Informationen stets verschlüsselt gespeichert und übertragen und liegen lediglich am Ort ihrer Erstellung oder ihrer Nutzung im Klartext vor.

3.5.1. Angreifermodell

Grundsätzlich kann eine Bedrohung durch Außenstehende vernachlässigt werden, denn diese können mit minimalem Aufwand (sie müssen lediglich ein *VANET*-fähiges Fahrzeug besitzen und evtl. registrieren) zu Insidern, also Teilnehmern des Systems – und damit mächtigeren Angreifern – werden. Insider verfügen nämlich zudem über das zur Kommunikation benötige kryptographische Material, wenn sie auch keinen direkten Zugriff darauf haben, da es in einem vertrauenswürdigen Hardwarebaustein hinterlegt ist (vgl. Abschnitt 2.3.2). Dies zwingt sie auch dazu, sich an gewisse Regeln zu halten und macht das Erzeugen gefälschter Nachrichten nahezu unmöglich. Dennoch können sie als *„honest but curious"* bezeichnet werden, worunter ein Angreifer zu verstehen ist, der zwar kein Protokoll und keine Regeln verletzt, aber dennoch im Rahmen seiner Möglichkeiten versucht, möglichst viele Informationen über andere Teilnehmer oder über Nachrichteninhalte herauszufinden.

Ein weiterer zu berücksichtigender Angreifer ist der Positionsdienst selbst. Wie bereits zuvor ausführlich dargelegt, besteht stets die Gefahr des Missbrauchs von Daten, wenn diese an einer zentralen Stelle zusammengeführt und gespeichert werden. Da der einzelne Benutzer kaum über Kontrollmöglichkeiten verfügt, sollte er auch möglichst kein Vertrauen in die zentralen Infrastruktureinheiten haben müssen, die möglicherweise zusammenarbeiten oder sogar von einer einzelnen Organisation betrieben werden. Allerdings ist davon auszugehen, dass ein häufiges Abweichen von den Protokollen durch den Positionsdienst den Teilnehmern früher oder später auffallen und zu negativen Konsequenzen führen würde. Zudem stellen zentrale Datenbanken ein lohnendes Ziel für dritte Angreifer dar, deren Abwehr niemals vollständig garantiert werden kann.

Für die in diesem Konzept eingesetzten Mix-Kaskaden gilt lediglich die gängige Annahme über ihre Vertrauenswürdigkeit: Wenigstens ein Mix jeder Kaskade arbeitet nicht mit allen anderen zusammen um die Kommunikationsverbindung aufzudecken (vgl. Abschnitt 2.1.4). Somit sollte sichergestellt werden, dass die Mixe der Kaskade jeweils von unterschiedlichen Organisationen betrieben werden.

3.5.2. Notation

In diesem Abschnitt wird die folgende Notation verwendet:

- c_j bezeichnet den öffentlichen Schlüssel des Mix j.
- d_j ist der zu c_j zugehörige private Schlüssel von Mix j.
- $k_{i,j}$ ist ein symmetrischer Schlüssel, der von Teilnehmer i erzeugt und von Mix j zur Verschlüsselung genutzt wird.

- $c_j(M)$ bezeichnet die asymmetrische Verschlüsselung einer Nachricht M mit dem Schlüssel c_j.
- $\{M\}$ beschreibt eine von einem Benutzer i mehrschichtig verschlüsselte Nachricht M zum Transport über eine Mix-Kaskade (vgl. Abschnitt 2.1.4) bestehend aus n Mixen ($j = [1..n], n \geq 2$).
  Es ist somit eine Kurzschreibweise für $c_1(k_{i,1}, c_2(k_{i,2}, \ldots c_n(k_{i,n}, M) \ldots))$. Der symmetrische Schlüssel $k_{i,j}$ steht jeweils Mix j zur Verschlüsselung einer Nachricht an Benutzer i zur Verfügung; Teile einer Nachricht werden jeweils mit einem Komma konkateniert.
- $[M]$ steht für eine mehrfach (n-fach) mit symmetrischen Schlüsseln $k_{i,j}$ ($j = [1..n]$, $n \geq 2$) verschlüsselte Nachricht M. Somit ist dies eine Kurzschreibweise für $k_{i,n}(\ldots k_{i,2}(k_{i,1}(M) \ldots)$.
- UID_i ist die eindeutige Kennung von Teilnehmer i (vgl. Abschnitt 2.3.1).
- P_i repräsentiert das aktuell gültige Pseudonym von Teilnehmer i. Das darauf folgende Pseudonym wird mit P_i' bezeichnet.
- loc_i ist die aktuelle Geoposition von Teilnehmer i. Die darauf folgende gültige Position wird von loc_i' symbolisiert.
- MP_i ist das aktuelle zufällig gewählte Mapping-Pseudonym des Teilnehmers i.
- Falls nicht explizit darauf hingewiesen wird, werden die Begriffe *Teilnehmer*, *Fahrzeug*, *Benutzer* und *Knoten* synonym verwendet.

3.5.3. Überblick und Architektur

Das System lehnt sich an ein Konzept von Federrath an, das für Mobilfunknetze vorgestellt wurde (vgl. [FJP96, Fed99a, Fed99b]).

Zum Einsatz kommt – ähnlich wie bei Mobilfunknetzen – eine zweischichtige hierarchische Architektur bestehend aus einer zentralen Einheit (bezeichnet als *Central Location Service, CLS*, ähnlich dem *Home Location Register, HLR* in Mobilfunknetzen) und vielen räumlich verteilten Komponenten (bezeichnet als *Local Location Service, LLS*, ähnlich dem *Visitor Location Register, VLR*), siehe Abbildung 3.5. Jeder *LLS* ist für eine bestimmte Region zuständig, die als *Service Area, SA* bezeichnet wird. In jeder *SA* befindet sich eine Reihe von *Roadside Units, RSU*, über die eine Kommunikation zwischen diesem Backbone und dem eigentlichen *VANET* möglich ist. Diese *RSU* sind reine Sende- und Empfangsmasten und verfügen über keinerlei Datenspeicher.

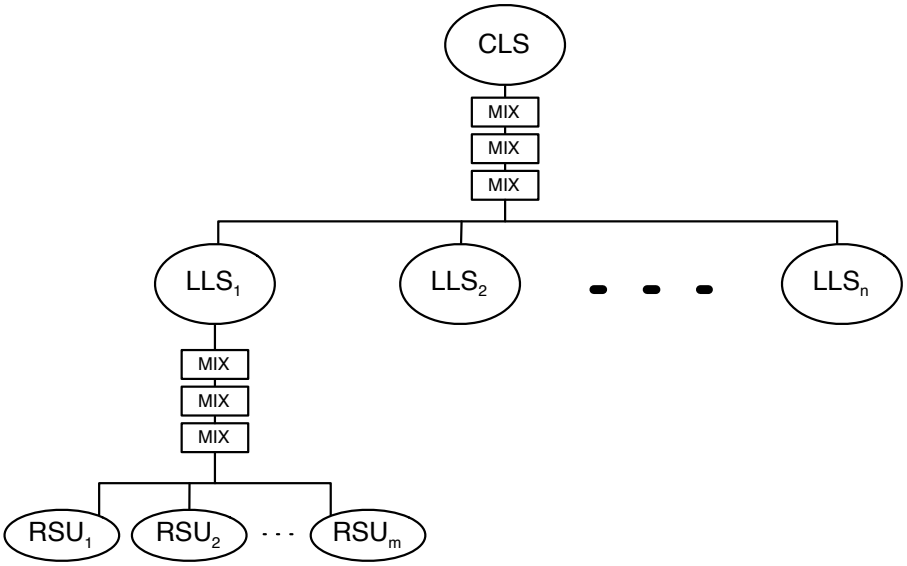


Abbildung 3.5.: Architektur des Backbones

Der *CLS* kommuniziert über eine Mix-Kaskade bestehend aus l Mixen ($l \geq 2$) mit n *LLS*. Jeder dieser *LLS* ist seinerseits über eine Mix-Kaskade mit m *RSU*s verbinden. Somit wird jede Nachricht zwischen einem Teilnehmer und dem für ihn zuständigen *LLS* (da er sich in dessen *SA* befindet) von einer Kaskade gemixt, eine Nachricht an den *CLS* sogar von zweien (vgl. Abbildung 3.5). Zudem ist die Route von einem Teilnehmer zu jeder Instanz eindeutig festgelegt, was aus den Eigenschaften einer Baumstruktur (die hier abgebildet wird) resultiert; dort ist ebenfalls ein Pfad von einem Blatt[2] in Richtung Wurzel[3] stets eindeutig (vgl. [Wir75]).

Mit der hierarchischen Unterteilung in *CLS* und *LLS* geht eine Trennung der Informationen (und damit des Vertrauens) einher: Der *CLS* kennt in diesem Konzept die Identitäten aller Teilnehmer des *VANET*s, verfügt jedoch nicht über Informationen über ihre Aufenthaltsorte. Den *LLS* sind die groben Aufenthaltsorte der bei ihnen verwalteten Teilnehmer bekannt, da sich diese in ihren Zuständigkeitsbereichen aufhalten müssen; sie kennen von ihnen jedoch nur ein Pseudonym, welches sie nicht mit ihren eindeutigen Identitäten, UID_i, in Einklang bringen können.

Um erreichbar zu sein, muss sich ein Teilnehmer bei seinem aktuell zuständigen *LLS* und dem *CLS* registrieren. Während er sich fortbewegt, müssen regelmäßig Updates

[2] Als Blatt werden in Baumstrukturen Knoten an dem verzweigten Ende bezeichnet, die ihrerseits keine Nachfolger besitzen.

[3] In einer Baumstruktur versteht man unter der Wurzel das Ursprungselement, dem alle anderen Knoten hierarchisch untergeordnet sind und das keinen Vorgänger besitzt. Es wird in der graphischen Darstellung meist oben abgebildet.

über die veränderte Position an den Positionsdienst übertragen werden. Findet die Bewegung lediglich innerhalb des aktuellen *SA* statt, so muss zur Aktualisierung der Position oder des Pseudonyms lediglich der *LLS* kontaktiert werden. Erst wenn die *SA* gewechselt wird, muss eine Update-Nachricht an den *CLS* gesendet werden.

Soll ein Nachrichtenaustausch eingeleitet werden, sendet ein anfragender Teilnehmer seine Position und Kennung über sein *LLS* an den *CLS*. Von dort werden diese Daten dem angefragten Teilnehmer über dessen *LLS* weitergeleitet. Jener kann nun entscheiden, ob er einen Nachrichtenaustausch einleiten möchte oder die Kontaktanfrage einfach verwirft.

Damit die Funktionalität des Systems sichergestellt werden kann, ist eine Public-Key-Infrastruktur (*PKI*) zwingend erforderlich. Dies bedeutet, dass es eine oder mehrere vertrauenswürdige Zertifizierungsstellen (*Certification Authorities*, *CA*s) gibt, die die öffentlichen Schlüssel der eingesetzten Mixe unterschreiben und damit sicherstellen, dass diese authentisch sind. Zudem wird im Folgenden angenommen, dass jeder Teilnehmer über sämtliche (unterschriebene) öffentliche Schlüssel der Mixe seines eindeutigen Pfades bis zum *CLS* verfügt oder diese jederzeit erhalten kann. Er kann die Signaturen dieser Schlüssel auch zweifelsfrei auf ihre Gültigkeit hin untersuchen und somit sicher sein, dass nur der entsprechende Mix die für ihn verschlüsselten Nachrichten entschlüsseln kann.

3.5.4. Sicheres Positionsupdate

Bei dem sicheren Positionsupdate sind zwei Fälle zu unterscheiden: Updates, die nur den *LLS* betreffen und solche, die zusätzlich auch den *CLS* mit einbeziehen. Letztere sind immer dann notwendig, wenn das Fahrzeug initialisiert wird (zum Beispiel beim Starten) oder wenn der Zuständigkeitsbereich des aktuellen *LLS* – die *SA* – gewechselt wird. Wird dagegen nur ein Positionsupdate innerhalb der gleichen *SA* oder ein Pseudonymwechsel durchgeführt, ist es nicht notwendig, den *CLS* zu kontaktieren. Die Gründe für Pseudonymwechsel werden im Laufe der Arbeit noch ausführlich diskutiert und sind Gegenstand von Kapitel 4.

Initialisierung oder *SA*-Wechsel Um einen Teilnehmer zu initialisieren oder im Falle eines *SA*-Wechsels müssen sowohl der zuständige *LLS* als auch der *CLS* ein Update der gespeicherten Datensätze – beziehungsweise einen initialen Datensatz – erhalten. Der dabei durchgeführte und im Folgenden beschriebene Ablauf ist in Abbildung 3.6 schematisch dargestellt.

Teilnehmer *A* möchte seine Position initialisieren oder bei dem *CLS* updaten und erzeugt dazu die folgende Nachricht, die er an die nächstgelegene *RSU* sendet:

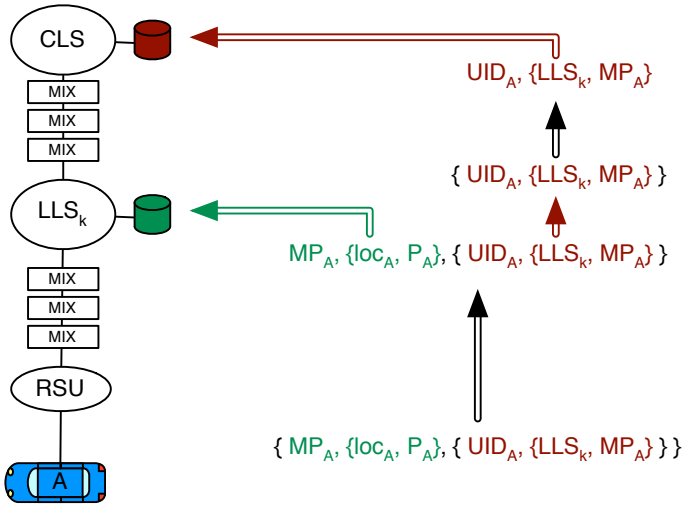


Abbildung 3.6.: Sicheres Positionsupdate (*LLS* und *CLS*)

$\{MP_A, \{loc_A, P_A\}, \{UID_A, \{LLS_k, MP_A\}\}\}$. Diese komplexe Nachricht besteht aus zwei Teilen, von denen der erste ($MP_A, \{loc_A, P_A\}$) für den aktuellen *LLS* (hier mit dem Index k identifiziert) bestimmt ist und der zweite ($UID_A, \{LLS_k, MP_A\}$) für den *CLS*. Diese beiden Teile sind zudem für die zu durchlaufenden Mix-Kaskaden schichtweise verschlüsselt.

Die *RSU* leitet diese Nachricht bei Erhalt an die Mix-Kaskade des zuständigen *LLS* weiter. Dieser ist, wie bereits eingangs erwähnt, durch die Baumstruktur der Architektur eindeutig festgelegt und eine Adressierung kann somit entfallen. Auf dem Weg zum *LLS* entschlüsselt jeder Mix der durchlaufenen Kaskade die Nachricht und leitet sie an den jeweils nächsten Mix weiter. Der letzte Mix übergibt die Nachricht, die nunmehr folgendermaßen lautet, an den *LLS*: $MP_A, \{loc_A, P_A\}, \{UID_A, \{LLS_k, MP_A\}\}$. Es wurde beim Durchlaufen der ersten Mix-Kaskade also die äußerste Verschlüsselungsschicht entfernt.

Die erste Hälfte der Nachricht – $MP_A, \{loc_A, P_A\}$ – wird im *LLS* gespeichert und der restliche Teil – $\{UID_A, \{LLS_k, MP_A\}\}$ – wird an die Mix-Kaskade des *CLS* weitergeleitet. Bei der Speicherung der Informationen im *LLS* wird MP_A als eindeutiger Primärschlüssel betrachtet, unter dem die verschlüsselten Daten $\{loc_A, P_A\}$ abgelegt sind. Es ist leicht zu ersehen, dass der *LLS* somit nur Informationen über das Mapping-Pseudonym MP_A des Teilnehmers A erhält, nicht jedoch über seine eindeutige Kennung UID_A, sein aktuell gültiges Pseudonym P_A oder seine Position loc_A, da diese Informationen ausschließlich verschlüsselt am *LLS* eintreffen.

Beim Durchlauf der ebenfalls durch die hierarchische Struktur eindeutig festgelegten Mix-Kaskade des *CLS* wird der zweite Teil der Nachricht wiederum sukzessive

entschlüsselt und schließlich an den *CLS* übermittelt. Die dort angekommene Nachricht – $UID_A, \{LLS_k, MP_A\}$ – wird in der Datenbank des *CLS* gespeichert, wobei dieses Mal die UID_A als Primärschlüssel dient und der verschlüsselte zweite Teil der Nachricht – $\{LLS_k, MP_A\}$ – darunter abgelegt wird. Der *CLS* erfährt somit die eindeutige Kennung UID_A des Teilnehmers, nicht jedoch seine genaue Position loc_A, geschweige denn den für diesen Teilnehmer zuständige LLS_k, was Rückschlüsse über die *SA* auf den ungefähren Aufenthaltsort des Fahrzeugs zulassen würde. Sämtliche Hinweise auf den räumlichen Ursprung der Nachricht können durch die Mix-Kaskade effektiv unterdrückt werden, und selbst die Überwachung sämtlicher Kommunikationsstrecken liefert keine neuen Informationen.

Nachdem der letzte Teil der Nachricht beim *CLS* eingetroffen ist, ist der Benutzer bei dem Positionsdienst erfolgreich registriert und kann nun kontaktiert werden (vgl. Abschnitt 3.5.5).

Positionsupdate innerhalb der *SA* oder Pseudonymwechsel Wann immer ein Teilnehmer A lediglich sein Pseudonym P_A zu P'_A wechselt und/oder ein Update der Position von loc_A zu loc'_A an den Positionsdienst meldet, ist eine Nachricht an den zuständigen *LLS* ausreichend. Der Teilnehmer erzeugt dazu die Nachricht $\{MP_A, \{loc'_A, P'_A\}\}$ und übermittelt sie durch das Ad-hoc-Netz an die nächste *RSU*. Von dort wird die Nachricht durch die Mix-Kaskade des *LLS* transportiert und dabei entschlüsselt, so dass der *LLS* somit die Nachricht $MP_A, \{loc'_A, P'_A\}$ erhält. Den verschlüsselten zweiten Teil speichert dieser nun unter dem im Klartext enthaltenen Mapping-Pseudonym MP_A ab und ersetzt damit den alten Eintrag.

Damit ist der Update-Vorgang abgeschlossen und der Teilnehmer ist nun unter seinem neuen Pseudonym P'_A an seiner neuen Position loc'_A erreichbar. Da das Mapping-Pseudonym MP_A gleich geblieben ist, besteht keine Notwendigkeit, den *CLS* ebenfalls zu kontaktieren. Der gesamte Vorgang ist in der Abbildung 3.7 schematisch dargestellt.

3.5.5. Sichere Positionsanfrage

Nachdem sich ein Teilnehmer beim Positionsdienst registriert hat, ist er für andere Teilnehmer kontaktierbar. Möchte Teilnehmer B den zuvor registrierten Teilnehmer A kontaktieren, so läuft der folgende Prozess ab, der in Abbildung 3.8 dargestellt ist.

Zunächst erzeugt Teilnehmer B die folgende Nachricht: $\{\{UID_A, loc_B, P_B\}\}$. Er muss also zur Kontaktaufnahme die eindeutige Kennung des anzufragenden Teilnehmers UID_A sowie seine eigene Position loc_B und das eigene Pseudonym P_B zur

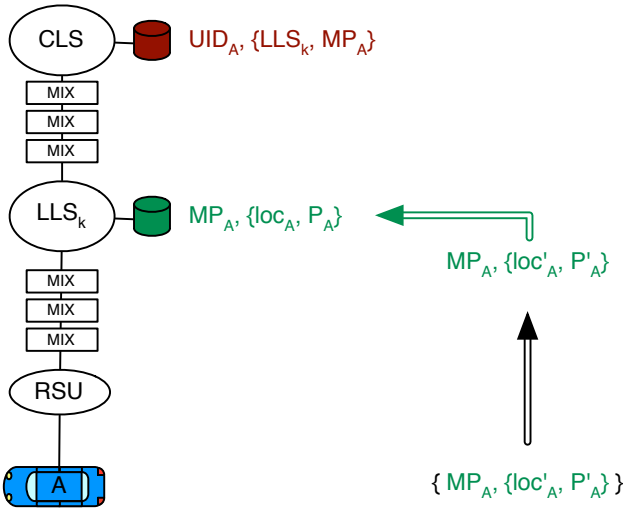


Abbildung 3.7.: Sicheres Positionsupdate (nur *LLS*)

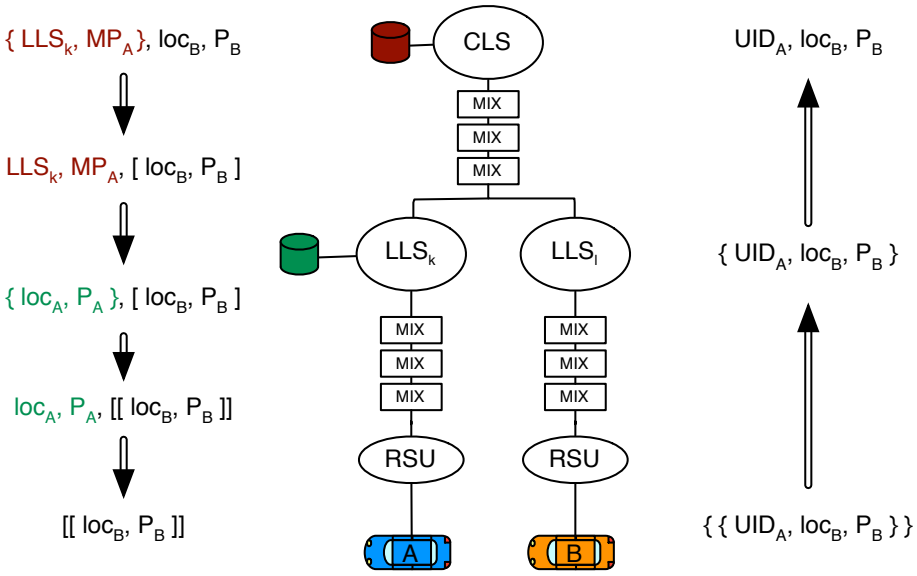


Abbildung 3.8.: Sichere Positionsanfrage

Verfügung stellen. Diese werden für zwei Mix-Kaskaden verschlüsselt und zur nächstgelegenen RSU übermittelt. Von dort wird die Nachricht durch die Mix-Kaskade des für B zuständigen LLS geleitet, und die äußerste Verschlüsselungsschicht wird entfernt. Die Nachricht $\{UID_A, loc_B, P_B\}$ wird nun durch die Mix-Kaskade des CLS übermittelt und die innere Verschlüsselungsschicht wird dabei ebenfalls aufgelöst. Der CLS erhält somit die Klartextnachricht UID_A, loc_B, P_B.

In der Datenbank des CLS ist unter dem Eintrag der UID_A der Datensatz $\{LLS_k, MP_A\}$ gespeichert. Dieser wird um den Rest der Nachricht von B – loc_B, P_B – ergänzt und durch die Mix-Kaskade des CLS geleitet. Dabei wird die Verschlüsselungsschicht von $\{LLS_k, MP_A\}$ entfernt und jeder passierte Mix j verschlüsselt mit dem in der entschlüsselten Nachricht vorgefundenen symmetrischen Schlüssel $k_{A,j}$ den zweiten Teil der Nachricht – loc_B, P_B – für A symmetrisch. Nach dem letzten Mix dieser Kaskade sieht die Nachricht also folgendermaßen aus: $LLS_k, MP_A, [loc_B, P_B]$. Anhand der nun im Klartext vorliegenden Kennung des für den Teilnehmer A zuständigen LLS_k kann die Nachricht an diesen zugestellt werden.

Der LLS_k erhält also die Nachricht $MP_A, [loc_B, P_B]$ (die für die Weiterleitung an ihn verwendete Adresse LLS_k ist nach der Zustellung nicht mehr relevant) und findet in seiner Datenbank unter dem Eintrag für den Benutzer mit dem Mapping-Pseudonym MP_A den Datensatz $\{loc_A, P_A\}$. Diesen sendet er ergänzt um den erhaltenen symmetrisch verschlüsselten Teil $[loc_B, P_B]$ durch die ihm zugeordnete Mix-Kaskade.

In der Kaskade wird erneut der erste Teil der Nachricht, $\{loc_A, P_A\}$, ent- und der zweite Teil, $[loc_B, P_B]$, verschlüsselt. Nach der Kaskade liegt somit die Nachricht in der folgenden Form vor: $loc_A, P_A, [[loc_B, P_B]]$. Anhand der nun im Klartext vorhandenen Position von A, loc_A, kann die diesem Ort nächstgelegene RSU ausgewählt und ihr die Nachricht mit Hilfe des Pseudonyms P_A zur Übermittlung an A zugestellt werden. Der Teilnehmer A erhält also die Nachricht $[[loc_B, P_B]]$, die er entschlüsseln kann, da er die zur Verschlüsselung verwendeten symmetrischen Schlüssel selbst erzeugt und dem Positionsdienst zur Verfügung gestellt hat.

Der Teilnehmer A hat somit nach erfolgreichem Durchlaufen des Verfahrens das aktuelle Pseudonym und die aktuelle Position von B erhalten, der um Einleitung eines Nachrichtenaustausches bittet. Nur dann, wenn sich A einverstanden erklärt und Kontakt mit B aufnimmt, erfährt dieser die Position und das Pseudonym von A. Der angefragte Benutzer hat somit die vollständige Kontrolle darüber, wer seine Position oder sein aktuell gültiges Pseudonym erfährt.

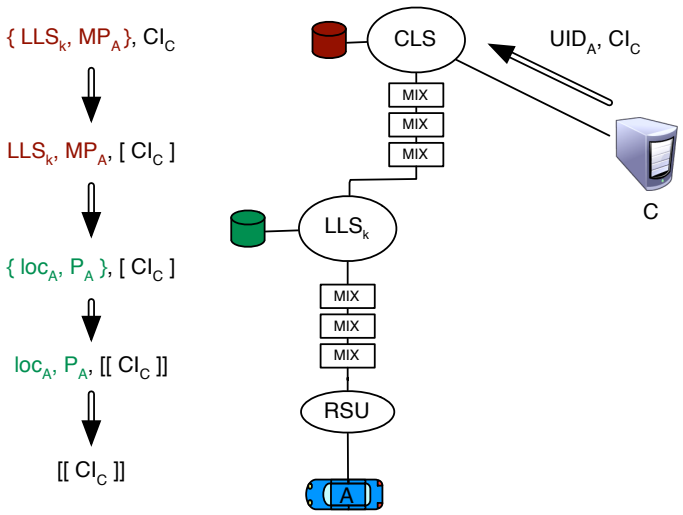


Abbildung 3.9.: Effizentere Positionsanfrage ohne Verschleierung des Absenders

3.5.6. Effizientere Positionsanfrage ohne Verschleierung des Absenders

Besteht keine Notwendigkeit, den Absender einer Positionsanfrage zu schützen, da es sich beispielsweise dabei nicht um ein Fahrzeug sondern um einen Dienstleister handelt, der lediglich in das *VANET* eingebunden ist, kann die Positionsanfrage vereinfacht werden (vgl. Abbildung 3.9). Ein Beispiel für solch eine Situation könnte die Mitteilung von aktuellen Staumeldungen an einen Abonnenten sein.

Der Dienstleister C, der eine Positionsanfrage (also eine Bitte um Kontaktaufnahme) an den Teilnehmer A senden möchte, übermittelt an den *CLS* über eine Verbindung (z. B. dem Internet) die eindeutige Kennung von A – UID_A – und Kontaktinformationen CI_C anhand derer A eine Verbindung zu C herstellen kann. Da sich C nicht im Ad-hoc-Teil des *VANET*s befindet, könnte CI_C beispielsweise eine IP-Adresse darstellen. Diese Kommunikation sollte, obwohl sie nicht durch eine Mixkaskade verschleiert wird, verschlüsselt stattfinden, da der Inhalt der Nachricht, obwohl diese zuordenbar sein darf, dennoch schützenswert ist. Diese Verschlüsselung kann aber relativ einfach mit einem spontan über ein Schlüsselaustauschprotokoll – wie beispielsweise dem Diffie-Hellman-Merkle-Verfahren (vgl. Abschnitt 2.1.3.1) – erzeugten Schlüssel durchgeführt werden.

Der weitere Kommunikationsablauf erfolgt analog der oben beschriebenen Positionsanfrage: Der *CLS* ersetzt die UID_A mit dem unter ihr in seiner Datenbank abgelegten Datensatz $\{LLS_k, MP_A\}$ und leitet diese Informationen ergänzt um CI_C an die Mix-Kaskade. Dort wird der erste Teil der Nachricht ent-, der zweite Teil

verschlüsselt und dem für den Teilnehmer A zuständigen LLS als $MP_A, [CI_C]$ zugestellt. Der LLS ersetzt erneut das Mapping-Pseudonym MP_A durch den bei ihm gespeicherten Datensatz $\{loc_A, P_A\}$ und übergibt die Nachricht an den ersten Mix der Mix-Kaskade. Beim Durchlaufen der Kaskade wird wieder ent- bzw. verschlüsselt und zu guter Letzt die Nachricht dem Teilnehmer A in der Form $[[CI_C]]$ mit Hilfe der entschlüsselten Informationen loc_A und P_A zugestellt.

3.5.7. Evaluation

3.5.7.1. Performance

Das vorgestellte Konzept eines hierarchisch strukturierten und teilweise verteilten, aber zentral organisierten Positionsdienstes erzeugt durch den Einsatz der Mix-Kaskaden und der daraus resultierenden Verwendung von asymmetrischer Kryptographie einen Overhead gegenüber einem Positionsdienst ohne Schutz der Aufenthaltsorte.

Ein Aspekt beruht auf der Expansion von Nachrichten bei asymmetrischer Verschlüsselung. Bei dem Einsatz dieser Kryptosysteme müssen die jeweiligen Blocklängen berücksichtigt werden, das heißt, wenn beispielsweise *RSA* mit einer Schlüssellänge von 1024 bit verwendet wird, so muss die Länge von Klar- und Schlüsseltext jeweils ein ganzzahliges Vielfaches dieser Schlüssellänge sein. Da dies in der Regel nicht gegeben sein dürfte, muss der letzte begonnene Block mit Zufallsdaten aufgefüllt werden, was zu einer Vergrößerung der Nachrichten führt. In Ad-hoc-Fahrzeugnetzen – und so auch in dem vorgestellten Konzept – kommen meist sehr kurze Nachrichten zum Einsatz. Es ist beispielsweise davon auszugehen, dass ein einfaches Positionsupdate, bestehend aus dem Mapping-Pseudonym MP, der Position loc und dem Pseudonym P nicht länger als ca. 300 bit (für die drei Attribute sollten Längen von jeweils ca. 100 bit ausreichen) ist. Da jedoch die Position und das Pseudonym separat asymmetrisch verschlüsselt werden, umfasst der Schlüsseltext allein dieser beiden Felder bereits mindestens einmal die Blockgröße. Ergänzt man dies nun um das Mapping-Pseudonym, so werden für die Verschlüsselung dieser kombinierten Nachricht bereits zwei Blöcke benötigt (auch wenn der zweite Block nicht annähernd mit Nutzdaten gefüllt wird). Somit müsste diese kurze Nachricht bereits – je nach eingesetzter Schlüssellänge des Kryptosystems – auf das mindestens sechsfache vergrößert werden.

Dieser Effekt kann jedoch durch den Einsatz von auf elliptischen Kurven basierender Kryptographie (*ECC*, vgl. Abschnitt 2.1.3.3) verringert werden, da sich damit sowohl die Verarbeitungsgeschwindigkeit, als auch die Schlüssellängen (und damit die

Blockgrößen) deutlich reduzieren lassen. So bietet ein *ECC*-Schlüssel mit der Schlüssellänge von 160 bit ein vergleichbares Sicherheitsniveau wie ein *RSA*-Schlüssel mit 1024 bit. Dies begrenzt den Effekt der Nachrichtenexpansion deutlich.

Ein weiteres Problem stellt die Übertragung der Nachrichten durch Mix-Kaskaden und die dort notwendigen Operationen dar. Dies führt zu einer langsameren Übermittlung und kann sich zudem als Flaschenhals des Systems entpuppen. Die Erfahrungen mit Kaskaden des AN.ON-Systems[4] zeigen jedoch, dass bereits heute das effektive Mixen von Nachrichten innerhalb weniger hundert Millisekunden möglich ist. Diese Zeiträume sind bei den nicht hochkritischen Nachrichten von und zu Positionsdiensten tolerierbar.

Eine Abschätzung, wie viele Teilnehmer durch eine Mix-Kaskade betreut werden können, stellt sich dagegen als sehr schwierig heraus. Die Nachrichten sind an sich recht kurz und der technische Fortschritt führt nach wie vor zu einer steten Steigerung der Rechenkapazität. Damit dürfte die Anzahl der verarbeitbaren Nachrichten bei der Einführung eines *VANET*s bereits erheblich sein. Weiterhin ist es möglich, die Zuständigkeitsbereiche der *LLS* (und damit die Anzahl an Teilnehmern pro zugehöriger Mix-Kaskade) zu verkleinern und eine größere Anzahl einzusetzen. Mit Hilfe dieser Stellschraube kann eine Abwägung der Performance und der Kosten getroffen werden. Es ist zudem denkbar, dass die Kaskade vor einem *LLS* durch eine Reihe von Kaskaden mit vorgeschaltetem Lastverteiler ersetzt werden. Diese Redundanz würde ebenfalls die Kosten und die Performance erhöhen. Zudem kann der Einsatz spezieller Krypo-Prozessoren die Verarbeitungsgeschwindigkeit der Nachrichten (Entschlüsselungsoperationen) ebenfalls verbessern.

Einen weiteren Engpass – und einen *Single Point of Failure* – stellt der zentrale *CLS* dar. Dieser muss sämtliche Positionsanfragen, Positionsinitialisierungen und regelmäßige Updates der Teilnehmer verarbeiten können. Diese Komponente dürfte die größte Last zu tragen haben. Als eine mögliche Lösung für Kapazitätsengpässe kann neben der Vorhaltung einer großen redundanten Infrastruktur auch eine flexibel wachsende Umgebung wie ein Cloud-System zum Einsatz kommen. Bereits heute müssen in Mobilfunknetzen die zentralen Komponenten (z. B. das *HLR*) eine enorme Anzahl an Anfragen bearbeiten; so wurden beispielsweise im Jahr 2011 im Durchschnitt in Deutschland mehrere Millionen Kurzmitteilungen pro Stunde[5] gesendet. Bezieht man noch die Vielzahl an Ein- und Ausbuchunsvorgängen sowie die große Anzahl an Telefonaten[6], die innerhalb kürzester Zeit vermittelt werden,

[4] http://www.jondos.org, abgerufen am 20.06.2012

[5] Laut Bundesnetzagentur wurden 2011 ca. 55 Mrd. Kurzmitteilungen verschickt, was einem Durchschnitt von ca. 6,3 Mio. Nachrichten pro Stunde entspricht (siehe [Bun11]).

[6] Die Bundesnetzagentur meldet für 2011 ca. 107 Mrd. abgehende und ca. 86 Mrd. ankommende Gesprächsminuten. Dies entspricht rund 12,2 Mio. abgehenden und rund 9,8 Mio. ankommenden Gesprächsminuten pro Stunde (siehe [Bun11]).

mit ein, so wird deutlich, dass hier bereits trotz einer Vielzahl an nötigen Aktionen eine technische Realisierbarkeit und eine organisatorische Erfahrung im Mobilfunkbereich existiert, die sich auf ein Ad-hoc-Fahrzeugnetz übertragen lassen.

3.5.7.2. Privatsphäre

Der Schutz der Privatsphäre des beschriebenen Konzepts ist äußerst umfassend. Der Einsatz von Mixen auf allen Übertragungswegen stellt – in Kombination mit einer mehrschichtigen Verschlüsselung – sicher, dass eine Verkehrsanalyse, also die Überwachung aller Kommunikationswege, nicht zu einer Aufdeckung von Kommunikationsbeziehungen führt. Dabei wird lediglich die gängige Vertrauensannahme gegenüber Mixen getroffen, dass bei einer Kaskade aus n Mixen lediglich $n-1$ Mixe zusammenarbeiten dürfen. Ist diese Bedingung erfüllt, so ist die Kommunikationsverbindung nicht nachvollziehbar.

Weiterhin werden sämtliche Daten nur verschlüsselt in den Komponenten des Positionsdienstes gespeichert. Dabei kommen jeweils unterschiedliche Identifikatoren zum Einsatz, so dass selbst eine Zusammenarbeit der verteilten Dienste nicht zu einer Gefährdung der Privatsphäre eines Teilnehmers führen kann. So liegt am *CLS* ein verschlüsselter Datensatz unter dem eindeutigen Identifier *UID* vor; am *LLS* sind jedoch alle Datensätze der Benutzer unter den jeweiligen Mapping-Pseudonymen *MP* hinterlegt, von denen aus es ohne Kontextwissen keine Verbindung zu den *UID*s der Teilnehmer gibt. Die Position der Teilnehmer wird lediglich auf unterster Ebene des Systems entschlüsselt, wo allerdings die Teilnehmer nur unter dem relativ flüchtigen Pseudonym P bekannt sind.

Lediglich bei Anfragen an den Positionsdienst wird der anfragende Teilnehmer gegenüber dem Dienst exponiert, da der *CLS* in der Folge die Information erhält, welche Anfrage gestellt wird. Zudem erhält der angefragte Teilnehmer ebenfalls diese Information, um entscheiden zu können, ob er eine Kontaktaufnahme einleiten möchte. Allerdings sind diese Informationen flüchtig und werden im Positionsdienst nicht gespeichert.

Dem Anfragenden, der bei klassischen Positionsdiensten leicht als Angreifer agieren kann, wird somit etwas weniger Schutz zuteil, dafür bedarf es in der Folge keiner weiteren Schutzvorkehrungen, um das Erstellen von Bewegungsprofilen zu verhindern. Der Angefragte kann nach jeder Positionsanfrage entscheiden, ob er eine Kommunikation mit dem Anfragenden führen möchte und – dies ist bei positionsbasiertem Routing praktisch unvermeidbar – damit auch seinem Gegenüber den eigenen Aufenthaltsort preisgibt.

Der vorgestellte Positionsdienst ist jedoch durch die aufwändigen Operationen, die die einzelnen Komponenten durchführen müssen, relativ anfällig für Angriffe auf

die Verfügbarkeit (*DoS*-Angriffe). Eine Gruppe von Angreifern, die eine sehr große Anzahl an Anfragen innerhalb kürzester Zeit stellt, kann damit möglicherweise zu einer hohen Auslastung der Mixe und der Datenbanken führen. Diesem Problem könnte man lediglich beikommen, indem Anfragen an den Dienst digital signiert werden müssen, was jedoch auch im Regelfall zu einer höheren Belastung durch die Prüfung der Unterschriften bei den Komponenten führt. Allerdings wäre man dann in der Lage, die Angreifer zweifelsfrei zu identifizieren und anschließend Sanktionen gegen sie zu erlassen.

3.6. Fazit

Die eingangs angestellten Untersuchungen zeigen die große Gefahr, die mit dem Einsatz von positionsbasiertem Routing einhergeht: Ein Angreifer kann mit sehr einfachen Mitteln jeden beliebigen Teilnehmer des Netzes (zumindest jene, die die Möglichkeit der Kontaktaufnahme einräumen) beobachten und überwachen. Dieses Problem erstreckt sich zudem auch auf die Betreiber der Positionsdienste, die in der Regel über die umfassenden Informationen verfügen.

Die in der Literatur bekannten Gegenmaßnahmen beinhalten große Schwächen: Entweder ist die Performance problematisch oder es kann mit der Technik kein ausreichendes Schutzniveau erreicht werden. Weiterhin bietet keines der vorgeschlagenen Konzepte einen umfassenden Schutz der Positionsinformationen gegenüber dem Betreiber des Dienstes. Daher wurde ein eigener Ansatz zur Lösung dieser Probleme entwickelt.

Mit dem vorgestellten datenschutzfreundlichen Positionsdienst zeigt sich, dass es möglich ist, die gewünschte und benötigte Funktionalität bereitzustellen und dabei die Privatsphäre der einzelnen Teilnehmer effektiv zu schützen. Dieses Konzept kann jedoch nur umgesetzt werden, wenn die Bereitschaft existiert, die dafür nötigen höheren Kosten gegenüber einem einfachen Positionsdienst ohne Schutz vor der Erstellung von Bewegungsprofilen, zu investieren.

Kapitel 4

Sicherer Pseudonymwechsel

Zu Beginn dieses Kapitels wird untersucht, wie eine Verfolgbarkeit von Teilnehmern mit Hilfe regelmäßig gesendeter Beacons und den darin enthaltenen Pseudonymen in Ad-hoc-Fahrzeugnetzen zustande kommt, welche Konsequenzen sie haben kann und welche Schutzmechanismen bei Pseudonymwechseln wirken müssen. Anschließend wird eine Reihe von in der Literatur vorgestellten Konzepten beschrieben und auf ihre Wirksamkeit hin untersucht. Aus den dabei gewonnenen Erkenntnissen wird abschließend ein eigener Vorschlag für einen sicheren Pseudonymwechsel vorgestellt und evaluiert.

4.1. Bedrohung der Privatsphäre

Für eine gezielte Kommunikation mit anderen Teilnehmern ist in Netzen generell ein Merkmal erforderlich, das den oder die Empfänger oder Absender einer Nachricht identifiziert (vgl. Abschnitt 2.2.1.3). Dieses Merkmal muss, insbesondere um Zurechenbarkeit (vgl. Abschnitt 2.1.2.4) herzustellen, zweifelsfrei an die Identität eines VANET-Teilnehmers (vgl. Abschnitt 2.3.1) gekoppelt werden. Es wäre somit grundsätzlich denkbar, eine eindeutige *ID* des Teilnehmers, die nach gängiger Annahme in der Literatur (vgl. u. a. [Plö09, SPF08, HCL04, RH05, RJP+06, RH07, MKW08, LLZ+08, PP05]) auch stets vorhanden ist, für seine gesamte Kommunikation zu verwenden.

4.1.1. Sicht des Angreifers

Wie bereits in Abschnitt 2.3.3.1 ausführlich dargestellt, werden spezielle Nachrichten in *VANET*s, sogenannte *Beacons*, in sehr kurzen Abständen (meist 100 bis 300

ms) gesendet. Sie enthalten neben den Informationen über die Bewegung des Fahrzeugs (Position l, Geschwindigkeit und Richtung $\vec{v}$, Beschleunigung $\vec{a}$) und dem Zeitpunkt des Sendens t auch eine Kennung des Fahrzeugs (ID). Ein Beacon ist somit ein Tupel $(ID, t, l, \vec{v}, \vec{a})$, dessen Daten von Angreifern zur Verfolgung von Teilnehmern genutzt werden können. In verschiedenen Publikationen werden Beacons nicht verschlüsselt, sondern sind lediglich um eine digitale Signatur zur Authentisierung ergänzt. In diesem Fall stehen alle in den Beacons enthaltenen Informationen jedem zur Verfügung. Aber selbst wenn Beacons symmetrisch verschlüsselt übertragen werden und das Schlüsselmaterial stets geschützt in einem *TPM* verbleibt (vgl. Abschnitt 2.3.2), schützt dies nicht vor einem Missbrauch der Daten. Anwendungen wie Fahrassistenzsysteme benötigen die entschlüsselten Daten und somit müssen diese das *TPM* verlassen und zur Verfügung gestellt werden. Diese Schnittstelle kann natürlich auch von einem Angreifer genutzt werden und somit gelangt er trotz verschlüsselter Übertragung auf der Luftschnittstelle an die in den Beacons enthaltenen Informationen.

Um die Privatsphäre von Teilnehmern zu verletzen, also um herauszufinden, wer sich wann wo aufgehalten und bewegt hat, gilt es für einen Angreifer nach Gerlach [Ger06] zwei Teilprobleme zu lösen: (1) Verfolgen eines Fahrzeugs (bzw. einer *VANET*-Identität) und (2) Verknüpfen der *VANET-ID* mit der Identität des Benutzers. Ma et al. beschreiben die gleichen Herausforderungen und bezeichnen sie als „*Tracking*“ und „*Identification*“ [MKW08].

Verfolgen von Fahrzeugen (*Tracking*) Ziel der Verfolgung von Fahrzeugen ist es, sich Wissen über Aufenthaltsorte von Teilnehmern zu bestimmten Zeitpunkten anzueignen. Laut Gerlach sind dafür vier Schritte erforderlich:

1. *Errichten eines Netzes von Empfängern*: Der Angreifer benötigt eine Form der Überwachung in den zu beobachtenden Gebieten. Dies können beispielsweise Empfänger für in VANETs gesendeten Beacons und andere Nachrichten sein.
2. *Verbinden der verteilten Empfänger mit einer zentralen Datenbank.*
3. *Speichern der Daten*: In der zentralen Datenbank werden alle von den Empfängern wahrgenommenen Nachrichten gespeichert.
4. *Auswertung der Daten*: Die zentral gesammelten Daten können in einem letzten Schritt kombiniert, durchsucht und gefiltert werden, um so Wissen zu generieren.

Dieses generierte Wissen könnte beispielsweise folgender Form sein: „*Das Fahrzeug mit der ID x war zu Zeitpunkt t an Ort l.*“ oder „*Der Teilnehmer mit der ID y bewegt sich jeden Sonntagabend von Ort l zu Ort m.*“. Auf den ersten Blick erscheint dies nicht besonders problematisch, da kein Personenbezug existiert.

Welche Informationen jedoch über Personen allein anhand ihrer Bewegungsprofile ermittelt werden können, zeigen Iqbal und Lim mit Hilfe von aufgezeichneten GPS-Spuren (vgl. [IL10]): Sie konnten beispielsweise mit hoher Wahrscheinlichkeit ermitteln, zu welcher Berufsgruppe einer Universität jemand gehört, oder ob er Geschwindigkeitsübertretungen begeht. Durch den fehlenden expliziten Personenbezug ist jedoch der Nutzen der Daten für den Angreifer eingeschränkt und er wird anstreben, diese Datenspur mit der Identität einer Person zu verknüpfen.

Verknüpfen der Informationen (*Identification*) [Ger06] beschreibt vier Möglichkeiten, wie ein Angreifer einen Personenbezug bei mit *VANET*-Identitäten versehenen Datenspuren herstellen kann:

1. *Nutzung einer vorhandenen Datenbank*: In den meisten Konzepten, die für *VANET*s entwickelt wurden, wird es als notwendig erachtet, dass Teilnehmer mit Fehlverhalten (Missbrauch des Netzes oder Vandalismus) zur Verantwortung gezogen werden können. Damit dies gelingen kann, muss eine Datenbank geführt werden, mit deren Hilfe *VANET*-Identitäten den echten Identitäten (Name, Adresse) von Teilnehmern zugeordnet werden können[1]. Kann sich ein Angreifer Zugriff auf diese Daten verschaffen, so kann er beliebige Zuordnungen vornehmen.

2. *Identifikation über Aufenthaltsorte*: Dieser Angriff basiert darauf, dass es häufig bereits allein auf Basis des Aufenthaltsortes möglich ist, auf die Identität einer Person zu schließen. Dies wurde ausführlich von Gruteser und Grunwald in [GG03] vorgestellt und diskutiert. Ein einfaches Beispiel ist die Erkenntnis, dass das Fahrzeug auf dem privaten Parkplatz oder in der Garage einer Person dieser vermutlich auch gehört. Werden also ausreichend genaue Daten erfasst, so kann dies bereits zur Identifikation verwendet werden. Krumm zeigt in seiner Arbeit [Kru07], wie mit hoher Wahrscheinlichkeit der Wohnort (und damit die Identität einer Person) mit Hilfe von GPS-Tracks ermittelt werden konnte. Der Median der Abweichung vom tatsächlichen Wohnort lag dabei bei ca. 60 Metern, eine Zuordnung zu einer Person gelang mit einfachen Heuristiken in mindestens 5% aller Fälle. Ein noch beunruhigenderes Bild zeichnen Golle und Partridge: Sie konnten zeigen, dass es in sehr vielen Fällen möglich ist, allein mit Kenntnis von Wohn- und Arbeitsort auf eine Person zu schließen [GP09]. Diese beiden Paare lassen sich relativ leicht unter der Annahme, dass der Wagen nachts zu Hause und tagsüber an der Arbeitsstätte geparkt ist, ermitteln. Selbst mit einer Auflösung mit einer Genauigkeit des US-amerikanischen ZIP-

[1] Dies ist mit der aktuellen Eintragung des Kfz-Kennzeichens in das Zulassungsregister der Straßenverkehrsbehörden vergleichbar.

Code-Systems[2] konnten 5% der Arbeitnehmer eindeutig zugeordnet werden und beinahe ein Viertel der untersuchten Personen waren in Anonymitätsgruppen von kleiner oder gleich *5*.

3. *Schlüsse aus externen Datenbanken*: Die Methode von Angreifern, sich weitere Datenbanken nutzbar zu machen, wird im deutschsprachigen Raum in der Regel als „Einbringung von Kontextwissen“ bezeichnet. Beobachtet der Angreifer beispielsweise ein Fahrzeug mit einem bestimmten Kennzeichen und kann er gleichzeitig eine Nachricht in einem sozialen Netz auffinden, in dem eine Person bekannt gibt, der Halter eines Fahrzeuges mit genau diesem Kennzeichen zu sein, so ist die Identifikation trivial. Diese externen Datenbanken, die sich Angreifer zunutze machen können, reichen von öffentlichen oder teilöffentlichen Datenquellen wie Webseiten und sozialen Netzen hin zu über Kunden erhobene Informationen zu Marketingzwecken im Rahmen von Bonusprogrammen.

4. *Anfrage an den Teilnehmer*: Der einfachste und offensichtlichste Weg, die wahre Identität eines Teilnehmers zu erfahren, ist, sich bei ihm danach zu erkundigen. Dies kann im Rahmen von Bezahlverfahren, Bonuskarten oder Anwendungen, die einen Klarnamen erfordern, stattfinden. Obwohl der Benutzer sich darauf verlässt, dass seine Daten vertraulich behandelt und nur zu dem gewünschten Zweck eingesetzt werden, können diese natürlich dennoch durch die vertraute Partei missbraucht oder aber durch externe Personen oder Organisationen gestohlen werden.[3]

Welchen Weg ein Angreifer in einem bestimmten Fall wählt, hängt davon ab, welche Informationen ihm zur Verfügung stehen, welche Rolle er in dem System einnimmt und welche Möglichkeit ihm am vielversprechendsten erscheint. Unabhängig davon, wie es zu einer Verkettung zwischen Kennung im *VANET* und der Identität des Benutzers kommt, ist diese nicht mehr rückgängig zu machen. Von nun an (und bei aufgezeichneten Daten auch rückwirkend) können alle Bewegungen dieser Person offengelegt werden. Der einzige Ausweg aus diesem Problem ist, für die Kommunikation im Ad-hoc-Fahrzeugnetz kein statisches Merkmal, sondern wechselnde Pseudonyme, also Merkmale, die für Außenstehende nicht mit einer eindeutigen *ID* in Verbindung gebracht werden können, zu verwenden, die verhindern sollen, dass überhaupt solch umfassende Profile erstellt werden können. Trotzdem sollte eine

[2] Laut *United States Postal Service* (*USPS*) gibt es über 42.000 ZIP-Codes in den USA (vgl. http://faq.usps.com/, abgerufen am 27.01.2012). Das *U.S. Census Bureau* meldet für den Juli 2009, den Zeitpunkt der Untersuchung von Golle und Partridge, eine Einwohnerzahl von rund 307.000.000 Menschen in den Vereinigten Staaten [U.S09]. Damit ergibt sich eine durchschnittliche Einwohnerzahl pro ZIP-Code von rund 7.300.

[3] Diese illegalen Aktionen können zwar – sofern sie bekannt werden – juristisch sanktioniert werden, führen jedoch zu einem unwiederbringlichen Verlust an Vertraulichkeit der Informationen. Denn sind Daten einmal bekannt, kann dieses Wissen nicht mehr entzogen werden.

eine eindeutige Zuordnung zwischen Pseudonym und Identität existieren, die bei Fehlverhalten offen gelegt werden kann.

4.1.2. Pseudonyme in VANETs

Pseudonyme in Ad-hoc-Fahrzeugnetzen können, ebenso wie eindeutige Identifikatoren (vgl. Abschnitt 2.3.1), ganz unterschiedlich ausgestaltet sein. Grundsätzlich lassen sich zwei Arten unterscheiden: Bitketten ohne syntaktische Relevanz und asymmetrische kryptographische Schlüsselpaare. Sollen Bitketten für Pseudonyme zum Einsatz kommen, muss eine Vereinbarung mit einer vertrauenswürdigen Instanz (*Trusted Third Party, TTP*) geschlossen werden, die die wahre Identität hinter dem jeweiligen Pseudonym kennt und eine Zurechenbarkeit (vgl. Abschnitt 2.1.2.4) herstellt. Für die Art und Weise, wie Bitketten als eindeutige Pseudonyme gebildet werden können, gibt es eine Reihe unterschiedlicher Ansätze. Sie könnten einfach von einer *TTP* erzeugt und zugewiesen werden (vgl. [PF07]) oder durch Teilnehmer selbst erzeugt und von einem mit der *TTP* geteilten Geheimnis abgeleitet werden (beispielsweise mit Pseudozufallszahlenfolgen [Sch08, SPF08, Plö09] oder Hashketten [SWL10]).

Eine weit verbreitete Alternative ist der Einsatz von asymmetrischer Kryptographie (vgl. Abschnitt 2.1.3.2). Dabei wird, ebenfalls wie bei dem Einsatz von Zertifikaten als *ID*s, der beglaubigte öffentliche Schlüssel eines Teilnehmers als Pseudonym verwendet. Dieser kann von einer Zertifizierungsstelle (vgl. u. a. [PBH+07, PBH+08]), durch den Teilnehmer selbst (vgl. u. a. [GGS04]) oder durch eine Gruppe von Teilnehmern (vgl. u. a. [CPHL07, GBW07, SSBP09]) beglaubigt worden sein.

Unabhängig davon, wie die verwendeten Pseudonyme konkret ausgestaltet werden, ist es notwendig, sie in regelmäßigen Abständen zu wechseln. Über einen langen Zeitraum verwendet, würden sie nur eine weitere Form eines statischen identifizierenden Merkmals darstellen. Durch einen Pseudonymwechsel dagegen beschränken sich die Verkettungsmöglichkeiten von Nachrichten auf den Zeitraum, in dem ein Pseudonym verwendet wurde.

Zu beachten ist jedoch, dass der für Fahrzeugkommunikation vorgesehene Standard *IEEE 802.11p* [IEE10] Merkmale für untere Protokollschichten vorsieht: MAC-Adressen und IPv6-Adressen. Wird ausschließlich das Pseudonym auf einer hohen Protokollschicht gewechselt, so kann ein Beobachter Nachrichten mit altem und neuem Pseudonym über die gleich gebliebenen Kennungen auf unteren Protokollebenen verketten. Auch hier ist ein gleichzeitiger Wechsel also erforderlich. Diese Problematik wird in verschiedenen Publikationen (u. a. [LHV07, FFBA07, GMP+08, PBH+08]) ausführlich diskutiert. Fonseca et al. beschreiben in [FFBA07] zudem zwei weitere Schwierigkeiten, die es im Zusammenhang mit Pseudonymwechseln zu lösen

gilt: die Notwendigkeit eines Nachschlagedienstes für die aktuell gültigen Pseudonyme der Teilnehmer sowie die Folgen, die ein Pseudonymwechsel auf die Nachrichtenübermittlung hat (vgl. auch die Untersuchung von Schoch et al. [SKL+06]). Zudem sei darauf hingewiesen, dass es Ansätze gibt, Teilnehmer in Funknetzen anhand impliziter Merkmale der eingesetzten Geräte („*Radio Fingerprint*“ oder „*Signalprint*“) zu erkennen (vgl. [PGG+07, FC06]). Diese Probleme sollen hier jedoch nicht weiter berücksichtigt werden.

4.1.3. Herausforderung beim Pseudonymwechsel

Aus der vorangegangenen Betrachtung ist leicht ersichtlich, dass Pseudonyme häufig (oder zumindest regelmäßig) gewechselt werden sollten. Dieser Pseudonymwechsel muss jedoch unter besonderen Bedingungen durchgeführt werden, da, würde das Pseudonym einfach nur durch ein anderes ersetzt, ein Angreifer auch hier eine Verkettung zwischen altem und neuem Pseudonym erreichen könnte. Dieser Angriff ist besonders leicht auf Basis der regelmäßig gesendeten *Beacons* möglich (vgl. Abschnitt 2.3.3.1). Betrachtet der Angreifer das letzte Beacon mit dem alten Pseudonym und schätzt anhand der enthaltenen Informationen wie Geschwindigkeit und Beschleunigung ab, wo sich der Beobachtete bei seinem nächsten Beacon aufhalten wird, kann er mit nahezu hundertprozentiger Sicherheit den gesuchten Teilnehmer trotz geändertem Pseudonym wiederfinden. Denn selbst wenn der verfolgte Teilnehmer kurz nach dem Senden eines Beacons eine sehr große Änderung seiner Geschwindigkeit durch eine Verzögerung von $-10\,\mathrm{m/s^2}$ durchführt, beträgt bei einem Sendeintervall der Beacons von 200 ms seine Abweichung[4] von der erwarteten Position nur 10 cm (vgl. [Sch08, Plö09], auch diskutiert in [BHV07]). Dies ist, berücksichtigt man die Ausmaße eines Fahrzeugs, völlig ausreichend, um einen Teilnehmer eindeutig zu identifizieren.

Abbildung 4.1 zeigt eine Situation, in der zwei Teilnehmer dicht beisammen fahren, gleichzeitig ihr Pseudonym wechseln und zusätzlich die Positionen tauschen. Dennoch kann ein Angreifer bei alleiniger Betrachtung der Beacons die beiden Fahrzeuge unterscheiden und die neuen und alten Pseudonyme korrekt zuordnen. Erst wenn die Intervalllänge zwischen dem Senden zweier Beacons deutlich erhöht wird, kann der Angreifer keine gesicherte Aussage mehr treffen. Mit Intervallen von mehreren Sekunden könnte somit eine Unsicherheit über die genaue Bewegung eines Teilnehmers geschaffen werden. Alternativ dazu wäre es auch denkbar, die Unschärfe der Telematikdaten im Beacon selbst zu erhöhen. So könnte der Aufenthaltsort l mit einer geringeren Auflösung übermittelt werden (beispielsweise nur noch auf mehrere Meter genau). Auf diese Weise wäre es für einen Angreifer deutlich schwieriger,

[4] $\text{Abweichung} = v_0 \cdot t - (v_0 \cdot t + \frac{a}{2}t^2) = -\frac{a}{2}t^2 = -\left(-\frac{10\,\mathrm{m}}{2\,\mathrm{s}^2}\right) \cdot (0,2\,\mathrm{s})^2 = 0,1\,\mathrm{m}$

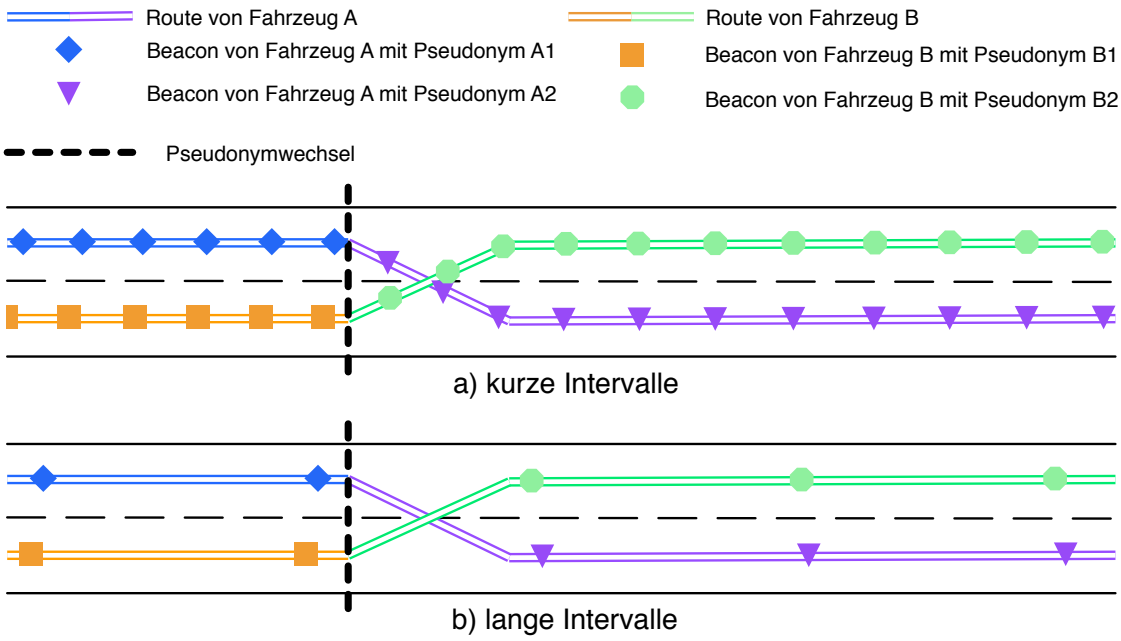


Abbildung 4.1.: Verfolgbarkeit durch Beacons (nach [Sch08, Plö09])

die Bewegung eines Teilnehmers zu antizipieren und ihn somit zu verfolgen. Allerdings bergen beide Ansätze ein Problem in sich: Wenn ein Angreifer die Position und Bewegung eines Teilnehmers nicht mehr exakt ermitteln kann, ist dies auch für Fahrassistenzsysteme in anderen Fahrzeugen nicht mehr möglich. Somit würde ein wichtiger Aspekt und Treiber des *VANET*s, die Erhöhung der Verkehrssicherheit, möglicherweise nicht mehr realisiert werden können.

Bei genauerer Betrachtung ist jedoch nur einmalig eine längere Pause zwischen zwei Beacons während des Pseudonymwechsels erforderlich, die von Eichler generisch als *quiet-time* beschrieben wird [Eic07]. Es besteht somit die Chance, das Pseudonym unbeobachtet zu wechseln und trotzdem sind über die meiste Zeit Fahrassistenzsysteme voll funktionsfähig. Abbildung 4.2 zeigt eine sehr günstige Situation, in der während einer kurzen Funkstille beim Pseudonymwechsel zwei Fahrzeuge unbeobachtbar ihre Positionen tauschen und ein Angreifer somit nicht entscheiden kann, welcher Teilnehmer nun welches Pseudonym trägt.

Ein Pseudonymwechsel ist also erforderlich, um das syntaktisches Verketten („*syntactic linking*“, [BHWW09]), also mit Hilfe des gleichen Pseudonyms, zu verhindern; dieser muss jedoch unter besonderen Umständen, zum Beispiel einer längeren Funkstille, stattfinden, da ansonsten ein semantisches Verketten („*semantic linking*“, [BHWW09]) auf Basis der Inhaltsdaten (Position, Geschwindigkeit, etc.) der Beacons stattfinden könnte. Im Folgenden soll eine Reihe von Konzepten genau-

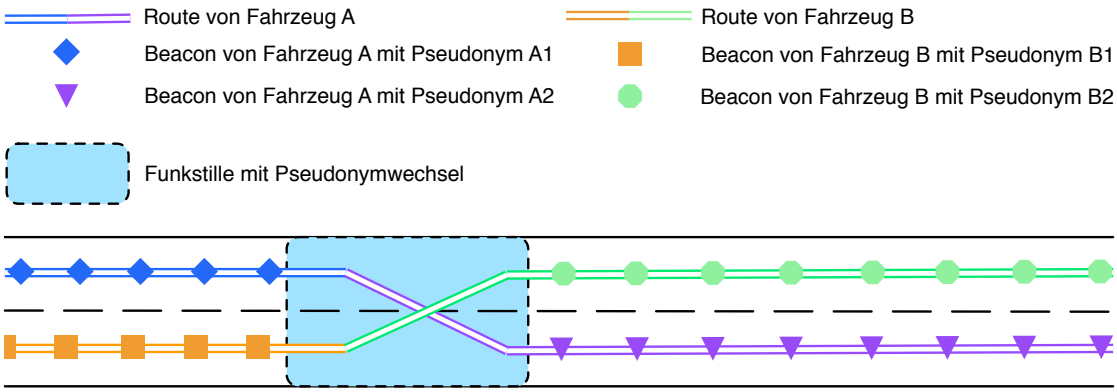


Abbildung 4.2.: Pseudonymwechsel bei Funkstille (nach [Sch08, Plö09])

er untersucht werden, die beschreiben, wie eine solche Funkstille ausgestaltet und koordiniert werden kann. Es kristallisieren sich dabei drei Prinzipien heraus: eine räumliche (*spatial*) oder eine zeitliche (*temporal*) Koordination der Funkstille sowie benutzerzentrierte (*user-centered*) Konzepte. Bei den räumlichen Ansätzen wird durch zentrale Festlegung ein Konsens über einen Bereich geschaffen, in dem Funkstille eingehalten wird, bei den zeitlichen Ansätzen dagegen finden Funkstillen grundsätzlich unabhängig von dem Aufenthaltsort der Teilnehmer zu bestimmten Zeitpunkten statt. Benutzerzentrierte Konzepte ermöglichen es Teilnehmern, unabhängig von ihren Aufenthaltsorten oder von der Zeit spontan eine Funkstille zum sicheren Pseudonymwechsel zu initiieren, wann immer sich aus ihrer Sicht eine gute Gelegenheit dazu bietet oder aber wenn sie es zum Schutz ihrer Privatsphäre für notwendig erachten.

Im Folgenden sollen die aus der Literatur bekannten Ansätze untersucht und bewertet werden. Dabei beschränkt sich diese Arbeit auf die relevanten Konzepte (diese sind in Abbildung 4.3 dargestellt) und berücksichtigt keine Publikationen, die (1) die Problematik nur unzureichend berücksichtigen, wie beispielsweise Hussain et al., die in [HKO09] als Maßnahme zum Schutz der Privatsphäre Beacons ohne Pseudonyme nutzen und dabei übersehen, dass die Verkettung auch mit Hilfe von Telematikdaten vonstatten gehen kann, (2) nicht von regelmäßig gesendeten Nachrichten ausgehen, wie Chaurasia et al., in [CVTA09], und somit die Verkettungsmöglichkeit ebenfalls im Wesentlichen auf die Pseudonyme reduzieren, oder (3) lediglich eine Verkettung mehrerer Fahrten (*trips*) zu verhindern suchen, wie Rass et al., in [RFSK08], was nach den in Abschnitt 4.1.1 beschriebenen Untersuchungen von Golle und Partridge [GP09] zur Verhinderung der Identifizierung unzureichend ist.

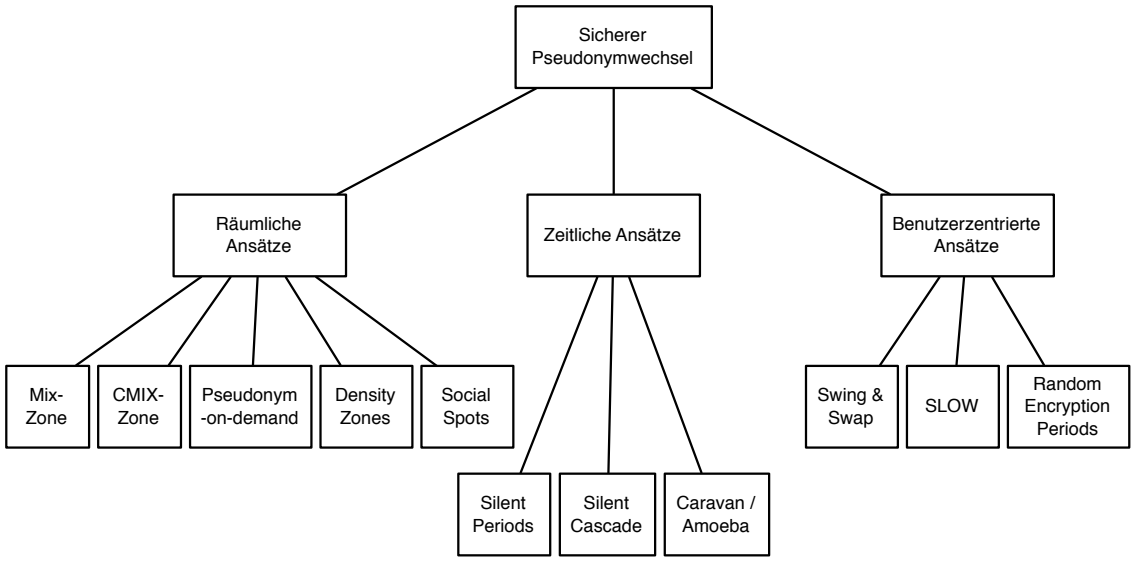


Abbildung 4.3.: Überblick über die untersuchten Ansätze

4.2. Räumliche Ansätze

4.2.1. Mix-Zone nach Beresford und Stajano

Beresford und Stajano beschreiben die dargelegte Problematik in einem ähnlichen Szenario in [BS03, BS04]. Sie betrachten dabei Benutzer, die Positionsdienste an bestimmten Orten nutzen (Anwendungszonen). Beispiele hierfür wären ein System zur Lokalisierung am Arbeitsplatz, eine Preisauskunft in einem Café oder Abfluginformationen am Flughafen. Die Anwendungsanbieter sind nicht vertrauenswürdig und müssen als zusammenarbeitend angesehen werden. Eine Überwachung in der Anwendungszone ist somit unvermeidbar, da der Benutzer ja auch den Nutzen aus dem angebotenen Dienst ziehen möchte. Eine Beobachtung der Aufenthaltsorte (und daher möglicherweise der Handlungen) über mehrere Anwendungszonen hinweg gilt es jedoch zu verhindern, da dies keinen Mehrwert für den Benutzer bietet und lediglich sensible Informationen über ihn missbraucht werden können.

Bewegt sich ein Benutzer von einer Anwendungszone in eine andere, dann ersetzt er zwischen den Zonen sein bisheriges Pseudonym durch ein noch nicht verwendetes. Dieser durchquerte Bereich zwischen den Anwendungszonen wird von den Autoren als Mix-Zone bezeichnet. Mit diesem Namen beziehen sie sich auf Chaums Kommunikationsmixe (vgl. Abschnitt 2.1.4), da die Mix-Zone wie ein Mixknoten wirkt und hier anstelle von Nachrichten die Teilnehmer umsortiert, so dass diese für einen Angreifer von gleichzeitig in der Mix-Zone befindlichen anderen Teilnehmern theo-

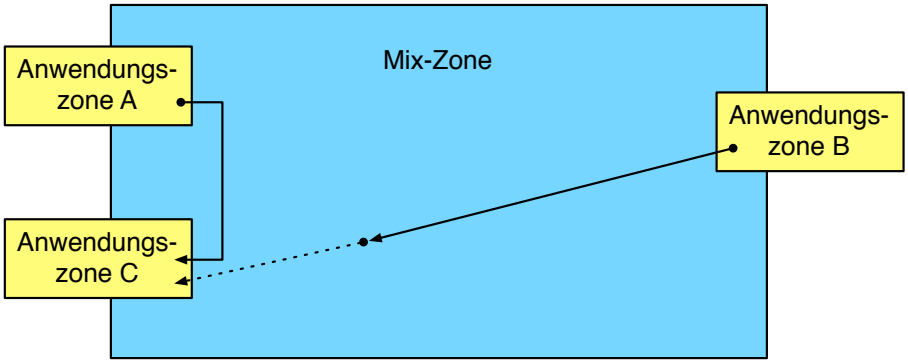


Abbildung 4.4.: Mix-Zone nach Beresford und Stajano (nach [BS03])

retisch nicht mehr unterscheidbar sind.

Allerdings weisen Beresford und Stajano darauf hin, dass die Mix-Zone in bestimmten Situationen keinen Schutz bietet (vgl. Abbildung 4.4). So kann es passieren, dass zwei Benutzer von unterschiedlichen Anwendungszonen (A und B) kommend eine Mix-Zone gemeinsam durchqueren um eine dritte Anwendungszone (C) aufzusuchen. Ist nun die zurückgelegte Strecke des einen Teilnehmers deutlich länger als die des anderen, können die Benutzer mit diesen Zusatzinformationen bereits wieder unterschieden und ihre alten und neuen Pseudonyme somit verkettet werden.

Um ihre theoretischen Überlegungen experimentell zu untersuchen, nutzten Beresford und Stajano das in den Räumen der AT&T Labs Cambridge installierte Active Bat-System [WJH97]. Bei diesem System wird jeder Mitarbeiter mit einer kleinen Einheit ausgestattet, die Ultraschall emittiert. Über verschiedene im Gebäude platzierte Ultraschallsensoren kann durch Triangulation die Position der Einheit und damit der Aufenthaltsort des Mitarbeiters auf 3 cm genau bestimmt werden. Die Autoren werteten insgesamt 3,4 Millionen Datensätze aus und simulierten Mix- und Anwendungszonen im Gebäude. Der genaue Versuchsaufbau und die detaillierten Ergebnisse können in [BS03] nachgelesen werden.

Die Autoren stellen fest, dass sich die Bewegungsprofile der Benutzer für einen Außenstehenden teilweise vorhersagbar darstellen, dass also für einen gegebenen Teilnehmer in Abhängigkeit seines Eintrittspunktes in die Mix-Zone die Wahrscheinlichkeiten für alle Austrittspunkte nicht gleichverteilt ist. Konkret untersuchten sie beispielsweise den Fall, dass zwei Teilnehmer die Mix-Zone aus entgegengesetzten Richtungen betreten und sie wiederum in entgegengesetzten Richtungen verlassen (vgl. Abbildung 4.5). In dieser Situation können nur zwei Fälle eingetreten sein:

1. Beide Teilnehmer haben ihren Weg fortgesetzt und die Mix-Zone in der ihrem Eintrittspunkt gegenüberliegenden Seite verlassen oder

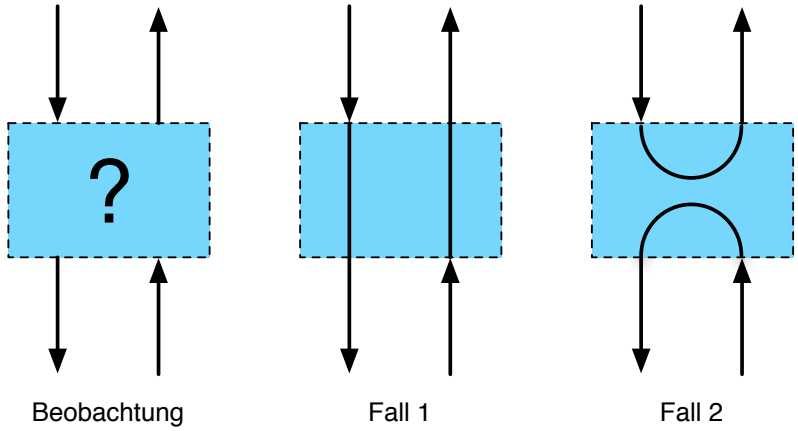


Abbildung 4.5.: Untersuchung zur Bewegung in einer Mix-Zone [BS03] (nach [Sch08, Plö09])

2. beide Teilnehmer haben eine 180°-Drehung vollzogen und die Mix-Zone in die Richtung verlassen, aus der sie sie betreten hatten.

Die Daten aus der Untersuchung belegen das, was man intuitiv vermuten würde: Die Wahrscheinlichkeit, dass *Fall 2* eintritt ist äußerst gering, in der Untersuchung lag sie bei 0,1 %. Ein Angreifer kann somit davon ausgehen, dass wahrscheinlich *Fall 1* eingetreten ist und hat damit die Schutzfunktion der Mix-Zone untergraben.

Die Untersuchung zeigt anschaulich, dass sich das Maß der Anonymitätsgruppe, also nach gängiger Auffassung die Anzahl an gleichzeitig in der Zone befindlichen Benutzer, nur bedingt dazu eignet, den Grad des Schutzes einer Mix-Zone zu beschreiben, wenn die Wahrscheinlichkeiten der Austrittspunkte nicht für jeden Teilnehmer die gleiche Wahrscheinlichkeit besitzen. In dieser Situation eignet sich nach Aussage der Autoren die Shannon'sche Entropie wesentlich besser.

In einer folgenden Veröffentlichung ([BS04]) diskutieren die Autoren im Detail eine Möglichkeit zur Quantifizierung der erreichbaren Anonymität in einer Mix-Zone. Das hier beschriebene Konzept wird von Beresford in seiner Dissertation [Ber05] noch detaillierter analysiert, aber nicht erweitert.

Bewertung Beresford und Stajano schaffen mit der Mix-Zone in [BS03] ein Konzept, das sich in der Folge als vielversprechend für die Anwendung in VANETs herauskristallisiert hat. Eine ganze Reihe von Arbeiten, die sich mit Datenschutz in Ad-hoc-Netzen auseinandersetzen, machen Vorschläge, die auf der Mix-Zone aufbauen. Desweiteren wurde natürlich der Name an sich ebenfalls durch diese ursprüngliche Arbeit geprägt.

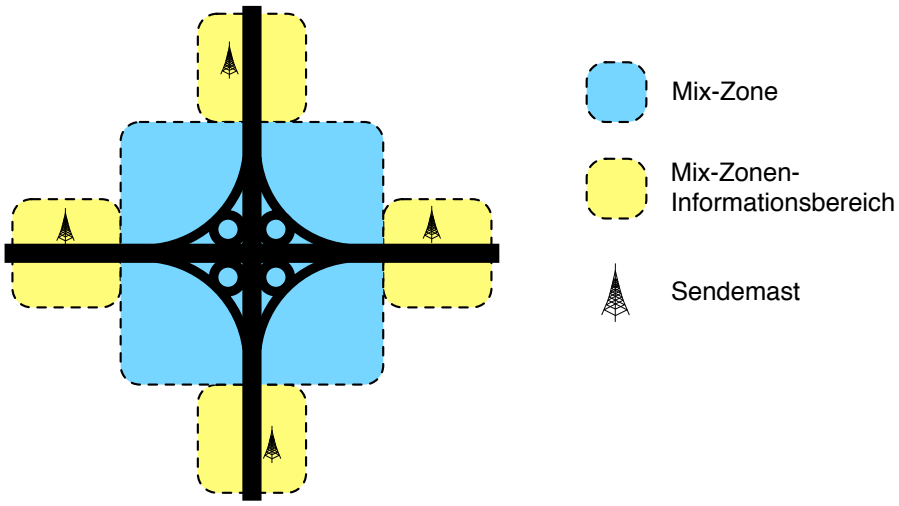


Abbildung 4.6.: Aufbau einer Mix-Zone nach [Sch08, Plö09]

Obwohl das Konzept keine Beacons sondern lediglich Location-based Services betrachtet, ist der Bogen relativ leicht zu schlagen. Interessante Ansätze geben auch bereits die Überlegungen zur Untersuchung der mit Hilfe einer Mix-Zone erzeugten Privatsphäre und damit die Erkenntnis, dass die Kenngröße der Anonymitätsgruppe nur unter bestimmten Umständen genutzt werden kann.

4.2.2. Die Mix-Zone für Ad-hoc-Fahrzeugnetze

Ausgehend von den Arbeiten von Beresford und Stajano lässt sich das Prinzip der Mix-Zone auf das Einsatzgebiet der Ad-hoc-Fahrzeugnetze zuschneiden. Dieses Prinzip wurde unter anderem von Plößl et al. (vgl. [Sch08, SPF08, Plö09]) und – mit kleinen Unterschieden – von Freudiger et al. (vgl. [FRF+07, FSH09]) vorgestellt.

Platziert wird eine Mix-Zone an Orten, an denen eine möglichst gute Durchmischung der Fahrzeuge möglich ist. Dies sind in der Regel größere Kreuzungen oder Knotenpunkte von Fernstraßen wie beispielsweise Autobahnkreuze. Gerade Straßenabschnitte eignen sich kaum, da es hier nur zu geringen Vermischungen bei Überholvorgängen kommt. Insbesondere bei sehr dünnem (kaum Überholvorgänge) oder sehr dichtem Verkehr (kaum Veränderung der relativen Position) bieten derart platzierte Mix-Zonen keinen Schutz. Idealerweise sollte es also mehrere Ausgänge aus der Mix-Zone für jedes eingefahrene Fahrzeug geben, die im Optimalfall auch noch mit der gleichen Wahrscheinlichkeit genutzt werden.

Der Aufbau einer Mix-Zone an einem Autobahnkreuz kann nach [Sch08, Plö09] wie in Abbildung 4.6 dargestellt aussehen. In der Umgebung der eigentlichen Mix-

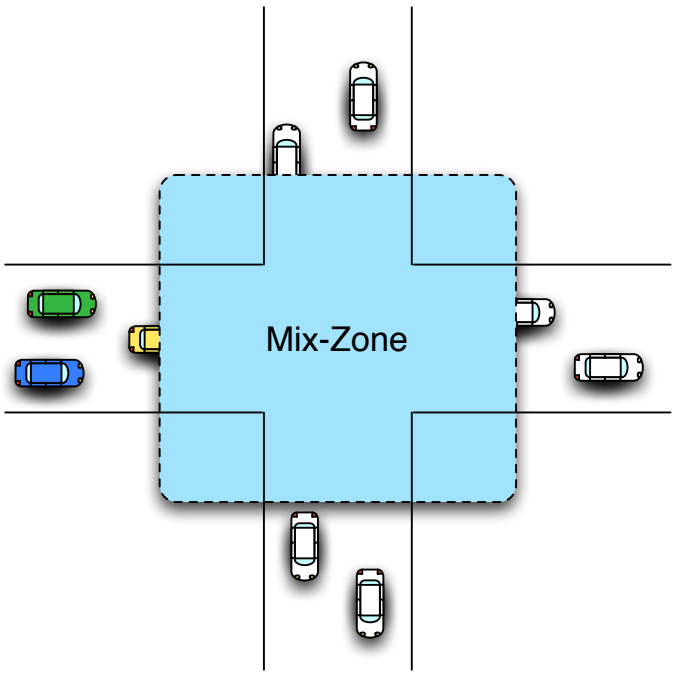


Abbildung 4.7.: Prinzip der Mix-Zone

Zone, die das gesamte Kreuz abdeckt, werden über *RSU*s die Informationen über die Zone gebroadcastet. Dies schließt vor allem die genaue Ausdehnung der Mix-Zone ein, so dass jeder Teilnehmer auch tatsächlich beim Einfahren in die Zone das Senden einstellen kann. Auf diese Weise kann sichergestellt werden, dass alle Teilnehmer über die Lage und Ausdehnung der jeweiligen Mix-Zone informiert sind und Änderungen relativ einfach und unkritisch durchgeführt werden können.

Bei der Ausdehnung einer Mix-Zone ist es unerlässlich, dass sie den gesamten Bereich eines Knotenpunkts einschließlich aller Ein- und Ausfädelstreifen abdeckt. Ansonsten könnte ein Angreifer durch die Beobachtung von Spurwechseln seines Opfers möglicherweise Hinweise über die Richtung erhalten, in die das Opfer die Mix-Zone verlassen wird.

Grundsätzlich fahren also Fahrzeuge in die Mix-Zone ein und wechseln dort ihr Pseudonym, während sie Funkstille halten. Dadurch vermischen sie sich mit anderen Fahrzeugen in der Zone. Wenn die Fahrzeuge die Zone verlassen, kann sie ein Angreifer nicht mehr unterscheiden (vgl. Abbildung 4.7).

Bewertung Der durch die Mix-Zone erreichte Vermischungseffekt der Fahrzeuge hängt in erster Linie von den Charakteristika des darunter liegenden Straßenabschnitts ab. Dieser müsste vermutlich empirisch ermittelt werden (vgl. u. a. die Un-

tersuchung von Freudiger et al. [FSH09]).

Problematisch an der Mix-Zone ist allerdings, dass durch die erzwungene Funkstille keine Beacons mehr anderen Teilnehmern für deren Fahrassistenzsysteme zur Verfügung gestellt werden. Besonders in komplexen Kreuzungen mit vielen Einfädel-, Verzögerungs- und Beschleunigungsvorgängen, die sich ideal für Mix-Zonen eignen, könnten die Assistenzsysteme zur Unfallvermeidung oder -abmilderung ihre größte Wirkung entfalten. Dieser Effekt könnte sich umso schlimmer auswirken, je mehr Fahrer gelernt haben, sich auf die Assistenzsysteme zu verlassen. Diese Problematik ließe sich nur dann entschärfen, wenn die Fahrer deutlich auf die Mix-Zone hingewiesen werden. Das ist natürlich nicht unbedingt wünschenswert, da eine Mix-Zone idealerweise für den technisch uninteressierten Fahrer unsichtbar bleibt.

4.2.3. Die CMIX-Zone

Die soeben skizzierte Problematik der Mix-Zone, dass durch Funkstille auch keine Fahrassistenzsysteme zur Verfügung stehen, versuchen Freudiger et al. in [FRF+07] mit der *Cryptographic Mix Zone* (*CMIX*-Zone) zu lösen. Dabei ist die Grundidee, dass die Fahrzeuge in der Zone weiterhin Beacons versenden, diese jedoch verschlüsselt werden. Für einen Außenstehenden ist somit kein Erkenntnisgewinn mehr vorhanden, da dieser nicht auf die enthaltenen Informationen wie Position, Geschwindigkeit und Beschleunigung zugreifen kann.

Die Positionierung der *CMIX*-Zone entspricht der zuvor vorgestellten Mix-Zone. Sie wird idealerweise an Orten platziert, die eine möglichst starke Vermischung der Fahrzeuge unterstützt sowie mehrere Ausgänge besitzt.

Bei der Einfahrt in die *CMIX*-Zone erfragt ein Teilnehmer mit Hilfe des *Key Establishment Mechanism* über eine RSU einen symmetrischen Schlüssel *SK*, der genutzt wird, um die Beacons zu verschlüsseln. Dieser Schlüssel steht allen Fahrzeugen in der Mix-Zone zur Verfügung, die damit sämtliche Beacons entschlüsseln und zur Kenntnis nehmen können.

Fahrzeuge, die sich knapp außerhalb der *CMIX*-Zone aufhalten (in der sog. *Extended Mix-Zone*), können die aus der Zone gesendeten Beacons natürlich ebenfalls nicht entschlüsseln. Dennoch ist es in Hinblick auf die Verkehrssicherheit möglicherweise notwendig, dass sie diese Informationen zur Kenntnis nehmen können. Zu diesem Zweck erhalten auch sie bereits in diesem erweiterten Bereich den symmetrischen Schlüssel SK. Die von ihnen ausgesendeten Beacons werden jedoch bis zum Einfahren in die *CMIX*-Zone weiterhin nicht verschlüsselt.

Auf diese Weise kann die Verkehrssicherheit durch die vollständige Funktionalität der Fahrassistenzsysteme sowohl in als auch an den Rändern der *CMIX-Zone* so

weit wie möglich sichergestellt werden.

Bewertung Auf den ersten Blick bietet die *CMIX*-Zone eine Möglichkeit des unbeobachtbaren Vermischens von Fahrzeugen in *VANET*s und löst zudem die Probleme in Hinblick auf die Verkehrssicherheit, die beim Einsatz einer einfachen Mix-Zone durch die erzwungene Funkstille auftreten.

Die Autoren berücksichtigen jedoch lediglich einen Angreifer, der außerhalb der *CMIX*-Zone agiert. Mit sehr geringem Aufwand (der Angreifer benötigt lediglich ein angemeldetes Fahrzeug) kann dieser an der Gruppenverschlüsselung mit *SK* und der Zone teilnehmen und somit ebenfalls alle Nachrichten entschlüsseln. In dieser Situation ist kein unbeobachtbarer Pseudonymwechsel mehr möglich. Weiterhin ist natürlich auch der Betreiber der *CMIX*-Zone in der Lage, alle Teilnehmer zu beobachten und ihre Nachrichten zur Kenntnis zu nehmen. Freudiger et al. legen diesbezüglich keine Annahmen dar und berücksichtigen dies nicht in ihrem Angreifermodell.

Es ist davon auszugehen, dass ein rationaler Angreifer alle durch ihn realistischerweise durchführbaren Angriffe erwägt und den vielversprechendsten davon auswählt. Daher ist es sehr problematisch, anzunehmen, dass er den soeben beschriebenen sehr einfachen Angriff nicht durchführen würde. Berücksichtigt man zudem, dass einige Ansätze für *VANET*-Sicherheitsinfrastrukturen (z. B. [Plö09]) ohnehin grundsätzlich eine symmetrische Verschlüsselung von Beacons vorsehen, dann ergibt sich durch die *CMIX*-Zone nur ein sehr geringer Mehrwert, da die Beobachtung von Pseudonymwechseln nur minimal erschwert wird.

4.2.4. Pseudonym-on-demand

Ma et al. befassen sich in ihrer Publikation *Pseudonym-on-demand* [MKW08] zwar überwiegend mit der Vergabe von Pseudonymen, dennoch ist ihnen die Schwierigkeit, Pseudonyme unverkettbar zu wechseln, bekannt und sie skizzieren als Lösung eine Variante der Mix-Zone. Diese weist sehr große Ähnlichkeit mit der *CMIX*-Zone von Freudiger et al. (vgl. Abschnitt 4.2.3) auf.

In dem Konzept von Ma et al. wird für einen Pseudonymwechsel Kontakt zu einer Instanz außerhalb des Ad-hoc-Fahrzeugnetzes benötigt. Demzufolge finden die Wechsel in dem Einzugsgebiet und mit Hilfe einer *RSU* statt. Diese *RSU* stellt allen Fahrzeugen in Reichweite neben den Informationen über die Ausdehnung der Mix-Zone einen symmetrischen Gruppenschlüssel zur Verfügung, mit dem alle Nachrichten verschlüsselt und somit vor Außenstehenden geschützt werden. Laut den Autoren soll die Größe der Mix-Zone die Kommunikationsreichweite (beispielsweise 300 m) nicht

überschreiten, damit alle Fahrzeuge im Zonenbereich von der *RSU* den benötigten Schlüssel erhalten können.

Das Ende der Mix-Zone wird in diesem Konzept jedoch nicht strikt festgelegt, sondern nur durch die maximale Ausdehnung der Zone beschrieben. Einem Teilnehmer steht es jedoch frei (und er wählt auch zufällig), zu welchem Zeitpunkt er das unverschlüsselte Senden wieder aufnimmt, solange er die Verschlüsselung nicht über die Grenzen der Mix-Zone hinaus fortsetzt. Dies führt laut den Autoren zu einem besseren Schutz der Privatsphäre, da die Verkettung der Pseudonyme nun erschwert sei.

Für die Positionierung der Mix-Zone sprechen Ma et al. keine weitere Empfehlung aus; aus ihrer Publikation geht hervor, dass sie auch einen Einsatz auf einem einspurigen und geraden Streckenabschnitt für möglich halten.

Bewertung In Hinblick auf den Schutz vor Verkettung von Pseudonymen durch einen Angreifer trägt die Publikation mit dem Vorschlag des variablen Endes der Mix-Zone bei. Dies scheint jedoch nur auf den ersten Blick eine Verbesserung zu sein. Durch klar definierten Grenzen der Mix-Zone, die zudem für jeden Teilnehmer gleichermaßen Gültigkeit besitzen, wird keinerlei Information über den Einzelnen preisgegeben. Es gilt lediglich darauf zu achten, dass – sollte es keine zentrale Taktung für den Zeitpunkt der ausgesendeten Beacons geben – Teilnehmer nicht an ihrem asynchronen Senderhythmus festhalten und auf diese Weise dem Angreifer Informationen bereitstellen. Sendet ein Teilnehmer beispielsweise ein Beacon zum Zeitpunkt t, so wird sein zweites Beacon (unter der Annahme eines konstanten Sendeintervalls von 100 ms) zum Zeitpunkt $t + 100$ ms gesendet werden. Ein zweiter Teilnehmer – in einem anderen Rhythmus – übermittelt sein erstes Beacon zum Zeitpunkt $t + 50$ ms und sein zweites zu $t + 150$ ms. Behalten beide Teilnehmer stets diesen Rhythmus bei und verzichten lediglich während des Pseudonymwechsels auf eine bestimmte Anzahl an Beacons, können sie dennoch von einem Angreifer auf Grund ihres verschobenen Sendeintervalls unterschieden werden. Diesem Problem lässt sich auch bei statischen Mix-Zonen-Grenzen begegnen, indem alle Teilnehmer ihr erstes Beacon nach der Mix-Zone exakt bei Überfahren der Grenze aussenden und somit einen neuen Rhythmus beginnen.

Die beschriebene Alternative mit einem variablen Ende der Mix-Zone ist nicht zu bevorzugen, da eine Verkürzung der Zone stets potentiell zu einer geringeren Privatsphäre führt. Die von Ma et al. beschriebene Mix-Zone hat bereits eine sehr geringe Ausdehnung (maximal die Übertragungsreichweite), die in einer Größenordnung von 300 m geschätzt wird. Sollte die Mix-Zone nun nicht in einem städtischen Kreuzungsbereich platziert werden, so ergibt sich bei dieser Größe nur eine sehr geringe Chance, dass es zu einer effektiven Vermischung der Teilnehmer kommt (vgl. auch

die Diskussion in [Sch08, Plö09]). Wird diese Ausdehnung nun durch ihre Variabilität verringert, so dürfte es für einen Angreifer relativ einfach möglich sein, alte und neue Pseudonyme von Teilnehmern zu verketten. Ein Angriff könnte durch ein variables Ende der Mix-Zone nur dann erschwert werden, wenn die Ortsauflösung der in den Beacons gesendeten Positionen sehr gering ist und somit der Zeitpunkt des Sendens nicht besonders viel über die aktuelle Position eines Teilnehmers aussagt. In diesem Fall könnte die Unsicherheit für einen Angreifer weiter gesteigert werden. Wie bereits in Abschnitt 4.1.3 dargelegt wurde, ist das nicht praktikabel.

Auch behalten die Kritikpunkte der *CMIX*-Zone weiterhin ihre Gültigkeit: Durch den Verzicht auf eine Funkstille kann zwar die Verkehrssicherheit weiterhin uneingeschränkt gewährleistet werden, jedoch kann ein Angreifer sehr einfach durch Einfahren in die Mix-Zone die Verkettung alter und neuer Pseudonyme vornehmen. Die mehrfach erwähnte geringe Größe der von Ma et al. vorgestellten Mix-Zone erleichtert diesen Angriff noch: Der Angreifer muss sich nach dem Einfahren in die Zone nicht bewegen, da der gesamte relevante Bereich per Definition innerhalb seiner Empfangsreichweite liegen muss. Es ist also davon auszugehen, dass diese Variante keinerlei Schutz bietet.

4.2.5. Density Zones

Befinden sich zum Zeitpunkt eines Pseudonymwechsels zu wenige Teilnehmer nahe beieinander, so ist der Wechsel unter Umständen wirkungslos. Diese bereits mehrfach umrissene Erkenntnis verwenden Song et al. in ihrer Publikation [SWL10] als Ausgangspunkt. Um diesem Problem zu begegnen, definieren sie eine Variante der Mix-Zonen – bei ihnen als *Density Zones* bezeichnet – in Straßenabschnitten, die idealerweise Kreuzungen beinhalten.

Teilnehmer, die sich in einer *Density Zone* befinden, zählen anhand von Beacons, wie viele andere Fahrzeuge sich nun ebenfalls in der Zone aufhalten. Wird ein festgelegter Schwellwert k_i überschritten, so wird das Pseudonym gewechselt. In einer in der gleichen Arbeit beschriebenen Erweiterung wird berücksichtigt, dass nicht alle anderen Fahrzeuge auf Grund ihrer Bewegung geeignet sind, die eigene Privatsphäre zu verbessern. So dürften beispielsweise Fahrzeuge, die sich in die Gegenrichtung bewegen, für einen Angreifer auch nach einem Pseudonymwechsel unterscheidbar sein. Wird dies berücksichtigt, so bezieht ein Teilnehmer zur Kalkulation der Fahrzeugdichte nur entsprechend geeignete andere Fahrzeuge mit ein.

Als Angreifer wird ein globaler passiver Angreifer angenommen, der laut den Autoren Fahrzeuge in der *Density Zone* auf Grund des Pseudonymwechsels nicht unterscheiden kann. Er darf jedoch alle Nachrichten in der Zone verfolgen und kann zudem

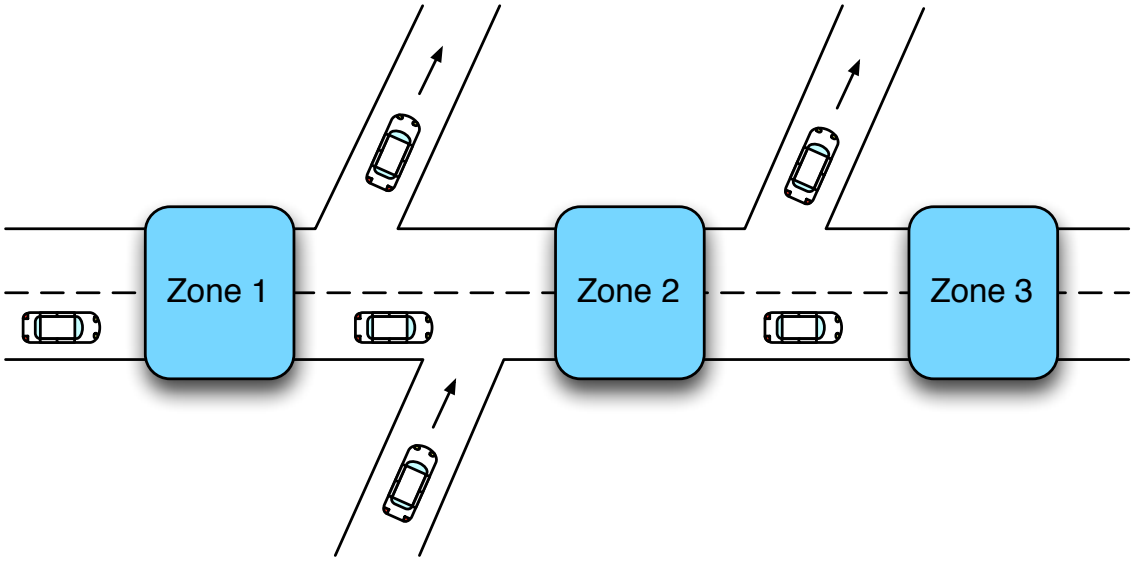


Abbildung 4.8.: Simulationsaufbau nach [SWL10]

Sensoren an den Ein- und Ausgängen der *Density Zone* zur Überwachung von Ein- und Ausfahrereignissen installieren.

Die Arbeit von Song et al. umfasst umfangreiche Betrachtungen sowohl analytischer Art als auch auf Simulationen basierend. Die *Density Zone* wird in den simulativen Untersuchungen der Mix-Zone (vgl. Abschnitt 4.2.2) und AMOEBA (vgl. Abschnitt 4.3.3) gegenübergestellt. Simuliert wurde ein Straßenabschnitt mit drei Zonen sowie einer Zu- und zwei Abfahrten außerhalb der Zonenbereiche (vgl. Abbildung 4.8). Jedes Fahrzeug durfte beim Durchfahren des Abschnitts lediglich einmal das Pseudonym wechseln. Im Fall von AMOEBA (Details in Abschnitt 4.3.3) war der Wechsel nur erlaubt, wenn auf gleicher Höhe ein Fahrzeug in die Straße einfährt, im Fall der Mix-Zone wurde zufällig eine der drei durchfahrenen Zonen mit jeweils einer Wahrscheinlichkeit von $1/3$ gewählt. Bei der Untersuchung der *Density Zones* konnten die Teilnehmer dagegen wählen, wann sie ihr Pseudonym wechseln wollten und hatten dabei einen Schwellwert von $k_i = 10$ (es müssen sich also gleichzeitig mit dem Betrachteten neun weitere Fahrzeuge in der Zone aufhalten) zugrunde gelegt. Die Ergebnisse der Autoren sprechen eine eindeutige Sprache: Die *Density Zone* ist den anderen Konzepten weit überlegen und kann die Zahl der erfolgreichen Verfolgung auf zwischen 50% und 1% (je nach Variation der Simulationsparameter) senken, wohingegen die beiden anderen Konzepte jeweils bei ca. 70% bis 12% verharren (Details siehe [SWL10]).

Bewertung Trotz der grundsätzlich interessanten Idee, die Entscheidung für oder gegen einen Pseudonymwechsel von der Verkehrsdichte abhängig zu machen, überwiegen die zu kritisierenden Aspekte an der Arbeit von Song et al. Das erste Problem zeigt sich bereits im Angreifermodell. So berücksichtigen die Autoren einen globalen Angreifer, der auch Nachrichten in der *Density Zone* mitlesen kann – immerhin existiert ja keine Funkstille (Mix-Zone, vgl. Abschnitt 4.2.2) oder Verschlüsselung (*CMIX*-Zone, vgl. Abschnitt 4.2.3) –, nehmen dann aber an, dass ein Pseudonymwechsel grundsätzlich erfolgreich ist, nur weil sich eine bestimmte Anzahl an Teilnehmern in der Nähe aufhält. Bereits eingangs wurde in Abschnitt 4.1.3 gezeigt, dass eine längere Zeitspanne ohne Überwachung durch einen Angreifer unerlässlich ist, um überhaupt eine Chance auf einen erfolgreichen – weil unbeobachtbaren – Pseudonymwechsel zu haben. In der von Song et al. beschriebenen Form ist die *Density Zone* somit wirkungslos.

Die Analyse der Autoren in ihrer Simulation ist ebenfalls problematisch. Die von dem Angreifer verwendete Heuristik zur Verfolgung von Teilnehmern basiert auf Ein- und Ausgangsereignissen, die in Bezug zueinander gesetzt werden sollen. Dies ist jedoch wiederum eine wenig sinnvolle Modellierung, da ein Angreifer in der Regel die erfolgsversprechendste Herangehensweise wählen dürfte. Diese basiert, wenn ihm – wie von den Autoren angenommen – sämtliche Nachrichten auch innerhalb der *Density Zone* zur Verfügung stehen, nicht auf der Verknüpfung von Ein- und Austrittsereignissen sondern vielmehr auf der Verkettung von Nachrichten (Beacons) auch in der Zone (vgl. Abschnitt 4.1.3).

Weiterhin wurde in der Simulation das von Song et al. vorgestellte Konzept bevorzugt. Dies wird im direkten Vergleich mit der Mix-Zone deutlich: Die Autoren erlauben nur einen Pseudonymwechsel in einer der drei definierten Zonen auf dem simulierten Straßenabschnitt, und diese Zone wird bei der Verwendung als Mix-Zone zufällig gewählt. Bei der Verwendung der *Density Zones* dagegen wird der für einen Pseudonymwechsel am besten – oder zumindest gut – geeignete herangezogen. Damit entsteht natürlich zwangsläufig ein deutlicher Vorteil für die *Density Zones*. Dies ist als umso problematischer zu werten, als sich Vertreter der Mix-Zonen ebenfalls Gedanken darüber machen, welcher Ort optimalerweise für eine Mix-Zone verwendet werden soll (vgl. u. a. [BS04, FSH09, Plö09]) und dies gerade nicht dem Zufall überlassen möchten.

Mit diesen zwei Punkten, der unplausiblen Heuristik des Angreifers bei der *Density Zone* und der starken Bevorzugung gegenüber den anderen Konzepten in der Simulation, konnte scheinbar die Qualität des von Song et al. beschriebenen Ansatzes gezeigt werden. Nach genauerer Betrachtung muss der Ansatz jedoch in dieser Form verworfen werden.

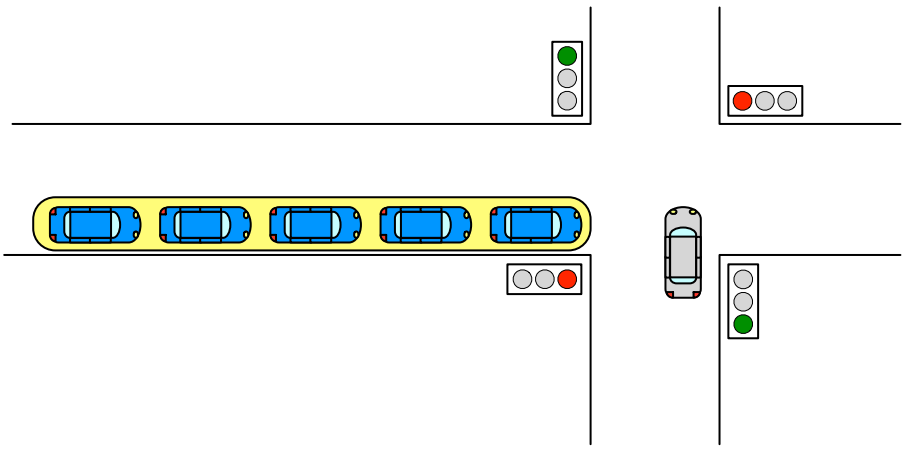


Abbildung 4.9.: *Small Social Spot* nach [LLL+11]: Der gelb hinterlegte Bereich umreißt die Anonymitätsgruppe.

4.2.6. Social Spots

Lu et al. schlagen in ihrer Arbeit [LLL+11] als Lösung für das Problem eines sicheren Pseudonymwechsels die Verwendung sogenannter *Social Spots* vor. Darunter verstehen die Autoren einen Bereich, in dem sich eine größere Anzahl an Fahrzeugen einfindet und in dem – ähnlich der Mix-Zone – die dort verorteten Teilnehmer gemeinsam ihre Pseudonyme wechseln. Grundsätzlich unterscheiden die Autoren zwei Klassen dieser *Social Spots*, nämlich kleine und große.

Unter *Small Social Spots* (vgl. Abbildung 4.9) verstehen Lu et al. Orte, an denen sich eine gewissen Anzahl an Fahrzeugen kurzfristig einfindet. Dies könnte beispielsweise der Bereich vor einer roten Ampel an einer Kreuzung sein. Die Autoren argumentieren, dass, wenn alle Fahrzeuge gleichzeitig ihr Pseudonym wechseln, bevor die Ampel auf „grün“ schaltet, die Teilnehmer eine Anonymitätsgruppe bilden und von einem Angreifer nicht mehr unterscheidbar sind. Auf diese Art könnte ohne größeren Aufwand ein regelmäßiger Pseudonymwechsel stattfinden.

Large Social Spots (vgl. Abbildung 4.10) dagegen sind Orte, an denen sich eine größere Anzahl an Fahrzeugen durchaus auch eine längere Zeit aufhält. Als Beispiel führen die Autoren den Parkplatz eines Einkaufszentrums an: Fahrzeuge fahren mit einem bestimmten Pseudonym ein und werden abgestellt. Nach einer zufälligen Zeitspanne werden sie wieder gestartet und verlassen den Parkplatz mit einem neuen Pseudonym. Auf diese Art und Weise bildet sich nach Aussage von Lu et al. eine Anonymitätsgruppe über alle gemeinsam auf dem Parkplatz befindlichen Fahrzeuge.

Abschließend kalkulieren die Autoren analytisch die durchschnittlichen Größen der

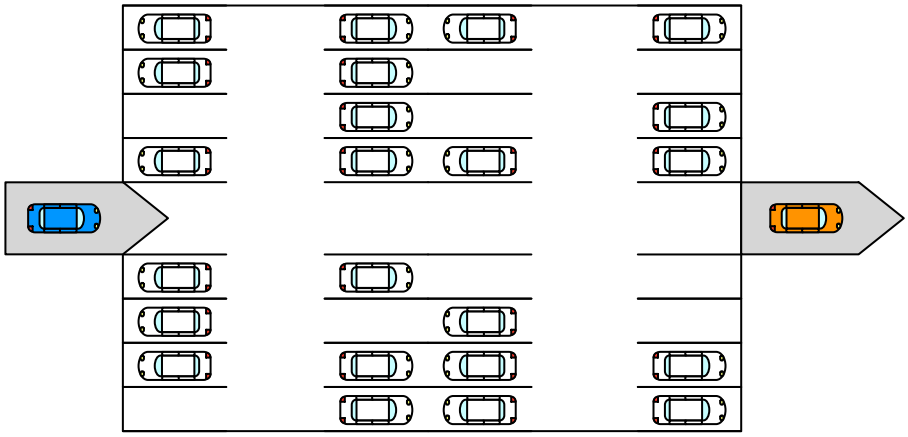


Abbildung 4.10.: *Large Social Spot* nach [LLL+11]

durch die beschriebenen *Social Spots* entstehenden Anonymitätsgruppen und erhalten, je nach Verweildauer und Fahrzeugdichte in diesem *Social Spot*, Anonymitätsgruppengrößen zwischen fünf und 30 (*Small Social Spots*) sowie zwischen zwei und 39 (*Large Social Spots*).

Bewertung Die Idee der *Social Spots* bietet einige Vorteile: Es ist nicht notwendig, in einem Bereich eine Mix-Zone einzurichten (mit unterstützender Infrastruktur), sondern Teilnehmer erreichen und durchfahren regelmäßig für einen Pseudonymwechsel – scheinbar – geeignete Orte. Ein ähnlicher Ansatz wird in der Folge der Arbeit noch diskutiert (vgl. Abschnitt 4.4.2).

Das entscheidende Problem, das von Lu et al. in ihrem Konzept nicht berücksichtigt wird, ist die Auflösung des Ortes eines Teilnehmers, also die Genauigkeit, mit der dessen Position bestimmt (und in *Beacons* übertragen) werden kann. Mangels Implementierungen ist es aktuell nicht möglich, hier konkrete Aussagen zu treffen, jedoch versprechen die Ankündigungen des Satellitensystems *GALILEO* (vgl. [Gal03, Eur08]) eine Genauigkeit von wenigen Metern (*Open Service*) bis in den Zentimeterbereich (*Commercial Service*). Bei dieser Ortsauflösung gibt es nur eine sehr geringe Unsicherheit für einen Angreifer bei der Zuordnung von Nachrichten zu den physischen Fahrzeugen. Eine etwas länger andauernde Funkstille könnte hier wiederum etwas Unschärfe erzeugen; dies ist jedoch von den Autoren nicht vorgesehen.

Auch auf Parkplätzen führt eine Auflösung des Ortes im Meterbereich zu einer sehr guten Unterscheidbarkeit der Fahrzeuge. Die Mix-Eigenschaften, die Lu et al. Parkplätzen zuschreiben, sind de facto nicht existent, da der Angreifer Einsicht in den *Social Spot* nehmen und Teilnehmer auch innerhalb dessen verfolgen kann. Hier könnte

man jedoch durch die Einrichtung einer Funkstille auf dem gesamten Parkplatz eine gute Mix-Funktionalität erzielen. Diese wäre jedoch auch nicht ausreichend, um die Privatsphäre von Teilnehmern effektiv zu schützen, denn ist es möglich, Start- und Zielort von Fahrten zu erkennen (dies wäre der Fall, wenn ein effektiver Pseudonymwechsel nur mit dem Abstellen von Fahrzeugen auf Parkplätzen durchgeführt würde). So kann möglicherweise eine Zuordnung des beobachteten Teilnehmers zu einer Person (Identifizierung) erfolgen (vgl. [GG03], diskutiert in Abschnitt 4.1.1). Weiterhin kann gerade das Fahrtziel einer Person eine schützenswerte und dem Angreifer vorzuenthaltene Information sein.

Insgesamt stellt sich somit der Einsatz von *Large Social Spots* und die Möglichkeiten, Unverkettbarkeit mit Hilfe von *Small Social Spots* herzustellen als unzureichend dar.

4.3. Zeitliche Ansätze

4.3.1. Silent Periods

In ihrer Veröffentlichung [HMYS05] argumentieren Huang et al., dass das Einrichten von einer Vielzahl an Mix-Zonen unpraktikabel ist. Stattdessen schlagen sie den Einsatz sogenannter *Silent Periods* vor. Auch hier soll eine Funkstille dafür sorgen, dass Pseudonymwechsel von Teilnehmern unbeobachtet ablaufen können. Die Arbeit ist dabei jedoch nicht speziell auf VANETs zugeschnitten, sie ist befasst sich vielmehr allgemein mit *Location-based Services* in mobilen Netzen. Es wird also von einer viel freieren Bewegung der Knoten ausgegangen.

Der Grundgedanke ist, dass alle Pseudonyme über eine gewisse Gültigkeitsdauer verfügen, nach der sie gewechselt werden müssen. Um den sicheren Wechsel (vgl. Abschnitt 4.1.3) zu ermöglichen, senden die Teilnehmer für eine gewisse Zeitspanne keine Nachrichten, die das alte oder neue Pseudonym enthalten. Diese als *Silent Period* bezeichnete Phase soll die Unverkettbarkeit der Pseudonyme sicherstellen. Wechseln mehrere Teilnehmer (nahezu) gleichzeitig ihre Pseudonyme, so entsteht somit eine *virtuelle Mix-Zone* (*„virtual mix zone“*, [HMYS05]).

Die *Silent Period* besteht aus zwei Komponenten, einer konstanten und einer variablen Zeitspanne. Die konstante Zeitspanne dient dazu, die räumliche Verknüpfung zwischen dem alten und dem neuen Pseudonym aufzulösen, die variable dagegen soll die zeitliche Verkettung verhindern. Dies leuchtet sofort ein, wenn man berücksichtigt, dass der Pseudonymwechsel nun nicht koordiniert abläuft und somit zwei Teilnehmer, die sich in räumlicher Nähe befinden und mit einer zeitlichen Überlappung ihre *Silent Periods* durchlaufen, leicht anhand des Beginns und des Endes

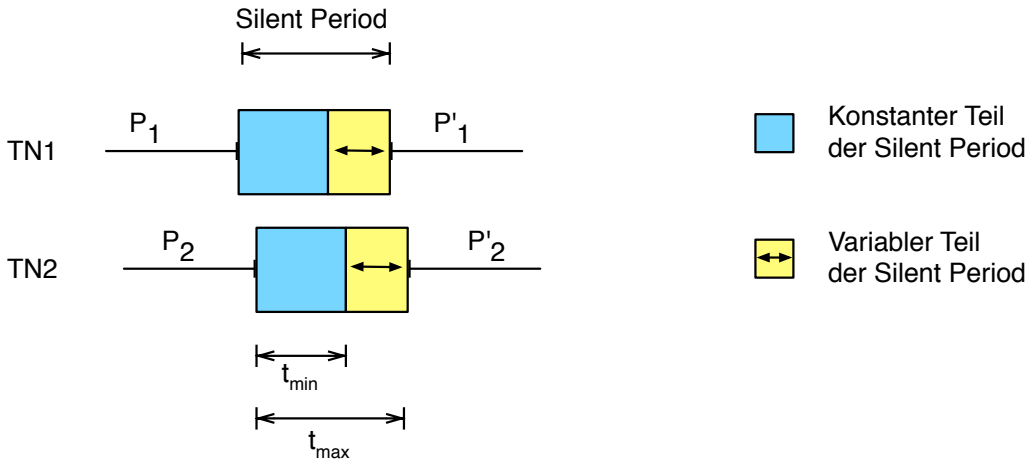


Abbildung 4.11.: Prinzip der *Silent Period* (nach [HMYS05])

einer zeitlich konstanten *Silent Period* unterscheidbar wären. Erst durch eine variable Komponente entsteht eine gewisse Unsicherheit für den Angreifer, da nun bei zwei oder mehr Teilnehmern nicht mehr die zuerst begonnene *Silent Period* auch zwingend zuerst enden muss.

In Abbildung 4.11 ist dieser Ablauf dargestellt. Die *Teilnehmer 1* und *2* (*TN 1* und *2*) wechseln mit einer Überlappung der *Silent Periods*, aber nicht synchron ihre Pseudonyme (von P_1 zu P_1' bzw. von P_2 zu P_2'). Da die variablen Teile ihrer *Silent Periods* einen Überlappungsbereich aufweisen, kann ein Angreifer in den meisten Fällen keine eindeutige Zuordnung zwischen den neuen und alten Pseudonymen vornehmen. Die Dauer einer *Silent Period* kann sich dabei stets zwischen t_{min} (entspricht dem konstanten Anteil) und t_{max} (entspricht der Summe aus konstantem und maximalen variablen Anteil) bewegen.

In einer Erweiterung, die Huang et al. in [HYMS05] veröffentlichen, argumentieren sie, dass eine bessere Vermischung der Teilnehmer durch eine Koordination der *Silent Periods* mehrerer Teilnehmer zu einer besseren Unbeobachtbarkeit führen. Dies haben auch ihre Simulationen gezeigt.

Die Autoren schlagen vor, ein Protokoll einzuführen, in dem jeder Teilnehmer, der sein Pseudonym wechseln möchte, dies nicht einfach tut, sondern eine Koordinierungsstelle darüber informiert. Diese Zentralinstanz sammelt eine Reihe von Wechselanfragen und sendet dann zu einem von ihr gewählten Zeitpunkt an alle Wechselwilligen eine Nachricht, dass sie nun in die *Silent Period* überwechseln sollen. Huang et al. erhoffen sich von dieser Vorgehensweise eine bessere Wirkung der *Silent Period*.

Bewertung Die Wirksamkeit der von Huang et al. beschriebenen *Silent Period* hängt noch stärker als die der Mix-Zone von externen Faktoren wie der Gültigkeitsdauer der Pseudonyme und der Dichte der Knoten ab. Gerade in VANETs fällt es einem Angreifer leichter, die voraussichtliche Position bei der Wiederaufnahme des Sendens seines Opfers auf Grund der erzwungenen Bewegung entlang des Straßennetzes zu antizipieren, sofern der Pseudonymwechsel nicht in einem Kreuzungsbereich stattfindet. Dies ist jedoch hier nicht sichergestellt, da sich aus rein statistischen Gründen die meisten Teilnehmer zum Zeitpunkt des Ablaufs ihres Pseudonyms (oder zu dem Zeitpunkt, den die Zentralinstanz für den Pseudonymwechsel festlegt) nicht in einem zur Vermischung besonders geeigneten Straßenabschnitt wie einem Autobahnkreuz befinden. Die Anzahl der zur Vermischung geeigneten anderen Teilnehmer ist, bedingt durch die durch das Straßennetz beschränkte Bewegungsfreiheit der Fahrzeuge, ebenfalls relativ gering. So kommen auf geraden Autobahnabschnitten nur die Fahrzeuge in unmittelbarer Nähe in Frage. Durch die vermutlich relativ gleich verteilten Ablaufzeitpunkte der Pseudonyme ist es abhängig von der Länge der *Silent Period* (die aus Verkehrssicherheitsgründen möglichst kurz gehalten werden sollte) eine statistische Frage, ob innerhalb eines geeigneten Intervalls einer der in Frage kommenden anderen Teilnehmer ebenfalls einen Pseudonymwechsel durchführt. Insgesamt ist der Schutz in *VANET*s somit als eher schwach einzustufen.

Selbst die vorgestellte Erweiterung um einen koordinierten Eintritt in die *Silent Period* bringt nur wenig zusätzlichen Schutz. Die Wahrscheinlichkeit, dass mehrere Fahrzeuge zur gleichen Zeit ihr Pseudonym wechseln, ist nun zwar als etwas höher einzuschätzen, allerdings gelten alle Restriktionen, die sich aus dem Straßenverlauf und der Verkehrsdichte ableiten, nach wie vor.

Auch aus Verkehrssicherheitsgründen kann durch die *Silent Period* in Ad-hoc-Fahrzeugnetzen eine neue Problematik entstehen: Da Fahrzeuge zu einem selbst gewählten Zeitpunkt das Senden ihrer Beacons einstellen, kann dieser Moment natürlich von anderen Teilnehmern kaum vorhergesehen werden. Damit verschwindet ein Fahrzeug unerwartet aus dem Wahrnehmungsbereich der Fahrassistenzsysteme, ohne dass der Fahrer darauf vorbereitet werden kann. In einer Mix-Zone ist es dagegen offenkundig, dass die Systeme für einen klar definierten Zeitraum nicht oder nur eingeschränkt zur Verfügung stehen.

4.3.2. Silent Cascade

Basierend auf der von ihnen vorgeschlagenen (*Random*) *Silent Period* (vgl. Abschnitt 4.3.1) stellen Huang et al. in [HYMS06] *Silent Cascade* als Erweiterung vor. Ausgehend von der Problematik, dass gewisse Dienste (wie beispielsweise IP-Telefonie) Qualitätsanforderungen (*Quality of Service*, *QoS*) an die Übertragung

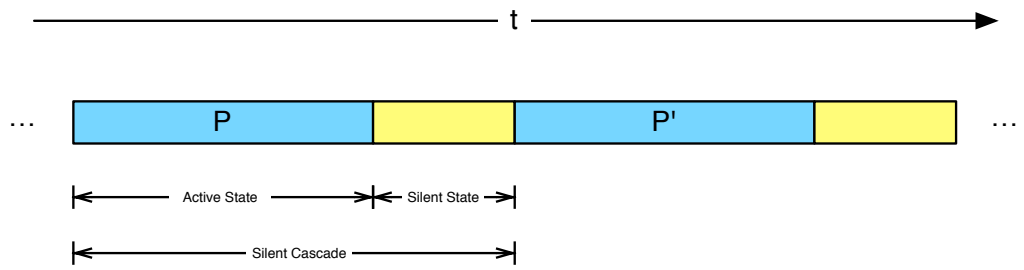


Abbildung 4.12.: *Silent Cascade* (nach [HYMS06])

stellen und diese bei einer Funkstille wie in der *Silent Period* nicht gewährleistet werden kann, versuchen die Autoren einen Trade-Off zwischen Dienstnutzung und Unbeobachtbarkeit zu bieten.

Das Grundkonzept bleibt dabei weitgehend unverändert, auch wenn die Autoren andere Begrifflichkeiten verwenden. Eine *Silent Cascade* (vgl. Abbildung 4.12) bezeichnet nun eine Abfolge von einer Phase, in der Kommunikation möglich und ein Teilnehmer unter einem bestimmten Pseudonym P erreichbar ist (*Active State* oder *Active Period* genannt) und einer weiteren Phase, in der Funkstille gewahrt und das Pseudonym gewechselt wird (*Silent State* oder *Silent Period* genannt).

Die Länge der beiden Phasen – *Active State* und *Silent State* – wird durch die QoS-Anforderung der genutzten Dienste bestimmt, welche somit sichergestellt werden kann. Die Autoren argumentieren nun, dass eine Reihe von Funkstillen in der Nähe anderer Teilnehmer jeweils eine kleine lokale Mix-Zone bilden (wie auch bei der *Silent Period*), und diese sich gemeinsam als ein Mix-Netzwerk (bzw. eine Mix-Kaskade) interpretieren ließen.

Nach diesen Überlegungen evaluieren Huang et al. ihr Konzept mit Hilfe einer Simulation, die den Zusammenhang zwischen einem geforderten QoS und der Unverkettbarkeit von Pseudonymen über den zusätzlichen Parameter *Delay*, der die Verzögerung bis zum Erreichen eines bestimmten Levels an Unbeobachtbarkeit abbildet, aufzeigen soll.

Bewertung Wie auch in ihrer vorangegangenen Arbeit [HYMS05] betrachten Huang et al. hier kein Ad-hoc-Fahrzeugnetz, sondern ein generisches drahtloses Kommunikationsnetzwerk. Aus ihren Ausführungen lässt sich schließen, dass sie eine beliebige und freie Bewegung der Knoten zu Grunde legen und somit die wesentliche Einschränkung eines *VANET*s in Hinblick auf die Unvorhersehbarkeit der Bewegungen nicht weiter berücksichtigen. Somit ist die Gültigkeit der Aussagen für Ad-hoc-Fahrzeugnetze fraglich.

Zudem ist die Argumentation der Autoren – und im Speziellen ihre Aussagen bezüglich der Verbesserung gegenüber der zuvor vorgestellten *Silent Period* – nicht komplett schlüssig. Huang et al. beschreiben das einmalige Durchführen des *Silent States* (vergleichbar mit der in Abschnitt 4.3.1 beschriebenen *Silent Period*) formal als einen Kommunikationsmix, die Wiederholung dieses Vorgangs über die Zeit als Mix-Netzwerk. Unklar bleibt in diesem Zusammenhang jedoch, warum nicht auch eine *Silent Period* auf exakt die gleiche Art und Weise beschrieben werden könnte. Auch hier gibt es eine Abfolge von Funkstillen, die jeweils eine lokal und temporal begrenzte Mix-Zone ausgestalten und somit als Mix-Netzwerk betrachtet werden können. Vor allem liegt der Verdacht nahe, dass gerade durch eine mehrfache Iteration dieser Funkstille erst ein bestimmtes und gewünschtes Niveau an Unbeobachtbarkeit erzeugt werden kann.

Dies wird von den Autoren als Argument für ihr Integrationskonzept von QoS in das System gewertet. Im Prinzip lässt sich die Erkenntnis auf zwei wesentliche Punkte reduzieren: (1) Die Länge der Intervalle mit und ohne Kommunikation (*Active State* bzw. *Silent State*) wird durch die QoS-Anforderungen der genutzten Anwendungen bestimmt. (2) Je höher die QoS-Anforderungen sind, also je kürzer und seltener eine Funkstille eingehalten werden soll, desto länger dauert es, bis ein gewünschtes Niveau an Unbeobachtbarkeit erreicht werden kann.

Dies zeigt somit lediglich den Trade-Off, in dem sich die ganze Diskussion bewegt oder zu bewegen scheint: Je datenschutzfreundlicher das Ad-hoc-Fahrzeugnetz ausgestaltet werden soll, desto schwieriger ist es, bestimmte Dienste effektiv nutzen zu können, und umgekehrt. *Silent Cascade* [HYMS06] kann also keinen wesentlichen Beitrag dafür liefern, entweder die Unbeobachtbarkeit zu verbessern oder – bei konstantem Niveau an Unbeobachtbarkeit – mehr Funktionalität zu bieten.

4.3.3. CARAVAN und AMOEBA

Sampigethaya et al. greifen in [SHL+05] das von Huang et al. vorgestellte Konzept der *Silent Period* auf, betrachten es eingehend und bauen es in ein umfassenderes Konzept zum Schutz der Privatsphäre in VANETs ein, das einige interessante Ideen beinhaltet.

Bei der Betrachtung der *Silent Period* wird argumentiert, dass die maximale Zeitspanne für eine Funkstille den Zeitraum zwischen zwei Beacons umfasst. Dabei gehen die Autoren von einer Zeitspanne von mehreren hundert Millisekunden aus (vgl. [SHL+05]). Die Ergebnisse einer durchgeführten Simulation stellen sich allerdings nicht besonders positiv für die Unverfolgbarkeit der Teilnehmer dar: Die durchschnittliche Größe der Anonymitätsgruppe bleibt bei Längen der *Silent Period* von bis zu 500 ms kleiner als *2*. Erst bei einer Erhöhung der Zeitspanne auf über zwei

Sekunden kann statistisch eine ausreichende Unbeobachtbarkeit beim Pseudonymwechsel gewährleistet werden, die aber zu Lasten der Verkehrssicherheit geht.

Als Konsequenz aus den ernüchternden Simulationsergebnissen diskutieren Sampigethaya et al. einen anderen Ansatz. Dabei berücksichtigen sie nicht länger einen globalen Angreifer, der sämtliche Broadcast-Nachrichten verfolgen kann, sondern wechseln zu einer Beobachtung mit Hilfe von RSUs. Die Überlegung, die dabei zu Grunde liegt, ist, dass zwischen jeweils zwei RSUs immer ein gewisser Abstand besteht. Werden nun Beacons mit einem Senderadius ausgestrahlt, der kleiner ist als der Abstand zwischen zwei RSUs, so können sie unter Umständen nicht von den RSUs empfangen und somit vom Angreifer nicht zur Kenntnis genommen werden. Es entsteht somit ein nicht überwachter Bereich, in dem Fahrzeuge ihre Pseudonyme wechseln.

Die Simulationsergebnisse der Autoren zeigen, dass für die von ihnen gewählten Daten der zuletzt beschriebene Ansatz gegenüber einer *Silent Period* im Vorteil ist, was in erster Linie daran liegt, dass die unbeobachtbare Zeitspanne größer ist. Andererseits scheint die *Silent Period* ein deutlich besseres Verhältnis zwischen unbeobachteter Zeit und Anonymität zu bieten, da bei dem von den Autoren als *Simple Tracking* bezeichneten Vorgehen des Angreifers eine zweisekündige *Silent Period* ein vergleichbares Maß an durchschnittlicher Unbeobachtbarkeit liefert wie Abstände zwischen RSUs von zwei Kilometern.

Für Nachrichtentypen, die seltener gesendet werden als Beacons, zum Beispiel Daten von *Location-based Services* (*LBS*) oder Verkehrsinformationen (wie z. B. Verkehrsdichte), schlagen Sampigethaya et al. den Einsatz von Gruppenbildung mit einem Proxy vor. Dieses Prinzip wurde unter anderem bereits für ein Routing-Schema in Ad-hoc-Netzen vorgeschlagen (vgl. [LJDC+00]).

Fahrzeuge, die sich nahe beieinander befinden und sich in die gleiche Richtung bewegen, haben häufig eher geringe relative Geschwindigkeiten. Daher können sie über einen gewissen Zeitraum eine Gruppe bilden. Diese soll all diejenigen Fahrzeuge umfassen, die gegenseitig ihre Broadcastnachrichten empfangen können. Ein Teilnehmer wird dabei zum *Group Leader* (*GL*) bestimmt. Dieser *GL* fungiert als Proxy und kommuniziert anstelle aller Gruppenmitglieder. Auf diese Weise kann zum einen Übertragungskapazität auf der Luftschnittstelle eingespart werden, weil redundante Nachrichten nun nicht mehr gesendet werden, und zum anderen haben alle Gruppenmitglieder außer dem *GL* eine lange Phase ohne Kommunikation, also somit eine lange *Silent Period*, in der sie ihr Pseudonym unbeobachtbar wechseln können.

Bei der Nutzung von *LBS* findet die Kommunikation über den *GL* statt (vgl. Abbildung 4.13). Zu diesem Zweck wählt der Teilnehmer (TN_1) ein Pseudonym (welche Anforderungen dabei erfüllt werden müssen, wird nicht näher ausgeführt) und erzeugt einen *Application Request*, den er an den *Group Leader* (*GL*) schickt. Die-

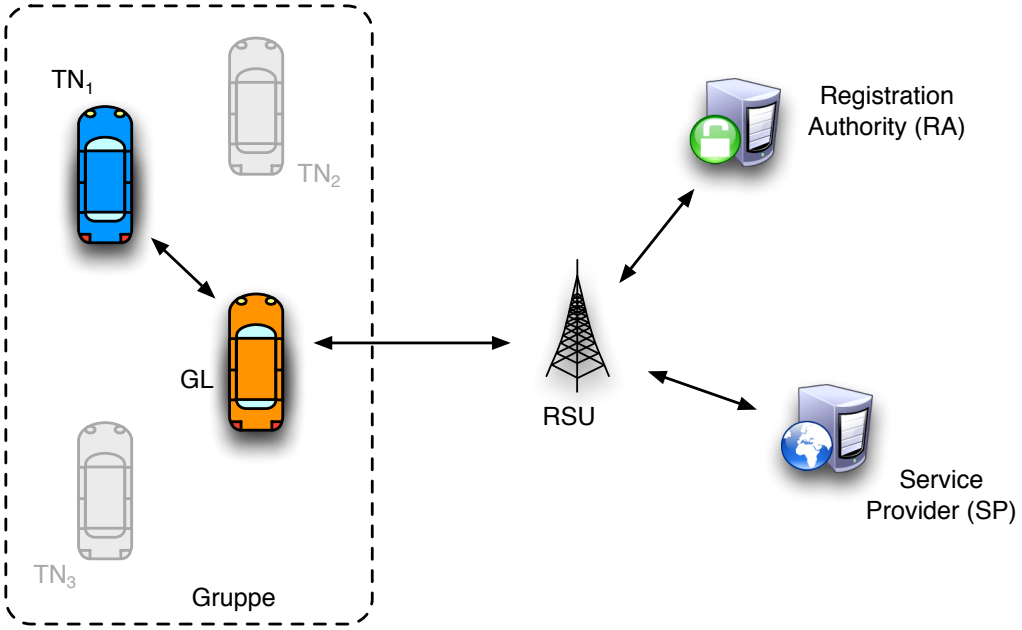


Abbildung 4.13.: Nutzung von *LBS* in CARAVAN (nach [SHL+05])

ser leitet ihn unter seiner eigenen Absenderadresse an die *Registration Authority* (*RA*), eine vertrauenswürdige Stelle, weiter. Dort wird der Request validiert und anschließend ein Sitzungsschlüssel k erzeugt, der sowohl an den TN_1 als auch an den *Service Provider* (*SP*) des angeforderten Dienstes übermittelt wird. Mit Hilfe dieses Schlüssels kann nun eine Ende-zu-Ende-verschlüsselte Kommunikation zwischen dem TN_1 und dem *SP* über den *GL* und eine RSU stattfinden. Da TN_1 ein neu gewähltes Pseudonym für die Kommunikation mit dem *SP* gewählt hat, kann er weiterhin Beacons unter seinem ursprünglichen Pseudonym versenden und die Kommunikationen können dennoch nicht verkettet werden.

Ein globaler Beobachter kann möglicherweise dennoch durch eine zeitliche Korrelation zwischen einer Nachricht von TN_1 zu *GL* und der unmittelbar darauf folgenden Weiterleitung von *GL* zur RSU erkennen, dass TN_1 das kommunizierende Fahrzeug ist. Um diesem Problem entgegenzuwirken, schlagen Sampigethaya et al. vor, dass der *GL* zusätzlich die Funktionalität eines Kommunikationsmixes erbringen soll: Er wartet auf mehrere *LBS*-Anfragen unterschiedlicher Teilnehmer seiner Gruppe und leitet diese dann gemeinsam und umsortiert an die RSU weiter.

In der Publikation [SLHP07], die das auf CARAVAN aufbauende Konzept AMOEBA vorstellt, bieten die Autoren lediglich eine ausführlichere Analyse der Angriffe auf VANETs; die zuvor beschriebenen Prinzipien werden nicht verändert oder weiter ausgestaltet.

Bewertung Sampigethaya et al. diskutieren ausführlich die Einschränkungen der *Silent Period*. Ein Pseudonymwechsel zwischen zwei Beacons ohne zusätzliche Funkstille ist, wie bereits eingangs in Abschnitt 4.1.3 beschrieben, problematisch, da eine Unbeobachtbarkeit auf Grund der räumlichen Verkettung nicht gewährleistet ist. Das Ausdehnen der *Silent Period* auf mehrere Sekunden würde wiederum die Verkehrssicherheit gefährden, da Fahrassistenzsysteme nicht zuverlässig arbeiten könnten. Das gleiche Problem besteht in einer Mix-Zone, jedoch ist dort für jeden Teilnehmer kommunizierbar, dass Fahrassistenzsysteme nicht zur Verfügung stehen. Die *Silent Period* dagegen kann ohne Vorwarnung überall auftreten und dabei werden nicht nur die in die Funkstille eintretenden Fahrzeuge, die von ihren Systemen vorgewarnt werden können, sondern auch alle umliegenden Fahrzeuge beeinträchtigt.

Die Kommunikation über einen *Group Leader* als Proxy, der sogar noch als Kommunikationsmix fungieren kann, bietet eine interessante Lösung für das Problem der Überwachung durch einen *Service Provider*. Dieser erhält nun durch ein nur für die Kommunikation zwischen Teilnehmer und *SP* gewähltes Pseudonym kaum noch Informationen über den *TN*, und auch seine Position ist ihm nur grob bekannt. Allerdings reichen diese Informationen natürlich auch, um grobe Bewegungsprofile zu erstellen und eine Verkettung kann möglicherweise auf Inhaltsdaten erfolgen.

Die Autoren können mit ihrem Konzept somit nur einen Schutz vor *Service Providern* bieten, ein Angreifer, der Beacons vor Ort überwacht, kann nach wie vor jeden Teilnehmer anhand seiner Telematiknachrichten verfolgen.

4.4. Benutzerzentrierte Ansätze

4.4.1. Swing & Swap

Mit dem Konzept *Swing & Swap* stellen Li et al. in [LSHP06] eine weitere Ergänzung zu der *Silent Period* vor. Das Hauptaugenmerk liegt hier darin, dem einzelnen Teilnehmer Kontrolle über Pseudonymwechsel einzuräumen und eigene Entscheidungen zu ermöglichen. Dieser nutzerzentrierte Ansatz soll sich in den beiden Teilkonzepten – *Swing* und *Swap* – widerspiegeln und dabei eine bessere Unverkettbarkeit als die *Silent Period* bieten.

Swing *Swing* bietet Teilnehmern die Möglichkeit, einen Pseudonymwechsel einzuleiten, wenn die Lebensdauer ihres aktuell verwendeten Pseudonyms abgelaufen ist oder der aktuelle Level an Unbeobachtbarkeit einen selbstgewählten Schwellwert unterschreitet. Trifft einer dieser beiden Fälle zu, so wird überprüft, ob sich in der Nähe des Teilnehmers andere Knoten aufhalten und zudem ein Wechsel der

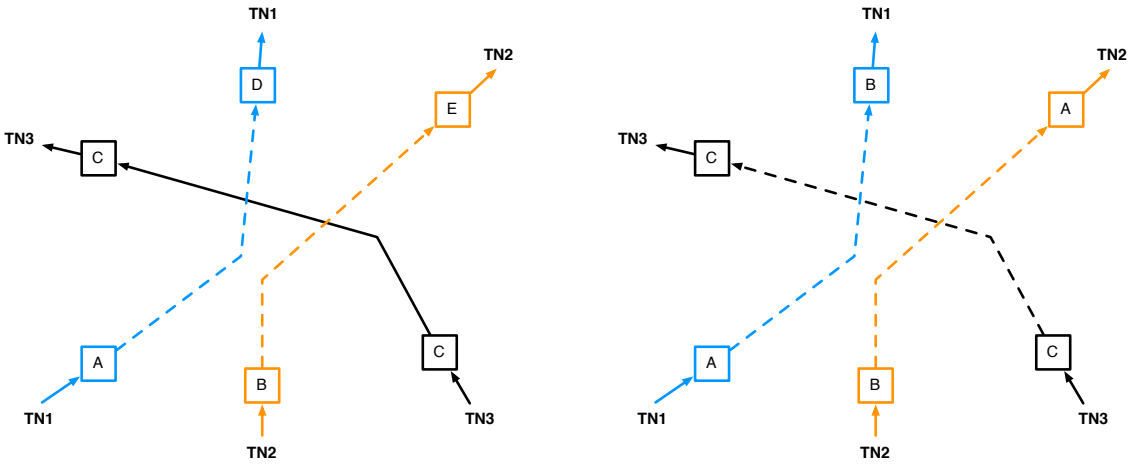
Bewegungsrichtung und/oder der Geschwindigkeit bevorsteht, da nur unter diesen beiden Bedingungen ein unbeobachtbarer Pseudonymwechsel erfolgreich durchgeführt werden kann. Alternativ dazu kann auch eine eingegangene *Update Message* (siehe unten) zusammen mit einem bevorstehenden Wechsel von Bewegungsrichtung und/oder Geschwindigkeit einen Pseudonymwechsel auslösen.

Hat sich ein Teilnehmer entschieden, einen Pseudonymwechsel durchzuführen, so sendet er eine *Update Message*, um andere Teilnehmer darauf hinzuweisen und sie zu motivieren, ebenfalls einen Pseudonymwechsel durchzuführen. Anschließend beginnt er eine *Silent Period*, wie sie auch bereits in den vorangegangenen Abschnitten beschrieben wurde. Während dieser Funkstille wechselt der Teilnehmer nun sein Pseudonym und kann anschließend unter diesem neuen Pseudonym das Senden wieder aufnehmen. Andere Teilnehmer in der Nähe, die sich auf Grund der *Update Message* nun ebenfalls entscheiden, eine Funkstille einzugehen und ihr Pseudonym zu wechseln, ermöglichen das Bilden einer Anonymitätsgruppe und erhöhen somit die Unbeobachtbarkeit aller Beteiligten.

Trotz der Nutzerzentrierung dieses Konzepts kann durch die *Update Message* eine Koordination herbeigeführt werden, wodurch bessere Ergebnisse als mit einer reinen *Silent Period* erzielt werden können.

Swap Das Konzept *Swap* soll den nutzerzentrierten Pseudonymwechsel *Swing* ergänzen und die erreichbare Unbeobachtbarkeit maximieren. Dazu wird hier das Augenmerk auf die Teilnehmer gelegt, die sich aus unterschiedlichen Gründen bei Erhalt einer *Update Message* dagegen entscheiden, ebenfalls einen Pseudonymwechsel durchzuführen. Auch sie sollen zu einem gewissen Grad geschützt werden und auch zur Vergrößerung der entstehenden Anonymitätsgruppe beitragen. Zur Erläuterung des Prinzips soll das Beispiel in Abbildung 4.14 nach Li et al. ([LSHP06]) herangezogen werden.

In Abbildung 4.14a bewegen sich drei Teilnehmer in einem näheren Umfeld. Alle von ihnen ändern ihre Richtung und sind somit Kandidaten für einen unbeobachtbaren Pseudonymwechsel nach *Swing*. Wie oben beschrieben, ist eine Voraussetzung für die Teilnahme an dem *Swing*-Protokoll entweder ein abgelaufenes Pseudonym oder ein unzureichender eigener Level an Unbeobachtbarkeit. In diesem Beispiel sei die Bedingung für die *Teilnehmer* (*TN*) *1* und *2* erfüllt, jedoch nicht für *Teilnehmer 3*. Die *Teilnehmer 1* und *2* wechseln somit ihr Pseudonym nach Protokoll (*TN1*: von *A* zu *D* und *TN2*: von *B* zu *E*) und wahren eine vorübergehende Funkstille (unterbrochene Linie). Ein Angreifer hat nach der Funkstille nun eine Unsicherheit darüber, welche Pseudonyme die *Teilnehmer 1* und *2* jetzt tragen; *Teilnehmer 3* hat keinerlei Gewinn an Privatsphäre erhalten. Zudem besteht die Anonymitätsgruppe nun nur aus *TN1* und *TN2* und hat somit nicht die maximal mögliche Größe erreicht.



(a) Situation in *Swing* **(b)** Situation in *Swap*

Abbildung 4.14.: *Swing* und *Swap* nach [LSHP06]. Kästen stellen Pseudonyme dar, unterbrochene Linien bedeuten Funkstillen.

Um die Anonymitätsgruppe zu maximieren, schlagen Li et al. vor, dass die *Teilnehmer 1* und *2*, statt ein neues Pseudonym zu wählen, ihre Pseudonyme mit einer bestimmten Wahrscheinlichkeit (diese könnte, um die Unsicherheit des Angreifers zu maximieren, $0,5$ betragen) austauschen. Dies ist in Abbildung 4.14b dargestellt. Da der *Teilnehmer 3* ebenfalls in der Nähe der beiden pseudonymtauschenden Knoten seine Richtung ändert, kann er einfach in die Anonymitätsgruppe aufgenommen werden, indem er kurzzeitig eine Funkstille wahrt. Ein Pseudonymwechsel oder -tausch ist für ihn nicht erforderlich, da ein außenstehender Angreifer nicht erkennen kann, welche Teilnehmer untereinander (wenn überhaupt) ihr Pseudonym getauscht haben.

Um diesen Pseudonymtausch zu koordinieren, broadcastet ein Teilnehmer, dessen Pseudonym abgelaufen ist oder dessen Level an Unbeobachtbarkeit ein Minimum unterschreitet, einen sogenannten *Exchange Request*, also eine Tauschanfrage. Diese Anfrage beinhaltet einen öffentlichen asymmetrischen Schlüssel dieses Teilnehmers. Alle empfangenen Fahrzeuge antworten mit einer *Exchange Response*, die neben einer Zu- oder Absage für einen Pseudonymwechsel einen öffentlichen Schlüssel des Absenders enthält und für den anfragenden Teilnehmer asymmetrisch verschlüsselt ist. Der initiierende Teilnehmer wählt nun aus den Zusagen einen beliebigen anderen Teilnehmer zufällig zu seinem Tauschpartner und broadcastet eine für diesen verschlüsselte Nachricht, die die Information enthält, ob nun tatsächlich das Pseudonym getauscht werden soll (dies findet ja nur mit einer bestimmten Wahrscheinlichkeit statt). Die beiden Tauschpartner sowie alle in der Nähe befindlichen Teilnehmer kön-

nen nun eine gewisse Zeit Funkstille wahren und die Pseudonyme können gewechselt werden. Während des Tauschs werden auch für die Zurechenbarkeit verantwortliche Stellen des *VANET*s, beispielsweise eine *Certification Authority* oder eine *Trusted Third Party*, kontaktiert und über den Tausch informiert. Nachdem eine Phase der Funkstille durchlaufen wurde, konnte somit die Anonymitätsgruppe maximiert werden und ein Angreifer kann nicht mehr unterscheiden, welcher Teilnehmer nun welches Pseudonym führt.

Evaluation der Autoren Li et al. evaluieren ihre Konzepte analytisch und mit Hilfe von Simulationen und stellen es insbesondere der *Silent Period* nach Huang et al. [HMYS05] gegenüber. Die Simulationen wurden jeweils für alle drei Konzepte – *Swing*, *Swap* und *Silent Period* – einmal mit einer freien und zufälligen Bewegung der Knoten (vergleichbar Fußgängern auf großen Plätzen) nach dem *Random Waypoint Model* [YLN03] und mit beschränkten Bewegungen (vergleichbar mit Fahrzeugen auf Straßen) nach dem *Manhattan Model* [BSH03] durchgeführt. Bestimmt wurden in den Simulationen jeweils die Entropie der Anonymitätsgruppen in Abhängigkeit der Knotendichte (Details siehe [LSHP06]).

Es zeigt sich, dass *Swap* in allen Fällen überlegen war, gefolgt von *Swing*, das auch noch akzeptable Ergebnisse liefert. *Silent Period* zeigt sich dagegen teilweise deutlich abgeschlagen. Die erhöhte Unbeobachtbarkeit erkauft man somit mit einem erhöhten Overhead; hier ist *Swap* deutlich aufwändiger als *Swing*; *Silent Period* benötigt keinerlei Koordination. Die erreichbare Unbeobachtbarkeit war jeweils bei freier Bewegung deutlich höher als bei beschränkter Bewegung. Dies ist ein intuitives Ergebnis, da ein Angreifer unter Berücksichtigung der Bewegungsbeschränkungen eine Reihe von möglichen Kandidaten eliminieren und die Privatheit damit einschränken kann. Als Lösung für diese Problematik schlagen die Autoren vor, bei der Koordinierung von *Swing* oder *Swap* den eigenen Bewegungsvektor mitzuliefern, um in Frage kommende Kooperationspartner zuverlässiger identifizieren zu können und damit zugleich die eigene Unbeobachtbarkeit besser einschätzen zu können.

Bewertung Mit *Swing & Swap* gelingt es Li et al., die in Abschnitt 4.3.1 beschriebene *Silent Period* deutlich zu verbessern. Dies rührt überwiegend daher, dass bei diesen Konzepten eine Koordination der Kommunikationspartner zum gemeinsamen Wechsel der Pseudonyme stattfindet und somit die Wahrscheinlichkeit für einen gleichzeitigen Eintritt in eine Funkstille nicht mehr ein statistisches Problem ist. Diese Verbesserung wird mit einem gewissen Overhead erkauft, der durch den Austausch der Nachrichten zur Koordination erzeugt wird. Dies dürfte jedoch vermutlich insbesondere bei *Swing* kein großes Problem darstellen.

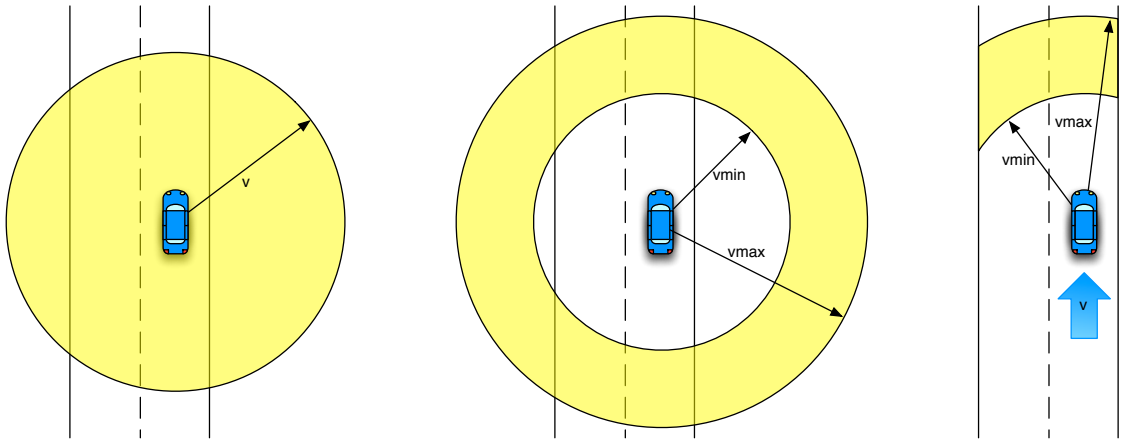
Bei *Swap* gibt es weitere Probleme, die von den Autoren nicht adressiert werden. So stellt sich die Frage, welche Konsequenzen ein Pseudonymtausch auf Routingprotokolle und somit die Zustellung von Nachrichten hat. Das Problem der Zurechenbarkeit konnten Li et al. in vielen Fällen dadurch lösen, dass bei einem Pseudonymtausch auch immer eine Zentralinstanz kontaktiert und informiert werden muss. Allerdings stellen auch sie die Frage, wie gerade in *VANET*s trotzdem eine Zurechenbarkeit garantiert werden kann, da unter anderem ein sofortiger Kontakt zu einem zentralen Dienst möglicherweise nicht immer garantiert werden kann und zudem in vielen Fällen Pseudonyme einen sehr starken Bezug zu den Identitäten der einzelnen Teilnehmer besitzen.

Weiterhin weist die Argumentation der Autoren bezüglich *Swap* Ungereimtheiten auf: Ausgangspunkt war die Situation, in der einzelne Benutzer nicht an einem Pseudonymwechsel nach *Swing* teilnehmen. Diese sollen mit Hilfe von *Swap* ebenfalls in die Anonymitätsgruppe einbezogen werden. Betrachtet man jedoch die Motivation der Benutzer, auf einen Pseudonymwechsel zu verzichten, bleiben im Wesentlichen zwei Fälle übrig: 1) der Teilnehmer fühlt sich ausreichend geschützt (sein Level an Unbeobachtbarkeit ist ausreichend hoch) oder 2) der Teilnehmer nutzt gerade einen Dienst und möchte daher nicht in eine Funkstille eintreten.

Nun ist es bei *Swap* zwar nicht erforderlich, das Pseudonym zu ändern um einen Zugewinn an Unbeobachtbarkeit zu erzielen, jedoch muss kurzzeitig eine Funkstille gewahrt werden, in der keinerlei Dienste verfügbar sind. In dem ersten beschriebenen Fall wird ein Teilnehmer keine Veranlassung sehen, sich an einer Funkstille zu beteiligen, im zweiten Fall wird er sich sogar dagegen wehren, da er sonst einen Verbindungsabbruch bei dem genutzten Dienst hinnehmen müsste. Ohne Funkstille trägt er jedoch nicht zu der Anonymitätsgruppe bei und das Konzept kann seinen Zusatznutzen nicht entfalten. Zudem gibt es für einen Teilnehmer kaum einen Grund, zwar eine Funkstille zu wahren, jedoch nicht bereit zu sein, das eigene Pseudonym zu ändern. Insofern bietet *Swap* nicht den von den Autoren dargestellten Mehrwert gegenüber einer einfachen koordinierten Funkstille wie bei *Swing*.

Sowohl *Swing* als auch *Swap* haben das Problem, dass die entstehende Zone ohne Kommunikation sehr klein ausfällt. Dies gilt insbesondere bei beschränkten Bewegungsmöglichkeiten der Knoten, wie es in *VANET*s durch das Straßennetz und Verkehrsregeln der Fall ist. Dies soll in Abbildung 4.15 anschaulich gemacht werden.

Abbildung 4.15a zeigt eine naive Herangehensweise an die möglichen Aufenthaltsorte eines Teilnehmers nach einer Funkstille ausgehend von seiner Startposition, die durch das Fahrzeug markiert ist: Man zieht einen Kreis mit dem Radius der innerhalb der Funkstille zurücklegbaren Strecke um den Teilnehmer und kennt nun den Bereich, in dem das Fahrzeug sich bei Wiederaufnahme des Sendens aufhalten muss. Dieser Bereich lässt sich jedoch deutlich einschränken, indem man berück-



(a) Theoretischer Idealfall **(b)** Berücksichtigung der Maximalbeschleunigung **(c)** Realität

Abbildung 4.15.: Beschränkte Aufenthaltsmöglichkeiten von Teilnehmern in *VANETs*

sichtigt, dass ein sich bewegendes Fahrzeug gewissen physikalischen Phänomenen wie beispielsweise Trägheit oder Reibung ausgesetzt ist. Daraus lässt sich abhängig von der aktuellen Geschwindigkeit eine minimale und maximale Geschwindigkeit am Ende der Funkstille ermitteln, und man erhält somit einen Kreisring, der die nun noch möglichen Aufenthaltsorte markiert (vgl. Abbildung 4.15b). Dabei ist jedoch bislang weder die Bewegungsrichtung des Fahrzeugs noch die Einschränkung durch den Straßenverlauf berücksichtigt. Dies führt zu einem kleinen Kreisringsegment, welches die von dem Teilnehmer während der Funkstille tatsächlich erreichbaren Orte aufzeigt (vgl. Abbildung 4.15c). Die Größe dieses Bereichs ist sehr stark von der Dauer der Funkstille abhängig; ist diese zu kurz, befindet sich möglicherweise nur ein einziges Fahrzeug – das gesuchte – darin. Die Autoren geben dazu keine klaren Empfehlungen ab, jedoch weisen ihre Simulationsergebnisse der Anonymität in Abhängigkeit der Dauer der Funkstille bei beschränkter Bewegung keinen sehr starken Zusammenhang auf. Dies bedeutet, dass sich auch durch Ausdehnung der Funkstillendauer die Ergebnisse nicht wesentlich verbessern lassen.

Eine weitere Schwierigkeit dieses Konzepts ist die Tatsache, dass jeder Teilnehmer – um das Protokoll durchlaufen zu können – in der Lage sein muss, die eigene aktuelle Unbeobachtbarkeit abzuschätzen und einem gewünschten Level an Unbeobachtbarkeit gegenüberzustellen. Da jedoch der einzelne Teilnehmer gerade keine globale Sicht auf den Verkehr hat und somit nicht mit Hilfe von Analysen die Größe von Anonymitätsgruppen ermitteln kann, stellt sich die Frage, wie der Einzelne diese Informationen erhält. Weiterhin ist es sehr schwierig, die eigene gewünschte Unbe-

obachtbarkeit in konkreten Werten auszudrücken (insbesondere für Laien); es ist sogar fraglich, ob es nicht im Interesse des Einzelnen liegt, die eigene Unbeobachtbarkeit stets (unter gegebenen Rahmenbedingungen) zu maximieren. Diese Aspekte werden von den Autoren nicht behandelt und bleiben daher offen.

Insgesamt lässt sich somit sagen, dass mit *Swing* und *Swap* zwar interessante Ansätze zur Koordinierung benutzerzentrierter Funkstillen vorgestellt werden, diese jedoch noch einige nicht gelöste Probleme aufweisen und daher noch weiterer Entwicklung bedürfen.

4.4.2. SLOW

Mit dem Konzept *SLOW*, ein Akronym für *silence at low speeds*, lösen sich auch Buttyán et al. in [BHWW09] sowohl von der räumlichen als auch von der zeitlichen Festlegung der Funkstille. Sie verfolgen den Ansatz, dass Teilnehmer, wann immer sie eine bestimmte Geschwindigkeit (die Autoren schlagen 30 km/h vor) unterschreiten, das Senden aller Nachrichten einstellen. Können sie die Funkstille eine gewisse Zeit halten, so wechseln sie zudem ihr Pseudonym. Gerade im städtischen Umfeld fällt die Geschwindigkeit häufig unter die vorgeschlagene Marke und ein Pseudonymwechsel kann somit stattfinden. Unterstützend wirkt sich zudem aus, dass sich insbesondere in Kreuzungsbereichen zusätzlich zu der gewünschten geringen Geschwindigkeit einerseits mehrere Teilnehmer in unmittelbarer Nähe aufhalten und diese andererseits – je nach Situation – eine Reihe von möglichen Richtungsänderungen besitzen.

Die Argumentation für diesen Ansatz stützt sich auf die Erkenntnis, dass bei geringen Geschwindigkeiten weniger Unfälle passieren und diese zudem glimpflicher verlaufen (vgl. [BT98, KMMP97, LP99]). Somit ist es möglich, eine Funkstille ohne signifikanten Verlust der Verkehrssicherheit durchzuführen. Sollte dennoch ein Teilnehmer erkennen, dass sich ein Fahrzeug mit zu hoher Geschwindigkeit nähert, kann er die Funkstille brechen und eine Warnung aussenden. Mit diesem Prinzip wird möglicherweise etwas Privatsphäre zugunsten der Gefahrenabwehr aufgegeben.

Buttyán et al. können mit Hilfe von Simulationen zeigen, dass die Erfolgsrate eines Angreifers, ein Fahrzeug über eine Kreuzung hinweg zu verfolgen, durch den Einsatz von *SLOW* und einem Schwellwert von 30 km/h (je nach Verkehrsdichte) bereits auf Werte zwischen 30 % und 55 % fällt. Durchfahren die Fahrzeuge mehrere Kreuzungen, so bleibt die Erkennungsrate stets unter 20 %.

Neben ihren recht guten simulativen Ergebnissen räumen die Autoren ein, dass ihr Konzept nur in urbanen Regionen sinnvoll zur Anwendung kommen kann. Sollten sich Fahrzeuge außerhalb von Stadtgebieten bewegen, so wird die für *SLOW* benötigte Grenzgeschwindigkeit zu selten unterschritten, als dass ein ausreichender Schutz der Privatsphäre gegeben sein kann.

Bewertung Dieser Ansatz von Buttyán et al. erweist sich als äußerst interessant. In urbanen Gebieten kommt es so ohne weitere Organisation von außen zu Bereichen mit Funkstille an nahezu jeder Kreuzung. Insbesondere an komplexen (weil vielspurigen) und viel befahrenen Kreuzungen, die in der Regel mit Ampeln ausgestattet sind, kann man damit rechnen, eine ausreichend große Gruppe von anderen Teilnehmern mit Geschwindigkeiten unter dem Schwellwert anzutreffen. Somit kommt es – unterstützt durch die Anzahl an Ein- und Ausfahrten des Bereichs mit Funkstille möglicherweise zu einer recht guten Vermischung und somit Unvorhersehbarkeit für einen Angreifer.

Ungelöst bleibt jedoch auch bei diesem Konzept das Problem, das bei Ansätzen ohne zentrale Organisation stets vorhanden ist: Ohne dass sich Fahrer darauf einstellen können, werden Fahrassistenzsysteme unverfügbar. Die Autoren versuchen dem zu begegnen, indem weiterhin die Beacons der anderen Teilnehmer (soweit diese noch senden) überwacht werden. Im Falle einer Kollisionsgefahr überwiegt das Verhindern eines Unfalls dem Interesse an Privatsphäre. Dies ist grundsätzlich sicherlich sinnvoll, jedoch lässt es sich nicht auf alle denkbaren Fälle verallgemeinern. Beispielsweise könnte eine Kollisionsgefahr erst sehr kurzfristig (z. B. bei einem Spurwechsel) auftreten, so dass eine adäquate Reaktion durch den Fahrer nicht mehr möglich ist.

Weiterhin gibt es durchaus auch in Bereichen geringer Geschwindigkeit (dies trifft in manchen deutschen Kleinstädten auf einen Großteil der Straßen zu) ein berechtigtes Interesse, gewisse Dienste zu nutzen. Beispiele hierfür wären Tourismusinformationen, Flottenmanagement von öffentlichem Personennahverkehr oder Post- und Paketdienstleistern oder aktuelle Verkehrsinformationen. Dies wäre mit dem *SLOW*-Modell unter Umständen vielerorts nicht mehr möglich. Zudem besteht die Gefahr, dass bei einer aktuell bestehenden Kommunikation unvermittelt die Verbindung getrennt wird, wenn das Fahrzeug die Grenzgeschwindigkeit zum Beispiel durch einen Bremsvorgang unterschreitet. Die Verwendung unterschiedlicher Pseudonyme für unterschiedliche Nachrichtentypen (vgl. Abschnitt 2.3.3) ist zwar denkbar, jedoch müssen alle vorhandenen Pseudonyme eines Teilnehmers gleichzeitig gewechselt werden, da sonst eine Verkettung weiterhin möglich ist (vgl. [BHWW09]).

Die Simulationsergebnisse von Buttyán et al., die meist eine Erkennungsrate von unter 20 % liefern, machen einen sehr vielversprechenden Eindruck, jedoch wird dabei nur von einer geringen Auflösung der Position der Teilnehmer ausgegangen: Ein Angreifer kann nur die Fahrbahn eines Teilnehmers identifizieren, nicht jedoch die Spur, in der dieser sich befindet. Könnte die Spur ermittelt werden, so nimmt (bei Betrachtung von Abbiegespuren) die Unvorhersagbarkeit der Bewegung des betroffenen Teilnehmers deutlich ab und ein Angreifer könnte Pseudonyme möglicherweise wesentlich effektiver verketten. Wie genau die Auflösung modernerer Positionsdienste (vgl. Abschnitt 2.2.2) wie *GALILEO* in der Praxis sein wird, gilt es abzuwarten.

Dennoch empfiehlt es sich im Zweifelsfall pessimistische – also für einen Angreifer günstige – Annahmen zu treffen.

4.4.3. Random Encryption Periods

Wasef und Shen stellen in ihrer Publikation [WS10] ein Konzept vor, das sie als *Random Encryption Periods*, kurz *REP*, bezeichnen. Der Grundgedanke von *REP* ist ein nutzerzentrierter Ansatz, bei dem zur Verschleierung von Pseudonymwechseln Gruppenverschlüsselung (ähnlich der *CMIX*-Zone, vgl. Abschnitt 4.2.3) zum Einsatz kommt.

Die Autoren verwenden für ihren Ansatz Zertifikate als Pseudonyme, die von einer zentralen *Trusted Authority* (*TA*) ausgestellt und den Teilnehmern zur Verfügung gestellt werden. Weiterhin setzen sie die Verteilung eines symmetrischen Gruppenschlüssels k_g an alle Teilnehmer voraus.

Eine *Random Encryption Period* kann von jedem beliebigen Teilnehmer ausgelöst werden, sobald dieser sein Pseudonym (Zertifikat) ändern möchte. Dazu broadcastet er eine Aufforderung an alle anderen in der Nähe befindlichen Teilnehmer, für die ebenfalls gewählte und mitgesendete Dauer T_{REP}, in die *Random Encryption Period* zu wechseln. Jeder, der diese Nachricht empfängt, beginnt nun mit einer Verschlüsselung aller Nachrichten unter der Verwendung des gemeinsamen symmetrischen Schlüssels k_g. Der Initiator der *REP* wechselt nun sein Pseudonym und beobachtet alle anderen Beteiligten auf folgende Ereignisse:

Mindestens ein weiterer Teilnehmer der *REP* wechselt sein Pseudonym und

1. wechselt die Geschwindigkeit oder
2. wechselt die Spur oder Fahrtrichtung.

Ist eine der beiden Bedingungen vor Ablauf von T_{REP} eingetreten, so teilt der Initiator mit einem Broadcast allen Beteiligten mit, dass die *Random Encryption Period* beendet und eine Verschlüsselung der Nachrichten nicht mehr notwendig ist. Sollte dies nicht eingetreten sein, so verlängert der Initiator mit einer anderen Nachricht die *REP*.

Die Arbeit von Wasef und Shen befasst sich darüberhinaus umfassend mit Details über die Verteilung und den Widerruf von symmetrischen Schlüsseln. Diese Aspekte sollen jedoch an dieser Stelle unberücksichtigt bleiben, da sie einerseits nichts zur Qualität der *Random Encryption Period* beitragen und andererseits unnötig komplex erscheinen.

Den Abschluss des Papiers [WS10] bildet eine Simulation zur Bewertung des Schutzes der Privatsphäre, der von *REP* geboten wird. Wasef und Shen verwenden zwei

Simulationen – städtisches Umfeld und ein Highway-Abschnitt – und nutzen das *Manhattan* beziehungsweise das *Highway Mobility Model*. Die Ergebnisse scheinen sehr überzeugend; so kann mit Hilfe von *REP* bei einer Dauer T_{REP} von 300 ms bis 500 ms in urbanen Gebieten eine durchschnittliche Größe der Anonymitätsgruppe von größer *2*, auf dem Highway-Abschnitt sogar von *15* erreicht werden. Weiterhin zeigen die Autoren, dass die *Random Encryption Periods* ein sehr günstiges Verhältnis der Anzahl an Anonymitätsgruppen der Größe *1* gegenüber größeren Anonymitätsgruppen bieten und somit der *CMIX*-Zone, dem einzigen Konzept, dem *REP* gegenübergestellt wird, deutlich überlegen sind. Die variable Dauer der *REP* (vgl. oben) schwankte dabei im urbanen Szenario zwischen ca. 700 ms und ca. 900 ms, im Highway-Szenario waren keine Verlängerungen nötig.

Bewertung Wasef und Shen können mit ihrem Konzept lediglich einen Schutz vor einem Outsider bieten. Der symmetrische Schlüssel k_g, der allen (nicht ausgeschlossenen) Teilnehmern zur Verfügung steht, kann natürlich von einem Angreifer genutzt werden um Beobachtungen während einer *Random Encryption Period* durchzuführen. Dafür wird lediglich ein Fahrzeug benötigt. Dass Beobachtungen auf Basis verschlüsselter Kommunikation möglich sind, zeigen auch die Autoren in ihrem Protokoll. Dort hat der Initiator der *REP* alle Beteiligten zu überwachen und muss auf Richtungs- und Geschwindigkeitswechsel während der Pseudonymwechsel achten. Ähnlich wie die *CMIX*-Zone bieten auch *Random Encryption Periods* keinerlei Verbesserung gegenüber dem Konzept von Plößl (vgl. [Plö09]), bei dem grundsätzlich alle Beacons von ihren Sendern mit einem gemeinsamen symmetrischen Schlüssel verschlüsselt werden (vgl. auch Abschnitt 4.2.3).

Die guten Simulationsergebnisse sind dennoch verblüffend, da andere Autoren mit ihren Konzepten oft schlechtere Ergebnisse präsentieren. Ein genauerer Blick auf die Rahmenbedingungen klärt dies jedoch schnell auf. Das erste Problem der vorliegenden Simulation besteht darin, dass nur sehr kleine Straßenabschnitte betrachtet wurden. So simulierten die Autoren mit dem Highway-Modell lediglich einen drei Kilometer langen Abschnitt mit jeweils drei Fahrspuren in jede Richtung. Fahrzeuge bewegten sich mit einer konstanten Geschwindigkeit von 80 $^{\mathrm{km}}/_{\mathrm{h}}$ durch diesen Abschnitt und konnten ihn somit in lediglich *2:15* Minuten durchqueren. Die gesamte Simulation dauerte auch nur vier Minuten. Die Simulation der urbanen Gegend war etwas größer und umfasste ein 3×3 - Gitternetz aus Straßen mit einer Länge von jeweils drei Kilometern. Die Geschwindigkeit der Fahrzeuge betrugt dabei 50 $^{\mathrm{km}}/_{\mathrm{h}}$, die Simulationsdauer sechs Minuten. Während der Simulation hatte jeder Teilnehmer eine Wahrscheinlichkeit von 10 %, jede 300 ms das Pseudonym zu wechseln. Im Mittel wechselte also ein Teilnehmer sein Pseudonym nach jeweils drei Sekunden.

Kommt es zu einem Pseudonymwechsel, was bei dieser Simulation sehr häufig der

Fall sein dürfte, so umfasst die entstehende *Random Encryption Period* einen Radius von 1000 m. Das ist die von den Autoren zugrunde gelegte Sendereichweite der Teilnehmer, die gängigen Annahmen zufolge die maximale Sendereichweite für *VANET*s sein dürfte. Eine *REP* umfasst bei einer Kantenlänge von drei Kilometern also mindestens ein Neuntel bis ein Drittel des Simulationsgebiets (urbanes bzw. Highway-Szenario).

Die *REP* dauerte in den Simulationsergebnissen im Mittel stets unter einer Sekunde. Das bedeutet, dass ein Fahrzeug selbst im besten Fall während einer *REP* nur eine Strecke von 22 m zurücklegen kann (bei einer Geschwindigkeit von 80 km/h). Auf dieser kurzen Strecke kann bei es diesem Tempo zu keiner nennenswerten Vermischung unter Fahrzeugen oder Abweichung von der erwarteten Position eines beobachteten Fahrzeugs kommen. Wasef und Shen beziehen jedoch zur Bildung einer Anonymitätsgruppe alle Fahrzeuge innerhalb des Bereichs einer *Random Encryption Period* mit ein. Dies ist eine nach den Überlegungen aus Abschnitt 4.1.3 unzulässige Herangehensweise, da Fahrzeuge, die sich bis zu einen Kilometer voneinander entfernt befinden, von einem Angreifer nach einer Funkstille von einer Sekunde stets unterscheidbar sind.

Insgesamt lässt sich aus dieser Betrachtung folgern, dass mit den *Random Encryption Periods* keine guten neuen Ansätze vorgestellt wurden und zudem die Evaluation durch die Autoren sehr fragwürdige Ergebnisse liefert.

4.5. Fazit

In der Literatur existiert eine Vielzahl von Konzepten zur Durchführung eines unbeobachtbaren Pseudonymwechsels, welche sich nach ihrer Herangehensweise klassifizieren lassen. So existieren neben Ansätzen, die einen klar definierten Ort als Ausgangspunkt für eine koordinierte Funkstille nutzen, weitere, die einen bestimmten Zeitpunkt als Initiator wählen und jene, die es den Teilnehmern überlassen, Entscheidungen über Pseudonymwechsel zu treffen. Die Übergänge der beiden letzten Klassen sind relativ fließend, was eine genaue Einteilung erschwert: Je freier und individueller die Zeitintervalle, nach denen ein Pseudonym gewechselt werden muss, gewählt werden können, desto benutzerzentrierter ist ein Ansatz. In die letzte Klasse fallen noch weitere Konzepte, bei denen ein Pseudonymwechsel durch die eigene Geschwindigkeit oder dadurch, dass die Unbeobachtbarkeit auf oder unter einen Schwellwert gefallen ist, ausgelöst wird.

Grundsätzlich lässt sich keine generelle Aussage über die Überlegenheit eines Prinzips über ein anderes treffen; dazu bedürfte es einer Betrachtung im Rahmen der konkreten Umsetzung. Bei Ansätzen mit einem festen Ortsbezug lässt sich fehlende

Flexibilität – Teilnehmer, die diese definierten Orte nicht durchfahren, werden nicht geschützt – bemängeln. Darüberhinaus bedarf es komplexer Analysen zur Festlegung geeigneter Orte. Als vorteilhaft erweist sich jedoch, dass gezielt Bereiche mit günstigen Bedingungen für eine Vermischung von Teilnehmern – und damit eine möglichst hohe Unsicherheit für Angreifer – gewählt werden können. Weiterhin ist es möglich, die Funkstillen an Orten einzurichten, die trotz des notwendigen Wegfalls der Fahrassistenzsysteme die Reisenden nicht über Gebühr gefährden. Zudem können Fahrer durch die statischen Eigenschaften der Bereiche sehr gut auf diesen Wegfall vorbereitet werden.

Zeitlich abhängige Funkstillen bieten den Nachteil, dass sie, je nach Implementierung, unter Umständen überall und an jedem Ort auftreten können. Dadurch können Fahrer stark gefährdet werden, wenn sie sich gerade in Situationen befinden, in denen sie auf Fahrassistenzsysteme angewiesen sind bzw. sich auf diese verlassen. Pseudonymwechsel können nur erfolgreich sein, wenn es während einer Funkstille auch zu einer Vermischung von Teilnehmern kommt. Dies ist jedoch an vielen Orten kaum möglich, weil die Verkehrsdichte zu gering (beispielsweise ein Fahrzeug alleine auf einer Landstraße) oder zu hoch ist (z. B. bei Fahrzeugen, die im Stau stehen), oder Vermischung auf Grund der Gegebenheiten erschwert ist (z. B. ist es bei dichtem Verkehr auf einer Bundesstraße oft nicht möglich zu überholen und es existieren auch in längeren Bereichen keine Einmündungen). Trotzdem kann sich die Situation ergeben, dass Teilnehmer, die sich in für Pseudonymwechsel ungeeigneten Situationen befinden, trotzdem auf Grund der Protokolle eines Konzepts in eine Funkstille wechseln und auf Fahrassistenzsysteme verzichten müssen. Andererseits kann – im Gegensatz zu ortsfesten Bereichen für Funkstille und Pseudonymwechsel – sichergestellt werden, dass jeder Teilnehmer in regelmäßigen Abständen – unabhängig von der Reiseroute, die er nimmt – zumindest die Möglichkeit bekommt, sein Pseudonym unbeobachtbar zu wechseln. Weiterhin ist es nicht notwendig, Bereiche für eine Funkstille auszuwählen und festzulegen sowie mit Infrastruktur zu versehen.

Benutzerzentrierte Ansätze ermöglichen es Teilnehmern, wesentlich flexibler auf Situationen zu reagieren. So kann ein Pseudonymwechsel genau dann initiiert werden, wenn es sich entweder auf Grund der Straßen- und Verkehrssituation besonders anbietet oder dringend geboten ist, da ein Pseudonym bereits sehr lange verwendet wird oder die beobachtbare Strecke weit ausgedehnt ist. Zudem ist es leichter möglich, *QoS*-Standards für bestimmte Anwendungen durchzusetzen, da auf einen Pseudonymwechsel auch gegebenenfalls verzichtet werden kann, wenn dies eine bestimmte verwendete Anwendung in ihrer Funktionsweise einschränken würde. Nachteilig zeigt sich allerdings, dass es schwieriger sein dürfte, eine Kooperation der benachbarten Teilnehmer zu erlangen. Ist eine Funkstille zentral organisiert und gibt es einen Konsens darüber, so wird jeder Teilnehmer an der entsprechenden Funkstille teilnehmen. Bei benutzerzentrierten Ansätzen ist jedoch ein Fahrzeug darauf

angewiesen, dass zu einem selbst gewählten (nach eigenem Ermessen günstigen) Zeitpunkt auch andere Teilnehmer im näheren Umfeld bereit sind, einen Pseudonymwechsel mit Funkstille durchzuführen und dafür gegebenenfalls Nachteile in Kauf zu nehmen. Diese können das frühzeitige Verfallen von noch gültigen Pseudonymen oder der vorübergehende Ausfall von Diensten sein. Es besteht somit die Gefahr, dass die Anonymitätsgruppe bei spontan koordinierten Pseudonymwechseln tendenziell kleiner ausfällt. Weiterhin kann das Einhalten einer Funkstille andere Fahrer unerwartet – und möglicherweise ohne dass dies ihnen bewusst ist – ihrer Fahrassistenzsysteme berauben. Auch dies kann zu bedrohlichen Situationen führen.

Tabelle 4.1 fasst die beschriebenen und diskutierten Ansätze auf einen Blick zusammen und führt dabei eine grobe Bewertung durch. Berücksichtigt werden dabei die folgenden Aspekte:

- *Gewährleistung der Verkehrssicherheit*: Das eigentliche Ziel der Entwicklung eines Ad-hoc-Fahrzeugnetzes ist die Erhöhung der Verkehrssicherheit. Daher ist es unerlässlich, dass die Einschränkung der Assistenzsysteme, die der Erhöhung der Verkehrssicherheit dienen, möglichst gering ist. Erstaunlicherweise findet dieser wichtige Aspekt in vielen Untersuchungen zu der Effektivität von Techniken zum sicheren Pseudonymwechsel keine Berücksichtigung (z. B. [JBH11, FSH09]).
- *Schutz der Privatsphäre*: In diesem Kapitel wurden bereits ausführlich die Probleme und Folgen bei einer Verkettbarkeit von Nachrichten, insbesondere Beacons, dargelegt. Selbstverständlich müssen sich alle Ansätze, die die Privatsphäre schützen wollen, an ihrer Effektivität messen lassen.
- *Unabhängigkeit von anderen Teilnehmern*: Teilnehmer in Ad-hoc-Fahrzeugnetzen können nicht als altruistisch angenommen werden, und daher kann eine Kooperation ohne einen Nutzen für alle Beteiligten nicht vorausgesetzt werden. Ansätze, bei denen eine explizite Abhängigkeit von der Kooperation anderer Teilnehmer besteht, können daher möglicherweise nicht für einen Schutz der Privatsphäre garantieren. Diese Abhängigkeit kann vermieden werden, wenn zentral organisierte Regeln bestehen, so dass ein Abweichler Nachteile in Kauf nehmen muss.
- *Unabhängigkeit von Infrastruktur*: Der Idealfall eines Ad-hoc-Fahrzeugnetzes kommt gänzlich ohne Infrastruktur aus. Diese ist, auf Grund der hohen Kosten für ihre Installation, als ein Hemmnis zu betrachten. Daher wäre es wünschenswert, dass Ansätze nicht auf Infrastruktur angewiesen sind.
- *Restriktion der Bewegung auf Straßen berücksichtigt*: Fahrzeuge müssen Einschränkungen in ihrer Bewegungsfreiheit hinnehmen, die durch Straßen und Verkehrsregeln entstehen. Konzepte zum Schutz der Privatsphäre müssen die

daraus resultierende verringerte Unsicherheit eines Angreifers in Betracht ziehen.

- *Für Beacons ausgelegt*: Ein Angreifer wird stets seine Aufmerksamkeit auf den anfälligsten Angriffspunkt legen. Im Kontext der Verkettung von Pseudonymen ist dies das sehr häufige Senden von Beacons, da alle anderen Nachrichten wesentlich seltener übermittelt werden dürften. Eine Beobachtung von Teilnehmern auf Basis anderer Nachrichten ist somit deutlich schwieriger. Daher muss ein Konzept, das die Privatsphäre schützen soll, regelmäßig und kontinuierlich übertragene Nachrichten berücksichtigen.

Die Bewertungen in Tabelle 4.1 geben eine Abschätzung auf Basis der vorangegangenen Diskussion und analytischen Betrachtung der Ansätze wieder. Insbesondere gilt es, die tatsächliche Effektivität beim Schutz der Privatsphäre empirisch zu untersuchen. Dies erfolgt mit Hilfe einer Simulationsumgebung in Kapitel 5.

SLOW sticht als sehr vielversprechendes Konzept hervor. Allerdings besteht ein großes Manko im Bereich des Schutzes der Privatsphäre: Die Beendigung der Kommunikation findet gemäß Protokoll nur bei sehr geringen Geschwindigkeiten statt. Somit könnte eine Verfolgung über eine längere Reise mit höherer Geschwindigkeit (beispielsweise auf einer Autobahn) erfolgreich durchgeführt werden. Möglicherweise sollte zu *SLOW* eine Ergänzung gefunden werden, die sich insbesondere in Bereichen höherer Geschwindigkeiten eignet. Würde *SLOW* somit ausschließlich in urbanen Gebieten zum Einsatz kommen, würde sich der Schutz der Privatsphäre deutlich verbessern (Angabe des zweiten Wertes).

Einen guten Schutz der Privatsphäre scheinen Mix-Zonen und *Silent Periods* zu bieten. Das Konzept der *Silent Periods* – ebenso wie die Ergänzung *Silent Cascades* – krankt jedoch an den darin unberücksichtigten Besonderheiten in der Kommunikation in Ad-hoc-Fahrzeugnetzen sowie in den eingeschränkten Bewegungsmöglichkeiten der Teilnehmer. Somit rückt die Mix-Zone erneut in den Fokus der Betrachtung.

Die Notwendigkeit von Infrastruktur im Konzept der Mix-Zone ist nicht unbedingt wünschenswert, aber auch kein zentrales Hindernis. Es ist davon auszugehen, dass in regelmäßigen Abständen *RSU*s eine Kommunikation zwischen dem VANET und anderen Netzen ermöglichen sollen. Einige Sicherheitskonzepte für VANETs (vgl. u. a. [Plö09, RH05, PBH+08]) erfordern ebenfalls eine ausreichende Verbreitung von Infrastruktur. Die größte Schwäche der Mix-Zone ist der Wegfall von Systemen zur Erhöhung der Verkehrssicherheit in kritischen Bereichen. Ansätze wie die CMIX-Zone versuchen dies mit einer Aufrechterhaltung der Kommunikation zu kompensieren, was jedoch wieder zu einer deutlichen Reduktion des Schutzes vor der Verkettung von Pseudonymen führt.

	Gewährleistung der Verkehrssicherheit	Schutz der Privatsphäre	Unabhängigkeit von anderen Teilnehmern	Unabhängigkeit von Infrastruktur	Restriktionen durch Straßen berücksichtigt	Für Beacons ausgelegt
Mix-Zone	o	+	+	-	+	++
CMIX-Zone	++	-	+	-	+	++
Pseudonym-on-demand	++	--	+	--	o	++
Density Zones	++	--	+	o	+	++
Social Spots	o	-	+	+	+	++
Silent Periods	-	+	o	+	--	-
CARAVAN / AMOEBA	+	-	--	-	o	--
Silent Cascade	-	o	o	+	-	o
Swing	-	o	--	++	o	++
Swap	-	o	-	++	o	++
SLOW	+	o / +	+	+	+	++
Random Encryption Periods	++	--	--	+	++	++

Tabelle 4.1.: Übersicht über Ansätze zum sicheren Pseudonymwechsel

4.6. Die ProMix-Zone

In diesem Abschnitt soll nun, basierend auf den vorangegangenen Überlegungen zu Mix-Zonen, ein Konzept vorgestellt werden, das die Möglichkeit bietet, einen sicheren Pseudonymwechsel durchzuführen, während Fahrassistenzsysteme weiterhin zur Verfügung stehen. Dabei wird ein Regler angeboten, mit der eine feine Abstimmung zwischen den konkurrierenden Zielen der Verkehrssicherheit und dem Schutz der Privatsphäre ermöglicht wird. Der bereits als *ProMix-Zone* in [SFF11] veröffentlichte Ansatz basiert auf der Verwendung eines Proxys zur Selektion und Weiterleitung von Beacons. *ProMix-Zone* kürzt somit *„proxy-based mix zone“* ab.

4.6.1. Angreifermodell und Vorüberlegungen

Das in diesem Konzept zu Grunde gelegte Angreifermodell basiert im Wesentlichen auf den vorangegangenen Betrachtungen. Es wird ein globaler passiver Angreifer berücksichtigt, der sämtliche unverschlüsselte Kommunikationsverbindungen überwachen und zudem die Rolle eines Teilnehmers des *VANET*s annehmen kann. Letzteres ermöglicht ihm – durch Teilnahme an Gruppenverschlüsselung – auch so geschützte Nachrichten zur Kenntnis zu nehmen. Da er nicht in der Lage ist, kryptographische Systeme grundsätzlich zu brechen oder in realistischen Zeitdimensionen mit Hilfe vollständiger Enumeration den Klartext von effektiv verschlüsselten Nachrichten zu bestimmen, bleiben ihm diese verschlossen, sofern er nicht über einen geeigneten Schlüssel zur Entschlüsselung verfügt. Weiterhin besitzt der Angreifer keine Möglichkeit, die Position von einzelnen Teilnehmern mit Hilfe von Triangulation oder anderen Angriffen auf der physikalischen Schicht zu bestimmen. Diese Annahme wird getroffen, da derartige Angriffe zwar einerseits schwierig und aufwändig bei sich schnell bewegenden Fahrzeugen sind, andererseits der einzige effektive Schutz davor in völliger Funkstille besteht. Der beschriebene Angreifer zeigt sich insgesamt als sehr – möglicherweise unrealistisch – stark. Gelingt es jedoch, ein System vor diesem Angreifer zu schützen, so kann getrost davon ausgegangen werden, dass das System auch vor schwächeren Angreifern mindestens das gleiche Schutzniveau bietet.

Betreiber der Infrastruktur der ProMix-Zone sind auch nicht grundsätzlich als vertrauenswürdig einzustufen. Sie folgen jedoch dem Protokoll, da ein offensichtliches Fehlverhalten die Funktionalität der Zone beeinträchtigen und zu gefährlichen Verkehrssituationen führen kann. Damit würden die Infrastrukturbetreiber jedoch entlarvt und könnten sanktioniert werden. Allerdings haben sie die Möglichkeit, alle zur Verfügung stehenden Nachrichten zur Kenntnis zu nehmen und damit möglicherweise Pseudonymwechsel nachzuverfolgen. Aus diesem Grund wird an dieser

Stelle eine gängige Annahme aus dem Bereich der Kommunikationsmixe herangezogen: Betreiber der einzelnen Mix-Zonen kooperieren nicht untereinander; zumindest ein Betreiber einer Mix-Zone auf der Route eines Fahrzeugs ist vertrauenswürdig.

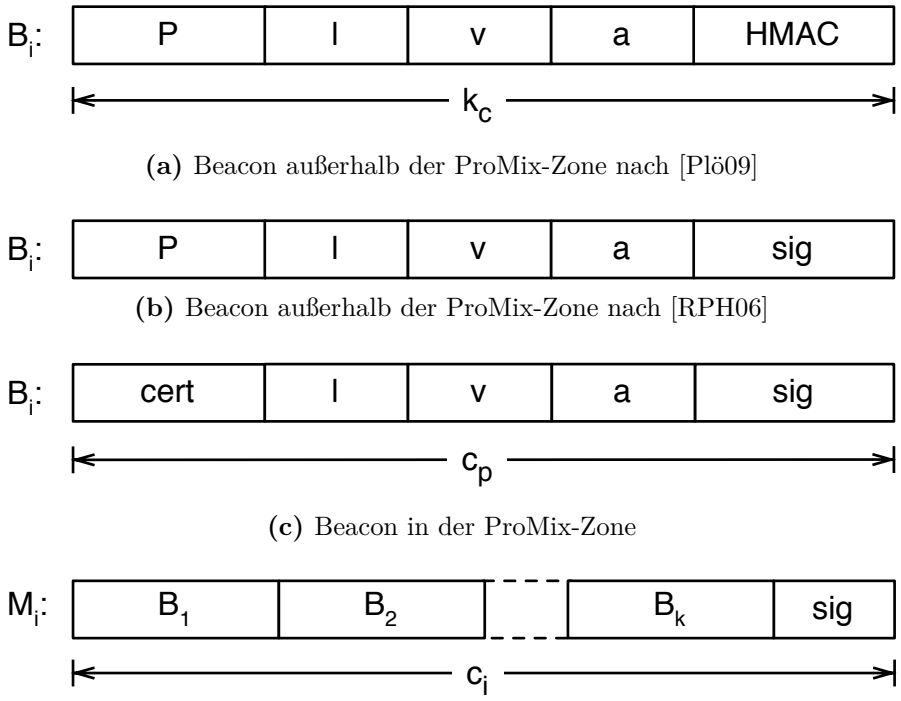
Das Konzept soll sich an den oben dargestellten Aspekten messen lassen, wobei ein besonderes Augenmerk auf der Aufrechterhaltung der Fahrassistenzsysteme zur Erhöhung der Verkehrssicherheit sowie dem effektiven Schutz der Privatsphäre der Teilnehmer durch die Möglichkeit eines sicheren, unbeobachtbaren Pseudonymwechsels liegt. Weiterhin ist zu berücksichtigen, dass die Verteilung von Informationen über die Verkehrslage mit Hilfe von Beacons als sehr zeitkritisch einzuschätzen ist. Sämtliche Protokolle, die erneutes Senden, Umkodieren oder Ver- und Entschlüsseln von Nachrichten beinhalten, müssen auf ihre Performance untersucht werden. Weiterhin ist es wünschenswert, den Kommunikationsoverhead auf der in Bezug auf ihre Bandbreite und Übertragungskapazität beschränkten Luftschnittstelle möglichst gering zu halten.

4.6.2. Architektur und Konzept

Es wird angenommen, dass jeder Teilnehmer i über eine Reihe von asymmetrisches Schlüsselpaaren, jeweils bestehend aus einem öffentlichen Schlüssel c_i und einem privaten Schlüssel d_i, sowie einem zugehörigen Zertifikat $cert_i$, für die Verwendung in den ProMix-Zonen verfügt. Ein Schlüsselpaar und das entsprechende Zertifikat werden nur für das Durchfahren einer einzigen ProMix-Zone verwendet. Sollte eine Wiederverwendung stattfinden, kann möglicherweise anhand des Zertifikates oder des öffentlichen Schlüssels eine Verkettung stattfinden. Sind alle Schlüsselpaare aufgebraucht, so müssen neue erzeugt und die zugehörigen Zertifikate bei einer Zertifizierungsstelle (*Certificate Authority*, *CA*) beantragt werden.

Diese Zertifikate werden von einer zentralen *CA* herausgegeben und können durch den Proxy in der ProMix-Zone verifiziert werden. Mit Hilfe des zugehörigen Schlüsselpaares (c_i, d_i) werden in der ProMix-Zone Beacons signiert, unabhängig davon, ob die Kommunikation außerhalb der Zone generell verschlüsselt und symmetrisch abgesichert stattfindet (vgl. u. a. [Plö09]) oder ob dort ebenfalls asymmetrische Schlüssel zur Integritätssicherung eingesetzt werden (vgl. u. a. [RPH06]). Weiterhin können an den Teilnehmer i gesendete Nachrichten mit c_i für ihn verschlüsselt werden. Der private Schlüssel d_i ist in einem *TPM* (vgl. Abschnitt 2.3.2) sicher abgelegt.

Jeder Teilnehmer i sendet außerhalb der ProMix-Zone alle 200 ms ein Beacon $B_{i,t}$, das neben dem aktuellen Pseudonym $P_{i,t}$ die Position $l_{i,t}$, Geschwindigkeit und Richtung $\overrightarrow{v_{i,t}}$ und Beschleunigung $\overrightarrow{a_{i,t}}$ enthält. Diese Beacons werden als Singlehop-Nachrichten ausgesendet und nicht weiterverbreitet. Fahrzeuge in der Nähe des Senders empfangen diese Beacons und werten sie für ihre Fahrassistenzsysteme aus.



(a) Beacon außerhalb der ProMix-Zone nach [Plö09]

(b) Beacon außerhalb der ProMix-Zone nach [RPH06]

(c) Beacon in der ProMix-Zone

(d) Gesammelte Beacons für einen Teilnehmer

Abbildung 4.16.: Aufbau der Nachrichten

Die Struktur der Beacons hängt davon ab, ob die ProMix-Zone in die Infrastruktur nach Plößl (vgl. [Plö09], siehe Abbildung 4.16a) oder nach Hubaux et al. (vgl. u. a. [RPH06], siehe Abbildung 4.16b) eingebracht werden soll.

Zur Lösung der in Abschnitt 4.1.3 beschriebenen Probleme werden an Kreuzungen, Autobahnkreuzen und möglicherweise großen Zu- und Abfahrten von Schnellstraßen ProMix-Zonen eingerichtet. In Kombination mit dem Konzept *SLOW* (vgl. Abschnitt 4.4.2) kann sich die Platzierung auf Abschnitte außerhalb geschlossener Ortschaften beschränken. Die genaue Platzierung und Verteilung von Mix-Zonen jeglicher Variante hängt unter anderem sehr stark von den konkreten Verkehrsflüssen ab, die empirisch bestimmt werden müssen. Eine Untersuchung der Einflussfaktoren findet in Abschnitt 5.4 statt. Ziel sollte es sein, dass Fahrzeuge in regelmäßigen Abständen durch eine ProMix-Zone die Möglichkeit erhalten, ihr Pseudonym sicher zu wechseln um eine Nachverfolgbarkeit ihrer Route zu vermeiden.

Jede dieser ProMix-Zonen hat einen anderen Betreiber und besitzt eine Reihe vernetzter Infrastrukturkomponenten mit Recheneinheiten (möglicherweise auch Kryptoprozessoren) und die gesamte Zone abdeckende Sende-Empfangsgeräte, die im Fol-

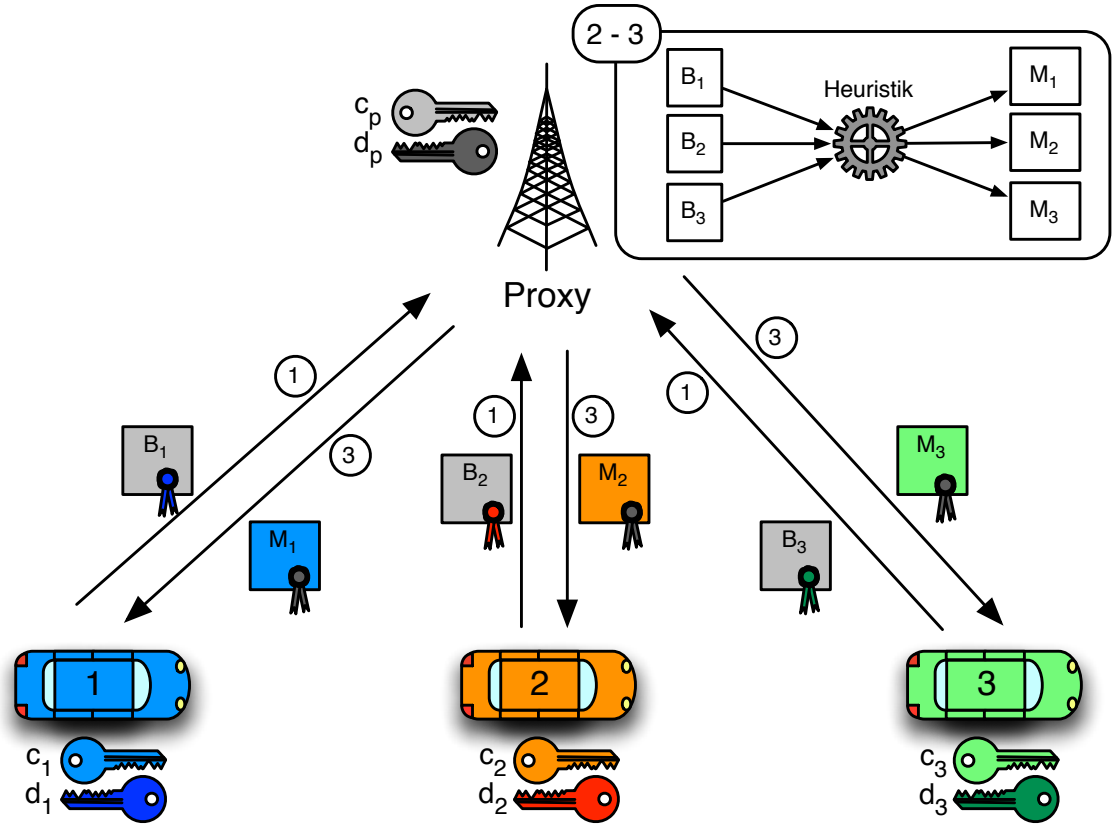


Abbildung 4.17.: Die drei Schritte der ProMix-Zone

genden unter den *Proxy* subsumiert werden. Der *Proxy* besitzt ein asymmetrisches Schlüsselpaar $c_p$und d_p mit einem zugehörigen, von der *CA* ausgestellten, Zertifikat $cert_p$. Dieses Zertifikat wird an den Grenzen der ProMix-Zone ausgesendet und kann von den Teilnehmern auf Validität überprüft werden. Jedes Fahrzeug in der Zone verfügt somit über den beglaubigten öffentlichen Schlüssel des *Proxys*.

Das in der ProMix-Zone zum Einsatz kommende Protokoll unterteilt sich in drei Schritte (vgl. Abbildung 4.17):

Schritt 1: Jeder Teilnehmer i in der ProMix-Zone sendet keine Beacons in der zuvor beschriebenen Form, sondern modifiziert diese wie folgt: Das Pseudonym $P_{i,t}$ wird durch das Zertifikat $cert_i$ des in der Zone eingesetzten Schlüsselpaares ersetzt und die Nachricht wird mit d_i signiert (vgl. Abbildung 4.16c). Das Beacon $B_{i,t}$ wird anschließend mit dem öffentlichen Schlüssel c_p des *Proxys* verschlüsselt und gesendet. Die Inhalte sämtlicher gesendeter Beacons können somit nur von dem Proxy zur Kenntnis genommen und ausgewertet werden. Dieser entschlüsselt und sammelt alle

Beacons $B_{i,t}$, $i \in [1 \dots n]$ der aktuell in der ProMix-Zone vorhandenen n Teilnehmer. Nach Ablauf eines Sendeintervalls von 200 ms hat der Proxy n Beacons empfangen – eines von jedem Teilnehmer.

Schritt 2: Zu diesem Zeitpunkt besitzt der *Proxy* vollständige Informationen über alle in der Zone vorhandenen Teilnehmer und die komplette daraus resultierende Verkehrssituation. Mit diesen Erkenntnissen kann er nun entscheiden, welcher Teilnehmer i, $i \in [1 \dots n]$ die Beacons, also die Informationen über die Position, Geschwindigkeit und Beschleunigung, welcher anderen Teilnehmer j, $j \in [1 \dots k], k < n$ benötigt. Der Ablauf und die Kriterien, nach welchen Beacons als relevant eingestuft werden, werden in Abschnitt 4.6.3 ausführlich diskutiert.

Schritt 3: Nachdem der *Proxy* die Auswahl an Beacons für einen Teilnehmer i zusammengestellt hat, entfernt er die Zertifikate $cert_j$ und die Signaturen aus den einzelnen Beacons und bündelt sie in einer neuen Nachricht $M_{i,t}$. Diese Nachricht wird nun mit d_p signiert und für den Empfänger mit dessen öffentlichen Schlüssel c_i verschlüsselt und anschließend an ihn übermittelt. Der Aufbau der Nachricht $M_{i,t}$ ist in Abbildung 4.16d dargestellt.

Jedes Fahrzeug erhält somit eine Nachricht $M_{i,t}$, die mit dem eigenen privaten Schlüssel d_i entschlüsselt werden kann und die die Menge an relevanten Beacons $B_{j,t}$ enthält. Da alle für die Verkehrssicherheit relevanten Nachrichten (anonymisiert) empfangen wurden, sind sämtliche Fahrassistenzsysteme selbst innerhalb der ProMix-Zone voll funktionsfähig.

Nachdem ein Teilnehmer die Zone verlassen hat, nimmt er das Senden der regulären Beacons wieder auf, nutzt nun aber ein anderes Pseudonym $P_{i,t'}$.

4.6.3. Auswahl relevanter Beacons

Der interessanteste und entscheidende Aspekt der ProMix-Zone ist die Auswahl an relevanten Beacons für einen Teilnehmer. Werden zu wenige Beacons zur Verfügung gestellt, kann die Funktionalität der Fahrassistenzsysteme nicht gewährleistet werden. Werden zu viele Nachrichten weitergeleitet, ist es für einen Angreifer einfacher, einen Teilnehmer zu verfolgen und die Privatsphäre wird bedroht. In den Grenzfällen verhält sich somit die ProMix-Zone entweder wie eine herkömmliche Mix-Zone ($k = 0$) oder wie eine CMIX-Zone, in der alle Teilnehmer jeweils die Beacons aller anderen erhalten ($k = n - 1$). Wichtig ist die Erkenntnis, dass die Route eines Fahrzeugs von denjenigen Teilnehmern nachvollzogen werden kann, die möglichst lückenlos dessen ausgesendete Beacons erhalten. Gelingt eine Verfolgung durch die

gesamte ProMix-Zone, so war der Pseudonymwechsel (gegenüber dem Verfolger) erfolglos und die Pseudonyme können verkettet werden.

Welchen Umfang k die Menge an einem Teilnehmer zur Verfügung gestellten Beacons annehmen muss, hängt sehr stark von der konkreten Implementierung der Fahrassistenzsysteme ab, da diese ja auf die übermittelten Daten zurückgreifen. Weiterhin ist ein Einflussfaktor, in welcher Verkehrssituation sich ein Teilnehmer bewegt.

Dennoch können Beacons der folgenden Teilnehmer unabhängig von Implementierungen stets als irrelevant betrachtet werden:

- Fahrzeuge, deren Wege sich nicht kreuzen und die sich nicht folgen; dies kann Fahrzeuge umfassen, die einander auf einer Schnellstraße oder Autobahn mit Mittelbebauung entgegenkommen oder von denen sich eines in einer Unterführung unter oder auf einer Brücke über der Fahrbahn des anderen befindet,
- Fahrzeuge, die sich entgegenkommen, aber einander bereits passiert haben oder
- Fahrzeuge, die in großem Abstand voneinander fahren und / oder eine größere Anzahl an anderen Fahrzeugen zwischen sich haben.

Einige andere Fahrzeuge dürften unabhängig von Implementierungen stets relevant sein:

- Fahrzeuge, deren Wege sich kreuzen,
- Fahrzeuge, die sich entgegenkommen und die noch aufeinander zufahren,
- Fahrzeuge, die direkt hintereinander herfahren sowie
- Fahrzeuge, die ein auffälliges und möglicherweise gefährliches Verhalten an den Tag legen, wie beispielsweise Falschfahrer.

In Abbildung 4.18 ist eine mögliche Situation aus Perspektive des schwarz eingefärbten Fahrzeugs dargestellt. Als Entscheidungsbasis für die vorgenommene Einteilung wurden die oben beschriebenen einfachen Regeln angewendet. In grün sind somit die relevanten, in rot die definitiv irrelevanten Fahrzeuge dargestellt. Eine gewisse Anzahl an Fahrzeugen (gelb dargestellt) kann nicht eindeutig zugeordnet werden, hier ist einerseits die konkrete Implementierung der Fahrassistenzsysteme entscheidend und andererseits empirische Untersuchungen über die Entstehung und den Verlauf gefährlicher Verkehrssituationen sowie über den Verkehrsfluss in einem konkreten Straßenabschnitt.

Je mehr andere Fahrzeuge die Beacons eines Teilnehmers erhalten, desto einfacher ist es für einen Angreifer unter diesen Fahrzeugen zu sein und eine Verfolgung durchführen zu können. Daher sollte die Anzahl an Fahrzeugen, die als relevant eingestuft werden, so gering gehalten werden, wie es die Verkehrssicherheit erlaubt.

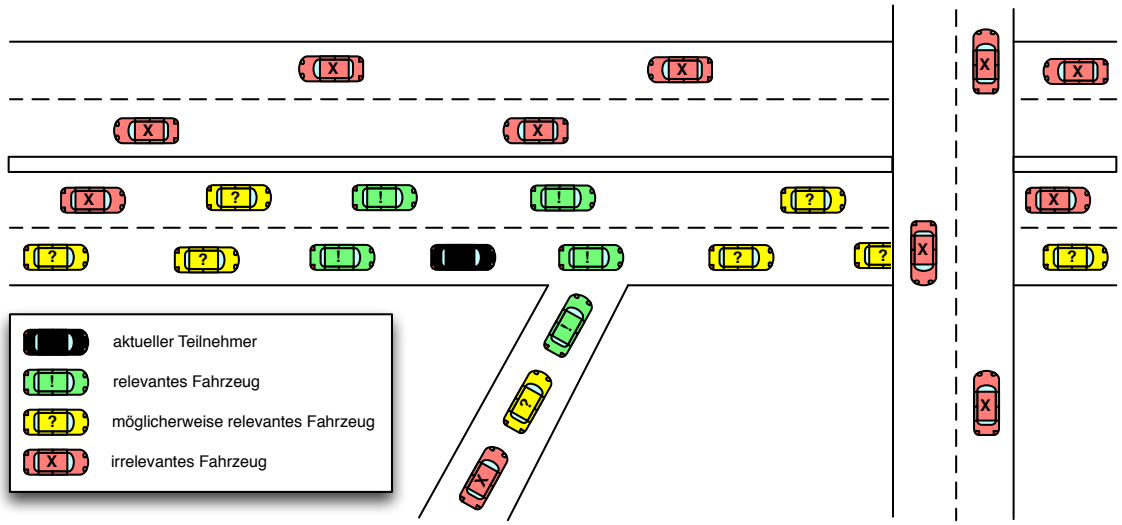


Abbildung 4.18.: Relevanz von Fahrzeugen aus Sicht eines Teilnehmers

4.6.4. Evaluation

4.6.4.1. Verkehrssicherheit

Das Konzept der ProMix-Zone ermöglicht es, eine Feinabstimmung zwischen den beiden konkurrierenden Zielen Verkehrssicherheit und Privatsphäre vorzunehmen. Dennoch sollte im Zweifel die Verkehrssicherheit Vorrang erhalten. Da auch in der ProMix-Zone Beacons gesendet und diese an die relevanten anderen Verkehrsteilnehmern weitergeleitet werden, stehen sämtliche Fahrassistenzsysteme zur Verfügung und die Verkehrssicherheit kann sichergestellt werden. Außerdem können natürlich in kritischen Situationen (beispielsweise bei Unfällen, glatter Fahrbahn, etc.) jederzeit Warnungen versandt werden, da in einer Situation, in der die Gesundheit von Verkehrsteilnehmern akut gefährdet ist, die Privatsphäre nur noch eine untergeordnete Rolle spielen darf.

4.6.4.2. Kommunikationssicherheit

Sowohl die in *Schritt 1* gesendeten Beacons als auch die Rückantworten des Proxys an die Teilnehmer aus *Schritt 3* sind asymmetrisch mit den öffentlichen Schlüsseln der jeweiligen Empfänger verschlüsselt und mit den privaten Schlüsseln der Sender digital signiert. Die Teilnehmer übermitteln bei jedem Beacon ein von einer vertrauenswürdigen Zertifizierungsstelle (*CA*) ausgestelles Zertifikat, mit dem der Proxy die Gültigkeit des öffentlichen Schlüssels nachvollziehen kann. Das – ebenfalls von der

CA ausgestellte – Zertifikat des Proxys wird im Randbereich der ProMix-Zone gebroadcastet und steht somit sämtlichen Teilnehmern zur Verifikation des öffentlichen Schlüssels des Proxys c_p zur Verfügung. Somit kann die Zurechenbarkeit der Nachrichten gewährleistet und ein Man-in-the-Middle-Angriff auf die Verschlüsselung effektiv unterbunden werden; die Vertraulichkeit der Nachrichten ist somit ebenfalls sichergestellt. Allerdings ist der Proxy in der aktuell beschriebenen Konfiguration anfällig für Denial-of-Service-Angriffe. Die Signatur der Nachrichten durch die Teilnehmer muss innerhalb der Verschlüsselung stattfinden, da ansonsten die Signatur Hinweise auf den Absender (bzw. sein aktuelles Pseudonym, den öffentlichen Schlüssel) geben kann. Somit ist es erforderlich, dass der Proxy alle erhaltenen Nachrichten probeweise entschlüsselt; Verwerfen von Nachrichten kann vorab nicht erfolgen.

4.6.4.3. Privatsphäre

Die ProMix-Zone bietet gegenüber Angreifern von außerhalb den gleichen Schutz wie eine herkömmliche Mix-Zone und damit von allen untersuchten Konzepten mit die beste Privatsphäre (vgl. Abschnitt 4.5). Eine Verkettung von Pseudonymen ist hier nur dann möglich, wenn die Verkehrsdichte zu gering ist, um ausreichend Unsicherheit für einen Beobachter zu erzeugen oder externes Wissen für Schnittmengenangriffe herangezogen wird. Könnte ein Angreifer beispielsweise die Route eines Teilnehmers antizipieren, da dieser eine bereits häufig zurückgelegte Strecke fährt, so reduziert sich die Anonymitätsgruppe einer Mix-Zone möglicherweise drastisch.

Ein vollständiger Schutz vor Angreifern, die in eine (Pro-)Mix-Zone einfahren, ist nur mit Funkstille möglich. Mit dem vorgestellten Konzept kann jedoch die Anzahl an Fahrzeugen, die die Nachrichten eines bestimmten Teilnehmers erhalten – und einen Pseudonymwechsel beobachten könnten – reduziert werden. Der tatsächlich erreichbare Schutz der Privatsphäre hängt sehr stark von der Anzahl an Fahrzeugen k ab, die die Beacons eines beobachteten Teilnehmers erhalten. Je kleiner k, desto schwieriger ist es für einen Angreifer, Teil dieser Gruppe an Fahrzeugen zu werden und die Nachrichten des Opfers erhalten. Bei sehr kleinem k muss ein Angreifer, um einen Teilnehmer durchgehend verfolgen zu können, diesem sehr nah folgen. Eine Verfolgung auf Sicht, bei der ein Fahrzeug einem anderen in sehr geringem Abstand folgt, ist jedoch jederzeit möglich und mit technischen Möglichkeiten niemals zu verhindern. Die vorgestellte ProMix-Zone kommt diesem möglichen Optimum sehr nahe. Allerdings gilt es zu berücksichtigen, dass eine zu starke Verringerung von k möglicherweise die Funktion von Fahrassistenzsystemen einschränkt. An dieser Stelle muss somit im Sinne der mehrseitigen Sicherheit (vgl. Abschnitt 2.1.1.2) ein Kompromiss eingegangen werden.

Der Betreiber der ProMix-Zone kann über sämtliche Daten des Proxys verfügen und erhält somit alle Informationen. Er ist also in der Lage, alle stattfindenden

Pseudonymwechsel zu beobachten und die alten und neuen Pseudonyme zu verketten. Allerdings wird dies nur in dem Moment zu einem großen Problem, in dem die Betreiber mehrerer Zonen zusammenarbeiten und ihre Informationen teilen. Dann wäre ein Teilnehmer exponiert und seine gesamte Route könnte nachvollzogen werden. In den Anforderungen wurde dieses Szenario jedoch – der gängigen Annahme über Kommunikationsmixe folgend – ausgeschlossen. In der Praxis ließe sich dies durch eine strikte Trennung der Betreiber realisieren, die ihrerseits sowohl finanzielle Einbußen (über Strafzahlungen) als auch einen Image-Schaden hinnehmen müssten, wenn sie gegen dieses Prinzip verstießen und Daten teilten. Als mögliche Kandidaten für Betreiber kämen neben staatlichen Datenschutzbehörden auch private Unternehmen und Vereinigungen in Frage, die sich für Datenschutz und Privatsphäre einsetzen.

4.6.4.4. Performance

Das Konzept der ProMix-Zone ermöglicht es nicht mehr, Beacons direkt von einem Teilnehmer zum anderen zu übertragen; stattdessen läuft alle Kommunikation über den Proxy ab. Weiterhin werden aufwändige asymmetrische Ver- und Entschlüsselungsoperationen sowie digitale Signatur- und Prüfvorgänge durchgeführt. Um die Realisierbarkeit der ProMix-Zone vor dem Hintergrund dieser Rechenkomplexität zu untersuchen, wurde eine Simulation dieser Vorgänge durchgeführt.

Die obere Grenze der tolerierbaren Verzögerung hängt von den Anforderungen an die Genauigkeit ab, in der die Position anderer Teilnehmer bekannt sein muss. Die Daten der Beacons B_i können nur dann als nützlich eingestuft werden, wenn mit ihnen die Verkehrssituation zu jedem Zeitpunkt hinreichend genau abgebildet werden kann. Geht man von einem Intervall von 600 ms aus, in dem Daten geliefert werden – was gleichzeitig dem maximalen Alter der Informationen entspricht –, so würde selbst bei einem sehr starken Bremsvorgang von $-10\,\mathrm{m/s^2}$ eine maximale Unsicherheit über den Aufenthaltsort (also die Abweichung in Fahrtrichtung) eines Fahrzeuges von weniger als zwei Metern bestehen[5]. Dies sollte unter Berücksichtigung der Ausdehnung der Fahrzeuge sowie des von Fahrern einzuhaltenden Sicherheitsabstands, völlig ausreichend sein. Die laterale Abweichung wird stets deutlich geringer sein, da eine Querbeschleunigung diesen Ausmaßes im Straßenverkehr kaum erreichbar ist.

Für die drei Schritte im Protokoll der ProMix-Zone stehen somit 600 ms zur Verfügung, die nicht überschritten werden dürfen. Ausgehend von der Annahme, dass alle Teilnehmer innerhalb von 200 ms jeweils ein Beacon verschicken, steht genau dieses Zeitfenster für die Bearbeitung von *Schritt 1* zur Verfügung.

[5] $\Delta s = a \cdot \frac{t^2}{2} = -10\,\frac{\mathrm{m}}{\mathrm{s}^2} \cdot \frac{0{,}6^2 s^2}{2} = -1{,}8\,\mathrm{m}$

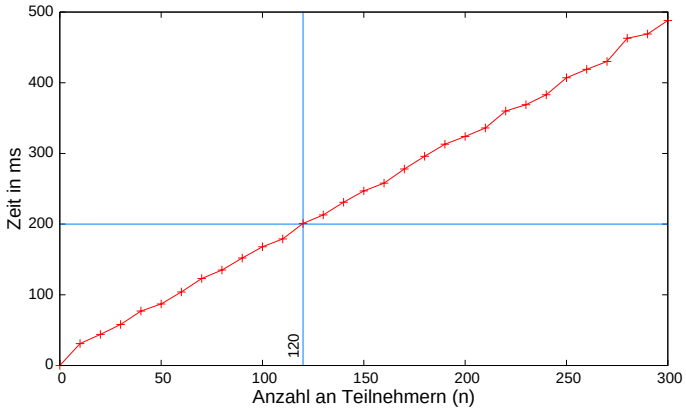


Abbildung 4.19.: Auswirkungen von n auf *Schritt 1*

Die Größe eines Beacons dürfte sich geschätzt auf 524 bytes belaufen. Dies ergibt sich aus einer Veranschlagung von 64 bit für die Position des Teilnehmers, 10 bit für seine Geschwindigkeit, 10 bit für die Beschleunigung, 9 bit für die Richtung[6] sowie 1024 bit für die digitale Signatur. Der Umfang erhöht sich weiterhin deutlich durch die Hinzunahme eines X.509-Zertifikats [CSF+08] mit einer Gesamtgröße von 3072 bit, das einen 1024 bit-RSA-Schlüssel und eine 1024 bit-RSA-Signatur enthält. Obwohl die Größe durch das Zertifikat den Umfang eines gewöhnlichen Beacons übersteigt, ist dies nicht kritisch, da diese nur im Umfeld einer ProMix-Zone verwendet werden. Zudem ließe sich die Größe durch den Einsatz von Kryptographie auf Basis elliptischer Kurven (*ECC*, vgl. Abschnitt 2.1.3.2) auf 212 bytes reduzieren, wenn eine Schlüssellänge von 192 bit zum Einsatz kommt.

Eine durchgeführte Simulation wurde in Java 1.6.0_20 für 64 Bit-Systeme implementiert und verwendet den *Bouncy Castle Cryptography Provider* in der Version 1.45[7]. Alle Tests wurden auf einem handelsüblichen Computer mit einer *Intel Core 2 Duo T9300*-CPU (2x 2.50 GHz) sowie mit 4 GB *DDR2 SDRAM* durchgeführt. Anstelle rein asymmetrischer Verschlüsselung wurde aus Performancegründen ein hybrider Ansatz gewählt: Die eigentliche Nachricht wurde stets symmetrisch verschlüsselt; lediglich der dazu genutzte symmetrische Sitzungsschlüssel wurde asymmetrisch chiffriert und dem Schlüsseltext beigefügt. Für die asymmetrische Kryptographie kam *RSA* mit einer Schlüssellänge von 1024 bit (vgl. Abschnitt 2.1.3.2), für die symmetrischen Operationen *AES* mit einer Schlüssellänge von 256 bit (vgl. Abschnitt 2.1.3.1) zum Einsatz.

[6] Alternativ ließe sich die Richtung auch, wie bereits in vorigen Untersuchungen beschrieben, in einen Geschwindigkeitsvektor einbeziehen. Für die Gesamtlänge dürfte dies nur eine geringe Auswirkung haben.

[7] http://www.bouncycastle.org, abgerufen am 20.03.2012

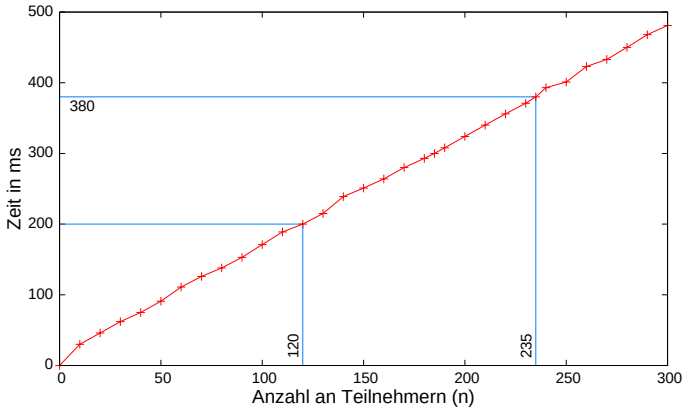


Abbildung 4.20.: Auswirkung von n auf *Schritt 3*

Abbildung 4.19 zeigt den Einfluss der Anzahl an Teilnehmern in einer ProMix-Zone (n) auf *Schritt 1*, der die Entschlüsselung der Beacons und die Validierung sämtlicher Signaturen erfordert. Es ist sofort ersichtlich, dass der Zeitverbrauch aus naheliegenden Gründen linear mit der Anzahl an Teilnehmern wächst. Berücksichtigt man nun die zuvor diskutierte maximale Zeit von 200 ms, die für diesen Schritt zur Verfügung steht, so konnte das Testsystem maximal 120 Teilnehmer gleichzeitig versorgen.

Die Auswirkungen der Teilnehmerzahl (n) auf *Schritt 3* des Protokolls, der aus Verschlüsselung und Signierung der Antwortnachrichten (M_i) besteht, sind in Abbildung 4.20 dargestellt. Auch hier wächst der Aufwand linear mit der Anzahl an Teilnehmern, da wiederum zwei rechenzeitintensive asymmetrische Kryptographieoperationen die Laufzeit dominieren. Kalkuliert man für *Schritt 2* – die Ermittlung der relevanten Beacons für jeden Nutzer – eine Zeitspanne von 20 ms, so stünden 380 ms für *Schritt 3* zur Verfügung. In dieser Zeit könnten die Nachrichten für 235 Teilnehmer von dem Testsystem erzeugt werden.

Die Simulationsergebnisse für *Schritt 1* limitieren jedoch in dem vorliegenden Fall die Anzahl an Teilnehmern auf 120. Die für sie bestimmten Nachrichten könnten in *Schritt 3* in ebenfalls 200 ms erzeugt werden, was somit weitere 200 ms für den zweiten Schritt sowie weitere Verzögerungen beispielsweise durch die Netzwerklatenz zur Verfügung stellen würde.

Eine weitere Betrachtung widmet sich der Länge der Beacons und ihrer Auswirkung auf die Durchführungsdauer der Operationen in *Schritt 1* (vgl. Abbildung 4.21). Die Laufzeit für Beacons kleiner oder gleich 12 kbyte ist nahezu konstant, da die Kosten der symmetrischen Entschlüsselungsoperationen, die Teil der hybriden Kryptographie sind, der zusätzlichen Daten geringer als die Messungenauigkeit in Bezug auf die aufwändigen asymmetrischen Operationen sind. Zwischen 12 und 200 kbyte steigt die Ausführungszeit leicht an, da die Laufzeit der symmetrischen Operationen nun

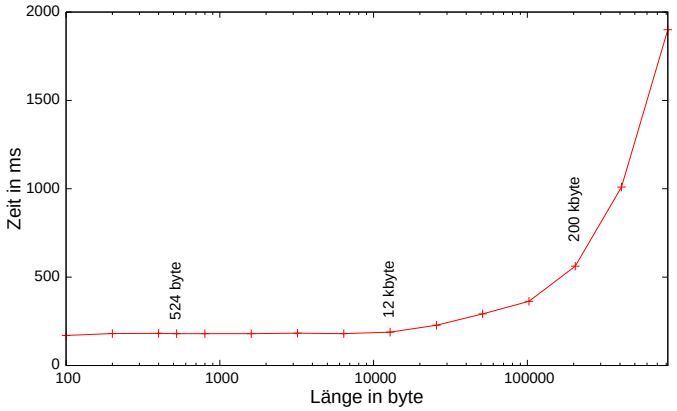


Abbildung 4.21.: Auswirkung der Größe der Beacons auf *Schritt 1*

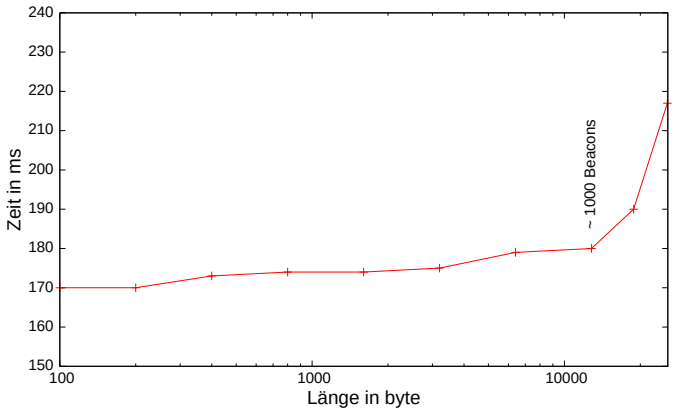


Abbildung 4.22.: Auswirkung von k auf *Schritt 3*

graduell an Bedeutung gewinnt. Jenseits der 200 kbyte skaliert die Ausführungsdauer annähernd linear mit der Größe der Beacons, da Kosten der symmetrischen Kryptographie nun den konstanten Aufwand der asymmetrischen Operationen übersteigen. Diese Bereiche spielen jedoch bei einer veranschlagten Beacongröße von 524 byte keine Rolle.

Abbildung 4.22 zeigt die Auswirkungen der durchschnittlichen Anzahl an Beacons (k) anderer Fahrzeuge, die an einen Teilnehmer verschickt werden, auf die Kosten der Kryptographieoperationen in *Schritt 3*. Erneut ergibt sich eine nahezu konstante Laufzeit für Gesamtlängen der Nachrichten (M_i) von bis zu 12 kbyte. Dies bedeutet, dass bei einer angenommenen Größe von 93 bit pro Beacon rund 1000 Beacons[8] ohne messbaren Einfluss auf die Laufzeit an jeden Teilnehmer verteilt werden könnten, da das umfängliche Zertifikat und die Signatur ja durch den Proxy entfernt werden.

[8] $\frac{12000\,\text{byte}}{93\,{}^{bit}/_{\text{Beacon}}} \approx 1032\,\text{Beacons}$

Berücksichtigt man die durch *Schritt 1* auf 120 Teilnehmer begrenzte Anzahl, die zugleich die obere Grenze für k ist, so hat die durchschnittliche Anzahl an Beacons, die an jeden Teilnehmer gesendet werden, in dem relevanten Bereich keinen messbaren Einfluss auf die Laufzeit.

Insgesamt zeigt die Simulation somit, dass die Laufzeit der durch den Proxy in einer ProMix-Zone durchzuführenden Schritte linear von der Anzahl an Teilnehmern n in der Zone sowie natürlich von der Sendehäufigkeit der Beacons abhängt. Die Größe der Beacons und die durchschnittliche Anzahl an relevanten Beacons pro Teilnehmer (k) hat in den zu erwartenden Bereichen lediglich einen geringen konstanten Einfluss und kann somit als in der Praxis irrelevant eingestuft werden.

Die Größe der Nachrichten wird durch die vorgestellten Mechanismen nur gering erhöht und der Overhead auf der Luftschnittstelle sollte daher nicht zu Übertragungsengpässen führen. Mit dem Einsatz von *ECC* könnte die Nachrichtenlänge jedoch noch einmal deutlich reduziert werden.

Obwohl *ECC* auch einen Performancevorteil gegenüber anderen asymmetrischen Verfahren verspricht (vgl. Abschnitt 2.1.3.3), stand zum Zeitpunkt der Simulation keine performante Java-Implementierung zur Verfügung. Insbesondere die getestete *Bouncy-Castle*-Version der *ECC* schnitt deutlich schlechter als die in den vorliegenden Simulationen genutzte *RSA*-Implementierung ab. Es kann also davon ausgegangen werden, dass mit geeigneten Implementierungen noch einmal deutliche Laufzeiteinsparungen erzielt werden können.

Weiterhin wurde die Simulation auf handelsüblicher PC-Hardware vor dem Hintergrund eines laufenden Betriebssystems (*MacOS X 10.6*) durchgeführt. Mit speziellen Kryptoprozessoren oder leistungsfähigeren Geräten ist nach den bereits sehr optimistisch stimmenden Simulationsergebnissen in der Praxis kein Performance-Engpass zu erwarten. Weiterhin skaliert die Laufzeit mit der Anzahl an Teilnehmern linear und die einzelnen Operationen sind voneinander unabhängig. Insofern ist eine Skalierung des Proxys sowie Parallelisierung einfach möglich.

4.6.5. Fazit

Mit der ProMix-Zone wurde ein Konzept vorgestellt, mit dem ein geschützter Pseudonymwechsel möglich ist, ohne dass auf das Senden von Beacons verzichtet werden muss, was zu einem Ausfall sämtlicher Fahrassistenzsysteme führen würde. Dies würde insbesondere in Bereichen mit einer hohen Vermischung von Fahrzeugen – und somit einer hohen Unsicherheit für einen beobachtenden Angreifer – möglicherweise zu einer Gefährdung der Verkehrsteilnehmer führen. Die mit der ProMix-Zone erreichbare Privatsphäre ist durch ihre Konzeption dennoch nur wenig geringer als bei

einer herkömmlichen Mix-Zone. Weiterhin bietet sie eine Möglichkeit, den Trade-Off zwischen Privatsphäre und Verkehrssicherheit für jede Zone selbst zu regeln, indem die Heuristik zur Auswahl der Nachrichten für andere Teilnehmer angepasst wird. So kann an besonders gefährlichen Verkehrsknotenpunkten die den Teilnehmern zur Verfügung gestellte Information umfassender sein als an Orten mit geringem Unfallrisiko. Die durchgeführten Simulationen zeigen, dass die ProMix-Zone selbst mit herkömmlicher PC-Hardware zu realisieren ist und somit keine herausragenden Kosten erzeugt.

Kapitel 5

Simulation von Ad-hoc-Fahrzeugnetzen

In diesem Kapitel werden zuerst der selbst entwickelte *VANET-Simulator* in seinen unterschiedlichen Bestandteilen im Detail beleuchtet und seine Designkriterien sowie der wissenschaftliche Unterbau dargestellt. In einem zweiten Teil wird der Simulator genutzt, um drei der in Kapitel 4 analytisch betrachteten Konzepte zum sicheren Pseudonymwechsel – Mix-Zone, Silent Period und SLOW –, in einer großen Reihe an Simulationen zu untersuchen. Dabei werden insbesondere die Einflussgrößen auf die Effektivität der Wirkungsweise beleuchtet und anhand verschiedener Szenarien evaluiert.

5.1. Motivation und Problemstellung

Ad-hoc-Fahrzeugnetze sind nach wie vor eine Zukunftstechnologie. Neben existierenden Prototypen der Automobilhersteller[1] gab es bereits eine großangelegte Studie von mit Funknetzkomponenten ausgestatteten Fahrzeugen im Großraum Frankfurt am Main unter der Bezeichnung SimTD[2]. Dieses Testfeld zur Untersuchung von *Car-to-X*-Kommunikation wird durch das *Car-2-Car-Consortium*[3] betreut, das sich aus vielen Partnern in der Industrie sowie einigen Hochschulen zusammensetzt. Allerdings wird auch in dieser Studie überwiegend die Grundfunktionalität von Ad-hoc-Fahrzeugnetzen, wie beispielsweise einfacher Nachrichtenaustausch von Infor-

[1] Ford demonstriert in einem Youtube-Video die Funktionsweise eines Prototyps: http://www.youtube.com/watch?v=orrd2qyY6lo, abgerufen am 15.03.2012.
[2] http://www.simtd.org, abgerufen am 15.03.2012
[3] http://www.car-to-car.org, abgerufen am 15.03.2012

mationen über Verkehrssituationen oder Warnungen, untersucht und evaluiert. Es existiert momentan keine Möglichkeit, flexibel und kostengünstig verschiedene theoretisch entwickelte und analytisch untersuchte weiterführende Konzepte zu erproben.

Die Untersuchungen von Techniken zur Realisierung eines sicheren Pseudonymwechsels in Kapitel 4 zeigen jedoch, dass eine rein analytische Betrachtungsweise sehr stark von dem zugrunde liegenden Modell beeinflusst wird. Dies beeinträchtigt insbesondere die Vergleichbarkeit der Konzepte, da in der Regel jeder Autor eine andere Vorstellung eines Ad-hoc-Fahrzeugnetzes hat. Besitzt man jedoch die Möglichkeit einer möglichst realitätsnahen Simulation von Fahrzeugbewegungen und Verkehrssituationen, so können unterschiedliche Ansätze in einem vergleichbaren Szenario untersucht werden.

5.2. Vorhandene Simulationslösungen

Es gibt eine ganze Reihe von Simulationslösungen, die über die Jahre vorgestellt und entwickelt wurden. An erster Stelle ist hier der überwiegend durch das *Institut für Verkehrssystemtechnik*[4] des *Deutschen Zentrums für Luft- und Raumfahrt*[5] (*DLR*) entwickelte Simulator *SUMO*[6] – *Simulation of Urban Mobility* – zu nennen. Dieser in der aktuellen Version von Behrisch et al. in [BBEK11] vorgestellte Verkehrssimulator bietet eine Fülle an Funktionen, die nach wie vor erweitert und ergänzt werden. Er simuliert Mikroverkehr, wobei jedes Fahrzeug individuell berechnet wird und autonome Entscheidungen trifft (vgl. [Kra98]). Nachteilig zeigt sich allerdings, dass es zum Zeitpunkt der Untersuchung der Simulatoren (Sommer 2008) keinerlei integrierte Komponenten für die Simulation von Fahrzeugkommunikation gab.

Einer der ersten Ansätze, *SUMO* um eine Komponente zur Simulation von Netzwerkverkehr zu erweitern, wurde mit der von Piorkowski et al. in [PRL+08] vorgestellten Anwendung *TraNS*[7] geschaffen. Dabei handelt es sich um ein Java-Programm, das die von *SUMO* erzeugten Fahrzeugbewegungen nutzt und dem Netzwerksimulator *ns-2*[8] in einem geeigneten Format zur Verfügung stellt. Der mit Hilfe von *ns-2* generierte Output wiederum kann von *TraNS* interpretiert und in Form von geänderten Verkehrssituationen an *SUMO* zurückgemeldet werden (vgl. Abbildung 5.1). In *SUMO* wurde zu diesem Zweck eigens eine Schnittstelle mit der Bezeichnung *Traffic Control Interface* (*TraCI*) geschaffen (vgl. [WPR+08]).

[4] http://www.dlr.de/ts, abgerufen am 15.03.2012
[5] http://www.dlr.de, abgerufen am 15.03.2012
[6] http://sumo.sourceforge.net, abgerufen am 15.03.2012
[7] http://lca.epfl.ch/projects/trans, abgerufen am 15.03.2012
[8] http://www.isi.edu/nsnam/ns/, abgerufen am 15.03.2012

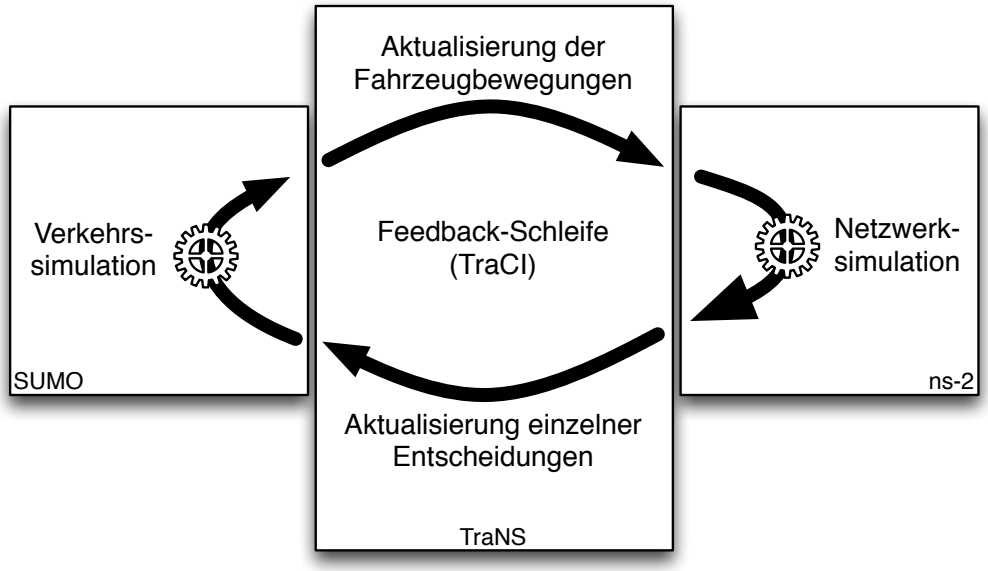


Abbildung 5.1.: Funktionsweise von TraNS (vgl. [PRL+08])

Einen vergleichbaren Ansatz nutzt das von Sommer et al. in [SYGD08, SD08, SGD11] vorgestellte Framework *Vehicles in Network Simulation*[9] (*Veins*). Hier wird *SUMO* mit dem Netzwerksimulator *OMNeT++*[10] (entstanden aus dem *Objective Modular Network Testbed*, *OMNeT*, siehe [Pon93, Var01]) gekoppelt. *Veins* führt, ebenfalls wie *TraNS*, abwechselnd einzelne Schritte der beiden Simulationskomponenten aus und spiegelt die Veränderungen an die jeweils andere Komponente. So kann auch hier eine Simulation eines kompletten *VANET*s durchgeführt werden.

VanetMobiSim[11], vorgestellt durch Härri et al. und Fiore et al. in [HFBF06, FHFB07], hat einen vergleichbaren Ansatz verfolgt und den Simulator *CANU Mobility Simulation Environment*[12] (*CanuMobiSim*), der Netzwerksimulatoren wie *ns-2* um Bewegungsmodelle ergänzt, erweitert und auf die Bedürfnisse der Simulation von Ad-hoc-Fahrzeugnetzen angepasst. Allerdings wurde dieses Projekt den Hinweisen auf seiner Webseite entsprechend seit 2007 nicht weiterverfolgt, auch wenn 2011 noch ein wiederaufgelegter Artikel über VanetMobiSim (vgl. [HFFB11]) erschienen ist.

Die beschriebenen Ansätze teilen auf Grund ihrer Ähnlichkeit auch ihre größte Schwäche: Die genutzten Simulatoren, wie beispielsweise *SUMO* oder *ns-2*, sind zwar

[9] http://veins.car2x.org/, abgerufen am 15.03.2012
[10] http://www.omnetpp.org/, abgerufen am 15.03.2012
[11] http://vanet.eurecom.fr/, abgerufen am 15.03.2012
[12] http://canu.informatik.uni-stuttgart.de/mobisim/, abgerufen am 15.03.2012

sehr ausgereift, aber auf die Bedürfnisse der *VANET*-Simulation nicht zugeschnitten. *ns-2* bietet einerseits sehr umfassende Konfigurationsmöglichkeiten, andererseits ist die Komplexität jedoch auch enorm hoch. Sämtliche Simulationen finden darüberhinaus auf sehr niedrigen Schichten des *OSI-Modells* statt und müssen entsprechend modelliert werden. *SUMO* auf der anderen Seite bietet zwar auf überschaubare Weise viele Optionen der Verkehrssimulation, allerdings ist die Einflussnahme mit einer Rückkopplung an Informationen sehr aufwändig und teilweise auch beschränkt. Insgesamt ist die Herausforderung, zu simulierende Aspekte kurzfristig zu implementieren und Untersuchungen durchzuführen, extrem hoch – verstärkt durch die Tatsache, drei unterschiedliche Anwendungen anpassen zu müssen.

Den einzigen Ansatz für eine Simulationsumgebung, die auf Ad-hoc-Fahrzeugnetze zugeschnitten ist, stellt *GrooveNet*[13], vorgestellt von Mangharam et al. in [MWS+05, MWR+06], dar. Jedoch scheint auch dieses Projekt seit 2006 nicht mehr weitergeführt zu werden und die Dokumentation ist äußerst dürftig. So ist neben einer fehlenden Beschreibung der tatsächlichen Funktionalität auch der gesamte Quellcode undokumentiert. Daher kann auch dieser Ansatz nicht für eine Weiterentwicklung genutzt werden.

Nach eingehender Betrachtung der vorhandenen Simulatoren musste festgestellt werden, dass kein vorhandenes Konzept vielversprechend erschien, in absehbarer Zeit eine Simulation vieler *VANET*-Funktionen und -Komponenten, insbesondere den Schutz der Privatsphäre betreffend, zu ermöglichen. So wurde 2008 entschieden, mit dem *VANET-Simulator* eine eigene Entwicklung eines Simulators für Ad-hoc-Fahrzeugnetze durchzuführen.

5.3. Eigene Entwicklung und Implementierung

Unter Beteiligung einer Reihe von studentischen Projekten und Qualifikationsarbeiten (herauszuheben sind dabei [Gru08, Tom09, Tom11]) wurde seit Sommer 2008 ein eigener, von Grund auf neu implementierter Simulator für Ad-hoc-Fahrzeugnetze entwickelt. In diesem Abschnitt sollen die wesentlichen Aspekte und Bestandteile, teilweise basierend auf den eben genannten Arbeiten, beleuchtet werden.

5.3.1. Anforderungen

Ziel der Eigenentwicklung eines Simulators für Ad-hoc-Fahrzeugnetze ist, eine Möglichkeit zu schaffen, unterschiedliche Aspekte der Fahrzeugkommunikation mit Hilfe

[13] http://www.seas.upenn.edu/∼rahulm/Research/GrooveNet/, abgerufen am 15.03.2012

möglichst realistischer Simulationen bei geringem Aufwand zu untersuchen. Dies schließt neben Betrachtungen der Basisfunktionalität, wie der Übermittlung von Nachrichten zwischen Teilnehmern, auch Bereiche der Sicherheitsinfrastrukturen und dem Schutz der Privatsphäre von *VANET*-Teilnehmern mit ein. Somit müssen die folgenden Anforderungen erfüllt werden:

- *Integration von Fahrzeug- und Kommunikationssimulation*: Bei einer Neuentwicklung soll natürlich das größte Manko der vorhandenen Lösungen ausgeschaltet werden: die Trennung zwischen der Simulation des Straßen- und des Datenverkehrs. Beides soll nun integriert in einer Simulationsumgebung durchgeführt werden.
- *Plattformunabhängigkeit*: Der neu entwickelte Simulator soll auf unterschiedlichen Systemen genutzt werden können.
- *Performanz*: Der Simulator soll auch umfassendere Simulationen mit mehreren tausend Fahrzeugen auf großen Straßennetzen möglichst in Echtzeit berechnen können. Dazu soll keine spezielle Hardware notwendig sein; als Basis sollen handelsübliche Einzelplatzrechner dienen. Zu diesem Zweck ist auf eine performante Implementierung Wert zu legen, außerdem sollen die Fähigkeiten von Multiprozessorsystemen nutzbar gemacht werden.
- *Leistungsfähige graphische Darstellung*: Neben einer performanten terminalbasierten Version soll auch eine leistungsfähige graphische Oberfläche angeboten werden. Dies erleichtert, in die Simulationsvorgänge Einblick zu nehmen und somit das Erkennen von Fehlern oder Problemen in der Simulation. Die graphische Oberfläche soll übersichtlich und leicht verständlich gestaltet werden und den Simulationsvorgang so gering wie möglich beeinträchtigen. Zudem soll sie den Benutzer möglichst umfangreich über das aktuelle Geschehen informieren.
- *Integration eines Karteneditors*: Viele Simulationsumgebungen bieten keinerlei Hilfestellung bei der Erstellung von Verkehrsnetzen an. Diese müssen teilweise mit eigens entwickelten Skripten oder sogar von Hand per Texteditor modelliert und erzeugt werden. Da dies sehr zeitaufwändig ist und zudem eine Beschäftigung mit dem zugrundeliegenden Modell erfordert, soll dieser Arbeitsschritt vermieden werden. Stattdessen soll dem Benutzer in der graphischen Oberfläche ein einfach zu bedienender Editor für Karten zur Verfügung gestellt werden.
- *Einfache Konfiguration*: Das Erstellen von Szenarien sowie das Festlegen von Simulationsparametern soll dem Benutzer möglichst einfach gemacht werden, jedoch soll auch die Konfiguration großer und umfangreicher Simulationen mit externen Tools oder Skripten ermöglicht werden.
- *Realistische Verkehrssimulation*: Die Simulation des Straßenverkehrs soll mög-

lichst realitätsnah stattfinden, um eine Übertragbarkeit der Simulationsergebnisse auf die reale Welt zu ermöglichen. Dafür sind folgende Teilaspekte nötig:

  - *Realistisches Kartenmaterial*: Der Simulator soll neben einer Möglichkeit zum Erstellen eigener Straßennetze eine Importmöglichkeit vorhandenen Kartenmaterials ermöglichen. Je realistischer das für Simulationen verwendete Straßennetz ist, desto besser werden sich die Simulationsergebnisse auf die reale Welt übertragen lassen.
  - *Mikroverkehrssimulation*: In vielen Vekehrsmodellen wird der Verkehr als eine Art Masse oder Flüssigkeit modelliert. Dies verhindert jedoch Aussagen über das einzelne Fahrzeug. Daher soll eine Mikroverkehrssimulation implementiert werden, in der jedes Fahrzeug individuell berechnet wird und eigene lokale Entscheidungen treffen kann. Dies erhöht zwar den Simulationsaufwand erheblich, jedoch ist nur auf diese Weise eine realistische Abbildung des Verkehrs und der Auswirkungen von Nachrichten auf die Fahrzeugbewegungen möglich.

- *Einfache Anpass- und Erweiterbarkeit*: Die Anwendung soll es möglichst leicht machen, Anpassungen oder Ergänzungen an den Simulationskomponenten vorzunehmen. So könnten beispielsweise das Verhalten der Verkehrsteilnehmer abgewandelt oder neue Nachrichtentypen hinzugefügt werden.
- *Gute Dokumentation*: Das gesamte Projekt (inklusive Quellcode) soll genau dokumentiert und beschrieben werden, damit sich die Einarbeitung möglichst einfach gestaltet und die Anpass- und Erweiterbarkeit sinnvoll realisiert werden kann.

Mit der Umsetzung dieser Anforderungen kann ein Simulator für Ad-hoc-Fahrzeugnetze geschaffen werden, der den vorhandenen Alternativen (vgl. Abschnitt 5.2) insbesondere durch den Fokus auf *VANET*s sowie eine einfache Konfigurier- und Anpassbarkeit überlegen ist.

5.3.2. Implementierungsdetails und Geschwindigkeitsoptimierung

Unter Berücksichtigung der soeben betrachteten Anforderungen an den Simulator bietet sich die Programmiersprache Java[14] aus den folgenden Gründen an:

- Java ist plattformunabhängig. Daher können in Java implementierte Anwendungen auf vielen unterschiedlichen Plattformen und Betriebssystemen, für die Java verfügbar ist, ausgeführt werden.

[14] http://www.java.com/de/, abgerufen am 21.03.2012

- Java erleichtert die Implementierung graphischer Oberflächen. Zu diesem Zweck steht eine große Anzahl an Klassen und Schnittstellen zur Verfügung, die genutzt werden können.
- Java-Implementierungen laufen relativ performant. Zwar stand Java lange Zeit in dem Ruf, durch Bytecodeinterpretation und die virtuelle Maschine, in der es ausgeführt wird, sehr schlecht zu performen, allerdings gab es in dieser Hinsicht bereits vor Jahren eine deutliche Verbesserung. Schon 2003 konnten Nikishkov et al. zeigen, dass auch aufwändige Operationen bei entsprechender Implementierung in Java nicht wesentlich rechenzeitintensiver sind als Implementierungen in der Programmiersprache *C* [NNS03]. Auch die Einführung von *HotSpot* – eine Erweiterung der JavaVM, die unter anderem häufig genutzte Codeabschnitte vorab in Maschinencode übersetzt und somit das laufzeitintensivere Interpretieren vermeidet (vgl. [PVC01]) – hat zu der heute deutlich besseren Performanz von Java beigetragen.

Obwohl Java inzwischen eine sehr gute Leistung bietet, wurde dennoch für die Programmierung des *VANET-Simulators* eine Reihe von Regeln aufgestellt und beachtet, um die Laufzeitintensität der Implementierung nicht unnötig zu erhöhen (vgl. [Gru08]):

- Es sollte möglichst vermieden werden, viele Objekte neu zu erzeugen. Dies steigert einerseits den Speicherverbrauch enorm, andererseits kann auch durch Wiederverwendung von Objekten (wo möglich) die Arbeit des *GarbageCollectors* reduziert werden.
- Funktionsaufrufe verursachen einen gewissen Overhead durch Bearbeitung der Parameter, Speicherung des aktuellen Status sowie der Rücksprungadresse und ähnlichen Operationen. Zwar werden sehr kleine Funktionen – wie *Getter* und *Setter* – durch den *HotSpot*-Compiler optimiert, jedoch findet dies bereits bei etwas größeren Funktionen nicht statt. Gerade in laufzeitkritischen Abschnitten sollten also Funktionsaufrufe vermieden und stattdessen redundanter Code geschrieben werden. Da Letzteres die Wartbarkeit der Anwendung stark einschränken kann, ist es sehr wichtig, dass hier eine genaue Abwägung durchgeführt wird und gegebenenfalls vorhandene Redundanzen durch Dokumentation im Quellcode deutlich hervorgehoben werden.
- Wo immer die Möglichkeit besteht, sind primitive Datentypen (z. B. *int*, *boolean*, *float*, ...) den entsprechenden Wrapper-Klassen (*Integer*, *Boolean*, *Float*, ...) und natürlich auch komplexen in Objekten abgebildeten Datentypen vorzuziehen, da sie deutlich weniger Speicher benötigen und performanter bearbeitet werden können.
- Obwohl Gleitkommazahlen (*float*, *double*) durch den wesentlich größeren Wertebereich einige Vorteile bieten, sollten sie, wo immer möglich, durch Ganzzah-

len *(int*, *long*) ersetzt werden, da Operationen auf diesen wesentlich schneller ablaufen.

- Listen, auf die eine Vielzahl von Zugriffen erfolgen, sollten stets mit Hilfe von *Arrays* anstelle von *ArrayLists* implementiert werden. Letztere sind zwar komfortabler in der Nutzung, erfordern aber stets ein Iterator-Objekt sowie Funktionsaufrufe (*hasNext()* und *next()*) beim Durchlaufen. Auch das Verwenden der *get()*-Methode der *ArrayList* ist nicht annähernd so performant wie der direkte Zugriff auf Inhalte eines *Arrays*.

Die vollständige Berücksichtigung dieser Regeln ist aufwändig und verschlechtert zum Teil deutlich die Les- und Wartbarkeit des Quellcodes. Daher werden sie nur an den relevanten, weil laufzeitkritischen, Stellen konsequent umgesetzt. Insbesondere Programmabschnitte, die viele hundert Male pro Sekunde durchlaufen werden, können so deutlich optimiert werden. Auch werden an einigen Stellen – beispielsweise bei Operationen durch Fahrzeuge – einige Codeabschnitte dennoch in Methoden ausgelagert, damit eine gewisse Kapselung der Funktionalität erhalten bleibt und Änderungen einfacher vorgenommen werden können.

Die Implementierung selbst wird in der Java-Entwicklungsumgebung Eclipse[15] und mit Java in der Version 6 durchgeführt. Eclipse wird mit den Plugins *JAutodoc*[16] zur vereinfachten Dokumentation, *JInto*[17] zur Bearbeitung der Sprachdatei, *Fat Jar*[18] zur Erstellung einer gebündelten Datei zur Veröffentlichung und Weitergabe sowie *eclox*[19] zur Generierung der *doxygen*-Dokumentation[20] erweitert.

Bei der Programmierung selbst wird auf eine konsequente sprechende Bezeichnung von Variablen-, Konstanten- und Methodenbezeichnungen geachtet. Dabei werden die üblichen Konventionen von Java (i. d. R. *CamelCase* (Binnenmajuskel), Großschreibung von Klassennamen und grundsätzliche Verwendung von Großbuchstaben für Konstanten) beachtet. Das von Java propagierte *Package*-System wurde ebenfalls konsequent eingesetzt und der Quellcode auf diese Weise nach Funktionsbereichen strukturiert.

[15] http://eclipse.org/, abgerufen am 22.03.2012
[16] http://jautodoc.sourceforge.net/, abgerufen am 22.03.2012
[17] http://www.guh-software.de/jinto.html, abgerufen am 22.03.2012
[18] http://fjep.sourceforge.net/, abgerufen am 22.03.2012
[19] http://home.gna.org/eclox/, abgerufen am 22.03.2012
[20] http://www.stack.nl/~dimitri/doxygen/, abgerufen am 22.03.2012

5.3.3. Modell und Implementierung der Karte

5.3.3.1. Die *Map* als Grundlage

Jede Simulation findet grundsätzlich auf einer Karte – *Map* bezeichnet – statt. Diese ist in einem zweidimensionalen kartesischen Koordinatensystem mit beliebigem Seitenverhältnis modelliert. Als Einheit wurden Zentimeter gewählt, da dies in Hinblick auf die Ausdehnung von Fahrzeugen eine ausreichende Auflösung bietet. Die Koordinaten liegen dabei für beide Achsen ausschließlich im positiven Wertebereich [0 . . . 2.147.483.647]. Die maximale Größe resultiert aus den Begrenzungen des *Integer*-Bereichs in Java; negative Werte werden nicht zugelassen, damit Berechnungen vereinfacht werden. Eine Karte kann somit eine maximale Kantenlänge von 21 474 Kilometern annehmen, was unter Berücksichtigung des Erdumfangs von rund 40 000 Kilometern absolut ausreichen dürfte. Weiterhin ist eine Projektion der gewölbten Erdoberfläche auf sehr große Flächen ohnehin sehr problematisch (vgl. [Ken94]).

5.3.3.2. Aufteilung in *Regionen*

Karten können offensichtlich sehr groß werden und auch eine Vielzahl an Objekten (Straßen, Fahrzeugen, etc.) enthalten. Damit diese effizient aufgefunden werden können, ist es nicht sinnvoll, sie unsortiert abzuspeichern. In diesem Fall müsste, wenn beispielsweise Fahrzeuge in der Nähe eines bestimmten Ortes benötigt werden, eine lineare Suche über alle Objekte der Karte durchgeführt werden. Daher empfiehlt es sich, eine räumliche Organisation in einer geeigneten Datenstruktur vorzunehmen.

Dieses in der Geoinformatik wohluntersuchte Problem bietet unterschiedliche Lösungsmöglichkeiten. So können beispielsweise Baumstrukturen wie die von Samet in [Sam89] vorgestellten *Quadtrees* oder *KD-Bäume* (*KD* steht für „k-dimensional“, vgl. [Bar05]) eingesetzt werden. Für eine detaillierte Beschreibung der Funktionsweise dieser Bäume sei auf das Buch von Bartelme [Bar05] verwiesen. Die erwähnten Bäume bringen zwar den Vorteil einer hierarchischen Gliederung der Karte und eines sehr schnellen Zugriffs auf die gespeicherten Objekte, sind jedoch auch mit deutlichen Nachteilen behaftet: So verursacht das Einsortieren von Objekten einen gewissen Overhead und es können – je nach Modellierung – sehr unausgeglichene (und damit langsame und tiefe) Baumstrukturen entstehen, die regelmäßig neu balanciert werden müssen. Erschwerend kommt bei der Entwicklung die hohe Dynamik der Fahrzeuge hinzu, die ständig ihre Position auf der Karte, und damit möglicherweise auch in der Baumstruktur, ändern.

Zur Lösung dieses Problems wurde eine Entscheidung für eine Aufteilung der Karte in eine Gitterstruktur getroffen. Die Karte wird dabei in ein regelmäßiges Gitter (*Regionen*) aufgeteilt und die Objekte werden entsprechend abgespeichert (vgl. [Bar05]).

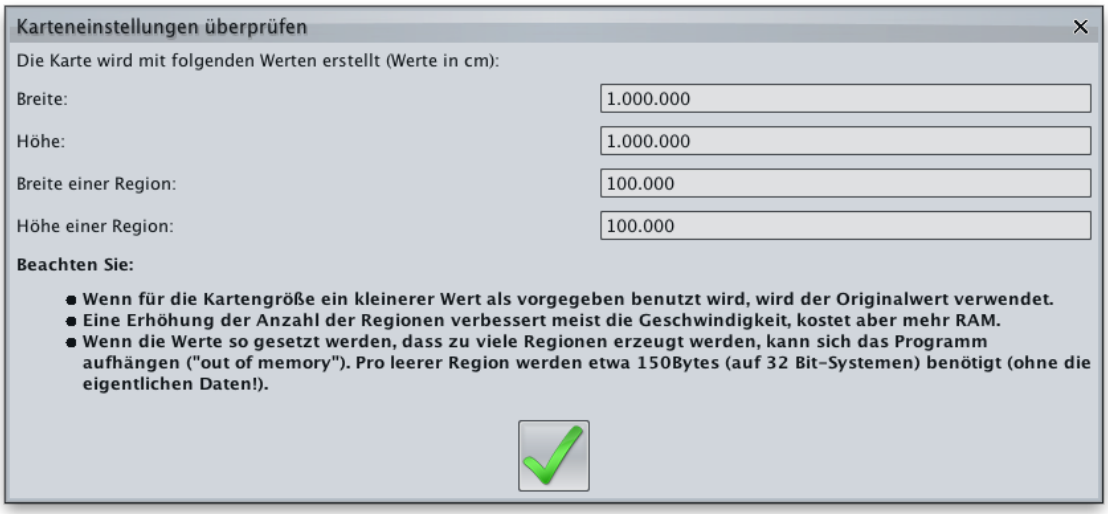


Abbildung 5.2.: *VANET-Simulator*: Anlegen einer neuen Karte (hier mit 100 Regionen)

Mit dieser in einem zweidimensionalen *Array* abgelegten Rasterung kann sehr einfach gearbeitet werden: Um beispielsweise die Region für einen bestimmten Punkt ausfindig zu machen, ist lediglich eine Ganzzahldivision der Koordinaten durch die Anzahl an Regionen in der jeweiligen Richtung durchzuführen. Möchte man dagegen Objekte an bestimmten Positionen ausfindig machen, ist innerhalb einer Region eine lineare Suche notwendig.

Bei der Implementierung stellen sich jedoch weitere Herausforderungen. Fahrzeuge bewegen sich sehr schnell über die Karte, weshalb sehr häufig überprüft werden muss, ob sie ihre alte Region verlassen und in eine neue eingefahren sind. Um diese Berechnung zu optimieren, werden bei der Erstellung der Regionen jeweils ihre Begrenzungen (*upperBound*, *lowerBound*, *leftBound* und *rightBound*) berechnet und abgespeichert. Somit kann die Überprüfung wesentlich schneller stattfinden.

Ist das abzuspeichernde Objekt nicht vereinfachend punktförmig, sondern eine Linie (wie beispielsweise Straßen), ist die Bestimmung der Region möglicherweise nicht mehr eindeutig, da sich das Objekt über mehrere Regionen erstrecken kann. In diesem Fall wird das Objekt zu sämtlichen Regionen hinzugefügt, durch die es verläuft. Dazu werden die Regionen von Start- und Endpunkt bestimmt und anschließend eventuell dazwischenliegende Regionen berechnet.

Über die Größe der Regionen kann ein *Time-Memory-Tradeoff* abgebildet werden. Je kleiner die Regionen sind, desto schneller können Objekte aufgefunden werden. Allerdings werden dann natürlich mehr Regionen benötigt und der Speicheraufwand steigt. Zudem besteht bei zu kleinen Regionen die Schwierigkeit, dass bei der Er-

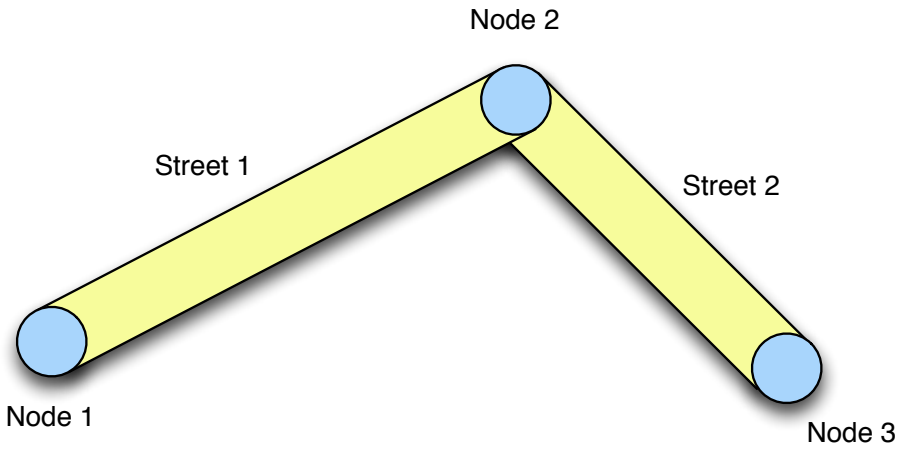


Abbildung 5.3.: *VANET-Simulator*: Modell des Straßennetzes

mittlung von Objekten in der Nähe eines bestimmten Orts eine Vielzahl an Regionen durchsucht werden muss, was ebenfalls den Rechenaufwand erhöht. Wählt man allerdings zu große Regionen, so muss sehr häufig eine große Anzahl an Objekten sequentiell durchsucht werden. Da ein Idealwert für die Größe der Regionen nicht generell angegeben werden kann, wird diese Ausdehnung bei der Erzeugung oder dem Öffnen einer Karte von dem Benutzer erfragt. Vorgeschlagene Standardwerte bewegen sich zwischen Kantenlängen von 50 000 und 100 000 cm (vgl. Abbildung 5.2).

5.3.3.3. Straßen

Das Straßennetz wird im *VANET-Simulator* als Graph aus Knoten (*Nodes,* Kreuzungs- und Start- bzw. Endpunkte) und Kanten (*Streets*, Straßenabschnitte) modelliert (vgl. Abbildung 5.3). Straßenabschnitte sind somit stets gerade und verbinden zwei Knoten. Gekrümmte Abschnitte müssen also über eine Reihe von sehr kurzen Straßenabschnitten, die in einem entsprechenden Winkel zueinander stehen, abgebildet werden.

Jeder *Node* ist über seine x- und y-Koordinate eindeutig bestimmt und erhält beim Laden der Karte eine eindeutige Kennung (*NodeID*), die für die Implementierung von Routingalgorithmen genutzt werden kann (vgl. Abschnitt 5.3.5.2). Auf jedem Punkt des Koordinatensystems kann nur maximal ein Knoten liegen. Neben der *NodeID* führt jeder Knoten noch eine Liste aller mit ihm verbundenen Straßen (*CrossingStreets*) sowie für eine performante Berechnung von Routen der Fahrzeuge noch eine weitere Liste, die ausschließlich von ihm wegführenden Straßen enthält. Damit können Routen alleine durch die Betrachtung der Knoten berechnet werden und eine Untersuchung der Attribute der Straßen ist nicht erforderlich (vgl.

Abschnitt 5.3.5.2).

Die Kanten des Graphen – die Straßenabschnitte – besitzen stets einen Start- und einen Endknoten (als *startNode* bzw. *endNode*) bezeichnet. Einbahnstraßen können dabei grundsätzlich nur von Start- zu Endknoten befahren werden, alle anderen Straßenabschnitte besitzen diese Restriktion nicht. Soll die Fahrtrichtung einer Einbahnstraße geändert werden, so müssen also lediglich Start- und Endknoten vertauscht werden.

Eine Straße kann mehrere Fahrspuren besitzen, deren Anzahl je Richtung in dem Attribut *laneCount* gespeichert wird. Somit wird in dem Straßenmodell davon ausgegangen, dass die Anzahl an Spuren in jede Fahrtrichtung gleich ist. Sollte dies nicht der Fall sein, kann die Straße über zwei nebeneinanderliegende Einbahnstraßen mit abweichenden Spurzahlen modelliert werden.

Die auf einer Straße befindlichen Fahrzeuge werden je nach ihrer Fahrtrichtung, aber unabhängig von ihrer Spur, in geordneter Reihenfolge in eine eigens konzipierte und optimierte Datenstruktur aufgenommen, die als *LaneContainer* bezeichnet wird. Diese wird, je nach Richtung der abgelegten Fahrzeuge, als *startToEndLane* bzw. *endToStartLane* bezeichnet und entspricht im Wesentlichen einer doppelt verketteten Liste. Die Java-Standardimplementierung *LinkedList* hat sich insbesondere bei Zugriffen als zu wenig performant herausgestellt, weswegen der *LaneContainer* neu entworfen wurde. Dieser enthält nur Verweise auf das erste (*head*) und das letzte Element (*tail*) der Liste und Informationen über die Anzahl an gespeicherten Objekten (vgl. Abbildung 5.4). Die Verbindungen zu den jeweils nächsten oder vorangegangenen Elementen führen die Objekte selbst mit, was die Zugriffe enorm beschleunigt: Bei der Java *LinkedList* muss, um ausgehend von einem Objekt auf das nächste zuzugreifen, dieses erste zunächst mit einem Aufwand von $O(n)$ in der Liste ausfindig gemacht werden. Von diesem ausgehend kann dann erst auf das nächste Element zugegriffen werden. Bei der eigenen Implementierung dagegen kann von einem Objekt direkt zum folgenden gegangen werden (Aufwand $O(1)$). Lediglich das (sortierte) Einfügen erfordert einen Aufwand von $O(n)$, jedoch befahren Fahrzeuge Straßen in der Regel von einem der beiden Enden her und werden somit zu Beginn oder am Ende einer Liste eingekettet. Alle für diese Operationen notwendigen Attribute und Methoden sind in der *LaneObject*-Klasse implementiert und müssen nur von allen auf den Straßen befindlichen Objekten geerbt werden.

5.3.3.4. Kartenmaterial von OpenStreetMap

Eine möglichst realistische Simulation des Straßenverkehrs wird mit der Nutzung von realen Straßennetzen erleichtert. Da in einigen Projekten im Internet das weltweite Straßenverkehrsnetz zusammengetragen und veröffentlicht wird, empfiehlt es

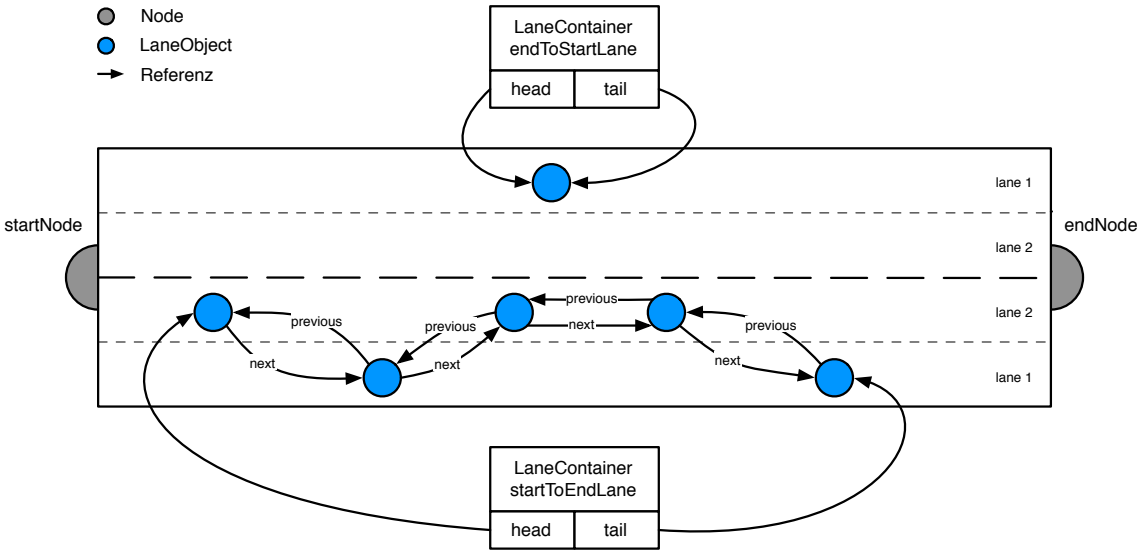


Abbildung 5.4.: *VANET-Simulator*: Prinzip des *LaneContainers* (nach [Gru08])

sich, auf diese Daten zurückzugreifen. Das umfangreichste dieser Projekte, die ihr Kartenmaterial mit freien Lizenzen zur Verfügung stellen, ist OpenStreetMap[21].

Über OpenStreetMap OpenStreetMap (detailliert vorgestellt in [HW08]) wurde 2004 mit dem Ziel ins Leben gerufen, ein freier und offener, durch *Crowdsourcing* getragener Gegenentwurf zu den kommerziellen Kartenprojekten (z. B. GoogleMaps[22] oder das von TomTom[23] gekaufte Tele Atlas) zu sein. Das Prinzip funktioniert ähnlich wie bei Wikipedia[24]: Jeder kann und soll zu diesem Projekt beitragen, wobei alle dort abgelegten Informationen über freie Lizenzen verfügbar[25] sind. Erstellt werden die Karten von Benutzern (nach eigenen Angaben hat OpenStreetMap bereits mehr als 730 000[26]), die mit GPS-Geräten Straßen und Wege abfahren oder -gehen, die aufgezeichneten GPS-Spuren um Metainformationen (wie beispielsweise Geschwindigkeitsbegrenzungen, Straßentyp, Fußgängerzonen, *Points-of-Interest* (*POI*), etc.) ergänzen und zum Projekt beisteuern.

Die Menge des Kartenmaterials wuchs während der letzten Jahre rasant und gewährleistet bereits sehr große Abdeckungen. Neis et al. stellten in ihrer neuesten

[21] http://www.openstreetmap.org/, abgerufen am 26.03.2012
[22] http://maps.google.com, abgerufen am 26.03.2012
[23] http://www.tomtom.com, abgerufen am 26.03.2012
[24] http://wikipedia.org, abgerufen am 26.03.2012
[25] Details unter http://wiki.openstreetmap.org/wiki/DE:Legal_FAQ, abgerufen am 26.03.2012
[26] Stand: August 2012, http://wiki.openstreetmap.org/wiki/Statistics, abgerufen am 22.08.2012

Untersuchung [NZZ12] fest, dass die in OpenStreetMap erfasste Gesamtstraßenlänge in Deutschland inzwischen um 27% größer ist als bei Tele Atlas. Allerdings weisen die Autoren auch noch auf einige Unzulänglichkeiten hin, die aber nach ihren Prognosen gegen Ende 2012 behoben sein sollten, wodurch OpenStreetMap auch für Fahrzeugnavigation vollständig konkurrenzfähig sein dürfte.

Kartenprojektion OpenStreetMap hält sämtliche Daten in Form von GPS-Koordinaten nach dem *World Geodetic System 1984* [WGS00] vor. Bei diesem standardisierten System wird ein Referenzellipsoid (*WGS84-Ellipsoid*) zugrunde gelegt und alle Punkte auf der Oberfläche werden in Längen- (*longitude*) und Breitengraden (*latitude*) angegeben. Da der *VANET-Simulator* jedoch einen flachen Kartenausschnitt mit metrischen Distanzangaben als Grundlage benutzt, ist es notwendig, eine Projektion der gewölbten Oberfläche aus OpenStreetMap auf diese zweidimensionale Ebene durchzuführen.

Kartenprojektionen sind ein sehr alter Forschungsgegenstand; bislang gibt es eine Vielzahl an Ansätzen, die alle Vor- und Nachteile besitzen. Einen guten Überblick über Projektionen bietet Kennedy in [Ken94]. Generell lassen sich drei Grundformen an Projektionen – konische, zylindrische und planare – unterscheiden, die jedoch alle nur eine realitätsnahe Abbildung in begrenzten Gebieten ermöglichen.

Die am weitesten verbreitete Projektion, die *Universal Transverse Mercator* (*UTM*)-Projektion (vgl. [SE90]), basiert – wie auch GPS – auf dem Referenzellipsoiden *WGS84* und weist damit eine sehr hohe Kompatibilität auf. Dies führt unter anderem dazu, dass viel EU-Staaten – so auch Deutschland – von dem von ihnen bisher genutzten Projektionssystem (für Deutschland ist es das *Gauß-Krüger-System*) zu dem zu *UTM* nahezu identischen *Europäischen Terrestrischen Referenzsystem 1989* (*ETRS89*) wechseln (vgl. [Gör10]).

Das von GPS – und damit auch von OpenStreetMap – eingesetzte Projektionssystem, UTM, nutzt eine konische Projektion. Dabei wird die Erdoberfläche in 60 nord-süd-verlaufende Streifen eingeteilt, die somit jeweils 6° umfassen. Die Nummerierung beginnt bei 180° westlicher Länge und verläuft in Richtung Osten. Von Süden nach Norden werden diese Streifen wiederum über 8° voneinander entfernte Breitenkreise in einzelne Felder unterteilt, die mit den Buchstaben C bis X bezeichnet sind. Für die Polarregionen (Buchstaben A und B sowie Y und Z) werden gesonderte Projektionstechniken angewendet (vgl. Abbildung 5.5 von Wikimedia Commons[27]). Weiterhin gibt es im südlichen Teil Norwegens sowie um Spitzbergen einige Besonderheiten, auf die hier jedoch nicht weiter eingegangen werden soll.

[27] http://commons.wikimedia.org/wiki/File:Utm-zones.jpg, abgerufen am 28.03.2012

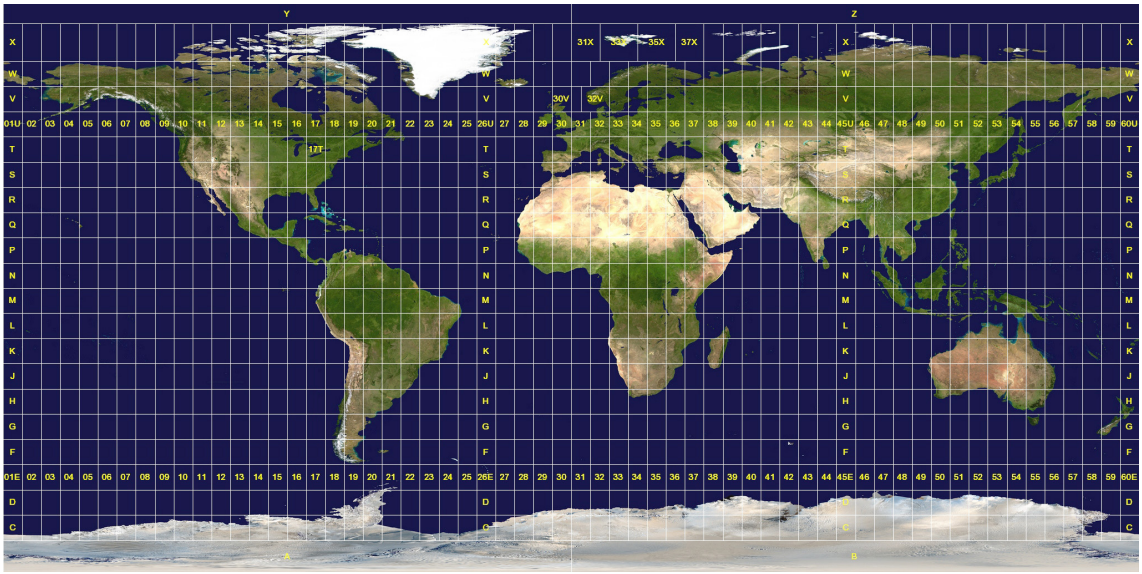


Abbildung 5.5.: Zonenaufteilung nach dem *UTM*-System (Quelle: Wikimedia Commons, CC-BY-SA 3.0)

Import von OpenStreetMap-Karten in den VANET-Simulator Die von OpenStreetMap gesammelten Karteninformationen können nicht nur online betrachtet und verlinkt werden, sondern lassen sich auch in verschiedenen Formaten exportieren, unter anderem in einer speziellen XML-Struktur[28]. Diese kann in den *VANET-Simulator* importiert werden.

Zum Export eines Kartenausschnitts kann entweder die Webseite des Projekts selbst (dies funktioniert aber nur bis ca. 50.000 Knoten) oder das von OpenStreetMap angebotene Tool *Osmosis*[29] verwendet werden. Mit diesem lassen sich Ausschnitte beliebiger Größe relativ komfortabel exportieren. Es werden allerdings auch regelmäßig Auszüge des gesamten Kartenmaterials oder einzelner Länder in Dateien bereitgestellt[30].

Der *VANET-Simulator* verfügt über eine Importfunktionalität, um Karten aus dem OpenStreetMap-Projekt für Simulationen nutzbar zu machen. Diese müssen dazu in das eigene Modell überführt werden. Der erste Schritt ist die Kartenprojektion (siehe oben), da, wie bereits beschrieben, alle Daten der OpenStreetMap-Karten in GPS-Koordinaten vorliegen, die in das plane Koordinatensystem übertragen werden müssen.

Zur Umrechnung von Koordinaten nach WGS84 in das *UTM*-System ist eine Reihe

[28] vgl. http://wiki.openstreetmap.org/wiki/DE:OSM_XML, abgerufen 28.03.2012
[29] http://wiki.openstreetmap.org/wiki/Osmosis, abgerufen am 28.03.2012
[30] siehe http://wiki.openstreetmap.org/wiki/Planet_dumps, abgerufen am 28.03.2012

highway	Geschwindigkeit	Färbung
motorway	130 km/h	blau
motorway_link	70 km/h	blau
trunk	110 km/h	grün
trunk_link	70 km/h	grün
primary	100 km/h	rot
primary_link	70 km/h	rot
secondary	100 km/h	orange
tertiary	90 km/h	gelb
road	70 km/h	weiß
unclassified	70 km/h	weiß
residential	30 km/h	weiß
living_street	10 km/h	weiß
service	10 km/h	weiß
unsurfaced	2 km/h	weiß
track	2 km/h	weiß

Tabelle 5.1.: Straßentypen aus OpenStreetMap, ihre geschätzte Maximalgeschwindigkeit im Simulator sowie ihre Färbung

von Kalkulationen notwendig. Die mathematischen Details sind in [GDA95] ausführlich dargelegt und wurden entsprechend implementiert. Aus Performanzgründen wurden dabei einige Werte vorgerechnet und zudem einige von Harm in [Har08] beschriebene Vereinfachungen vorgenommen. Die Ergebnisse sind jedoch weiterhin vollständig konsistent mit dem UTM-System.

Beim Import wird aus den minimalen und maximalen Längen- und Breitengraden die Kartengröße abgeleitet und zugrunde gelegt. Die von OpenStreetMap gelieferten Daten über das Straßennetz sind, wie auch bei dem eigenen Modell, in Knoten (*nodes*) und Kanten (*ways*) abgebildet. Daher ist die Übersetzung relativ einfach möglich. Annotiert sind diese Straßen unter anderem mit Informationen über Einbahnstraßen (*oneway*), ihren Straßennamen, die übernommen werden, und ihrem Straßentyp (im Element *highway*). Letzterer wird genutzt, um dort, wo keine maximale Geschwindigkeit (im Element *maxspeed*) mitgeliefert wird, diese zu schätzen (vgl. Tabelle 5.1) sowie eine Färbung für die Straße im VANET-Simulator festzulegen.

Weitere Implementierungsdetails zum Import von OpenStreetMap-Karten können der Arbeit von Gruber [Gru08] entnommen werden.

5.3.3.5. Speicherung der Karten

Einmal erstellte oder von OpenStreetMap importierte Karten sollen natürlich persistent speicherbar sein. Um dies zu ermöglichen, wird eine XML-Struktur verwendet, die sämtliche *Nodes* und *Streets* sowie alle Attribute aufnimmt. Das XML-Schema dieser Datenstruktur findet sich Anhang in Abschnitt B.1.

5.3.4. Szenarien als Sammlung der dynamischen Komponenten

Im Gegensatz zu der *Map*, die das für eine Simulation statische Kartenmaterial in sich aufnimmt, beinhaltet das Szenario (*Scenario*) alle dynamischen Objekte der Simulation – also Objekte, die beweglich sind (wie beispielsweise Fahrzeuge) oder solche, die mit anderen Objekten interagieren können (beispielsweise RSUs) – und alle Simulationsparameter. Alle Parameter der Szenarien sowie alle Objekte können über die Oberfläche erzeugt und manipuliert werden. Gespeichert werden die Daten – wie auch die *Map* – in einer XML-Struktur, deren XML-Schema im Anhang in Abschnitt B.2 zu finden ist.

Im Einzelnen werden die folgenden Informationen und Objekte im Szenario zusammengefasst, die in den nächsten Abschnitten detailliert erläutert werden:

- *Settin*gs: Die Konfigurationseinstellungen des Szenarios beeinflussen unter anderem, ob Kommunikation zwischen Fahrzeugen stattfindet und wie häufig Nachrichten ausgetauscht werden. Außerdem beinhalten sie die Konfigurationsparameter für jede der drei implementierten Techniken zum Schutz der Privatsphäre – Mix-Zonen, Silent Periods und SLOW, vgl. Abschnitt 5.3.8.
- *Vehicles*: Sämtliche Fahrzeuge (vgl. Abschnitt 5.3.5) sind mit ihren Attributen im Szenario hinterlegt.
- *MixZones*: Eventuell vorhandene Mix-Zonen sind mit ihren Positionen und Kerneigenschaften gespeichert (vgl. Abschnitt 5.3.8).
- *RSUs*: Die Positionen aller Roadside Units und ihrer Sendereichweiten werden vorgehalten (vgl. Abschnitt 5.3.6).
- *ARSUs*: Eventuell vorhandene von einem Angreifer kontrollierte RSUs (vgl. Abschnitt 5.3.6) sind Teil des Szenarios.
- *Events*: Der VANET-Simulator unterstützt Ereignisse der Simulation (vgl. Abschnitt 5.3.7), die ebenfalls im Szenario konfiguriert und hinterlegt werden.

5.3.5. Modell und Implementierung der Fahrzeuge

Die Fahrzeuge und ihre Simulation sind ein besonders wichtiger Aspekt des *VANET-Simulators*. Nur wenn es gelingt, die Fahrzeuge – und damit den gesamten Verkehrsfluss – möglichst realitätsnah abzubilden, ist es möglich, aus Simulationsergebnissen Schlüsse auf entsprechende Abläufe in der realen Welt zu ziehen. Die Fahrzeuge sind somit einer der komplexesten Bestandteile des Simulators. Die Details der Implementierung sollen in der Folge dargelegt werden.

5.3.5.1. Das Verkehrsmodell

Unter einem Verkehrsmodell versteht man die modellhafte Abbildung des Verhaltens und der Fortbewegung von Verkehrsteilnehmern in eine Reihe von Regelsätzen, die zwar hinreichend realistisch sind, jedoch durch Reduktion eine deutlich geringere Komplexität als die Gegebenheiten in der realen Welt aufweisen. In der Literatur über Verkehrsmodelle hat sich im Laufe der Jahre eine Unterscheidung nach drei Granularitätsstufen eingebürgert. Man differenziert in makroskopische, mesoskopische und mikroskopische Modelle (vgl. [Ros03, Kra98]). Makroskopische Verkehrsmodelle betrachten keine individuellen Fahrzeuge, sondern den Verkehr eines Straßenabschnitts in seiner Gesamtheit. Dieser wird als Flüssigkeit oder Gas modelliert und weist entsprechende Eigenschaften wie Dichte, Masse und Geschwindigkeit sowie davon abgeleitete Größen wie Massenfluss auf. Der Zu- und Abstrom von Fahrzeugen wird als *Verdampfung* und *Verflüssigung* modelliert; das einzelne Fahrzeug ist somit nicht relevant und kann auch nicht identifiziert werden. Demgegenüber stehen Mikroverkehrsmodelle, die ausschließlich einzelne Fahrzeuge mit ihren Eigenschaften (wie beispielsweise Geschwindigkeit, Masse, maximale Beschleunigung und Bremskraft etc.) betrachten und den Verkehrsfluss über ihre Gesamtheit abbilden. Als eine Art Kompromiss Zwischen wurden zwischen diesen beiden Extremen mesoskopische Verkehrsmodelle eingeführt, die einen Mittelweg beschreiten. In diesen Konzepten werden unterschiedliche Aspekte der anderen beiden Modelle kombiniert, um eine für den jeweiligen Anwendungsfall günstige Beschreibung des Verkehrs herbeizuführen. Für Beispiele sei hier auf die Arbeit von Rose [Ros03] verwiesen.

Für die Simulation im *VANET-Simulator* kommt nur eine Mikroverkehrssimulation in Frage, da im Rahmen der Untersuchung vor allem erhoben werden soll, wie einzelne Fahrzeuge in diesem System kommunizieren und Nachrichten austauschen, wie sie ihre Route auf Grund neu erhaltener Informationen über Verkehrslagen anpassen und nicht zuletzt, wie einzelne Teilnehmer von einem Angreifer beobachtet und verfolgt werden können. Mikroverkehrsmodelle bilden das Verhalten von Verkehrsteilnehmern in der Regel in einer Reihe von Untermodellen ab, die bestimmte

Ereignis	pro Zeiteinheit	pro zurückgelegte Strecke
Verkehrsinformationen	5 pro Sekunde	300 pro km
Beobachtungen des Fahrers	2 pro Sekunde	120 pro km
Entscheidungen des Fahrers	40 pro Minute	40 pro km
Handlungen des Fahrers	30 pro Minute	30 pro km
Fehler des Fahrers	1 je 2 Minuten	1 je 2 km
Riskante Situationen	1 je 2 Stunden	1 je 120 km
Beinaheunfälle	1 je Monat	1 je 2.000 km
Unfälle	1 in 7,5 Jahren	1 je 150.000 km
Unfälle mit Verletzung	1 in 100 Jahren	1 je 2 Mio. km
Tödliche Unfälle	1 in 2.000 Jahren	1 je 40 Mio. km

Tabelle 5.2.: Ereignisse eines durchschnittlichen Fahrers mit einer durchschnittlichen Geschwindigkeit von 60 km/h und einer Jahresfahrleistung von 20.000 km (Untersuchung von [HL91] nach [AK00])

Aspekte der Bewegung im Straßenverkehr (z. B. Verhalten bei Spurwechsel, Einhalten von Abständen, Erkennen von Hindernissen, etc.) beschreiben (vgl. [AK00]). Archer und Kosonen argumentieren, dass sich Unterschiede im Verhalten der Fahrer – leider noch nicht vollständig befriedigend – durch randomisierte Funktionen abbilden lassen und notwendig sind, um ein plausibles Abbild des Verkehrs zu liefern.

Die Modelle müssen jedoch eine gewisse Abstraktion des Fahrverhaltens der Verkehrsteilnehmer bieten, da in einer Simulation die Behandlung aller Aspekte eines Fahrzeugführers zwar theoretisch möglich ist, diese jedoch eine sehr hohe Komplexität aufweisen würde. So haben Häkkinen und Luoma 1991 in einer finnischsprachigen Studie ([HL91], vgl. [AK00]) untersucht, welche Ereignisse ein Fahrer beim Führen eines Fahrzeugs erlebt. Diese sind in Tabelle 5.2 zusammengefasst. Es ist leicht zu ersehen, dass eine detaillierte Abbildung all dieser Vorkommnisse äußerst schwierig ist.

In der Literatur existieren bereits mehrere verbreitete Mikrosimulationsmodelle. Im Folgenden sollen einige dieser Modelle auf ihre Eignung für den *VANET-Simulator* hin untersucht werden. Dabei werden die als am relevantesten eingeschätzten Modelle herangezogen; umfangreichere Überblicke finden sich bei Krauß ([Kra98]) oder Brockfeld und Wagner ([BW04]).

Anforderungen Aus den Anforderungen an den *VANET-Simulator* (vgl. Abschnitt 5.3.1) sowie begründet in den Simulationsvorhaben lassen sich die Anforderungen für das Verkehrsmodell ableiten:

- Hohe Performanz: Der *VANET-Simulator* soll zur Untersuchung größerer Straßenabschnitte mit einer Vielzahl an Fahrzeugen verwendet werden. Diese Simulationen sollen sich auch auf handelsüblicher PC-Hardware durchführen lassen. Es ist somit unerlässlich, dass das Verkehrsmodell eine performante Implementierung ermöglicht.
- Realistische Fahrzeugbewegungen: Wie bereits erläutert, ist es für die Aussagekraft der Ergebnisse unerlässlich, dass eine möglichst realitätsnahe Simulation durchgeführt wird. Das Modell muss also leisten, dass eine plausible Fahrzeugbewegung erzeugt wird, die jedoch nicht nur aus einer Reihe geklonter Fahrzeuge bestehen darf. Es sollte stattdessen auch unterschiedliche Charakteristika einzelner Fahrer berücksichtigen und abbilden.
- Hohe Vollständigkeit: Eine Reihe von Modellen betrachtet nur einzelne Aspekte der Fahrzeugbewegung wie beispielsweise das Einfädeln bei Zufahrten zu mehrspurigen Straßen. Mit dem *VANET-Simulator* sollen jedoch möglichst alle denkbaren Situationen des Straßenverkehrs simuliert werden können. Zu diesem Zweck ließen sich zwar Teilmodelle kombinieren, dies birgt jedoch einen größeren Aufwand in sich und wirft problematische Fragen bezüglich der Priorisierung einzelner Aspekte in Bereichen auf, in denen sich die Konzepte überschneiden.
- Kompatibilität zu den Kartenkonzepten: Das Verkehrsmodell soll in der Lage sein, sich in das Grundkonzept des Simulators – der Betrachtung einer Karte oder eines Kartenausschnitts mit Hilfe eines zweidimensionalen kartesischen Koordinatensystems – einzufügen.

Nagel-Schreckenberg-Modell Das von Nagel und Schreckenberg entwickelte Modell wurde 1992 in [NS92] vorgestellt und basiert auf Zellularautomaten (*Cellular Automata*, *CA*), deren Geschichte auf John von Neumann zurückgeht, der sie als formales Modell von selbstreproduzierenden Organismen vorgeschlagen hatte (vgl. [Sar00]). In den folgenden Jahren wurden die Automaten immer detaillierter untersucht, erweitert und in vielen verschiedenen Forschungsfeldern – unter anderem Biologie, Physik und Informatik – adaptiert.

Grundsätzlich basieren *CA*s auf einem eindimensionalen Array aus Zellen. Jede dieser Zellen hat zu einem bestimmten (diskreten) Zeitpunkt einen klar definierten Zustand, dessen Änderung in der Regel von den Zuständen anderer Zellen abhängt. Die Übergangsfunktion ist allen Zellen gemein.

Nagel und Schreckenberg adaptierten dieses Konzept zur Simulation von Straßenverkehr. Die Straße wird durch ein eindimensionales Array aus Zellen abgebildet, die jeweils entweder ein oder kein Fahrzeug beinhalten können. Die Geschwindigkeit v

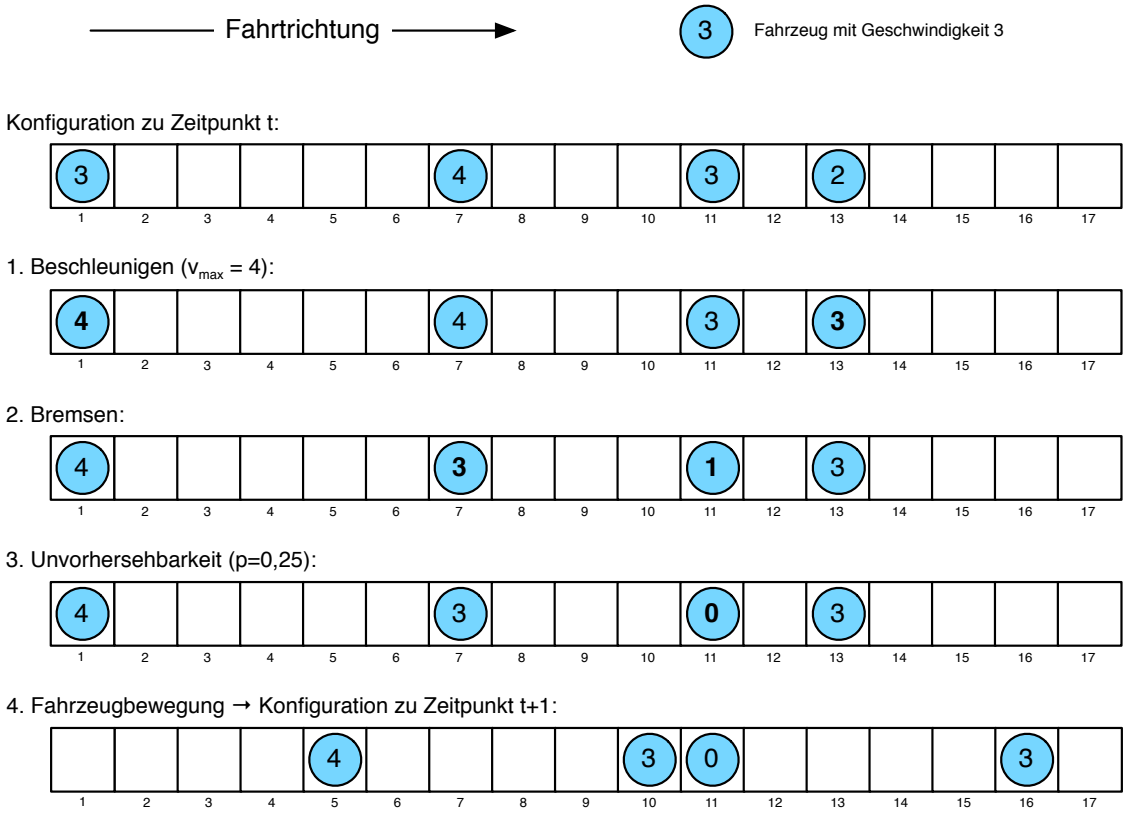


Abbildung 5.6.: Ablauf eines Simulationsschritts des Nagel-Schreckenberg-Modells

der Fahrzeuge wird in Form von Ganzzahlen repräsentiert (die Anzahl an Zellen, die das Fahrzeug pro Runde zurücklegen kann), und die Zeit wird in diskreten Schritten simuliert. Während eines Simulationsschritts wird ein Satz einfacherer Regeln angewendet (vgl. Abbildung 5.6):

1. Beschleunigung: Falls die Geschwindigkeit v eines Fahrzeugs kleiner als v_{max} und der Abstand zum nächsten Fahrzeug größer als $v + 1$ ist, so wird die Geschwindigkeit um 1 erhöht.
2. Bremsen: Falls ein Fahrzeug an der Stelle i ein anderes Fahrzeug an der Stelle $i + j$ sieht ($j \leq v$), so reduziert sich seine Geschwindigkeit auf $j - 1$. Damit wird eine Kollisionsfreiheit sichergestellt.
3. Unvorhersehbarkeit: Mit der Wahrscheinlichkeit p reduziert sich die Geschwindigkeit jedes Fahrzeugs (sofern sie größer als 0 ist) um 1.
4. Fahrzeugbewegung: Jedes Fahrzeug bewegt sich v Zellen weiter.

Dieses einfache Modell kann bereits einspurigen Verkehr überraschend realitätsnah abbilden (vgl. [NS92]), für andere Verkehrssituationen sind jedoch Erweiterungen

nötig. Diese ermöglichen es, unter anderem mehrspurigen Verkehr (vgl. [RNSL96]), komplexe Kreuzungen (vgl. [Nag93]), realistische Beschleunigung (vgl. [BJH96]) oder Ampeln (vgl. [BBSS01]) zu simulieren. Auf diese einzelnen Erweiterungen soll an dieser Stelle nicht detaillierter eingegangen und stattdessen auf die angegebenen Publikationen verwiesen werden.

Das Nagel-Schreckenberg-Modell bietet durch den einfachen Regelsatz und die Diskretisierung von Ort und Zeit eine sehr hohe Performanz. Dies wird insbesondere dadurch unterstützt, dass die Erweiterungen modular funktionieren und so genau die benötigten Komponenten hinzugenommen werden können. Nachteilig zeigt sich jedoch, dass die Granularität des Modells deutlich gröber ist als die des zu verwendenden Kartenmaterials: Eine Zelle entspricht der Summe aus der Größe eines Fahrzeugs und einem einzuhaltenden Sicherheitsabstand. Letzterer ist geschwindigkeitsabhängig; diese Variation ist jedoch im Modell nicht vorgesehen. Weiterhin gilt für alle Fahrzeuge der gleiche Regelsatz, der lediglich durch Zufallsvariablen variieren kann. Dies verhindert, bestimmten Fahrzeugen charakteristische Eigenschaften zuzuweisen oder Reaktionen auf eingegangene Nachrichten bei einzelnen Fahrzeugen zu simulieren.

SK-Modell Das SK-Modell wurde von Krauß im Rahmen seiner Dissertation [Kra98] entwickelt. Er motiviert die Konzeption eines neuen Modells mit der Feststellung, dass die bekannten Modelle in der Regel auf einen bestimmten Anwendungsbereich zugeschnitten sind und ihre Allgemeingültigkeit somit unklar ist. Sein Ziel ist es deshalb, ein generisches Modell zur Beschreibung von Mikroverkehr zu schaffen. Dabei greift er jedoch durchaus auf Aspekte des Nagel-Schreckenberg-Modells oder des Modells von Gipps (vgl. [Gip81, Gip86]) zurück.

Die Bewegung von Fahrzeugen in einem einspurigen Modell lässt sich nach Krauß relativ einfach mit vier Gleichungen beschreiben:

$$v_{safe}(t) = v_l(t) + \frac{g(t) - g_{des}(t)}{\tau_b + \tau} \tag{5.1}$$

$$v_{des}(t) = min\left[v_{max}, v(t) + a(v)\Delta t, v_{safe}(t)\right] \tag{5.2}$$

$$v(t + \Delta t) = max\left[0, v_{des}(t) - \eta\right] \tag{5.3}$$

$$x(t + \Delta t) = x(t) + v\Delta t \tag{5.4}$$

Diesem Modell liegt die Annahme zugrunde, dass Fahrzeuge nie schneller fahren als es die Verkehrssituation zulässt (v_{safe}), ansonsten jedoch versuchen, ihre Geschwindigkeit v möglichst bis v_{max} zu erhöhen. In Gleichung 5.1 wird auf Basis der Geschwindigkeit und des Abstands zu einem möglichen Vordermann (v_l und g), des

notwendigen Sicherheitsabstands g_{des}, der Reaktionszeit τ sowie der zu erwartenden Verzögerung des Vordermanns τ_b die maximale Geschwindigkeit unter Berücksichtigung der Verkehrssicherheitsanforderungen v_{safe} berechnet. Anschließend wird mit Gleichung 5.2 die angestrebte Geschwindigkeit v_{des} als das Minimum der maximal zulässigen Geschwindigkeit v_{max}, der maximal physikalisch möglichen Geschwindigkeit $v(t) + a(v)\Delta t$ und der aus Verkehrssicherheitsgründen maximalen Geschwindigkeit $v_{safe}(t)$ bestimmt. Die tatsächlich zum Einsatz kommende Geschwindigkeit $v(t + \Delta t)$ wird in Gleichung 5.3 aus der zuvor bestimmten gewünschten Geschwindigkeit v_{des} bestimmt, die um einen Zufallswert η reduziert wird, um die menschliche Abweichung von optimalem Fahrverhalten zu simulieren. Die Konstruktion über das Maximum aus der modifizierten gewünschten Geschwindigkeit und 0 stellt sicher, dass die Geschwindigkeit $v(t + \Delta t)$ niemals negativ wird. In einem letzten Schritt werden mit Gleichung 5.4 alle Fahrzeuge entsprechend ihren Geschwindigkeiten v ausgehend von ihrer aktuellen Position $x(t)$ an ihren neuen Ort $x(t + \Delta t)$ bewegt.

Im Laufe seiner Arbeit erweitert Krauß das Modell noch um mehrspurigen Verkehr. Für diese Beschreibungen sei auf seine Dissertation [Kra98] verwiesen. Dort belegt Krauß auch die realitätsgetreue Abbildung von Verkehrsflüssen mit Hilfe ausführlicher empirischer Untersuchungen.

Grundsätzlich unterscheidet sich das SK-Modell in seinem Ablauf nicht wesentlich von dem Nagel-Schreckenberg-Modell (NS-Modell): Es werden nach wie vor die Simulationsschritte des NS-Modells – Beschleunigung und Bremsen (bei *SK* über die zweite Gleichung, in die das Ergebnis der ersten einfließt), zufälliges Abbremsen, Fahrzeugbewegung – durchgeführt. Die Rasterung des Konzepts ist jedoch nun deutlich feiner geworden: Kommen im NS-Modell noch Bereiche, die ein Fahrzeug inklusive Sicherheitsabstand umfassen, als kleinste Einheit zum Einsatz, kann das SK-Modell mit beliebig genau aufgelösten Distanzen und Geschwindigkeiten arbeiten.

Diese gesteigerte Genauigkeit und Teilbarkeit der Distanzen und Geschwindigkeiten führt durch komplexere Berechnungen, beispielsweise die Ermittlung des Quotienten $v_{safe}(t)$, zu einer schlechteren Performanz im Vergleich zu den Ganzzahloperationen des NS-Modells. Dennoch ist der Rechenaufwand überschaubar und stellt kein Hindernis für den Einsatz dieses Modells dar.

Allerdings teilt das SK-Modell mit dem NS-Modell die fehlende Fähigkeit zur Simulation individueller Fahrzeuge auf Basis nicht geteilter Informationen – abweichende Regeln einzelner Teilnehmer finden in diesem Konzept keine Berücksichtigung.

IDM / MOBIL Das von Treiber und Helbing in mehreren Arbeiten (u. a. [THH00, TH02]) vorgestellte *Intelligent-Driver Model* (*IDM*) bietet eine realitätsnahe Simulation von Straßenverkehr auf einer Spur (Fahrzeugfolgemodell). In diesem

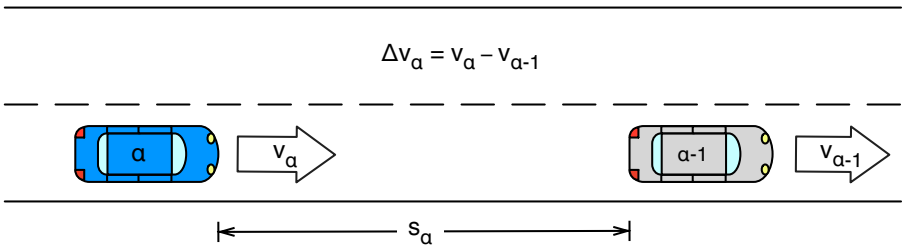


Abbildung 5.7.: Konzept des IDM nach [TH02]

Konzept, das nicht auf eine diskrete Zeiteinteilung angewiesen ist, ergibt sich die aktuelle Beschleunigung eines Fahrzeugs stets aus der eigenen Geschwindigkeit sowie der Geschwindigkeit von und dem Abstand zu einem eventuell vorausfahrenden Fahrzeug (vgl. Abbildung 5.7). Treiber und Helbig verwenden daher zur Ermittlung der Geschwindigkeit eines Teilnehmers die Berechnung in Gleichung 5.5.

$$\dot{v}_\alpha^{IDM}\left(v_\alpha, s_\alpha, \Delta v_\alpha\right) = a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*\left(v_\alpha, \Delta v_\alpha\right)}{s_\alpha}\right)^2\right] \tag{5.5}$$

Diese setzt sich nach [TH02] aus zwei Teilen, nämlich einem Beschleunigungsanteil $a_f(v) := a\left[1 - (v/v_0)^\delta\right]$ und einer Brems-Wechselwirkung $a_{int}\left(v, s, \Delta v\right) = -a\left[s^*\left(v, \Delta v\right)/s_\alpha\right]^2$zusammen. In ersterem wird die durchgeführte Beschleunigung im Idealfall (also ohne Hindernis) als von der aktuellen Geschwindigkeit v sowie einer Maximalgeschwindigkeit v_0 abhängiger Anteil der maximal möglichen Beschleunigung a ermittelt. Damit wird berücksichtigt, dass Fahrer umso stärker beschleunigen, je langsamer sie im Verhältnis zur gewünschten Maximalgeschwindigkeit fahren.

Der zweite Teil führt zu einer Verzögerungsreaktion, wenn der Abstand zu einem Vorderfahrzeug zu gering oder die Annäherungsrate zu hoch ist. Die Veränderung in der Beschleunigung ergibt sich dabei aus einem Vergleich des tatsächlichen Abstands zum Vorderfahrzeug s_α mit einem „effektiven Wunschabstand“ [TH02] $s^*(v, \Delta v)$. Dieser kann mit Hilfe von Gleichung 5.6 ermittelt werden.

$$s^*\left(v, \Delta v\right) = s_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}} \tag{5.6}$$

Die ersten beiden Summanden dieser Berechnung lassen den minimalen Abstand zwischen Fahrzeugen bei Stillstand s_0 sowie einen geschwindigkeitsabhängigen, da über die als konstant angenommene Reaktionszeit T ermittelten, Sicherheitsabstand

Parameter	Typischer Wert
Wunschgeschwindigkeit v_0	$120\,{}^{\mathrm{km}}/_{\mathrm{h}}$
Zeitlicher Sicherheitsabstand T	$1,4\,\mathrm{s}$
Maximale Beschleunigung a	$1,2\,{}^{\mathrm{m}}/_{\mathrm{s}^2}$
Komfortable Bremsverzögerung b	$1,5\,{}^{\mathrm{m}}/_{\mathrm{s}^2}$
Minimaler Abstand zum Vorderfahrzeug s_0	$2\,\mathrm{m}$

Tabelle 5.3.: Typische Parameter des IDM nach [TH02]

einfließen. Der dritte Summand spielt nur dann eine Rolle, wenn die Relativgeschwindigkeit zum Vorderfahrzeug Δv nicht null ist und wächst mit ihr. Die Parameter a und b bilden die maximal mögliche Beschleunigung beziehungsweise eine als komfortabel angenommene Bremsverzögerung. Ist eine stärkere oder schwächere als diese komfortable Verzögerung nötig, sorgt diese Berechnungsformel für eine Annäherung an diese Wunschverzögerung ohne dabei zu überschwingen. Damit kann ein dynamisches und – nach Einschätzung der Autoren – realistisches Fahrverhalten abgebildet werden, da es zu kontinuierlichen Bremsvorgängen führt und abrupte Änderungen bei Erreichen gewisser Abstände oder Geschwindigkeiten vermeidet.

Die Autoren definieren für die von ihnen gewählten Konstanten des Modells typische Parameter, die in Tabelle 5.3 zusammengefasst sind.

Um das lediglich für einspurige Straßen geeignete *IDM* für Mehrspurigkeit zu erweitern, stellen Treiber und Helbing in [TH02] die Spurwechselstrategie *MOBIL* – „Minimizing Overall Braking Induced by Lane-Changes“ – vor. Für die Details dieser Modellerweiterung sei auf die eben genannte Arbeit der Autoren verwiesen. Ein interessanter Aspekt von *MOBIL* soll jedoch nicht unerwähnt bleiben: Treiber und Helbing führen einen „Höflichkeitsfaktor“ ein, der die Aggressivität und die Risikobereitschaft bei Spurwechselvorgängen eines Fahrzeugführers abbildet.

Die Autoren zeigen in einer Vielzahl von Untersuchungen, dass ihre Konzepte auch in Grenzbereichen der Verkehrssimulation sehr gute und realitätsnahe Ergebnisse liefern. Das um *MOBIL* ergänzte *IDM* ermöglicht eine detaillierte Simulation des Fahrverhaltens mit beliebiger Genauigkeit. Einzelne Fahrzeuge können dabei gezielt über Parameter angepasst und somit individualisiert werden. Diese Genauigkeit und Anpassbarkeit muss jedoch durch eine Reihe von komplexen Berechnungen erkauft werden, was sich negativ auf die Performance auswirkt. Vorberechnungen besonders rechenintensiver Teile können leider nicht vorgenommen werden, da in sie stets aktualisierte Variablen Eingang finden.

Fazit Stellt man die drei detailliert analysierten Verkehrsmodelle den oben genannten Anforderungen gegenüber, so ergibt sich das in Tabelle 5.4 dargestellte Bild.

	Hohe Performanz	Realistische Fahrzeugbewegung	Hohe Vollständigkeit	Kompatibilität zu Kartenkonzept
NS-Modell	++	+ / -	+	–
SK-Modell	o	+ / o	+	+
ID-Modell	-	+ / ++	+	+

Tabelle 5.4.: Übersicht über die untersuchten Verkehrsmodelle

Das NS-Modell überzeugt durch die starke Diskretisierung und die Reduktion der Berechnungen auf Ganzzahloperationen mit einer sehr hohen Performance, dafür ist die Fahrzeugbewegung nur in ihrer Gesamtheit bei der Abbildung eines Verkehrsflusses realistisch, bei der Betrachtung einzelner Fahrzeuge sind in diesem Zusammenhang jedoch deutliche Einschränkungen hinzunehmen. Durch die starke Diskretisierung ist auch die Kompatibilität zu realistischem Kartenmaterial als sehr gering einzustufen, da sämtliche Straßenabschnitte ganzzahlige Vielfache von Fahrzeuglängen und einzuhaltenden Sicherheitsabständen sein müssen.

Ohne dabei das Grundkonzept des NS-Modells zu verändern kann das SK-Modell durch komplexere Berechnungen mit einer beliebigen Granularität der Positionen der Fahrzeuge aufwarten. Dies ermöglicht eine deutlich realistischere Bewegung auch einzelner Fahrzeuge ohne Einschränkungen bezüglich der Abbildung des gesamten Verkehrsflusses hinnehmen zu müssen. Jedoch findet diese Veränderung deutlich zu Lasten der Performanz statt. Zudem teilen sich auch bei dem SK-Modell alle Fahrzeuge den gleichen Regelsatz, und ein abweichendes Verhalten einzelner Verkehrsteilnehmer ist nicht abbildbar.

Die deutlich komplexesten Berechnungen werden bei der Anwendung des IDM durchgeführt, was zu einer sehr detaillierten Simulation des Fahrverhaltens einzelner Fahrzeuge führt. Die aufwändigen Berechnungen bieten jedoch nur eine schlechte Performanz. Bezüglich des Kartenmaterials zeigt sich auch hier eine sehr gute Eignung für realistische Distanzen.

Alle drei Modelle können eine ausreichend hohe Vollständigkeit bieten. Selbst wenn in den Grundformen häufig bei weitem nicht alle Aspekte des Straßenverkehrs abgebildet sind, so lassen diese sich mit Erweiterungen, die teilweise von den Autoren der Modelle selbst eingebracht werden, ergänzen. Dennoch zeigt sich – wie oben dargelegt –, dass keines der untersuchten Modelle uneingeschränkt Eingang in den Simulator finden kann. Daher wurde auf Basis der beschriebenen Konzepte und unter strenger Wahrung einer sehr guten Performanz eine eigene Variante der Mikroverkehrsmodelle umgesetzt.

Eigenes Modell Die Fahrzeuge im *VANET-Simulator* folgen – wie in anderen Verkehrsmodellen auch – einer Reihe von Regeln, die regelmäßig überprüft werden müssen. Die Granularität des Orts und der Zeit ist dabei theoretisch beliebig fein, da stetige Funktionen zum Einsatz kommen. Dennoch findet die Simulation letztendlich in diskreten Zeitschritten statt, die jedoch bis zu einem Minimum von einer Millisekunde frei gewählt werden können. Die räumliche Auflösung beträgt in der aktuellen Implementierung einen Zentimeter, was in Hinblick auf die Dimensionen eines Fahrzeugs ausreichend fein ist.

Bei seiner Erzeugung bekommt jedes Fahrzeug eine Reihe von Eigenschaften zugewiesen, beispielsweise seine maximale Geschwindigkeit, seine Beschleunigungs- und Verzögerungsrate sowie einen Start- und einen Zielort. Diese Attribute können sich von Fahrzeug zu Fahrzeug unterscheiden.

Die Bewegung der Fahrzeuge erfolgt nun auf Basis der folgenden Regeln:

1. Fahrzeuge überprüfen, ob sie aus irgendeinem Grund ihre Geschwindigkeit verringern müssen.
2. Fahrzeuge wechseln die Fahrspur, wenn sie auf einer anderen Spur besser vorankommen.
3. Gibt es für Fahrzeuge keinen Grund, langsam zu fahren, so versuchen sie möglichst schnell zu fahren.
4. Bevor Fahrzeuge bewegt werden, wird ihre Geschwindigkeit um einen zufälligen Wert reduziert, um eine realistischere Fahrweise zu simulieren.
5. Fahrzeuge werden abhängig von ihrer Geschwindigkeit, der Größe der Simulationsschritte und ihrer geplanten Route bewegt.

In einem ersten Schritt werden alle Situationen überprüft, in denen ein Fahrzeug die Geschwindigkeit verringern muss. Diese sind neben dem Erreichen des Zielorts die Annäherung an eine Kreuzung (vgl. Abschnitt 5.3.5.3) oder an ein Hindernis auf der Straße wie eine Blockade oder ein langsameres Fahrzeug. Als Entscheidungsgrundla-

ge für einen Bremsvorgang wird der Anhalteweg herangezogen, der stets als Abstand eingehalten werden soll. Gleichung 5.7 stellt die dazu verwendete Berechnung dar:

$$d = vT_r + \frac{v^2}{2b} \tag{5.7}$$

Der Anhalteweg d setzt sich somit aus der Summe des Reaktionsweges und des Bremsweges zusammen. Der Reaktionsweg ist diejenige Distanz, die ein Fahrzeug mit der Geschwindigkeit v innerhalb der Reaktionszeit T_r – also dem Zeitraum, der vergeht, bevor der Fahrer den Bremsvorgang einleitet – zurücklegt. Der eigentliche Bremsweg hängt ebenfalls von der aktuellen Geschwindigkeit v sowie der maximalen Bremsverzögerung b ab.

Die maximale Bremsverzögerung ist eine Eigenschaft des Fahrzeugs und kann somit unterschiedliche Werte annehmen; die Reaktionszeit T_r wurde einheitlich auf eine Sekunde („Schrecksekunde“) festgelegt.

Kann ein Fahrzeug den gewünschten Abstand d in Form des Anhalteweges bei der aktuellen Geschwindigkeit nicht mehr einhalten, so muss diese entsprechend reduziert werden. Bevor dies jedoch geschieht, überprüft das Fahrzeug, ob es den Bremsvorgang durch einen Spurwechsel vermeiden kann, da sich das Hindernis möglicherweise nur auf der eigenen Spur befindet.

Ist dieser Spurwechsel möglich, so wechselt das Fahrzeug die Spur (pro Simulationsschritt ist nur ein Spurwechsel um eine Spur nach links oder rechts möglich), andernfalls verringert es entsprechend der möglichen Verzögerung die Geschwindigkeit. Ist kein Hindernis vorhanden, zielen Fahrzeuge stets auf einen Wechsel auf eine weiter rechts gelegene Spur ab, um dem in vielen Ländern geltenden Rechtsfahrgebot[31] zu genügen.

Besteht für ein Fahrzeug keine Notwendigkeit, die Geschwindigkeit zu verringern, so beschleunigt es mit der eigenen Beschleunigungsrate a für die Dauer eines Simulationsschritts Δt. Dabei wird jedoch berücksichtigt, dass die Maximalgeschwindigkeit des eigenen Fahrzeugs v_{max} und die zulässige Höchstgeschwindigkeit der Straße v_{street}

[31] Das Rechtsfahrgebot besagt, dass ein Fahrzeug stets möglichst weit rechts zu fahren hat, sofern die Verkehrssituation dies zulässt. Geregelt ist dies beispielsweise in Deutschland in §2 Abs. 2 StVO (abrufbar unter http://www.gesetze-im-internet.de/stvo/), in Österreich in §7 Abs. 1 StVO (abrufbar unter http://www.jusline.at/Strassenverkehrsordnung_%28StVO%29_Langversion.html) und in der Schweiz in Art. 34 des Straßenverkehrsgesetzes (abrufbar unter http://www.admin.ch/ch/d/sr/741_01/index.html).

nicht überschritten wird. Zusammenfassend errechnet sich die neue Geschwindigkeit v' wie in Gleichung 5.8 dargestellt:

$$v' = min\left[v + a\Delta t, v_{max}, v_{street}\right] \tag{5.8}$$

In einem letzten Schritt, der Einfluss auf die Geschwindigkeit nimmt, wird diese bei jedem Fahrzeug um einen zufälligen Wert aus dem Intervall $[0\,{}^{\mathrm{cm}}/_{\mathrm{s}}; 6\,{}^{\mathrm{cm}}/_{\mathrm{s}}]$ reduziert, um eine zufällige Abweichung von perfektem Fahrverhalten abzubilden (siehe dazu auch [Kra98]). Die Geschwindigkeit darf dabei nicht in den negativen Bereich gelangen. Die Ermittlung dieses Wertes erfolgt über einen Pseudozufallszahlengenerator, der als Parameter die eindeutige Kennung des Fahrzeugs und den aktuellen Zeitpunkt erhält. Damit ist sichergestellt, dass die Simulation vollständig deterministisch und reproduzierbar durchgeführt wird.

Nachdem die Geschwindigkeit v für alle Fahrzeuge festgelegt ist, werden diese jeweils um die entsprechende Distanz $s = v\Delta t$ entlang ihrer Route bewegt, wobei Δt die Zeitspanne repräsentiert, die von einem Simulationsschritt abgedeckt wird. Damit ist eine Iteration der Simulation durchlaufen; mit dieser neuen Ausgangssituation können die einzelnen Schritte nun erneut durchgeführt werden. Je kürzer die Dauer eines Simulationsschritts gewählt wird, umso detaillierter werden die Bewegungen der einzelnen Fahrzeuge abgebildet, desto schlechter ist jedoch die Performanz, da in jedem Schritt sämtliche Berechnungen durchgeführt werden müssen.

5.3.5.2. Routing

Jedes Fahrzeug erhält bei seiner Erzeugung eine Liste von mindestens zwei Wegpunkten, die im Wesentlichen jeweils aus x- und y-Koordinaten eines Ortes bestehen, der um die Straße, auf der er sich befindet, ergänzt wird. Dies ist notwendig, da bei Brücken oder Tunneln ein Koordinatenpaar nicht zwingend einen Ort eindeutig definiert. Der erste Ort ist der Startort, an dem das Fahrzeug erzeugt wird, das anschließend alle weiteren Wegpunkte der Liste der Reihe nach abfährt um nach Erreichen des letzten Ortes entfernt zu werden. Somit können Fahrzeugströme ganz gezielt erstellt werden.

Zur Ermittlung der optimalen Route, derer sich ein Fahrzeug bedient, kommen eine Reihe bekannter Routingalgorithmen in Frage. Die Entscheidung zur Implementierung im *VANET-Simulator* fiel auf den A^*-Algorithmus (vgl. [Nil82, Ste94]), der als eine Verallgemeinerung des Dijkstra-Algorithmus (vgl. [Dij59]) angesehen wird. Der A^*-Algorithmus ist optimal, das heißt, er findet stets – sofern vorhanden – eine optimale Lösung, also einen kürzesten Weg. Dabei ist er jedoch im Gegensatz zum

Dijkstra-Algorithmus nicht darauf angewiesen, alle Knoten zu untersuchen, sondern findet die optimale Route – je nach Ausgestaltung der Hindernisse – wesentlich effizienter. *Greedy* Algorithmen, wie beispielsweise *Greedy Best-first-search* (vgl. [Pea84]), sind zwar in der Lage, mit noch geringerem Aufwand in einem Idealfall (d. h. der kürzeste Weg ist auch der direkteste) die optimale Lösung zu bestimmen; dies gelingt ihnen jedoch in komplexeren Systemen oft nicht. Abbildung 5.8 skizziert die Arbeitsweise der erwähnten Algorithmen.

Basis für die Berechnung einer Route ist die grob geschätzte benötigte Zeit. Der Algorithmus geht davon aus, dass eine Straße stets mit der maximal zulässigen Geschwindigkeit befahren werden und die benötigte Dauer für einen Straßenabschnitt daher mit der Division ihrer Länge durch die zulässige Höchstgeschwindigkeit bestimmt werden kann. Details zur konkreten Implementierung des A^*-Algorithmus für dieses Szenario werden von Gruber [Gru08] ausführlich beschrieben.

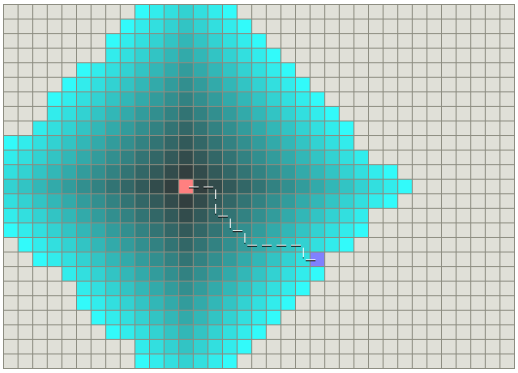
Die zurückzulegenden Routen der Fahrzeuge werden bei ihrer Erstellung vorberechnet, um die trotz günstigem Algorithmus aufwändigen Operationen nicht während der Simulation durchführen zu müssen. Prinzipiell wird die Kalkulation auf einer speziell angepassten und für die Routenberechnung optimierten Kopie des Straßennetzes durchgeführt und liefert für das Fahrzeug eine Liste von Knoten in der Reihenfolge, in der sie abgefahren werden müssen. Die Details der performanten Implementierung können der Arbeit von Gruber entnommen werden.

Zur Berechnung der Route kommt ein Interface zum Einsatz, das von einem zu verwendenden Routingalgorithmus lediglich implementiert werden muss. Dies ermöglicht einen einfachen Wechsel des Algorithmus und wirkt sich positiv auf die Wartbarkeit des Quellcodes aus.

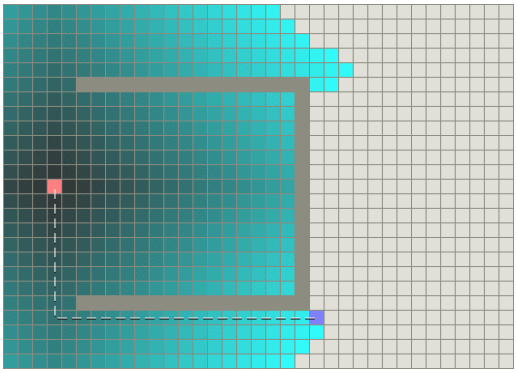
5.3.5.3. Kreuzungen

Besitzt ein *Node* des Straßensystems mehr als zwei abgehende oder ankommende Verbindungen (Straßen), so bildet er den Mittelpunkt einer Kreuzung. Bewegt sich ein einzelnes Fahrzeug über diesen Knoten hinweg, so entsteht für es kein Unterschied zu einem *Node*, der lediglich zwei Straßenabschnitte verbindet – das Fahrzeug kann sich ganz einfach darüber hinweg bewegen. Anders sieht die Situation aus, wenn sich mehrere Fahrzeuge einer Kreuzung nähern, da in diesem Fall ein Regelsystem über eine Priorisierung der Beteiligten bestehen muss, damit es nicht zu Kollisionen kommt. Dies ist eine relativ komplexe Aufgabe, da die Karten aus *OpenStreetMap* leider keinerlei Informationen über die Beschilderung in sich tragen und eine einfache „rechts-vor-links“-Regelung die Realität der meisten Straßen nicht abbildet.

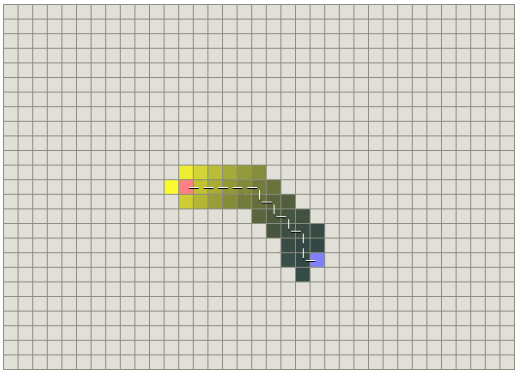
In einem ersten Schritt werden vor dem Start der Simulation alle Knoten daraufhin untersucht, ob sie eine Kreuzung bilden. Für die betroffenen Knoten wird im Fol-



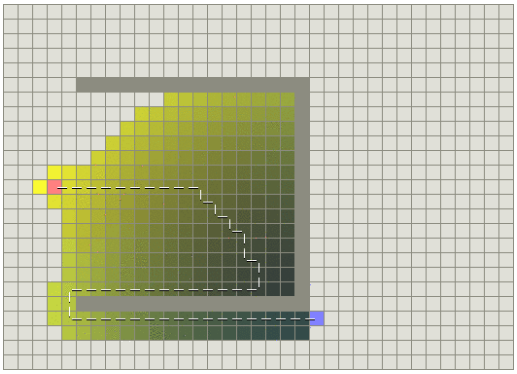
(a) Dijkstra ohne Hindernis



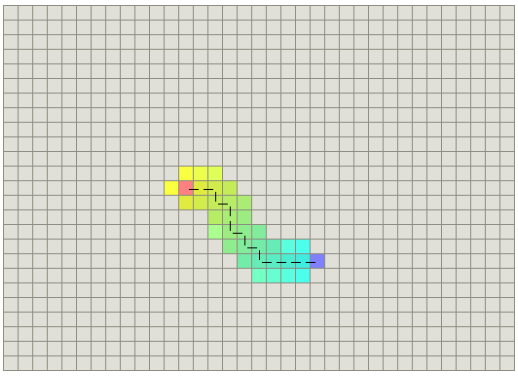
(b) Dijkstra mit Hindernis



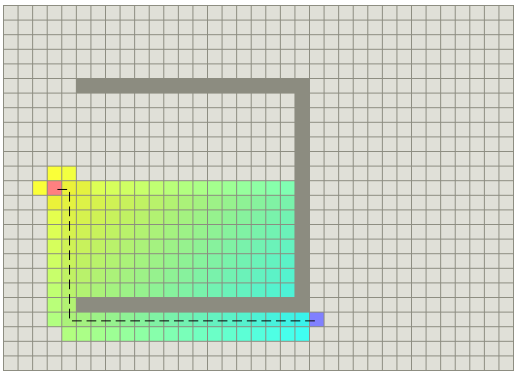
(c) Best-first-search ohne Hindernis



(d) Best-first-search mit Hindernis



(e) A^* ohne Hindernis



(f) A^* mit Hindernis

Abbildung 5.8.: Arbeitsweise von Dijkstra, Best-first-search und A^* (Quelle: [Pat11]). Untersuchte Knoten sind eingefärbt, die Linie stellt die gefundene Route dar.

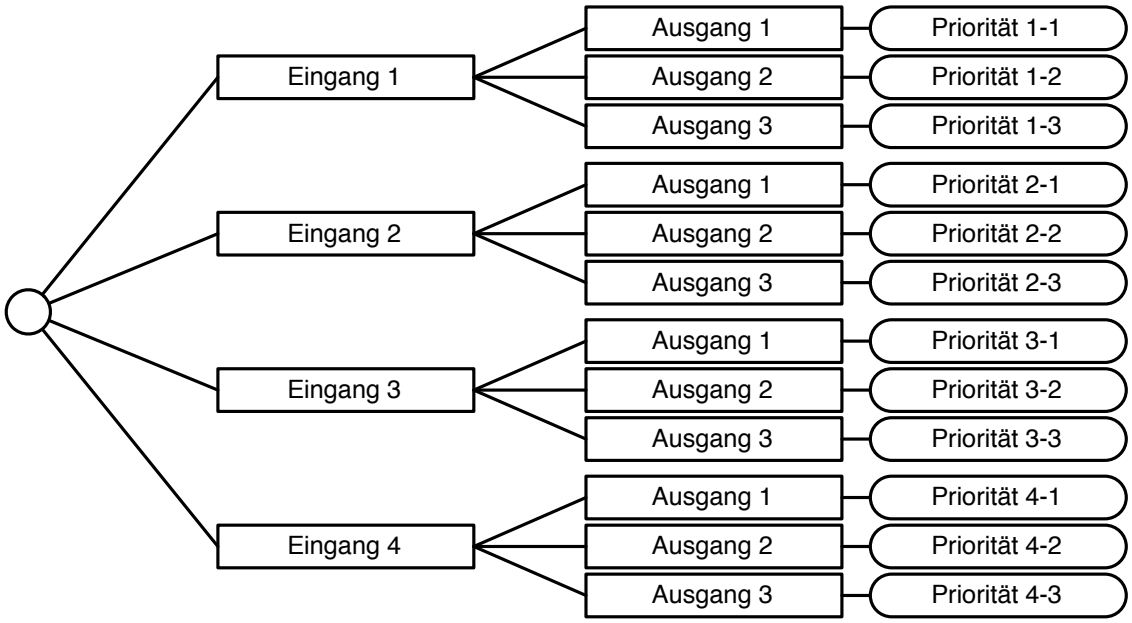


Abbildung 5.9.: *VANET-Simulator*: Logischer Aufbau der Datenstruktur für Prioritäten einer Kreuzung am Beispiel mit vier Eingängen und drei Ausgängen

genden eine Datenstruktur angelegt (realisiert über zwei geschachtelte *HashMaps*), in der alle möglichen Kombinationen aus ein- und ausgehenden Straßenabschnitten einen Eintrag – eine Priorität – erhalten (vgl. Abbildung 5.9).

Um diese Prioritäten zu ermitteln, gilt es zunächst, eine Vorfahrtsstraße (*priority street*) zu bestimmen. Dazu werden zusammengehörige Straßenabschnitte anhand ihres als Attribut hinterlegten Straßennamens identifiziert. Die Straße, zu der der Straßenabschnitt mit der höchsten zulässigen Geschwindigkeit gehört, wird als Vorfahrtsstraße festgelegt (vgl. Abbildung 5.10). Existieren keine zusammengehörigen Straßenabschnitte, so werden die beiden Straßensegmente mit der größten Maximalgeschwindigkeit gewählt und bilden gemeinsam die *priority street*. Sollten mehrere Straßen die gleiche Geschwindigkeitsbegrenzung aufweisen, so wird eine zufällige Straße ermittelt. Im Zusammenhang mit Einbahnstraßen kann sich hier ein Sonderfall ergeben, der dazu führen kann, dass ein drittes Straßensegment zur Vorfahrtsstraße hinzugenommen wird. Dieser selten auftretende und nicht problematische Fall wird von Gruber [Gru08] ausführlich beschrieben und soll hier keine weitere Betrachtung finden.

Ist die Vorfahrtsstraße definiert, so können alle Prioritäten ermittelt werden. Dabei kommen folgende Regeln zum Einsatz:

1. Priorität: Kommt ein Fahrzeug von einer Vorfahrtsstraße und verbleibt auf

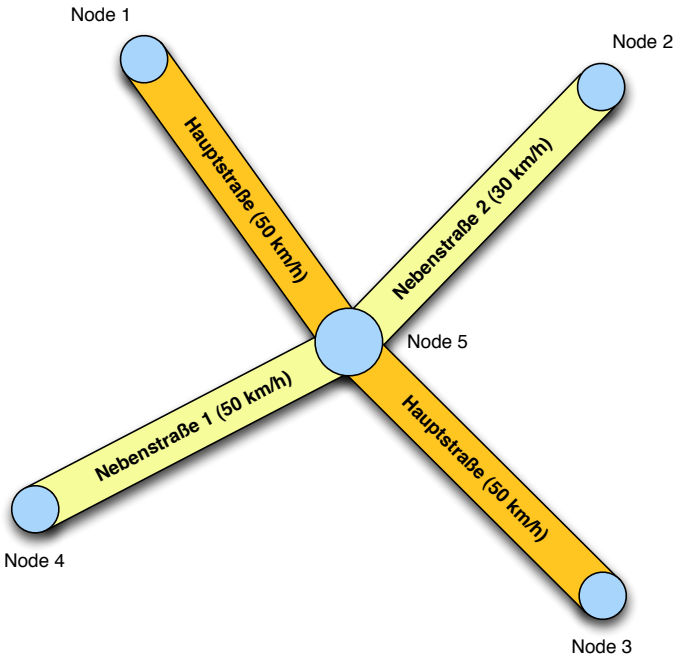


Abbildung 5.10.: *VANET-Simulator*: Beispiel der Ermittlung einer Vorfahrtsstraße

ihr, so kann es die Kreuzung ohne Geschwindigkeitsveränderung passieren.

2. Priorität: Biegt ein Fahrzeug von einer Vorfahrtsstraße auf eine Nebenstraße in Fahrtrichtung nach rechts ab, so hat es zwar Vorfahrt, muss jedoch für den Abbiegevorgang die Geschwindigkeit leicht verringern.
3. Priorität: Biegt ein Fahrzeug von einer Vorfahrtsstraße auf eine Nebenstraße in Fahrtrichtung nach links ab, so muss die Geschwindigkeit reduziert und auf den Gegenverkehr auf der Vorfahrtsstraße geachtet werden.
4. Priorität: Ein Fahrzeug möchte von einer Nebenstraße in Fahrtrichtung nach rechts auf eine Vorfahrtsstraße abbiegen. Dabei muss lediglich der Verkehr in die gewünschte Fahrtrichtung auf der Vorfahrtsstraße beachtet werden.
5. Priorität: Alle weiteren Fälle. Diese sind gegeben, wenn ein Fahrzeug von einer Nebenstraße kommt und eine Vorfahrtsstraße queren oder in Fahrtrichtung nach links auf sie einbiegen möchte. Es müssen die Verkehrsteilnehmer der Vorfahrtsstraße sowie aller Nebenstraßen beachtet werden.

Sobald sich ein Fahrzeug einer Kreuzung nähert, wird seine Priorität bestimmt. Erhält es *Priorität 1*, so kann es seinen Weg ungebremst fortsetzen; bei allen anderen Prioritäten muss die Geschwindigkeit auf 10 m/s reduziert werden. Wurde dem Fahrzeug *Priorität 2* zugewiesen, kann der Abbiegevorgang nun durchgeführt werden.

Fahrzeuge der *Priorität 3* reihen sich in eine Warteschlange (*junctionQueuePriority3*) ein, die der Prioritäten *4* und *5* in eine weitere (*junctionQueuePriority4*). In Warteschlangen eingereihte Fahrzeuge bremsen bei ihrer Annäherung an die Kreuzung weiterhin ab (gegebenenfalls bis zum Stillstand), bis sie von einer Funktion (*canPassJunction*) die Erlaubnis zum Passieren der Kreuzung erhalten. Dazu werden die folgenden Bedingungen überprüft:

- Fahrzeuge können nur in Abständen von $2,5$ Sekunden die Erlaubnis zur Einfahrt in die Kreuzung erhalten.
- Fahrzeuge dürfen nur dann in die Kreuzung einfahren, wenn ausreichend Platz auf den Vorfahrtsstraßen ist, so dass sich von dort annähernde Fahrzeuge nicht bremsen müssen. Bei den Prioritäten *3* und *4* muss dabei nur jeweils die relevante Fahrtrichtung untersucht werden; bei *Priorität 5* werden alle Richtungen betrachtet.
- Fahrzeuge aus der *junctionQueuePriority3* haben stets Vorrang gegenüber Fahrzeugen aus der *junctionQueuePriority4*.

Besitzt eine Kreuzung eine Ampel, so werden jeweils nur die Fahrzeuge mitberücksichtigt, die aktuell freie Fahrt erhalten.

5.3.5.4. Fahrzeugkommunikation

Neben der Bewegung der Fahrzeuge spielt natürlich ebenfalls ihre Kommunikation eine große Rolle. Im *VANET-Simulator* sind dafür aktuell die Übertragung von Beacons und Stauwarnungen (diese auf Basis eines generischen Nachrichtentyps) implementiert (vgl. Abschnitt 2.3.3). Die Sendereichweite der Fahrzeuge ist dabei eine Eigenschaft, die individuell festgelegt werden kann. Sämtliche Details der unteren Protokollschichten werden bei der Nachrichtenübermittlung ignoriert, ebenso wie Verschlüsselungs- oder Authentikationsaspekte. Weiterhin wird keinerlei denkbare Abschattung durch Gebäude oder andere Hindernisse berücksichtigt.

Beacons In den Einstellungen des Szenarios ist vorgegeben, wie groß das Intervall, in dem Beacons übertragen werden, ist. Jedes Fahrzeug sendet in diesem Rhythmus ein Beacon, das eine eigene Kennung, die eigene Position sowie die Geschwindigkeit enthält. Der Zeitpunkt des Sendens ist nicht für alle Fahrzeuge synchron, da dies zum einen nicht der Realität entsprechen würde und zum anderen die Rechenlast der Simulation nicht gleichmäßig über die Zeit verteilen würde. Stattdessen wird für alle Fahrzeuge ein Startzeitpunkt zwischen dem Zeitpunkt *null* und dem *Beacon-Intervall* anhand der Startposition ermittelt. Dieses Konzept kommt anstelle einer echt zufälligen Bestimmung zum Einsatz, damit der Zeitpunkt deterministisch ist

und somit bei einer erneuten Simulation mit den gleichen Parametern reproduziert werden kann.

Kommt in der Realität ein einfacher Broadcast beim Senden der Beacons zum Einsatz, so müssen die Empfänger dieser Nachricht in der Simulation dennoch bestimmt werden. Dies ist ein relativ aufwändiger Vorgang, da theoretisch alle Fahrzeuge daraufhin untersucht werden müssten, ob sie sich in dem runden Sendegebiet des übertragenden Fahrzeugs aufhalten. Um diesen Ablauf zu beschleunigen, werden die folgenden Schritte durchgeführt:

1. Berechnung einer rechteckigen Annäherung an die runde Sendereichweite und Ermittlung der vier Eckpunkte mit den Bezeichnungen *MapMinX*, *MapMaxX*, *MapMinY* und *MapMaxY*.
2. Auswahl der Regionen, die einen Schnittbereich mit diesem Rechteck aufweisen, da nur sie nach in Frage kommenden Fahrzeugen durchsucht werden müssen.
3. Erstellen einer Liste der möglichen Empfänger, die sich innerhalb des rechteckigen Schätzbereichs befinden, da diese Überprüfung mit Hilfe der Eckpunkte äußerst performant durchgeführt werden kann.
4. Berechnung des tatsächlichen Abstands der möglichen Empfänger von dem sendenden Fahrzeug mit Hilfe des Satzes des Pythagoras. Dabei wird auf die Berechnung der Wurzel verzichtet; stattdessen wird der Vergleich auf Basis der quadrierten Werte durchgeführt.

Wie bereits eingangs erwähnt, wird eine Verschlüsselung der Nachrichten nicht explizit abgebildet. Dennoch kann es bei der einen oder anderen Simulation notwendig sein, verschlüsselte Beacons als solche zu erkennen. Unter der Annahme, dass die eingesetzte Kryptographie von einem Angreifer nicht gebrochen werden kann, reicht es somit aus, lediglich die Information, dass eine Nachricht (sicher) verschlüsselt ist, zu übermitteln und dies bei der Auswertung der Daten entsprechend zu berücksichtigen.

Weitere Nachrichten Der *VANET-Simulator* bietet eine generische Implementierung verschiedener Nachrichten auf Basis einer abstrakten Klasse (*Message*) an. Diese Nachrichten, beziehungsweise die von ihnen abgeleiteten Implementierungen, werden gezielt an eine beliebige Anzahl an Fahrzeugen in einer bestimmten Region gesendet und dienen somit als Prototyp der Nachrichten von den Typen *Warnung und Alarmsignale* (vgl. Abschnitt 2.3.3).

Diese Nachrichten besitzen als zentrale Attribute eine Zielregion, für die sie bestimmt sind, sowie die Information, mit welcher Routingstrategie die Nachrichten an den

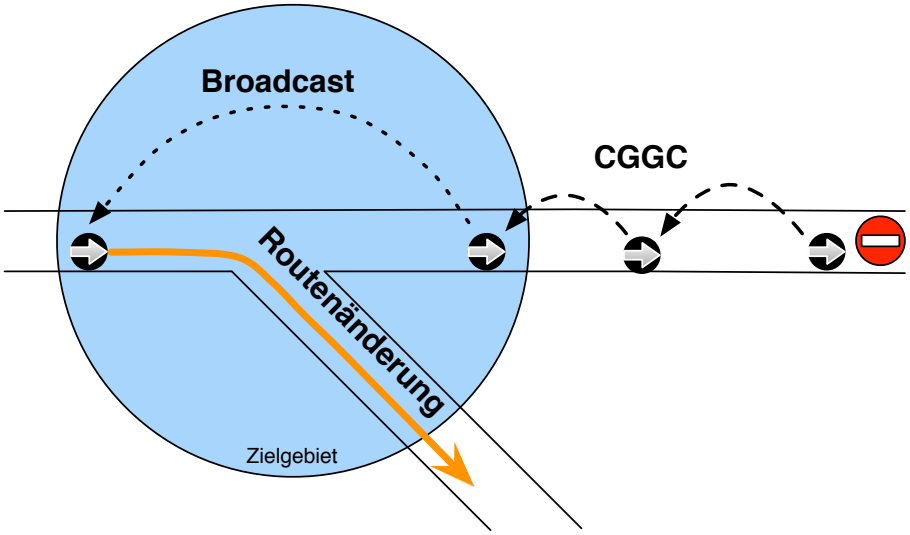


Abbildung 5.11.: *VANET-Simulator*: Stauwarnung

Zielort übermittelt und dort noch einmal gebroadcastet werden. Als Strategien werden aktuell *Cached Greedy GeoCast* (*GCCG*) und ein einfaches *Flooding* unterstützt (vgl. Abschnitt 2.2.3.2). Da die beiden Verfahren bereits in dem genannten früheren Abschnitt beschrieben wurden, soll auf sie an dieser Stelle, ebenso wie auf einige wenige umsetzungstechnische Besonderheiten, die bei Gruber [Gru08] nachgelesen werden können, nicht eingegangen werden.

Aktuell existiert nur die konkrete Implementierung eines Nachrichtentyps – der Stauwarnungen –, da Warnungen in der Regel in gefährlichen Situationen (wie beispielsweise bei Glätte und Unfällen) gesendet werden, die jedoch in der Simulationsumgebung momentan nicht abgebildet sind.

Ein Fahrzeug sendet eine Stauwarnung, wenn es für eine bestimmte Zeit auf einem Straßenabschnitt zum Stehen kommt. Die Nachricht wird an die letzte Kreuzung hinter der aktuellen Position gerichtet und dorthin in Abständen von 5 000 ms gesendet, bis sich das Fahrzeug wieder fortbewegt. An der Kreuzung besitzt die Nachricht einen Verbreitungsradius von 500 m, was es anderen Verkehrsteilnehmern ermöglicht, den Straßenabschnitt zu umgehen (vgl. Abbildung 5.11). Die Stauwarnung selbst besitzt nur eine begrenzte Gültigkeit; erhält ein Fahrzeug keine neuen Warnungen mehr, so wird die Straße wieder als frei angenommen.

Im Detail wird bei Erhalt einer Stauwarnung eine neue Routenberechnung durchgeführt, wobei die Länge des blockierten Straßenabschnitts für die Berechnung mit einem Malus von 20 km versehen und der Abschnitt somit, wenn möglich, gemieden wird. Würde dieser stattdessen in der Routenberechnung nicht mehr berücksichtigt werden, so könnte dies zu Problemen in Situationen führen, in denen er die einzige

mögliche Verbindung zum Ziel eines Fahrzeugs darstellt. In diesem Fall könnte keine gültige Route mehr gefunden werden.

5.3.5.5. Einfacher Angreifer

Als erster Proof-of-Concept wurde in den *VANET-Simulator* Funktionalität eingebracht, die es ermöglicht, ein einzelnes Fahrzeug lediglich anhand der gesendeten Beacons durch ein anderes Fahrzeug verfolgen zu lassen. Dieses verfolgende Fahrzeug bewegt sich nach Möglichkeit stets am Rande des Sendebereichs seines Opfers, um von ihm nicht gesehen zu werden. Sollte es das verfolgte Fahrzeug aus den Augen verlieren, so beschleunigt es und folgt einer angenommenen Route so lange, bis es sein Opfer wiedergefunden hat.

5.3.6. Road-Side-Units

Auch Road-Side-Units (*RSU*s), im Deutschen oft auch als Basisstationen bezeichnet, sind im *VANET-Simulator* als Teil des Szenarios implementiert. Sie besitzen ein Koordinatenpaar, das ihre Position definiert sowie eine individuelle Sendereichweite. In der aktuellen Version initiieren RSUs keinerlei Kommunikation; sie leiten jedoch erhaltene Nachrichten weiter und verhalten sich dabei exakt wie ein Fahrzeug. Sie dienen also in erster Linie dazu, eine Nachrichtenübermittlung auch dort sicherzustellen, wo die Verkehrsdichte dazu nicht ausreichen würde.

Ein weiterer Typ von RSUs, der ebenfalls im Szenario abgelegt ist, stellen die *ARSUs*, die von einem beschriebenen einfachen Angreifer (vgl. Abschnitt 5.3.5.5) kontrollierten RSUs, dar. Diese besitzen die gleichen Attribute wie reguläre RSUs, nehmen aber nicht an der Kommunikation teil. Es handelt sich dabei vielmehr um Empfänger, die das Gebiet, in dem ein einzelner Angreifer Nachrichten zur Kenntnis nehmen kann, über den Empfangsbereich seines Fahrzeugs hinaus erweitern.

5.3.7. Ereignisse

Zu bestimmten Zeitpunkten kann während einer Simulation ein Ereignis ausgeführt werden. Zu diesem Zweck existiert eine abstrakte Klasse *Event*, die geerbt werden muss und die Implementierung von drei Methoden vorschreibt. Details dazu können [Gru08] entnommen werden.

Aktuell ist nur die Blockade eines Straßenabschnitts sowie die Aufhebung einer solchen als Ereignis umgesetzt. Um einen Straßenabschnitt für Fahrzeuge ab einem bestimmten Zeitpunkt unpassierbar zu machen, kommt das *startBlockingEvent*

zum Einsatz. Sobald es aktiv ist, erzeugt es ein oder mehrere *LaneObjects* (vgl. Abschnitt 5.3.3.3), die unbeweglich sind und somit eine Spur an einer bestimmten Stelle blockieren. Fahrzeuge, die sich ihnen annähern, müssen abbremsen und anhalten, bis die Blockade wieder aufgehoben wurde. Diese bezieht sich immer auf einen bestimmten Punkt auf einem definierten Straßenabschnitt (ähnlich den Wegpunkten, vgl. Abschnitt 5.3.5.2), kann für eine oder beide Fahrtrichtungen Gültigkeit besitzen und dort eine oder mehrere Fahrspuren versperren.

Das Aufheben einer Blockade passiert mit Hilfe eines *stopBlockingEvents*, das sich immer auf exakt ein *startBlockingEvent* bezieht und sämtliche Hindernisse, die durch dieses verursacht werden, bei Aktivierung wieder entfernt. Diese Zweiteilung von Erzeugen und Aufheben einer Blockade wurde gewählt, um die Allgemeingültigkeit eines *Events* mit genau einem Aktivierungszeitpunkt beibehalten zu können.

Alle Ereignisse werden in der *EventList*, einer nach Zeit sortierten Warteschlange, gespeichert. Nach jedem Simulationsschritt wird diese Liste daraufhin überprüft, ob nun das nächste Ereignis ausgelöst werden muss. Ist dies der Fall, wird das Ereignis aktiviert und aus der Liste entfernt.

5.3.8. Konzepte zum Schutz der Privatsphäre

Das zu Beginn dieses Kapitels erläuterte Ziel der Simulation von *VANET*s ist die Untersuchung von Konzepten zum Schutz der Privatsphäre auf einer einheitlichen Basis. Zu diesem Zweck wurden diejenigen Ansätze, die als am vielversprechendsten in Hinblick auf das Erreichen einer Unverkettbarkeit zwischen neuen und alten Pseudonymen erscheinen, ausgewählt und, ebenso wie das selbstentwickelte Konzept *ProMix* (vgl. Abschnitt 4.6), implementiert.

5.3.8.1. Mix-Zone

Aus der Gruppe der räumlichen Ansätze wurde die Mix-Zone, wie sie von Plößl et al. beschrieben wurde (vgl. [Plö09, Sch08, SPF08]), ausgewählt, da dieser Ansatz unter den bestehenden Varianten den größtmöglichen Schutz der Privatsphäre verspricht (vgl. Abschnitt 4.5). Dies ist relativ einfach dadurch herzuleiten, dass ein komplettes Einstellen jeglicher Kommunikation trivialerweise mehr Schutz als reduzierte Kommunikation unter bestimmten Randbedingungen bieten muss.

In der Implementierung im *VANET-Simulator* sind Mix-Zonen stets an einem Knoten des Verkehrsnetzes verortet (idealerweise einem Kreuzungsmittelpunkt) und besitzen eine kreisrunde Ausdehnung mit einem definierten Radius. Bewegt sich ein Fahrzeug innerhalb dieses Gebiets, so sendet es keinerlei Nachrichten mehr aus und

wechselt sein Pseudonym. Erst nach dem Verlassen des Mix-Zonenbereichs nimmt das Fahrzeug das Senden wieder regulär – allerdings unter dem neuen Pseudonym – wieder auf. Optionalerweise ist es möglich, nur das Senden der Beacons zu unterdrücken und die anderen Nachrichten weiterhin zu übertragen, die ja möglicherweise eher selten und lediglich in besonderen Situationen gesendet werden.

Viele der in Abschnitt 4.2.2 beschriebenen Details zur Konfiguration und dem Betrieb der Mix-Zone werden – unter anderem aus Performanzgesichtspunkten – durch Abstraktion entfernt. So gibt es beispielsweise keine spezielle RSU an den Randbereichen der Zonen, die sich annähernde Fahrzeuge auf die Mix-Zone aufmerksam machen und Informationen über ihre Ausdehnung bereitstellen. Stattdessen besitzen alle Fahrzeuge jederzeit die vollständigen Informationen über die Konfigurationen aller Mix-Zonen.

Technisch gesehen wird bei jedem Simulationsschritt überprüft, ob sich ein Fahrzeug in einer Mix-Zone befindet, und diese Information wird zur weiteren Verwendung vorgehalten. Dabei wird über eine Einschränkung mit Hilfe der Regionen selbstverständlich darauf geachtet, dass nur in Frage kommende Paare aus Fahrzeugen und Mix-Zonen auf ihren Abstand hin untersucht werden müssen und dies nicht für alle Objekte vorgenommen werden muss.

5.3.8.2. ProMix-Zone

Als Erweiterung oder Spezialisierung der Mix-Zone kann die in Abschnitt 4.6 beschriebene ProMix-Zone verstanden werden. In ihrem Einzugsbereich wird die Kommunikation nicht vollständig eingestellt, sondern Beacons können weiterhin ausgetauscht werden. Allerdings werden sie in diesem Fall verschlüsselt über einen Proxy gesendet, der sie nur einer kleinen Gruppe zur Verfügung stellt. Die Empfänger werden mit Hilfe einer Heuristik ermittelt und als relevant eingestuft. Die Details dieses Konzepts sind in Abschnitt 4.6 ausführlich dargelegt.

Eine ProMix-Zone im *VANET-Simulator* ist grundsätzlich identisch zu einer soeben beschriebenen Mix-Zone, allerdings trägt sie als Attribut die Information in sich, dass eine verschlüsselte selektive Kommunikation stattfinden kann. Dies führt dazu, dass bei der Verarbeitung der Beacons eine andere Funktion verwendet wird, in der die Selektion stattfindet und zudem sämtliche innerhalb des ProMix-Bereichs gesendete Beacons die Information besitzen, dass sie verschlüsselt gesendet werden. Eine tatsächliche Verschlüsselung wird nicht durchgeführt, da dies zu einer sehr hohen Rechenlast führen würde und keinen echten Nutzen stiftet. Wird mit dem Simulator ein Angriff durchgeführt, so sind verschlüsselte Nachrichten – unter der Annahme, dass ein sicheres Kryptoverfahren zum Einsatz kommt – zu ignorieren.

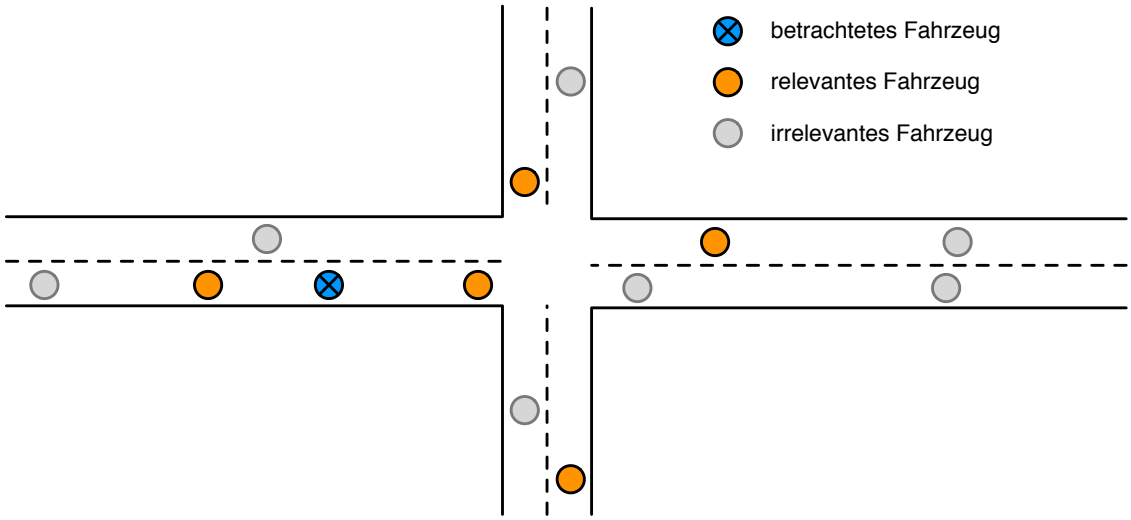


Abbildung 5.12.: *VANET-Simulator*: Relevante Fahrzeuge der implementierten ProMix-Zone

Für die Auswahl der relevanten Empfänger eines Beacons – also diejenigen, die die enthaltenen Informationen zur Kenntnis nehmen können – kommt aktuell eine sehr einfache Heuristik zum Einsatz: Die Fahrzeuge, die entweder dem betrachteten Teilnehmer direkt folgen, ihm direkt vorausfahren oder ihm entgegenkommen und ihn noch nicht passiert haben, erhalten die Nachrichten eines Teilnehmers. Alle weiteren Fahrzeuge werden ignoriert, also beispielsweise diejenigen, die sich auf einer parallel verlaufenden Straße oder einer Brücke über die Straße des betrachteten Teilnehmers bewegen. Ebenso können jene Fahrzeuge, die diesem zwar entgegenkommen, ihn aber bereits passiert haben, als irrelevant eingestuft werden.

Aus jeder dieser drei Gruppen wird das Fahrzeug ausgewählt, das dem Bezugsfahrzeug am nächsten ist und erhält von diesem somit alle Beacons. Diese Betrachtung wird natürlich kontinuierlich durchgeführt, so dass diese Informationen stets aktuell gehalten werden können.

Eine Besonderheit tritt in Kreuzungssituationen auf: Hier müssen die Fahrzeuge sämtlicher einmündender Straßenabschnitte separat betrachtet und jeweils die beschriebenen Gruppen gebildet werden, damit sichergestellt werden kann, dass beispielsweise alle auf die Kreuzung zufahrenden Fahrzeuge Informationen über ein auf oder bei der Kreuzung befindliches Fahrzeug erhalten (vgl. Abbildung 5.12).

5.3.8.3. Silent Period

Die Silent Period, ursprünglich von Huang et al. in [HMYS05] als *Random Silent Period* vorgestellt, verspricht in der Gruppe der zeitbasierten Ansätze den besten Schutz der Privatsphäre (vgl. Abschnitt 4.5). Allerdings wurde für den *VANET-Simulator* das Konzept dahingehend verändert, dass es eine global koordinierte Funkstille für alle Fahrzeuge gibt, die dann im Einzelnen nicht mehr zufällig initiiert wird. Durch diese Anpassung soll das größte Manko der *Random Silent Period* behoben werden: Im Grundkonzept entscheiden Fahrzeuge selbst über einen Pseudonymwechsel, wodurch dieser nur mit einer bestimmten Wahrscheinlichkeit in der Nähe eines anderen Teilnehmers stattfindet, der ebenfalls sein Pseudonym wechselt. Um jedoch Unsicherheit für einen Angreifer zu erzeugen, ist diese Nähe notwendig. Wechseln dagegen alle Fahrzeuge gleichzeitig ihr Pseudonym in einer koordinierten Funkstille, fällt diese Einschränkung weg. Mit dieser Designentscheidung soll das Konzept *Silent Period* in seiner bestmöglichen Ausprägung untersucht werden; der in der Realität entstehende Koordinierungsoverhead wird dabei ignoriert.

Die umgesetzte Silent Period ist sehr einfach zu koordinieren, da sie lediglich zwei Attribute besitzt: den Zeitraum, der zwischen zwei Funkstillen liegt, und deren Dauer. Mit Hilfe dieser beiden Parameter kann somit ein stetiger Wechsel zwischen einem Zeitraum, in dem Kommunikation stattfindet, und einem Zeitraum, in dem Funkstille gehalten wird, durchgeführt werden.

Bevor also in der Simulationskomponente die Nachrichten erzeugt und übermittelt werden, wird überprüft, in welchem der beiden Zeiträume man sich gerade befindet. Abhängig davon wird entschieden, ob der Prozess nun tatsächlich durchgeführt oder schlichtweg übersprungen wird, bis die Silent Period vorüber ist.

5.3.8.4. SLOW

Auch aus der Gruppe der benutzerzentrierten Ansätze wurde der vielversprechendste – *SLOW* von Buttyán et al. (vgl. [BHWW09]) – ausgewählt und implementiert. Das Prinzip, dass Fahrzeuge ihre Kommunikation einstellen und das Pseudonym wechseln, wenn sie sich langsamer als eine Schwellengeschwindigkeit fortbewegen, lässt sich sehr einfach umsetzen. Demzufolge wird zur Konfiguration von *SLOW* ein Parameter benötigt, der die Geschwindigkeit festlegt, unterhalb derer Fahrzeuge nicht mehr senden. Um jedoch zu vermeiden, dass Fahrzeuge, die im fließenden Verkehr nur für Sekundenbruchteile die Grenzgeschwindigkeit unterschreiten, einen unnötigen, weil ineffektiven, Pseudonymwechsel durchführen, wird ein weiterer Parameter verwendet, der bestimmt, wie lange eine Funkstille bereits andauern muss, um einen Pseudonymwechsel auszulösen. Dies entspricht somit auch den Überlegungen von Buttyán et al.

Fahrzeuge prüfen somit bevor sie kommunizieren, ob ihre aktuelle Geschwindigkeit auch tatsächlich größer als die Grenzgeschwindigkeit ist; andernfalls findet keine Nachrichtenübermittlung statt. Bewegt sich ein Fahrzeug ausreichend langsam um eine Funkstille zu wahren, so wird eine Zeitspanne abgewartet, ob die Funkstille lange genug für einen Pseudonymwechsel gehalten werden kann. Ist dies der Fall, so nimmt das Fahrzeug nach Erreichen einer höheren Geschwindigkeit die Übermittlung von Beacons unter einem neuen Pseudonym wieder auf.

5.3.9. Graphische Benutzeroberfläche

Der *VANET-Simulator* bietet neben einer terminalbasierten Interaktionsmöglichkeit zur möglichst performanten Durchführung auch umfangreicher und langwieriger Simulationen eine umfassende graphische Benutzeroberfläche an. Diese stellt neben einer detaillierten Darstellung der statischen und dynamischen Objekte der Simulation und Informationsfenstern über ihre Eigenschaften einen umfangreichen Editor zur einfachen Erstellung und Bearbeitung von Karten und Szenarien zur Verfügung.

5.3.9.1. Konzept der graphischen Oberfläche

Die graphische Benutzeroberfläche des *VANET-Simulators* wurde mit dem Ziel gestaltet, dass eine Bedienung der Anwendung nach kürzester Einarbeitungszeit möglich ist (vgl. Abbildung 5.13). Dabei wurde eine Zweiteilung des Hauptfensters gewählt: Der größte Teil wird stets durch eine Ansicht der Karte mit einer Einblendung der bereits verstrichenen Simulationszeit eingenommen (1), die Menüs sind in einer Seitenleiste am rechten Rand untergebracht und können über Karteireiter angewählt werden (2). Über die rechte Seitenleiste kann der Kartenausschnitt vergrößert und verkleinert sowie bewegt werden (3); im Hauptbereich ist dabei stets ein Maßstab eingeblendet (4). Alternativ ist Zoomen per Mausrad und ein Bewegen des Bildausschnitts per Drag-und-Drop möglich. Weitere Schaltflächen bieten Optionen für die Steuerung des Simulationsablaufs (5). Fahrzeuge und die Straßenabschnitte, auf denen sie sich gerade befinden, können ausgewählt werden (6), um in der Seitenleiste Detailinformationen zu ihnen anzuzeigen (7). Auch andere Objekte wie beispielsweise blockierte Straßenabschnitte werden stets angezeigt (8).

5.3.9.2. Darstellung der statischen und dynamischen Objekte

Java ermöglicht seit einigen Jahren mit Hilfe der *Java2D API* [32] sehr performante, hardwarebeschleunigte Grafiken darzustellen. Daher wird darauf in der graphischen

[32] http://java.sun.com/products/java-media/2D/index.jsp, abgerufen am 8.5.2012

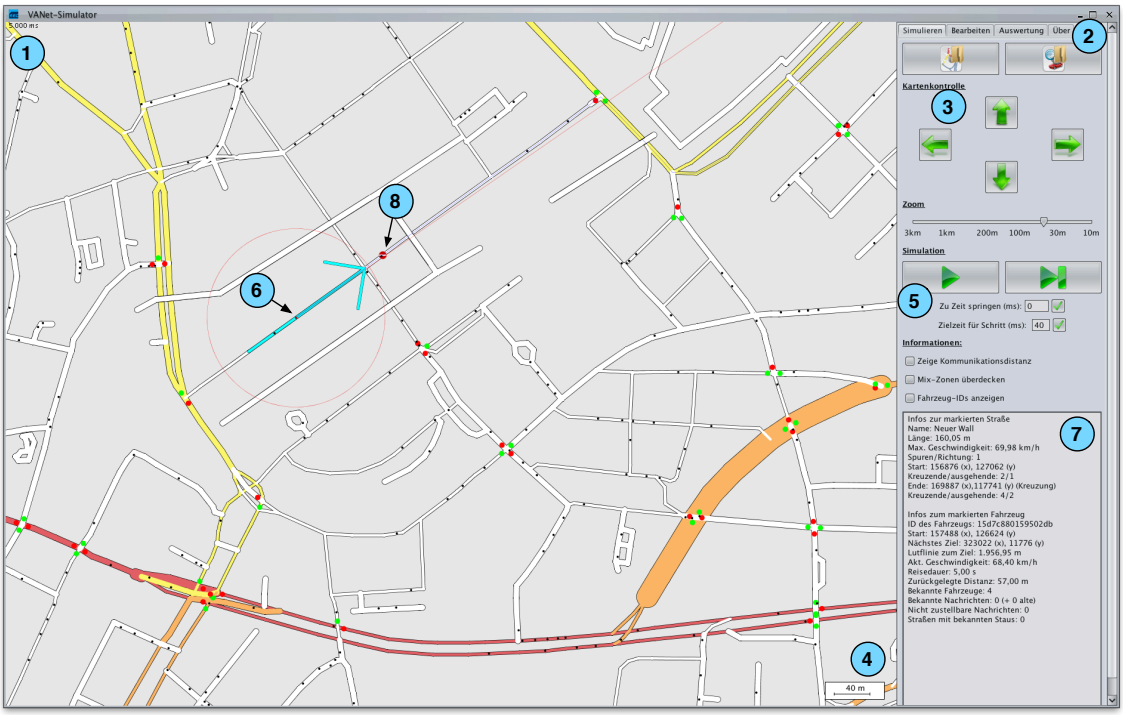


Abbildung 5.13.: *VANET-Simulator*: Graphische Oberfläche

Benutzeroberfläche des *VANET-Simulators* für eine effiziente Darstellung der Simulation zurückgegriffen.

Um die Geschwindigkeit weiter zu verbessern, werden – ähnlich der internen Struktur des *VANET-Simulators* – statische und dynamische Objekte unterschieden. Die statischen Objekte sind dabei diejenigen, die sich im Laufe eines Simulationsdurchgangs nicht verändern und somit nicht ständig neu berechnet werden müssen. In der aktuellen Version sind dies ausschließlich die Straßen; also die Bestandteile der Karte (vgl. Abschnitt 5.3.3). Die dynamischen Objekte umfassen somit unter anderem die Fahrzeuge, RSUs und Ereignisse – also alle Elemente, die laufend aktualisiert werden müssen.

Demzufolge wird die Ansicht des Straßenabschnitts nur dann neu berechnet, wenn der angezeigte Ausschnitt verschoben oder gezoomt wird. Andernfalls kann die zuvor gerenderte Ansicht zwischengespeichert und wiederverwendet werden. Dies ist besonders wichtig, da das Rendern der Anzeige – je nach Vergrößerung des Ausschnitts – bis zu einige hundert Millisekunden in Anspruch nehmen kann. Um die Verschiebung zu optimieren, wurde noch eine weitere Performanceoptimierung umgesetzt: Verwendet ein Benutzer die Schaltflächen oder Cursortasten, um den angezeigten Ausschnitt zu bewegen, so wird dieser exakt einen halben Bildschirm weit verscho-

ben. Somit kann die weiterhin angezeigte Hälfte der bereits berechneten Darstellung wiederverwendet und die Rechenlast somit auf annähernd die Hälfte gesenkt werden.

Der große Aufwand bei der Berechnung des angezeigten Kartenausschnitts beruht auf der relativ detaillierten Darstellung des Straßennetzes sowie der Komplexität, die relevanten Straßenabschnitte und ihre genauen Lagen aus der Gesamtmenge an Straßen und Kreuzungspunkten zu extrahieren. Straßen werden in Abhängigkeit ihrer Klassifikation eingefärbt und entsprechend der Anzahl an Fahrspuren, die sie besitzen, gezeichnet und mit einer schwarzen Konturlinie umgeben. An Kreuzungspunkten müssen die Straßenabschnitte natürlich ineinander übergehen und sollten keine abrupten Kanten aufweisen. Besonders aufwändig ist die Darstellung von Brücken, da hier eine eindeutige Festlegung getroffen werden muss, welche Straße die andere quert und somit deckend platziert werden muss.

Weiterhin ist der Detailgrad der Darstellung von der Größe des betrachteten Kartenausschnitts abhängig. Wenn ein kleiner Ausschnitt gewählt wird, so sind in der Regel nur relativ wenige Objekte zu sehen und die Darstellung kann mit Techniken wie Antialiasing verbessert werden. Ist dagegen ein sehr großer Ausschnitt mit hunderten von Straßenabschnitten sichtbar, so wird selbst auf die schwarze Konturlinie der Straßen verzichtet, um die Darstellung zu beschleunigen. In dieser Zoomstufe wäre es zudem ohnehin nicht möglich, derart hoch aufgelöste Details zu erkennen.

Die dynamischen Objekte müssen nach jedem Simulationsschritt neu gezeichnet werden, um die aktualisierte Situation adäquat darzustellen. Dabei ist zu beachten, dass ein Zeichenvorgang stets nur nach Abschluss sämtlicher Berechnungen eines Schritts stattfinden darf, da ansonsten möglicherweise ein inkonsistenter Zustand abgebildet werden könnte. Da diese sehr häufig durchgeführte Darstellung äußerst zeitkritisch ist – es können in der Zwischenzeit keine anderen Kalkulationen durchgeführt werden – wurde sie sehr stark geschwindigkeitsoptimiert. So werden beispielsweise in der aktuellen Version alle Fahrzeuge in Form eines Kreises mit dem Durchmesser von $2,5\,\mathrm{m}$ dargestellt, da ansonsten eine Berücksichtigung der Ausrichtung den Zeichenvorgang deutlich verlangsamen würde. Optionale Informationen, die in ihrer Berechnung sehr aufwändig sind (z. B. die Darstellung der Kommunikationsreichweite aller Fahrzeuge), können jederzeit (und sollten wenn möglich) ausgeblendet werden, um den Zeichenprozess zu beschleunigen.

Sind die einzelnen Zeichenschritte definiert, ist es unerlässlich, dass die Reihenfolge ihrer Ausführung berücksichtigt wird, da auf diese Weise festgelegt werden kann, welche Objekte andere in der Darstellung verdecken dürfen. So müssen Fahrzeuge beispielsweise unbedingt nach dem Straßensystem gerendert werden, da nur so sichergestellt werden kann, dass sie sich optisch auf und nicht hinter der Straße bewegen.

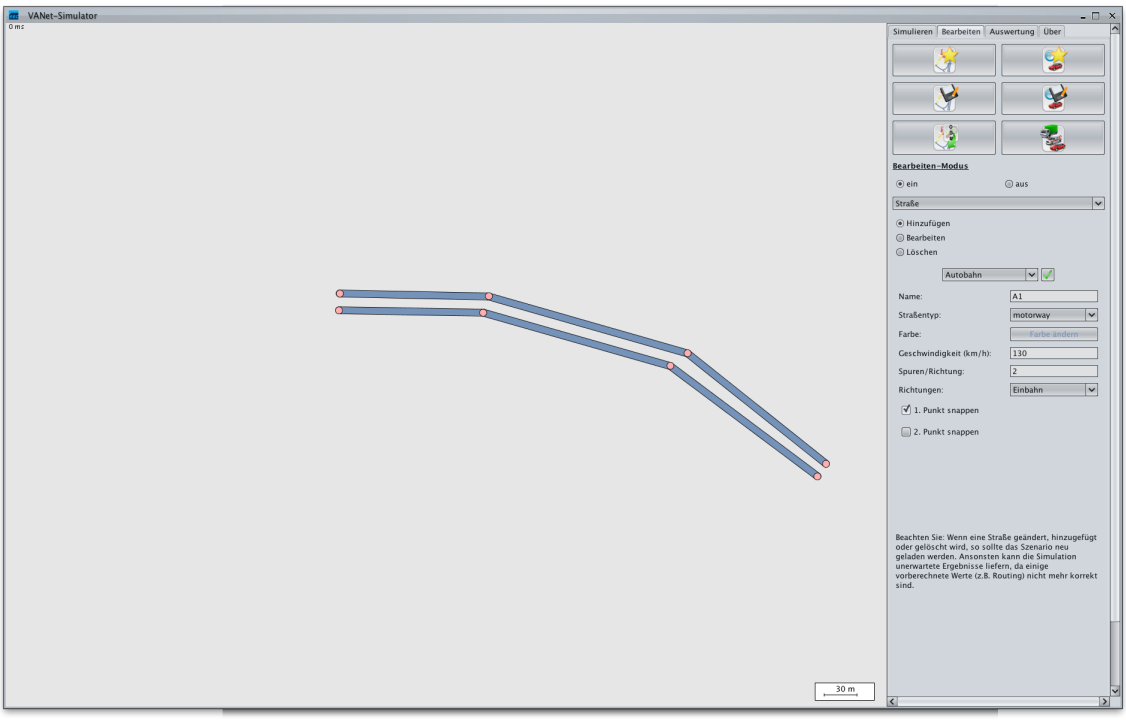
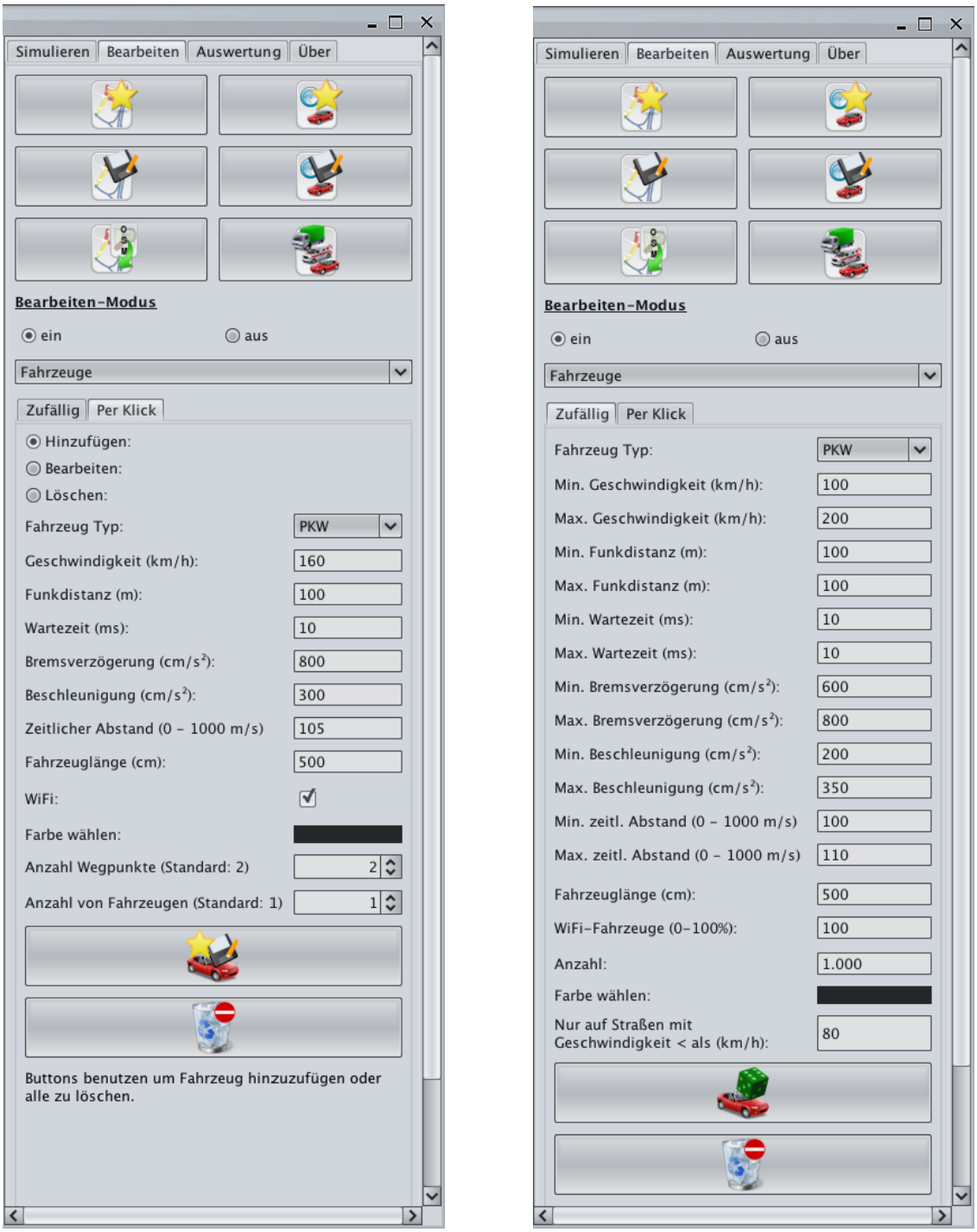


Abbildung 5.14.: *VANET-Simulator*: Erstellen von Karten

5.3.9.3. Szenarioeditor

Eine zentrale Komponente der graphischen Oberfläche des *VANET-Simulators* stellt der Szenarioeditor dar. Mit seiner Hilfe ist es möglich, vollständige Straßensysteme und Szenarien zu erstellen und anzupassen. Dabei können mit wenigen Mausklicks nahezu alle Parameter der Simulationen über eine übersichtliche und einfach zu bedienende Benutzerschnittstelle konfiguriert sowie erstellte Karten und Szenarien gespeichert bzw. geladen werden. Auch der OpenStreetMap-Import ist dort verortet.

- *Generelle Einstellungen*: Mit Hilfe der Oberfläche lassen sich die zentralen Optionen der Simulation konfigurieren, so beispielsweise das Intervall zwischen zwei Beacons, die Optimierung von Fahrtdauer oder Fahrtstrecke als Basis zur Berechnung der Routen der Fahrzeuge und ähnliche Einstellungen.
- *Karte und Straßennetz*: Der integrierte Szenarioeditor ermöglicht es, vollständig neue Straßennetze auf einer leeren Karte anzulegen. Dabei können sämtliche Parameter der Straßenabschnitte frei gewählt und Letztere anschließend mit zwei einfachen Mausklicks auf der Karte platziert werden (vgl. Abbildung 5.14). Damit es einfach und unkompliziert zu einer lückenlosen Verbindung zweier Abschnitte kommt, wurde ein sogenanntes *Snapping* ein-



(a) Manuelle Fahrzeugerstellung

(b) Automatische Fahrzeugerstellung

Abbildung 5.15.: *VANET-Simulator*: Erzeugen von Fahrzeugen

geführt, das bewirkt, dass ein Klick in die Nähe eines bereits vorhandenen Knotens diesen als Start- oder Endknoten des neu anzulegenden Straßenabschnitts verwendet. Weiterhin ist es selbstverständlich möglich, importierte Karten ebenfalls zu editieren, indem vorhandene Straßenabschnitte angepasst oder gelöscht sowie neue Abschnitte hinzugefügt werden können.

- *Ampeln*: Die durch einen Import erstellten Ampeln lassen sich in ihren Schaltzeiten manipulieren und können entfernt werden. Das Hinzufügen von neuen Ampeln mit entsprechenden Einstellungen ist ebenfalls möglich.

- *Fahrzeuge*: Vor der Simulation ist es offensichtlich notwendig, ausreichend viele Fahrzeuge zu erzeugen, die jeweils einen Start- und einen Zielort sowie möglicherweise weitere Zwischenstationen besitzen. Zudem verfügt jedes Fahrzeug über eine Reihe von Eigenschaften, die individuell festgelegt werden können. Der Szenarioeditor ermöglicht in einem zweistufigen Konzept zuerst das Konfigurieren und anschließend das Platzieren dieses Fahrzeugs per Mausklick auf der Karte (vgl. Abbildung 5.15a). Mit einer Reihe von Klicks kann die Route des Fahrzeugs festgelegt werden. Möchte man jedoch größere Regionen mit ausreichend vielen Fahrzeugen bestücken, so ist dies von Hand realistischerweise nicht mehr zu bewerkstelligen. Daher bietet der Szenarioeditor die Möglichkeit, eine definierte Anzahl an Fahrzeugen mit jeweils einem zufälligen Start- und Zielort sowie zufälligen Eigenschaften, deren Minimal- und Maximalwert festgelegt wurde, zu erzeugen (vgl. Abbildung 5.15b). Auf diese Weise kann innerhalb von Sekunden eine Vielzahl an Fahrzeugen in der Karte platziert werden. Dabei wird sichergestellt, dass die Fahrzeuge auch tatsächlich auf gültigen Straßenabschnitten erzeugt werden und mindestens eine gültige Route von ihrem Start- zu ihrem Zielort existiert.

- *RSUs*: Auch Road-Side-Units können per Klick auf der Karte platziert werden, nachdem ihre individuelle Funkreichweite festgelegt wurde.

- *Angreifer*: Der in Abschnitt 5.3.5.5 beschriebene einfache Angreifer lässt sich ebenfalls über den Szenarioeditor konfigurieren und, zusammen mit eventuell von ihm kontrollieren RSUs, platzieren. Weiterhin ist es möglich, das von ihm angegriffene und verfolgte Fahrzeug aus den zuvor erzeugten Fahrzeugen auszuwählen.

- *Ereignisse*: Der Szenarioeditor stellt eine Zeitleiste zur Verfügung, in der anstehende Ereignisse eingefügt, konfiguriert und entfernt werden können. Abbildung 5.16a zeigt dies beispielhaft anhand zweier Blockadeereignisse, von denen eines bereits nach kurzer Zeit wieder aufgehoben wird. Auch hier funktioniert die Platzierung der Blockaden sehr einfach über einen Klick auf die gewünschte Stelle der Karte.

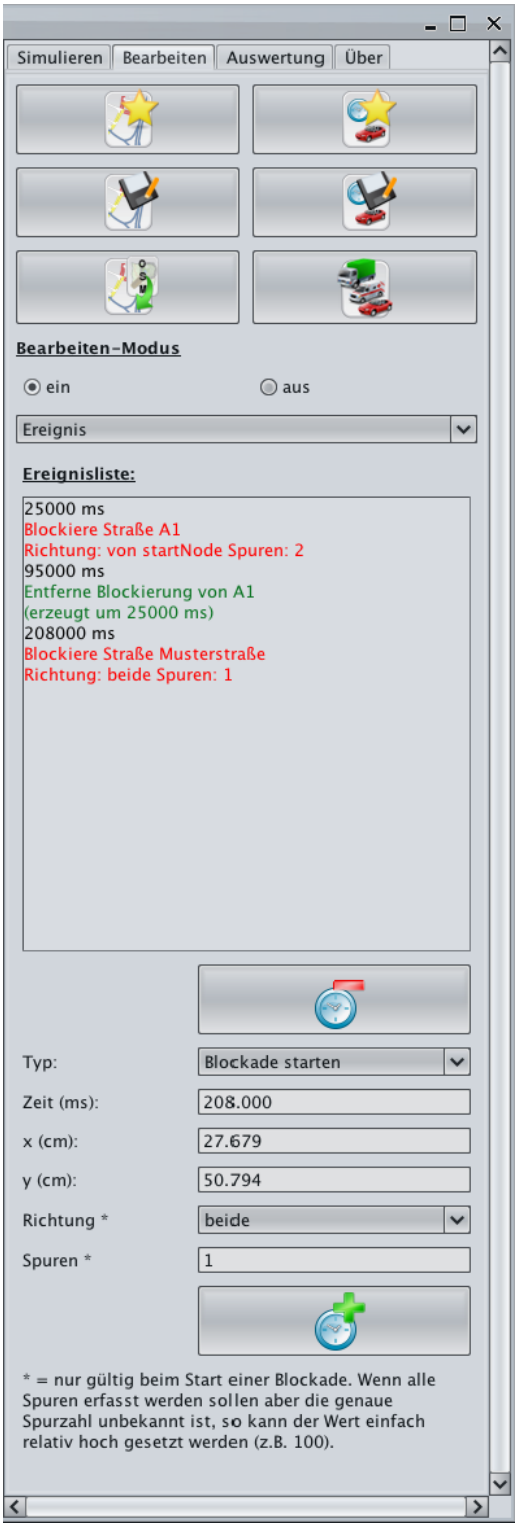
- *Techniken zum Schutz der Privatsphäre*: Wie bereits in Abschnitt 5.3.8 ausführlich dargelegt, wurden mehrere Konzepte zum Schutz der Privatsphäre von Teilnehmern zur Analyse in den VANET-Simulator integriert. Der Szenarioeditor bietet über seine Benutzeroberfläche die Möglichkeit, komfortabel entsprechende Einstellungen vorzunehmen (vgl. Abbildung 5.16b).
- *Logging*: Da konkrete komplexe Angriffe bisher mit Hilfe externer Werkzeuge auf Basis von detailliert protokollierten Simulationen stattfanden, wurde eine ausführliche Konfigurationsmöglichkeit der Logging-Funktionalität zur Erstellung der Protokolle in den Szenarioeditor integriert.

Insgesamt bietet der Szenarioeditor damit die Möglichkeit, sämtliche relevanten Einstellungen der Karte und der darauf basierenden Simulation komfortabel und einfach durchzuführen.

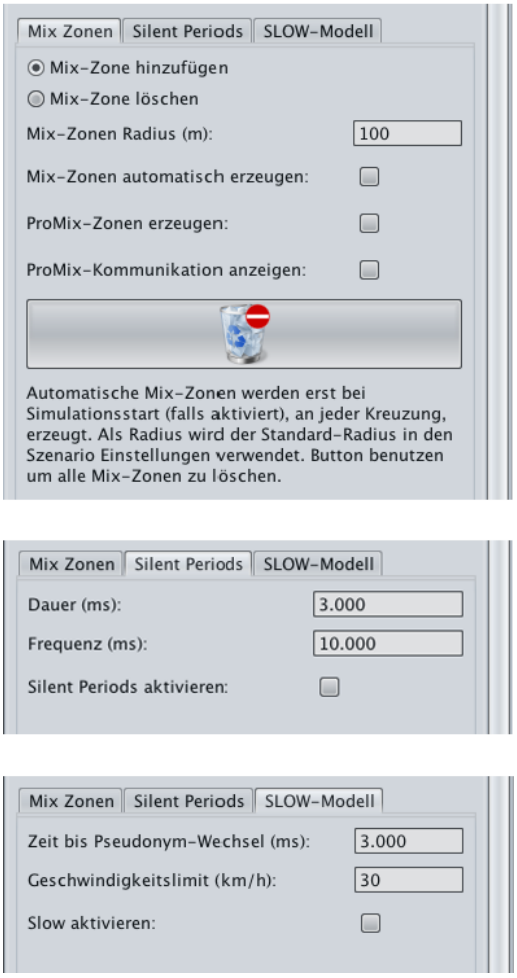
5.3.10. Simulation

Die Durchführung der eigentlichen Simulation wurde naheliegenderweise ebenfalls geschwindigkeitsoptimiert und auch für moderne Mehrkernprozessoren angepasst. Grundsätzlich existiert ein *SimulationMaster*, der den gesamten Ablauf der Simulation koordiniert und kontrolliert und zudem für alle Aufgaben zuständig ist, die sich nicht parallelisieren lassen. Neben ihm existieren eine Reihe von *WorkerThreads*, die nebenläufig viele Tätigkeiten während eines Simulationsschritts durchführen und regelmäßig synchronisiert werden. Abbildung 5.17 zeigt schematisch den Ablauf der Simulation: Eingangs erzeugt der *SimulationMaster* eine Reihe von Threads und initialisiert sie. Besitzt das Gastsystem des Simulators nur einen Prozessorkern, so wird auch nur ein *WorkerThread* erzeugt, da Nebenläufigkeit in diesem Szenario zu keinem Vorteil führt. Auf Mehrkernprozessoren dagegen werden doppelt so viele *WorkerThreads* erzeugt wie Kerne vorhanden sind, damit Laufzeitunterschiede zwischen den Threads nicht so stark ins Gewicht fallen, da weniger Leerlaufzeit auf die einzelnen Prozessorkerne entfällt. Die vorhandenen Regionen (vgl. Abschnitt 5.3.3.2) werden gleichmäßig auf die *WorkerThreads* verteilt, wobei mangels Teilbarkeit einzelne Threads bis zu eine Region mehr als andere Threads erhalten können (vgl. Abbildung 5.18). Jeder der erzeugten *WorkerThreads* führt sofort einige Initialisierungsschritte durch, wie beispielsweise das Berechnen von Kreuzungen (vgl. Abschnitt 5.3.5.3).

Um die Threads zu synchronisieren, werden nach der Durchführung von Aufgaben, die parallel ausgeführt werden können, sogenannte *CyclicBarriers* der Java-API eingezogen, die Threads dazu zwingen, mit ihrer Ausführung erst fortzufahren, wenn sie das Signal dazu bekommen. So steht auch zu Beginn im Anschluss an die Initialisierungsphase eine Barriere.



(a) Ereignisse



(b) Techniken zum Schutz der Privatsphäre

Abbildung 5.16.: *VANET-Simulator*: Weitere Konfigurationsmöglichkeiten

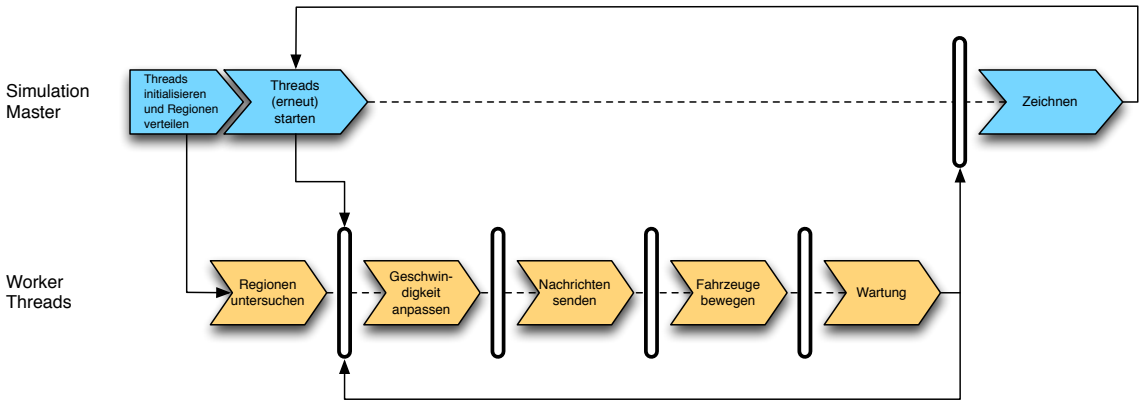


Abbildung 5.17.: *VANET-Simulator*: Ablauf der Simulation

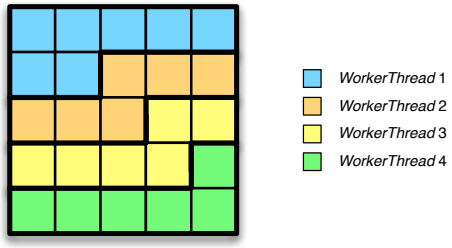


Abbildung 5.18.: *VANET-Simulator*: Beispielhafte Aufteilung der 25 Regionen einer Karte auf *WorkerThreads* bei zwei Prozessorkernen (nach [Gru08])

Das Signal zum Start des Simulation wird von dem *SimulationMaster* an die *WorkerThreads* gegeben, sobald diese alle initialisiert sind. Nun kann die parallele Ausführung beginnen. Jeder *WorkerThread* durchläuft zu Beginn den ersten Teil des Mikroverkehrsmodells für alle Fahrzeuge, um die Geschwindigkeiten anzupassen. Nach einer *Barrier* zur Synchronisation bearbeiten alle Threads den Nachrichtenversand der in ihren Regionen verorteten Fahrzeuge, worauf erneut eine Barriere folgt. In der nächsten Phase wird der zweite Teil des Mikroverkehrsmodells durchgeführt, die Bewegung der Fahrzeuge. Auch im Anschluss an diesen Schritt folgt eine Barriere, bevor abschließende Wartungsaufgaben wie beispielsweise die Überprüfung und gegebenenfalls das Schalten von Ampeln erfolgen. Nachdem auch dieser Schritt durchlaufen ist, ist die eigentliche Simulation für diese Runde abgeschlossen und die *WorkerThreads* warten auf das Signal zur Bearbeitung einer weiteren Runde. Bevor es durch den *SimulationMaster* erteilt wird, muss dieser noch sämtliche Zeichenaufgaben übernehmen und somit für die Darstellung des Simulationsschritts sorgen. Diese kann nicht an die Threads verteilt werden, da so möglicherweise inkonsistente Zustände ausgegeben würden.

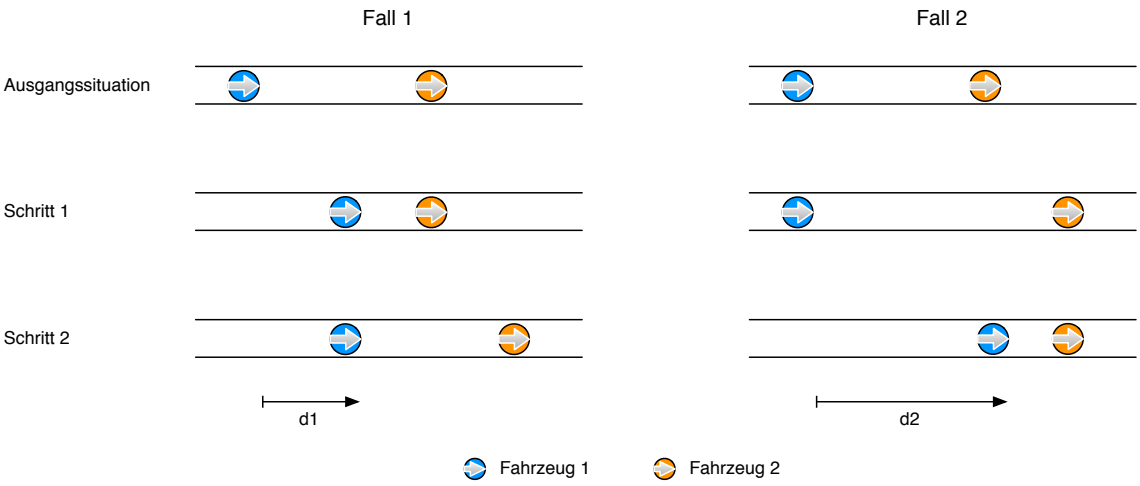


Abbildung 5.19.: *VANET-Simulator*: Mögliche Inkonsistenz bei der Simulationsdurchführung ohne Barrieren (nach [Gru08])

Die Barrieren sind nötig, da die Ausführungszeit und -reihenfolge von Threads nicht deterministisch ist und somit bei einer vollständigen Durchführung einer Simulationsrunde ohne Synchronisationspausen Situationen entstehen können, die bei einer erneuten Durchführung nicht reproduzierbar sind. Dies soll im Folgenden an dem in Abbildung 5.19 gezeigten Beispiel deutlich gemacht werden.

In diesem Beispiel nähert sich im Übergangsbereich zwischen zwei Regionen das schnellere *Fahrzeug 1* (blau) von hinten dem langsameren *Fahrzeug 2* (orange), wobei beide von unterschiedlichen *WorkerThreads* bearbeitet werden. Erfolgt nun zuerst eine Bearbeitung von *Fahrzeug 1* (*Fall 1*), so wird *Fahrzeug 2* als Hindernis wahrgenommen und das Fahrzeug muss seine Geschwindigkeit deutlich reduzieren, bevor es bewegt wird (*d1*). Anschließend wird *Fahrzeug 2* in einem weiteren Schritt ebenfalls bewegt.

Würde aber wie in *Fall 2* dargestellt zuerst *Fahrzeug 2* bewegt werden, so müsste *Fahrzeug 1* seine Geschwindigkeit nicht so stark reduzieren und könnte eine größere Distanz (*d2*) zurücklegen. Eine Vorhersagbarkeit darüber, welcher Fall tatsächlich eintreten wird, ist unmöglich. Somit kann eine erneute Durchführung der Simulation die jeweils andere Situation hervorrufen.

Diese Inkonsistenzen werden mit dem Einsatz der Barrieren erfolgreich verhindert, wodurch jede Simulation mit der gleichen Startkonfiguration vollständig reproduzierbar ist.

5.3.11. Fazit

Mit dem in Eigenentwicklung entstandenen *VANET-Simulator* konnte eine Umgebung geschaffen werden, die die Simulation von Straßenverkehr und dem darauf aufbauenden Ad-hoc-Fahrzeugnetz ermöglicht. Die Simulation selbst läuft dabei sehr performant und ist auch für größere Straßensysteme auf handelsüblichen Desktop-PCs möglich. Zudem gibt es keine zufälligen Elemente in der Durchführung, so dass identische Eingangsparameter stets zu einem reproduzierbaren Ergebnis führen. Sollten randomisierte Entscheidungen notwendig sein, um das Verhalten der Fahrzeuge nicht vorhersagbar zu gestalten, so kommt stets Pseudozufall zum Einsatz, dessen Startwerte mit Hilfe von Startkoordinaten der Fahrzeuge oder ihren eindeutigen Kennungen erzeugt werden.

Um eine möglichst realistische Verkehrssimulation zu ermöglichen, hat eine intensive Auseinandersetzung mit der Literatur im Bereich der Mikroverkehrsmodelle stattgefunden. Die dort entstandenen Erkenntnisse sind in die Entwicklung des Simulators eingeflossen. Der Import von Karten des Projekts OpenStreetMap ermöglicht die Simulation von Verkehr auf realistischen Straßensystemen und trägt somit zu einer verbesserten Übertragbarkeit der Ergebnisse in die reale Welt bei. Beim Import wird darauf geachtet, gängige Konventionen der Kartenprojektion und die in diesem Forschungsbereich bekannten Probleme zu berücksichtigen und somit eine standardkonforme Abbildung zu realisieren.

Die innere Struktur des Simulators ermöglicht es, auf einfache Art und Weise neue Funktionalitäten zu ergänzen oder andere Anpassungen vorzunehmen. Für die im Folgenden beschriebenen Untersuchungen wurden bereits drei Konzepte zum Schutz der Privatsphäre in Ad-hoc-Fahrzeugnetzen implementiert und geeignete Logging-Mechanismen integriert. Somit können aufwändige Analysen wie beispielsweise Angriffe auf diese Schutzmaßnahmen mit Hilfe externer Tools durchgeführt werden. Zugleich liefern diese Implementierungen ein anschauliches Beispiel, wie sich beliebige Aspekte der Ad-hoc-Fahrzeugnetze umsetzen und in die Simulation integrieren lassen.

Insgesamt bietet der aus der Eigenentwicklung hervorgegangene *VANET-Simulator* momentan eine der umfangreichsten Simulationsumgebungen für die Untersuchung von Schutzmaßnahmen gegen Verfolgung in Ad-hoc-Fahrzeugnetzen.

5.4. Untersuchung von Konzepten zum Schutz der Privatsphäre

5.4.1. Vorüberlegungen

Mit Hilfe des entwickelten *VANET-Simulators* sollen die in Kapitel 4 durchgeführten analytischen Untersuchungen um empirische Betrachtungen ergänzt werden. Einige der Ergebnisse wurden auch bereits in [Tom12] beschrieben. Dazu wurden – wie bereits in Abschnitt 5.3.8 beschrieben – die drei in Hinblick auf den Schutz der Privatsphäre effektivsten Konzepte in besonders datenschutzfreundlichen Varianten implementiert (vgl. Abschnitt 4.5). Die verkehrssicherheitstechnischen Implikationen, die aus einer derart restriktiven Konfiguration folgen, werden ignoriert, da mit den anstehenden Untersuchungen eine obere Schranke für den möglichen Schutz mit bekannten Techniken ermittelt werden soll.

Die in Abschnitt 4.6 vorgestellte *ProMix-Zone* wird in diesem Abschnitt nicht direkt untersucht (obwohl eine Implementierung besteht), da der Schutz vor einem Angreifer, der in die Zone einfährt und dem es dort gelingt, die Nachrichten seines Opfers zu erhalten und es somit verfolgen zu können, nicht möglich ist. Für einen außenstehenden Beobachter wiederum verhält sich die *ProMix-Zone* exakt wie die simulierte *Mix-Zone*. Betrachtet man also Letztere, erhält man automatisch eine obere Schranke für das mit Hilfe einer *ProMix-Zone* erreichbare Schutzniveau.

Die drei zu untersuchenden Konzepte – *Mix-Zone*, *Silent Period* und *SLOW* – werden mit plausiblen Einstellungen parametrisiert und einer Reihe von Simulationen unterworfen. Die Ausgangssituation (Start- und Zielorte der Fahrzeuge) wird dabei zufällig generiert. Um eventuell durch diese Methode entstehende Ausreißer zu kompensieren, wird jede einzelne Konfiguration mehrfach – fünffach – simuliert; es werden Mittelwerte der Ergebnisse verwendet.

Die Simulationen sollen sich auf den urbanen Raum beschränken. Bereits hier gibt es gravierende Unterschiede in den inneren Strukturen der Großstädte. Manche weisen ein akkurates, regelmäßiges Straßennetz auf, das aus Stadtplanungen im 19. und 20. Jahrhundert hervorgegangen ist (z. B. Manhattan in New York City, Brasilia oder Teile von Barcelona), wohingegen andere (insbesondere recht alte) Städte ein scheinbar chaotisches Straßensystem, das aus alten Gassen und Wegen hervorging, besitzen.

Um diesen Unterschieden Rechnung zu tragen, werden die Simulationen auf den Karten (diese stammen aus dem Projekt OpenStreetMap, vgl. Abschnitt 5.3.3.4) dreier unterschiedlicher Großstädte durchgeführt:

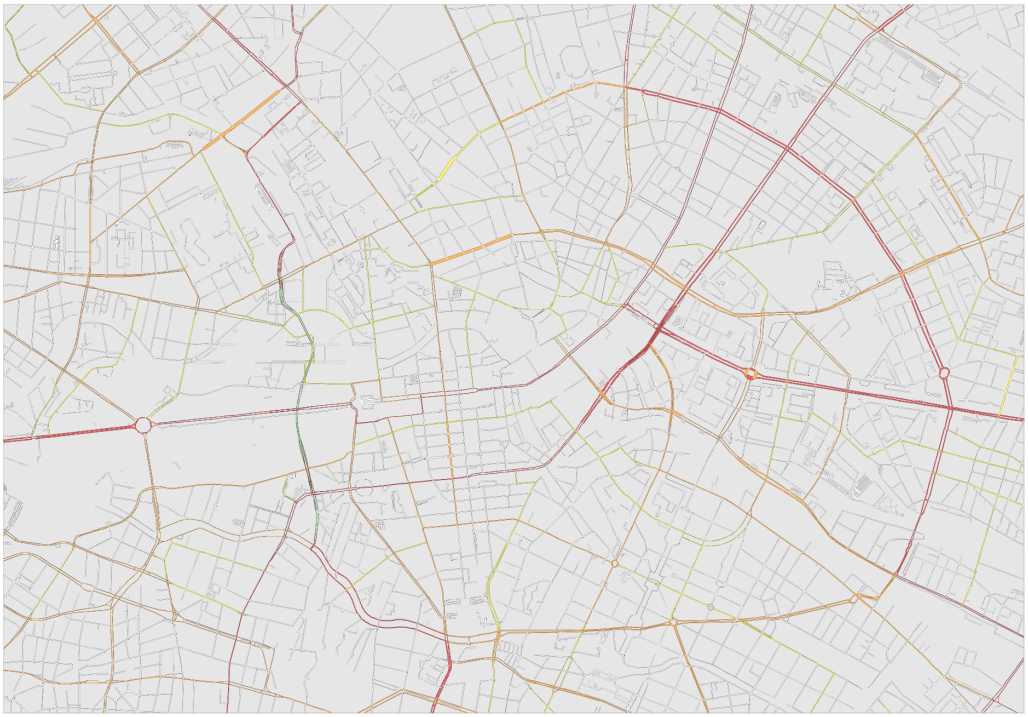


Abbildung 5.20.: Kartenausschnitt Berlin

- Die Innenstadt Berlins als Repräsentant eines unregelmäßigen (wenn auch nicht vollständig zufällig so entstandenen) Straßennetzes (siehe Abbildung 5.20),
- New York City – insbesondere der Stadtteil Manhattan – mit überwiegend auf der sprichwörtlichen „grünen Wiese“ entstandenen Verkehrswegen (siehe Abbildung 5.21) sowie
- Die Innenstadt Pueblas (genauer: Heroica Puebla de Zaragoza), der nach Einwohnerzahl viertgrößten Stadt Mexikos, die neben einer historischen und verschlungenen Altstadt auch große Flächen mit einem rechteckigen Straßennetz aufweist (siehe Abbildung 5.22).

Bei der Auswahl der Kartenausschnitte wurde darauf geachtet, insbesondere die markanten Regionen sowie entsprechende Einfallstraßen und Querverbindungen in die Simulation aufzunehmen. Dies führte zu Gesamtstraßenlängen von $853,91\,\text{km}$ (Berlin), $546,62\,\text{km}$ (New York) beziehungsweise $496,38\,\text{km}$ (Puebla). Die deutlich größere Gesamtlänge der simulierten Straßen in Berlin liegt in der höheren Dichte an Verkehrswegen gegenüber den anderen Städten begründet. Diese sind allerdings auch deutlich öfter als Einbahnstraßen oder mit einer geringeren Anzahl an Spuren konzipiert, was dazu führt, dass der Verkehrsfluss bei der gleichen Anzahl an

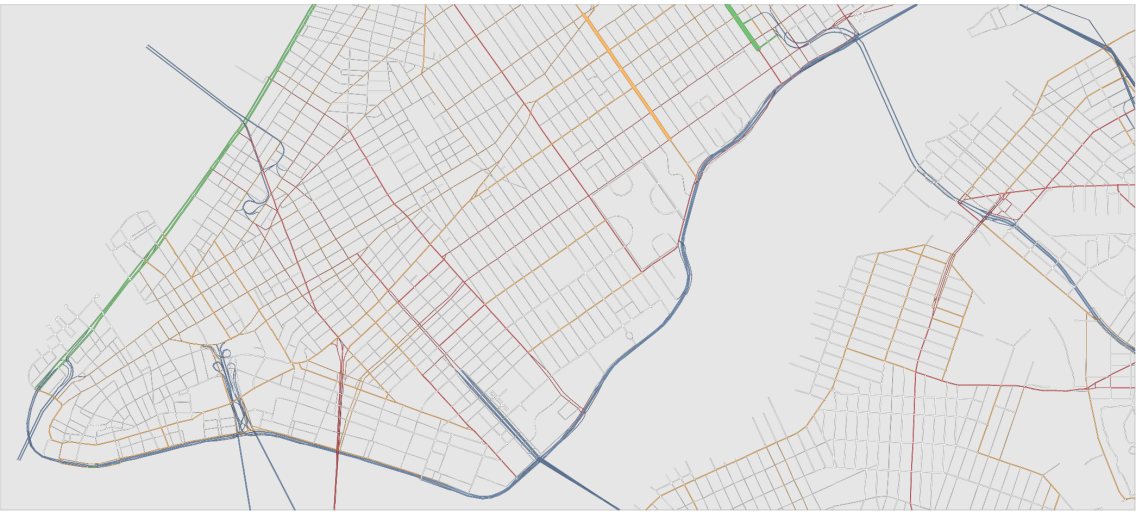


Abbildung 5.21.: Kartenausschnitt New York City

Fahrzeugen ähnlich bleibt.

In der Simulation soll ein globaler passiver Angreifer (*Global Passive Adversary*, *GPA*) simuliert werden. Dieser ist in der Lage, sämtliche gesendeten Nachrichten zur Kenntnis zu nehmen (*global*) ohne diese jedoch verändern oder selbst Nachrichten beisteuern zu können (*passiv*). In den Simulator wurde daher eine Logging-Komponente integriert, die alle gesendeten Beacons protokolliert. Dabei werden das aktuelle Pseudonym des Senders, seine Position und Geschwindigkeit übermittelt. Dies sind die Daten, die sich der Angreifer zunutze machen kann. Neben diesen Informationen wird auch noch die interne eindeutige Kennung des Fahrzeugs (vergleichbar mit der *UID*, vgl. Abschnitt 2.3.1) mit abgelegt. Diese steht dem Angreifer nicht zur Verfügung (er könnte damit ja eine sehr einfache Verkettung der unter unterschiedlichen Pseudonymen gesendeten Beacons durchführen), wird jedoch benötigt, um die korrekte Zuordnung von Pseudonymen und Teilnehmern zu kennen und so die Effektivität der durchgeführten Angriffe bestimmen zu können.

Die Auswertung der bei der Simulation angefallenen Daten erfolgt in einem separaten zweiten Schritt, da diese sehr rechenintensiv sind und somit die Simulation enorm beeinträchtigen würden. In einem realen Szenario würde ein Angreifer vermutlich ohnehin nicht versuchen, sämtliche Teilnehmer nach dem Durchlaufen einer Funkstille wieder zu identifizieren, sondern stattdessen das Ziel haben, ein bestimmtes verfolgtes Fahrzeug zu ermitteln. Die dafür notwendigen Berechnungen sind auf gängiger Hardware in sehr kurzer Zeit durchzuführen. Bei Erfolg kann der Angreifer sofort das wiederentdeckte Opfer weiterverfolgen.

Ziel der Untersuchungen soll somit sein, die oben genannten Konzepte zum Schutz

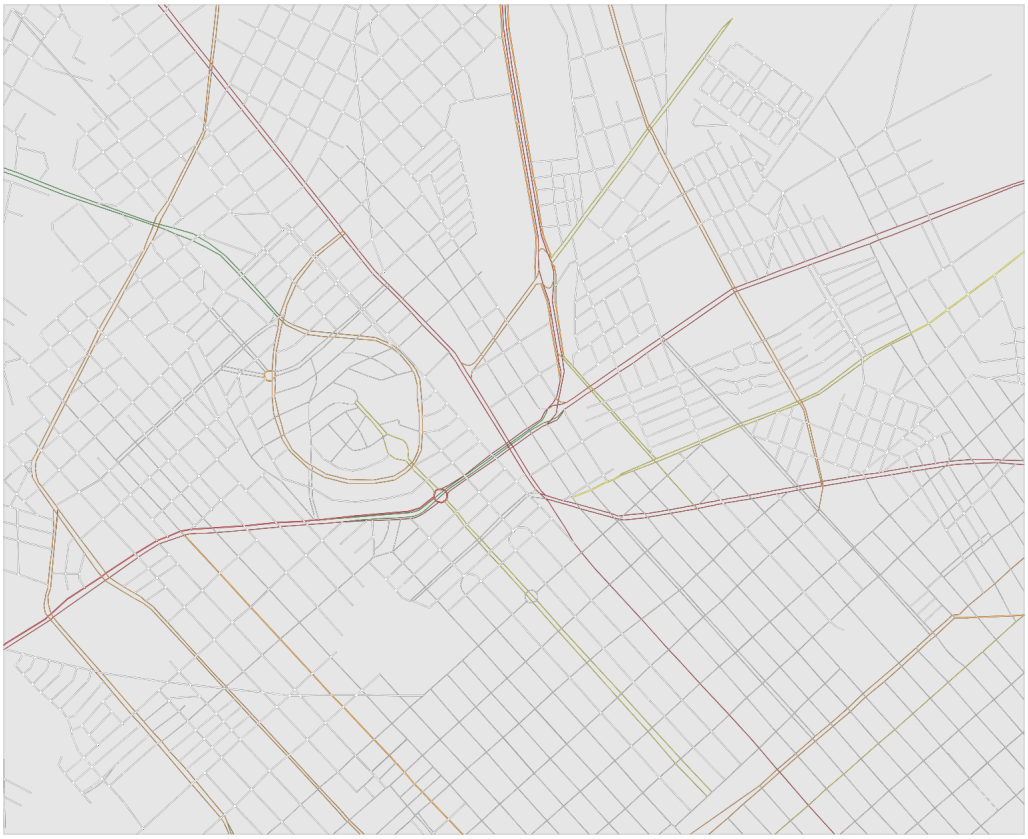


Abbildung 5.22.: Kartenausschnitt Puebla

der Privatsphäre auf den Prüfstand zu stellen und die Effektivität des von ihnen gebotenen Schutzes auf ihre Einflussfaktoren hin zu untersuchen. Dabei sollen insbesondere die folgenden Aspekte untersucht werden:

- Der Verkehrsfluss durch das Straßennetz, der mit Hilfe der drei unterschiedlichen Großstädte variiert wird,
- die Verkehrsdichte durch eine unterschiedliche Anzahl an Fahrzeugen in der Simulation (jeweils 1250, 2500, 5000, 10 000 und 20 000) sowie
- die Parameter der jeweiligen Konzepte.

Die Mix-Zonen wurden dabei so platziert, dass sie an möglichst unterschiedlichen Strukturen des Verkehrsnetzes verortet sind, damit auch hier verschiedene Einflüsse auf ihre Effektivität wie beispielsweise die Anzahl an Ein- und Ausfahrten sowie ihrer Spuren betrachtet werden können. Mit sehr hoher Wahrscheinlichkeit ist es möglich, mit Hilfe von umfangreichen empirischen Verkehrsanalysen optimalere Platzierungen für das Gesamtsystem (vgl. auch [FSH09]) zu finden.

Als Parameter für die (runde) *Mix-Zone* wird der Radius verändert und nimmt dabei Ausdehnungen von 25 m, 50 m, 100 m, 200 m und 500 m an, die *Silent Period* wird in ihrer Dauer in Zeitspannen von 1250 ms, 2500 ms, 5000 ms, 10 000 ms und 20 000 ms variiert, und bei *SLOW* wird die Grenzgeschwindigkeit zum Eintritt in die Funkstille auf die Werte 10 km/h, 20 km/h, 30 km/h und 40 km/k angepasst. Eine Grenze von 50 km/h ist in einem innerstädtischen Szenario nicht sinnvoll, da dies bei den meisten Fahrzeugen zu einer andauernden Funkstille führen würde.

Bei den Simulationen soll nur jeweils ein Parameter verändert werden (gemäß der Ceteris-paribus-Klausel), die anderen werden jeweils auf den mittleren Wert ihrer Spanne gesetzt; bei *SLOW* kommt dabei die von den Autoren vorgeschlagene Grenze von 30 km/h zum Einsatz.

Da die Fahrzeuge über die gesamte Karte verteilt starten, liefert die Simulation bereits nach wenigen Sekunden geeignete Ergebnisse. Von jedem Simulationsdurchgang werden über einen Zeitraum von 15 Minuten Daten erhoben.

5.4.2. Angriffe

Wie bereits einleitend dargelegt, soll die Bedrohung durch globale passive Angreifer (*GPA*) untersucht werden. Diese sind zwar vermutlich stärker als die meisten in der Realität agierenden Angreifer, dennoch ist es als Netzbetreiber möglich, eine derart große Verbreitung zu erreichen. Ist ein Schutz vor einem *GPA* möglich, so kann man mit Sicherheit davon ausgehen, dass auch vor schwächeren Angreifern ein effektiver Schutz besteht.

Die im Folgenden näher beschriebenen Angreifer müssen sich ausschließlich auf über Beacons empfangbare oder öffentlich zugängliche Informationen (wie beispielsweise Straßenverläufe) beschränken. In der Regel werden sie zudem lediglich auf einzelne Teilnehmer abzielen, die sie nach dem Durchfahren einer Funkstille wiederfinden möchten. Die hier modellierten Angreifer versuchen jedoch, die Pseudonyme möglichst vieler Teilnehmer vor und nach einem unter Funkstille stattfindenden Pseudonymwechsel zu verketten.

5.4.2.1. Angriffe auf Mix-Zonen

Angriffe auf Mix-Zonen folgen stets dem Prinzip, dass ein unter einem bestimmten Pseudonym in die Zone einfahrendes Fahrzeug nach der Funkstille und dem Pseudonymwechsel lediglich anhand der frei verfügbaren Informationen wiedergefunden werden soll. Dazu muss also zu einem Einfahrtszeitpunkt und -ort der passende Ausfahrzeitpunkt und -ort gefunden werden. Um dies zu erreichen, wurden zwei unterschiedlich komplexe Angriffsheuristiken – der einfache und der komplexe Angriff

auf die Mix-Zone – entwickelt. Die feste Position der Mix-Zonen ist dabei zugleich ein Fluch und ein Segen: Einerseits ist es zwar möglich, besonders günstige Orte für die Vermischung der Teilnehmer zu wählen, andererseits kann dafür der Angreifer ebenfalls statistische Untersuchungen über die Mix-Zone und die darunterliegende Kreuzung durchführen und sich dieses Wissen für seine Angriffe zunutze machen.

Einfacher Angriff Der Angreifer kann als Vorüberlegung für die Konzeption einer Heuristik davon ausgehen, dass ein Teilnehmer, der eine Mix-Zone durchfährt, eine Strecke zurücklegen muss, die ungefähr dem doppelten Radius r der Mix-Zone entspricht. Als durchschnittliche Geschwindigkeit in diesem Abschnitt könnte der Angreifer entweder die zuletzt bei einem konkreten Teilnehmer beobachtete Geschwindigkeit (die im Beacon übermittelt wird) oder die maximal zulässige Höchstgeschwindigkeit der Einfahrtstraße v_{street} als Näherung heranziehen. Die Entscheidung bei der Entwicklung dieses Angriffs fiel auf die zweite Möglichkeit, da sie von Teilnehmern unabhängig ist und somit einige Werte vorberechnet werden können.

Der Angreifer ist nach den eben angestellten Überlegungen in der Lage, die erwartete Zeit für das Durchfahren der Mix-Zone mit $t_{exp} = \frac{2r}{v_{street}}$ abzuschätzen. Prinzipiell muss er somit nur noch sämtliche Ausgänge der Mix-Zone beobachten und das Fahrzeug, das diese zu einem Zeitpunkt verlässt, der dem geschätzten am nächsten kommt, als das gesuchte annehmen.

Um die Genauigkeit der Vorhersage noch zu verbessern, kann der Angreifer einen Faktor für die Anpassung der Zeit (f_t) einführen. Dieser Faktor berücksichtigt, dass die bisherige Ermittlung der erwarteten Durchfahrtsdauer t_{exp} sehr optimistisch kalkuliert war, da möglicherweise Ampelanlagen oder andere Fahrzeuge, denen Vorfahrt zu gewähren ist, das Fortkommen des Opfers behindern können. Dieser Modifikator hängt somit sehr stark von der tatsächlichen Positionierung der Mix-Zone ab und kann durch den Angreifer mit Hilfe empirischer Untersuchungen ermittelt werden. Die Abschätzung der erwarteten Zeit t_{exp} erfolgt daher mit der in Gleichung 5.9 angegeben Berechnung.

$$t_{exp} = \frac{2r}{v_{street}} f_t \tag{5.9}$$

Untersucht man die aus einer Mix-Zone ausfahrenden Fahrzeuge, so ist es möglich, die Abschätzung weiter zu präzisieren. Tendenziell neigen Fahrzeuge dazu, in Kreuzungen auf Straßen, die ein schnelleres Fortkommen versprechen, zu wechseln. Dieses schnellere Fortkommen kann in einer höheren zulässigen Maximalgeschwindigkeit oder einer größeren Anzahl an Spuren begründet sein. Die Wahrscheinlichkeit, dass

Fahrzeuge eine Kreuzung auf einer größeren Straße verlassen als sie sie befahren haben, ist somit größer als im umgekehrten Fall. Daher ist es möglich, eine weitere Größe, die Gewichtung der zur Ausfahrt verwendeten Straße (als *streetSizeWeight*, w_s bezeichnet), einzuführen, die Fahrzeuge, die die Zone auf einer kleineren Straße verlassen als das Opfer sie betreten hat, mit einem Malus versehen.

Eine weitere Beobachtung zeigt, dass Fahrzeuge an Kreuzungen in der Regel dazu neigen, diese geradeaus zu überfahren anstatt abzubiegen oder gar zu wenden. Lediglich an Kreuzungen mit besonderen Verkehrsführungen (abknickende Vorfahrtsstraßen oder einige große innerstädtische Verkehrsknotenpunkte) zeigt sich eine deutliche Abweichung von diesem Verhalten, das jedoch meist anhand der Beschilderung oder der Anzahl an Abbiegespuren (auch für einen Angreifer) ersichtlich ist. Diese Überlegungen führen zu einem weiteren Modifikator, der Gewichtung von Abbiegevorgängen (als *streetTurnWeight*, w_t bezeichnet). Dieser Wert ist, wie die anderen Gewichte auch, von dem Verkehrsfluss durch die Mix-Zone abhängig. Ausfahrten aus der Zone, die mit einer hohen Wahrscheinlichkeit genutzt werden (weil sie beispielsweise der Einfahrt gerade gegenüber liegen), erhalten einen hohen Wert w_t, Straßen, die einen Abbiegevorgang oder gar eine Kehre erfordern, dagegen einen geringen Wert.

Berücksichtigt man diese neu eingeführten Modifikatoren, so kann der Angreifer einen *Score*, einen Vergleichswert für jedes ausfahrende Fahrzeug in Bezug zu seinem Opfer mit Hilfe der Gleichung 5.10 ermitteln. Dabei gibt t_{in} die Einfahrtszeit des Opfers in die Mix-Zone und t_{out} die Ausfahrtszeit des jeweils beobachteten Fahrzeugs an.

$$score = (1 - w_t)\,(1 - w_s)\,|t_{out} - t_{in} - t_{exp}| \tag{5.10}$$

Dieser *Score* wird für jedes aus der Mix-Zone ausfahrende Fahrzeug berechnet. Anschließend wird dasjenige mit dem kleinsten Wert als das gesuchte Fahrzeug angenommen. Der gesamte Ablauf ist in Algorithmus 5.1 im Pseudocode für ein Opfer dargestellt. Möchte der Angreifer für möglichst viele Fahrzeuge eine Verkettung zwischen der Ein- und der Ausfahrt herstellen, so muss er diesen Algorithmus für jedes einfahrende Fahrzeug durchführen.

Komplexer Angriff Der beschriebene einfache Angriff lässt sich mit Hilfe von empirischen Untersuchungen des Verkehrsflusses in der Mix-Zone (bzw. der darunterliegenden Kreuzung) verbessern. Durch die entsprechenden Analysen ist ein Angreifer in der Lage, die Bewegung eines Fahrzeugs mit deutlich höherer Präzision vorherzusagen und entsprechend – mit allerdings größerem Rechenaufwand – eine höhere Trefferchance zu erreichen.

Algorithmus 5.1 Angriff auf Mix-Zone (einfach)

$score \leftarrow \infty$
$t_{in} \leftarrow$ time of entry
$t_{exp} \leftarrow 2 * r_{mixzone} * v_{street} * f_t$
for all vehicles leaving mix zone **do**
  $t_{out} \leftarrow$ time of exit
  $w_t \leftarrow streetTurnWeight$ of used exit
  $w_s \leftarrow streetSizeWeight$ of used exit
  $score_{tmp} \leftarrow (1 - w_t) * (1 - w_s) * |t_{out} - t_{in} - t_{exp}|$
  if $score_{tmp} < score$ **then**
    $score \leftarrow score_{tmp}$
    $vehicleFound \leftarrow currentVehicle$
  end if
end for
return $vehicleFound$

In diesem Ansatz muss der Angreifer erneut ein Fahrzeug, das die Mix-Zone an Einfahrt i erreicht hat, mit jedem anderen Fahrzeug, das die Zone bei Ausfahrt j verlässt, vergleichen. Dabei kann er die (empirisch ermittelte) durchschnittliche Fahrzeit $t_{i,j}$ von i nach j sowie die (ebenfalls empirisch ermittelte) Wahrscheinlichkeit, dass sich ein Fahrzeug von i nach j bewegt, $p_{i,j}$, in seiner Heuristik berücksichtigen. Er ermittelt somit den *Score* für einen Kandidaten mit Hilfe von Gleichung 5.11.

$$score = \left(1 - \frac{p_{i,j}}{2}\right) |t_{out} - t_{in} - t_{i,j}| \tag{5.11}$$

In der Berechnung wird erneut die theoretische Fahrzeit des Kandidaten – ermittelt aus der Differenz zwischen der Einfahrtszeit des Opfers t_{in} und der Ausfahrtszeit des Kandidaten t_{out} – mit der erwarteten Dauer $t_{i,j}$ verglichen und das Ergebnis um einen Korrekturfaktor bereinigt. In diesen Faktor geht jedoch die Wahrscheinlichkeit der Route $p_{i,j}$ nur zur Hälfte ein, da Untersuchungen gezeigt haben, dass die geschätzte Zeit ein deutlich präziseres Indiz für das korrekte Fahrzeug ist als die generelle Wahrscheinlichkeit, einen bestimmten Weg zu wählen. Der Ablauf dieses Angriffs ist in Algorithmus 5.2 dargestellt. Sollen mehr Fahrzeuge als nur ein einzelnes angegriffen werden, so muss für jedes dieser Fahrzeuge der Ablauf erneut ausgeführt werden.

Algorithmus 5.2 Angriff auf Mix-Zone (komplex)

$score \leftarrow \infty$
$i \leftarrow$ used entrance of mix zone
$t_{in} \leftarrow$ time of entry
for all vehicles leaving mix zone **do**
  $j \leftarrow$ used exit of mix zone
  $t_{out} \leftarrow$ time of exit
  $t_{i,j} \leftarrow$ expected travel time from ports i to j
  $p_{i,j} \leftarrow$ probability of combination i, j
  $score_{tmp} \leftarrow (1 - \frac{p_{i,j}}{2}) \, |t_{out} - t_{in} - t_{i,j}|$
  if $score_{tmp} < score$ **then**
    $score \leftarrow score_{tmp}$
    $vehicleFound \leftarrow currentVehicle$
  end if
end for
return $vehicleFound$

5.4.2.2. Angriffe auf Silent Periods

Im Gegensatz zu der Mix-Zone findet bei Silent Periods die Funkstille unabhängig vom Aufenthaltsort der Fahrzeuge zu einem global festgelegten Zeitpunkt (in der implementierten Variante, vgl. Abschnitt 5.3.8.3) für alle Teilnehmer gleichzeitig statt. Dies führt aus Sicht des Angreifers zwar einerseits dazu, dass dieser keinerlei Analysen über die erwartete Bewegung von Teilnehmern an bestimmten Orten durchführen kann, andererseits aber finden viele Funkstillen und Pseudonymwechsel somit in Straßenabschnitten statt, die Teilnehmern keinerlei Abbiege- oder Vermischungsmöglichkeiten bieten, wie beispielsweise lange gerade Straßen.

Da bei der Silent Period die Zeit – anstelle des Ortes bei der Mix-Zone – der auslösende und bestimmende Faktor ist, muss für einen erfolgreichen Angriff die innerhalb dieser Zeit zurückgelegte Strecke möglichst exakt geschätzt werden, um ein in die Funkstille eintretendes Fahrzeug wiederauffinden zu können und somit das alte und das neue Pseudonym korrekt zu verketten. Für die Durchführung dieses Angriffs sind im Folgenden wiederum zwei unterschiedlich komplexe Heuristiken dargestellt.

Einfacher Angriff Die für alle Teilnehmer zu einem bestimmten Zeitpunkt durchgeführte Silent Period hat eine vorgegebene und dem Angreifer bekannte Dauer. Daher versucht dieser abzuschätzen, welche Strecke sein Opfer in der gegebenen Zeit zurücklegen konnte. Dazu gibt es für ihn – ähnlich wie bei dem einfachen Angriff auf Mix-Zonen – zwei Geschwindigkeiten, die er seiner Schätzung zugrunde legen kann:

Er könnte die letzte bekannte Geschwindigkeit des Fahrzeugs heranziehen (diese wird in einem Beacon übermittelt) oder die zulässige Höchstgeschwindigkeit der Straße (v_{street}), auf der sich sein Opfer befindet. Da sich zum einen die Geschwindigkeit des Fahrzeugs deutlich ändern kann und zum anderen in der Folge wieder eine Vielzahl an Angriffen auf unterschiedliche Teilnehmer durchgeführt werden soll, fiel die Entscheidung erneut auf die Höchstgeschwindigkeit der zu Beginn der Silent Period befahrenen Straße. Somit legt der betrachtete Teilnehmer innerhalb der Silent Period mit der Dauer t_{sp} die Strecke $d_{exp} = t_{sp} v_{street}$ zurück.

Auch in diesem Fall dürfte die Schätzung zu optimistisch sein, da die Durchschnittsgeschwindigkeit von Fahrzeugen in der Regel etwas unter der zulässigen Höchstgeschwindigkeit liegt. Dies resultiert aus Verkehrssituationen, in den sie anderen Fahrzeugen Vorrang einräumen müssen oder durch Ampeln aufgehalten werden. Somit wird – wie bereits zuvor beim Angriff auf Mix-Zonen – ein Korrekturfaktor der Distanz (f_d) eingeführt, der die Prognose verbessern soll. Somit ergibt sich die Schätzung aus Gleichung 5.12 für die zurückgelegte Distanz während der Silent Period.

$$d_{exp} = t_{sp} v_{street} f_d \tag{5.12}$$

Erneut werden in der Folge alle in Frage kommenden Fahrzeuge jeweils mit dem in die Silent Period eingefahrenen Opfer verglichen; für jede der möglichen Kombinationen wird ein *Score* berechnet. Dieser besteht in der einfachen Heuristik lediglich in der Abweichung der erwarteten Strecke (d_{exp}) von der beobachteten zurückgelegten Strecke (d_{obs}) und ist in Gleichung 5.13 ausformuliert.

$$score = |d_{exp} - d_{obs}| \tag{5.13}$$

Derjenige Kandidat, der die kleinste Abweichung aufweist, wird als das gesuchte Fahrzeug angenommen. Algorithmus 5.3 zeigt systematisch den Ablauf des Angriffs auf ein Fahrzeug; für jedes weitere anzugreifende Fahrzeug muss der Algorithmus erneut von Beginn ausgeführt werden.

Komplexer Angriff Der einfache Angriff weist im Rahmen seiner geringen Komplexität eine gewisse Schwäche auf; er berücksichtigt die Bewegungsrichtung der Fahrzeuge in keinster Weise. Ein in Bewegung befindliches Fahrzeug neigt in der Regel dazu, die Bewegungsrichtung zumindest die meiste Zeit über beizubehalten, wodurch eine Richtungsänderung gerade innerhalb der relativ kurzen Silent Period umso unwahrscheinlicher wird. Mit dem oben beschriebenen Algorithmus würde

Algorithmus 5.3 Angriff auf SilentPeriod (einfach)

$score \leftarrow \infty$
$pos_{in} \leftarrow$ position of last beacon before entering silent period
$t_{SP} \leftarrow$ duration of silent period
$v_s \leftarrow$ max speed of street
$d_{exp} \leftarrow t_{SP} * v_s * f_d$
$targetArea \leftarrow$ round area with radius d_{max} and center pos_{in}
for all vehicles ending silent period in $targetArea$ **do**
  $pos_{out} \leftarrow$ position of vehicle when ending Silent Period
  $d_{obs} \leftarrow$ distance between pos_{in} and pos_{out}
  $score_{temp} \leftarrow |d_{exp} - d_{obs}|$
  if $score_{tmp} < score$ **then**
    $score \leftarrow score_{tmp}$
    $vehicleFound \leftarrow currentVehicle$
  end if
end for
return $vehicleFound$

auch dann eine Verkettung hergestellt werden, wenn ein aus der Funkstille austretendes Fahrzeug exakt den geschätzten Abstand zu dem eintretenden Fahrzeug aufweist, auch wenn dieses sich in die entgegengesetzte Richtung bewegt. Somit entsteht eine Reihe von falschen Zuordnungen.

In einem komplexeren Angriff kann die Fahrtrichtung des angegriffenen Teilnehmers Berücksichtigung finden. Dazu wird anstelle einer Distanz der Ort pos_{exp} geschätzt, an dem sich ein Fahrzeug, das mit dem Geschwindigkeitsvektor $\vec{v}$ an Position p_{in} in die Funkstille eintritt, bei einer konstant angenommenen Fortbewegung nach der Dauer einer Silent Period t_{sp} aufhält. Der *Score* in diesem Angriff wird somit über die Abweichung der erwarteten Position von der beobachteten Position pos_{out} eines Kandidaten nach Gleichung 5.14 bestimmt.

$$score = |pos_{exp} - pos_{out}| \tag{5.14}$$

Diese lediglich auf die Position ausgerichtete Betrachtung läuft Gefahr, erneut einen Fehler zu machen: Ein entgegenkommendes Fahrzeug, das sich zufällig in der Nähe des erwarteten Ortes aufhält, könnte auch hier fälschlicherweise Berücksichtigung finden. Daher wird ebenfalls an dem beobachteten Punkt der Bewegungsvektor des Kandidaten betrachtet und die Richtung mit der Ausgangsrichtung des Opfers abgeglichen. Überschreitet der Winkel zwischen den beiden Vektoren einen Schwellenwert (*threshold*), so wird der Kandidat nicht berücksichtigt.

Algorithmus 5.4 Angriff auf SilentPeriod (komplex)

$score \leftarrow \infty$
$pos_{in} \leftarrow$ position of last beacon before entering silent period
$t_{SP} \leftarrow$ duration of silent period
$v_s \leftarrow$ max speed of street
$d_{exp} \leftarrow t_{sp} * v_s * d_f$
$targetArea \leftarrow$ round area with radius d_{max} and center pos_{in}
for all vehicles ending silent period in $targetArea$ **do**
  $pos_{out} \leftarrow$ position of vehicle when ending silent period
  $pos_{exp} \leftarrow$ position when moving from pos_{in} for d_{exp} while maintaining direction
  $score_{tmp} \leftarrow |pos_{exp} - pos_{out}|$
  $angle \leftarrow$ angle between directions of current and attacked vehicle
  if $score_{tmp} < score \wedge angle < threshold$ **then**
    $score \leftarrow score_{tmp}$
    $vehicleFound \leftarrow currentVehicle$
  end if
end for
return $vehicleFound$

Der komplexe Angriff auf Silent Periods ist in Algorithmus 5.4 in Pseudocode zusammengefasst. Die Berechnung der *targetArea* über die maximal mögliche Distanz wird dabei lediglich durchgeführt, um die Liste der Kandidaten vorab deutlich einzuschränken. Das infrage kommende Gebiet soll daher gerade noch die maximal erreichbaren Orte innerhalb von t_{sp} umfassen.

5.4.2.3. Angriff auf SLOW

Die Verkettung der Pseudonyme eines Teilnehmers vor und nach einer SLOW-Phase ist eine deutlich schwierigere Aufgabe als die zuvor beschriebenen Angriffe. Daher wurde für das SLOW-Szenario auch nur ein (komplexer) Angriff konzeptioniert und implementiert; ein einfacherer Ansatz würde schlichtweg nicht funktionieren.

Die Schwierigkeit liegt darin, dass SLOW weder einen koordinierten Ort noch eine koordinierte Zeitspanne bietet, die einen Ansatzpunkt für einen Angreifer darstellen. Da zudem gerade in Kreuzungsbereichen, wo eine besonders gute Vermischung von Teilnehmern stattfinden kann, viele Teilnehmer die Grenzgeschwindigkeit von SLOW v_{slow} unterschreiten und somit in eine Funkstille eintreten (anders etwa als bei einer Random Silent Period, vgl. Abschnitt 4.3.1), ist das Wiederauffinden eines angegriffenen Teilnehmers keine einfache Aufgabe. Theoretisch könnte sich ein Teilnehmer beliebig lange mit einer geringeren Geschwindigkeit als der Schwellenge-

schwindigkeit fortbewegen und somit auch eine beliebig lange Strecke zurücklegen. Dies führt dazu, dass die Anonymitätsgruppe enorm anwächst und theoretisch die Gesamtheit an Fahrzeugen umfasst.

In der Praxis zeigt sich jedoch, dass Teilnehmer in der Regel versuchen, sich mit möglichst großer Geschwindigkeit fortzubewegen und somit, sobald es die Verkehrssituation erlaubt, SLOW auch wieder verlassen. Ein Angreifer kann sich dies zunutze machen und zur Realisierung des Angriffs eine maximale Verweildauer von t_{max} annehmen. Somit kann ein verfolgtes Fahrzeug, das die Grenzgeschwindigkeit v_{slow} unterschreitet, höchstens die Distanz $d_{max} = t_{max} \cdot v_{slow}$ in Funkstille zurücklegen. Unter dieser Annahme muss der Angreifer lediglich diejenigen Teilnehmer als Kandidaten berücksichtigen, die in einer runden Region mit dem Radius d_{max} und dem Mittelpunkt pos_{in}, der Eintrittsposition des Opfers in die Funkstille innerhalb des Zeitraums t_{max}, aus einer Funkstille auftauchen.

Von diesen Kandidaten muss der Angreifer jeweils den Austrittszeitpunkt t_{out} sowie die Position bei Beenden der Funkstille pos_{out} betrachten. Daraus kann er zum einen die Distanz zu dem Eintrittsort des Opfers (d_{obs}) sowie die in dem Zeitraum zwischen t_{in} und t_{out} erwartete Distanz mit Gleichung 5.15 berechnen.

$$d_{exp} = (t_{out} - t_{in})\, v_{slow} f_t \tag{5.15}$$

Wie bei den vorigen Angriffen findet auch hier ein Korrekturfaktor f_t Berücksichtigung, da eine Kalkulation mit Hilfe der SLOW-Grenzgeschwindigkeit v_{slow} ja eine obere Schranke darstellt. Sobald diese Geschwindigkeit auch nur um den geringsten Wert überschritten wird, tritt das Fahrzeug aus der Funkstille aus. Zudem wird ein Teilnehmer – wie auch bereits oben dargelegt – durch die Verkehrssituationen auf der Straße aufgehalten und die erwartete Distanz nimmt ab.

Der zur Identifikation des wahrscheinlichsten Kandidaten herangezogene *Score* berechnet sich anschließend erneut aus der Differenz der erwarteten und der (angenommenen) beobachteten zurückgelegten Distanz des Kandidaten mit Gleichung 5.16. Der soeben beschriebene Ablauf ist in Algorithmus 5.5 dargestellt und muss wiederum bei Angriffen auf mehrere Teilnehmer für jedes potentielle Opfer individuell durchlaufen werden.

$$score = |d_{exp} - d_{obs}| \tag{5.16}$$

Algorithmus 5.5 Angriff auf SLOW

```
score ← ∞
t_in ← time of last beacon before entering SLOW
pos_in ← position of last beacon before entering SLOW
d_max ← t_max * v_slow
targetArea ← round area with radius d_max and center pos_in
for all vehicles leaving SLOW in targetArea between t_in and t_in + t_max do
    t_out ← time of first beacon after leaving SLOW
    pos_out ← position of first beacon after leaving SLOW
    d_obs ← distance between pos_in and pos_out
    d_exp ← (t_out − t_in) * v_slow * f_t
    score_tmp ← |d_exp − d_obs|
    if score_tmp < score then
        score ← score_tmp
        vehicleFound ← currentVehicle
    end if
end for
return vehicleFound
```

5.4.3. Metriken

Mit der Durchführung der Simulation und der Angriffe soll die Effektivität der untersuchten Schutzmaßnahmen gemessen werden. Dazu sollen die drei verbreitetsten Metriken zum Einsatz kommen und miteinander verglichen werden: der Angriffserfolg, die Anonymitätsgruppe und die Entropie. Die beiden letzteren würden sich sehr gut zu einer analytischen Betrachtung vorab eignen. Daher sollen sie daraufhin untersucht werden, wie gut sie den tatsächlichen Angriffserfolg der empirischen Studien wiedergeben.

5.4.3.1. Angriffserfolg

Die am einfachsten zu bestimmende Metrik ist der Angriffserfolg. Dabei werden die oben beschriebenen Angriffe durchgeführt und anschließend wird – mit Hilfe der für den Angriff nicht herangezogenen Zusatzinformationen über die Identitäten der Fahrzeuge – bestimmt, wie viele Fahrzeuge korrekt wiedererkannt werden konnten (N_a). Diese Zahl wird mit der Gesamtzahl an Fahrzeugen (N), die untersucht wurden, ins Verhältnis gesetzt, so dass sich die Berechnungsvorschrift aus Gleichung 5.17

für den Angriffserfolg P ergibt.

$$P = \frac{N_a}{N} \tag{5.17}$$

Der Angriffserfolg ist stark von der Heuristik des durchgeführten Angriffs abhängig. Dennoch gibt er einen guten Einblick, wie effektiv ein bestimmter definierter Angreifer die Privatsphäre der Teilnehmer gefährden kann.

5.4.3.2. Anonymitätsgruppe

Die Anonymitätsgruppe (vgl. u. a. [Swe02]) ist ein weit verbreitetes und relativ einfach zu bestimmendes Maß für die Unbeobachtbarkeit beziehungsweise Anonymität eines Teilnehmers in einer Gruppe Gleichartiger. Innerhalb einer bestimmten Region bilden während eines bestimmten Zeitfensters alle anwesenden Teilnehmer eine Anonymitätsgruppe, wenn sie von einem Angreifer nicht unterscheidbar sind. In der Theorie kann der Angreifer somit lediglich raten, welches der k Gruppenmitglieder der Gesuchte ist und hat damit (ohne weiteres Kontextwissen) eine Wahrscheinlichkeit von $\frac{1}{k}$ richtig zu liegen. Grundsätzlich wird in dieser Arbeit die Anonymitätsgruppe für ein Konzept zum Schutz der Privatsphäre als die durchschnittliche Anonymitätsgruppe aller Teilnehmer definiert.

Die Anonymitätsgruppe in einer Mix-Zone definiert sich durch die Anzahl an Fahrzeugen, die sich in einem bestimmten Zeitraum in ihr aufhalten. Es werden also je Teilnehmer die Ein- und Austrittszeiten t_{in} und t_{out} aus der Zone mit dem bekannten Radius r und ihrem Mittelpunkt $[x, y]$ bestimmt und die gleichzeitig in der Zone befindlichen Fahrzeuge gezählt. Die Zahlen werden über die n Teilnehmer, die die Mix-Zone jemals durchfahren, und die Simulationen entsprechend Gleichung 5.18 gemittelt.

$$k_{ges} = \frac{\sum_{i=1}^{n} k_i}{n} \tag{5.18}$$

Die Berechnung der Anonymitätsgruppe bei Silent Periods stellt sich als etwas komplexer und schwieriger heraus, da der Bereich, in dem die Teilnehmer eine Anonymitätsgruppe bilden, nicht einfach zu bestimmen ist. Prinzipiell kommen nach einer zeitlich koordinierten Funkstille sämtliche Fahrzeuge infrage, das gesuchte zu sein. Dabei würde jedoch eine beliebig große Geschwindigkeit für die Teilnehmer angenommen. In der Folge wird daher von einer Maximalgeschwindigkeit von 250 $^{\mathrm{km}}/_{\mathrm{h}}$

ausgegangen, da diese von den meisten Fahrzeugen durch technische Beschränkungen niemals überschritten werden kann.

Die Region, in der die Teilnehmer eine Anonymitätsgruppe bilden, wird somit wiederum kreisrund angenommen und erhält den Radius $r = 250\,{}^{\mathrm{km}}/_{\mathrm{h}} \cdot t_{sp}$, der sich aus der in der Zeit der Funkstille t_{sp} maximal zurücklegbaren Strecke ergibt. Als Mittelpunkt wird jeweils das aktuelle Fahrzeug gewählt, und entsprechend wird die durchschnittliche Anonymitätsgruppe über alle Fahrzeuge bestimmt.

Problematisch stellt sich die Anonymitätsgruppe bei SLOW dar: Da es weder einen klar umrissenen Bereich, in dem Funkstille gewahrt wird, noch eine maximale Zeitdauer gibt, kann ein Fahrzeug theoretisch beliebig lang in der Funkstille verbleiben und sich dabei zu jedem beliebigen Ort fortbewegen. Dazu muss lediglich die Schwellengeschwindigkeit v_{slow} stets unterschritten werden. Dies führt dazu, dass ein aus einer durch SLOW ausgelösten Funkstille auftauchendes Fahrzeug theoretisch jedes beliebige jemals in die Funkstille eingetretene Fahrzeug sein könnte. Dabei gibt es natürlich gewisse Beschränkungen hinsichtlich derjenigen Fahrzeuge, die in großer Distanz unmittelbar zuvor die Grenzgeschwindigkeit unterschritten haben. Dennoch vergrößert sich die Anonymitätsgruppe ab Beginn der Beobachtung sukzessive und breitet sich mit der Schwellengeschwindigkeit aus. Entsprechend bedeckt sie nach einer gewissen Zeit die komplette Beobachtungsregion und umfasst somit ab diesem Zeitpunkt alle Fahrzeuge. Daher ist eine Bestimmung dieser Größe für SLOW nicht sinnvoll.

5.4.3.3. Entropie

Die große Schwäche der Anonymitätsgruppe zeigt sich bereits bei der Betrachtung der Silent Periods, aber insbesondere auch bei SLOW: Bei einer Gruppe von k Teilnehmern in einer Anonymitätsgruppe wird davon ausgegangen, dass jeder von ihnen mit der exakt gleichen Wahrscheinlichkeit als der Gesuchte in Frage kommt. Dies ist jedoch eine enorme Vereinfachung und führt zu den oben geschilderten Problemen. Als Lösung wird oftmals die Entropie H (vgl. u. a. [Sha48]) betrachtet, die jedem einzelnen Kandidaten eine bestimmte Wahrscheinlichkeit zuordnet und über die in Gleichung 5.19 angegebene Berechnung bestimmt wird. Diese ist ein Maß für die Unvorhersehbarkeit oder Unsicherheit, die ein Betrachter (hier: der Angreifer) durch ein System (z. B. eine Mix-Zone) erfährt. Dabei repräsentiert p_i die Wahrscheinlichkeit, dass ein bestimmtes Objekt i das gesuchte Objekt aus der Grundgesamtheit N ist.

$$H = \sum_{i=1}^{N} p_i \log_2 (p_i) \tag{5.19}$$

Bei Mix-Zonen ist es relativ einfach, den Kandidaten Wahrscheinlichkeiten zuzuweisen, indem man sich an den Überlegungen zu dem komplexeren Angriff orientiert. Dabei wird durch eine empirische Untersuchung für jede Kombination aus Ein- und Ausfahrtspunkt (i und j) der Zone eine relative Häufigkeit ermittelt. Diese sich aus den Routen (i, j) ergebenden relativen Häufigkeiten $N_{i,j}$ können dann entsprechend Gleichung 5.20 in Wahrscheinlichkeiten $p_{i,j}$ umgerechnet werden (vgl. auch die Arbeiten [BS03, BS04] von Beresford und Stajano).

$$p_{i,j} = \frac{N_{i,j}}{\sum^{i,j} N_{i,j}} \tag{5.20}$$

Etwas schwieriger ist es, die Wahrscheinlichkeiten für die Fahrzeuge bei Silent Periods zu bestimmen. Dazu wird in der Folge die Differenz der Abweichung eines Kandidaten nach einer Funkstille der Dauer t_{sp} von der erwarteten Position (analog den Betrachtungen der Angriffe auf Silent Periods bestimmt über $|d_{exp} - d_{obs}|$) von der maximal möglichen Abweichung unter der Annahme einer Maximalgeschwindigkeit von 250 km/h (über $d_{max} - d_{exp}$) bestimmt. Somit wird diese Differenz d nach Gleichung 5.21 ermittelt.

$$d = max\,[0, (d_{max} - d_{exp}) - |d_{exp} - d_{obs}|] \tag{5.21}$$

Sollte die tatsächliche Abweichung eines Kandidaten größer als die unter den gegebenen Annahmen maximal mögliche Abweichung werden, so wird die Differenz auf null gesetzt (was dazu führt, dass der Kandidat eine Wahrscheinlichkeit von null zugewiesen bekommt und dadurch nicht mehr weiter berücksichtigt wird). Die Wahrscheinlichkeiten für jeden Kandidaten i ergeben sich dann über die Gesamtabweichung aller N Teilnehmer nach Gleichung 5.22.

$$p_i = \frac{d_i}{\sum_{i=1}^{N} d_i} \tag{5.22}$$

Die Ermittlung der Entropie bei SLOW führt – wie schon bei der Anonymitätsgruppe – zu Schwierigkeiten. Da es keine Begrenzung jeglicher Art für die Dauer oder den Ort eines in der Funkstille befindlichen Teilnehmers gibt, besitzt nach einiger Zeit jeder Teilnehmer im System eine Wahrscheinlichkeit von größer null, der Gesuchte zu sein. Sollte er der Gesuchte sein, ergibts sich die Wahrscheinlichkeit eines Kandidaten aus der Abweichung seiner potentiell zurückgelegten Strecke d_{obs} zwischen dem Eintrittszeitpunkt t_{in} des Gesuchten und dem Austrittszeitpunkt des

Kandidaten i aus der Funkstille t_{out} und der erwarteten Strecke d_{exp} im gleichen Zeitintervall $[t_{in}, t_{out}]$ (vgl. Gleichung 5.23 und Gleichung 5.24).

$$d = |d_{exp} - d_{obs}| \tag{5.23}$$

$$p_i = \frac{d_i}{\sum_{i=1}^{N} d_i} \tag{5.24}$$

Die anschließende Berechnung der Entropie mit Gleichung 5.19 erfordert jedoch die Berücksichtigung aller Fahrzeuge zu jedem beliebigen Zeitpunkt $t > t_{in}$. Selbst wenn man berücksichtigt, dass vermutlich die meisten Fahrzeuge nur eine begrenzte Zeit Funkstille wahren, ist diese Kalkulation auf Basis empirischer Daten mit tausenden Fahrzeugen, für deren Funkstille jeweils die Entropie über wiederum sämtliche Fahrzeuge zu bestimmen ist, nicht sinnvoll durchzuführen. Diese Abschätzungen können somit allenfalls analytisch erfolgen; die Simulationen bieten jedoch nicht die Möglichkeit, hier eine sinnvolle Entropie zu ermitteln. Auch die Autoren des Konzepts, Buttyán et al., ermitteln in ihrer Arbeit [BHWW09] lediglich die Erfolgsraten eines Angriffs auf die Teilnehmer und machen somit keinen Vorschlag, wie die Entropie in ihrem System bestimmt werden könnte.

5.4.4. Ergebnisse

Nach der Durchführung der 450 Simulationen – je Stadt und Konzept wurden zehn verschiedene Parametereinstellungen gewählt und jede Kombination wurde fünfmal angewendet – wurden die zuvor beschriebenen Angriffe angewandt und die Ergebnisse aufbereitet.

Die Metriken *Anonymitätsgruppe* und *Entropie* sollen in den Untersuchungen dem *Angriffserfolg* gegenübergestellt werden. Da dieser durch den Anteil der korrekt zugeordneten Teilnehmer ausgedrückt wird, werden die beiden anderen Metriken in eine entsprechende Form umgewandelt: Die Anonymitätsgruppe gibt die Anzahl k derjenigen Teilnehmer an, die aus Sicht eines Angreifers nicht unterscheidbar sind. Müsste dieser raten, so hat er eine Chance von $\frac{1}{k}$, den korrekten Teilnehmer zu treffen. Entsprechend wird in den folgenden Untersuchungen der Kehrbruch der Größe der Anonymitätsgruppe berücksichtigt.

Die Entropie gibt den Informationsgehalt in Bit einer Verteilung an. Potenziert man 2 mit dem Wert der Entropie, so erhält man die effektive Anzahl an Elementen dieser

Verteilung unter der Annahme der Ununterscheidbarkeit. Analog den Überlegungen im Zusammenhang mit der Anonymitätsgruppe wird in den folgenden Betrachtungen anstelle der Entropie H selbst der folgende Wert herangezogen: $\frac{1}{2^H}$.

5.4.4.1. Mix-Zonen

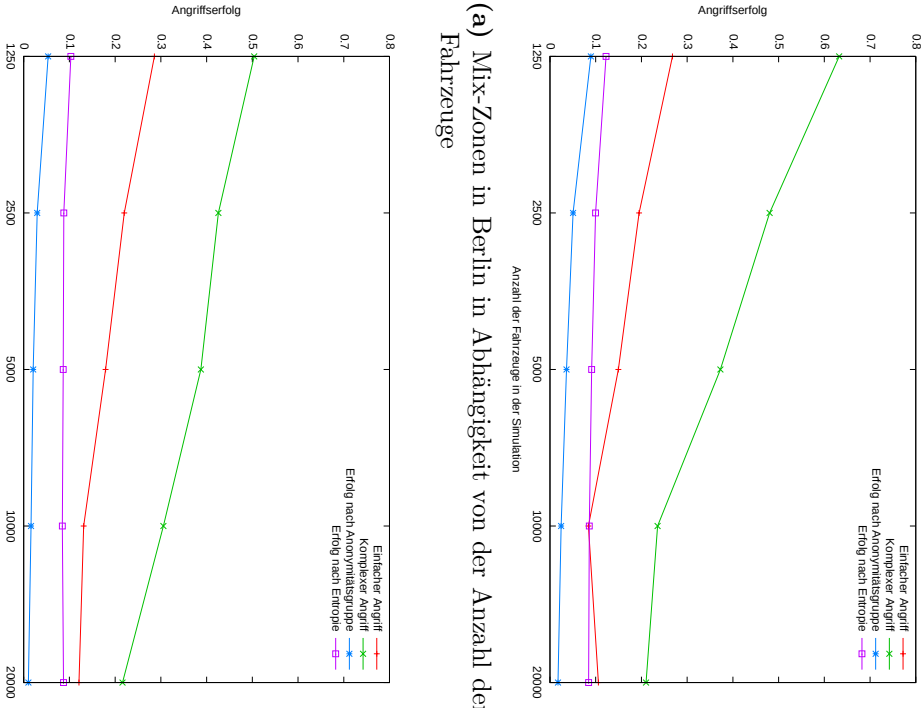
Überblick und Metriken Die Erfolgsraten der Angriffe auf Mix-Zonen (sowohl mit der einfachen als auch mit der komplexen Heuristik) sind in Abbildung 5.23 und Abbildung 5.24 zusammen mit den ermittelten Metriken *Anonymitätsgruppe* und *Entropie* in Abhängigkeit der untersuchten Einflussgrößen *Anzahl der Fahrzeuge* und *Größe der Mix-Zonen* in den drei Städten dargestellt.

Zuerst zeigt sich, dass der erwartete Zusammenhang zwischen der Anzahl an Fahrzeugen und dem Erfolg der Angriffe bestätigt werden konnte: Mehr Fahrzeuge verringern die Erfolgsaussichten eines Angreifers, der versucht, Pseudonyme vor und nach der Mix-Zone zu verketten. Ist die Verkehrsdichte größer, so kommen in der Regel mehr Fahrzeuge im Rahmen der Heuristik in Frage und die Fehlerrate steigt. Zudem gibt es mehr Vermischungsereignisse (Spurwechsel, Einschervorgänge, etc.) innerhalb der Mix-Zone und somit außerhalb des Wahrnehmungsbereichs des Angreifers.

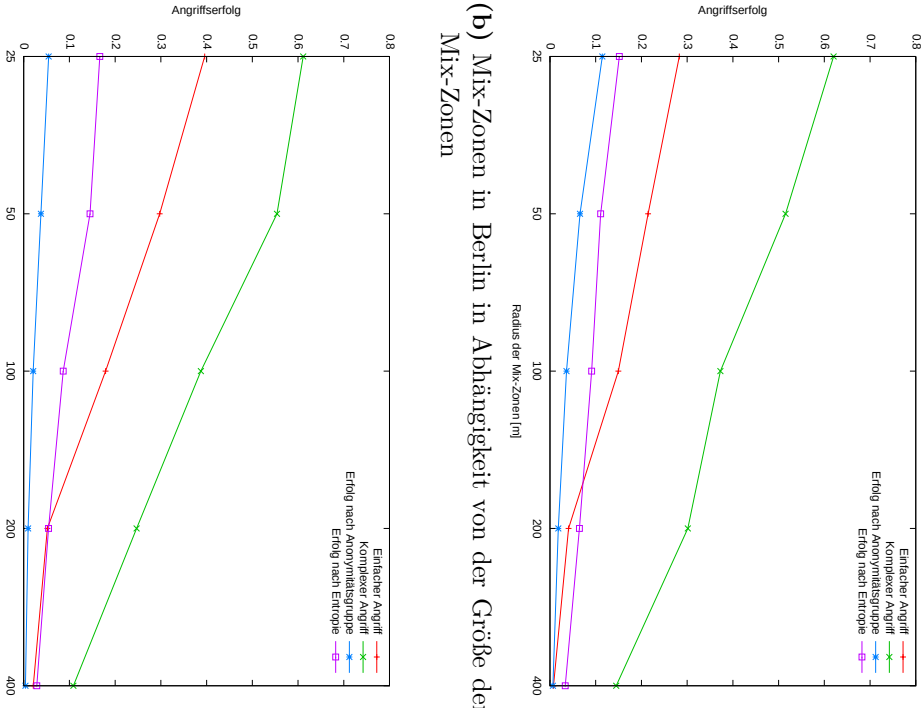
Auch die vermutete Beziehung zwischen der Größe der Mix-Zonen und dem Schutz, den sie bieten können, zeigt sich in den Simulationsergebnissen: Eine größere Mix-Zone kann den Teilnehmern mehr Schutz bieten, da die Wahrscheinlichkeit von Abweichungen von dem erwarteten Ausfahrtszeitpunkt mit einer längeren Durchfahrzeit steigt. Damit erhöht sich auch das Risiko für den Angreifer, eine falsche Zuordnung vorzunehmen. Weiterhin kann es vorkommen, dass bei einer größeren Ausdehnung der Mix-Zone weitere Ein- und Ausfahrten entstehen und somit wiederum ein größerer Unsicherheitsfaktor für den Angreifer hinzukommt.

Wesentliche Unterschiede zwischen den Städten konnten dagegen nicht nachgewiesen werden; die Kurven weisen einen relativ ähnlichen Verlauf auf. Die Unterschiede in den Angriffserfolgen können in dem Mix-Zonen-Szenario stark von der Platzierung der Zonen selbst abhängig sein, wodurch eine Aussage an dieser Stelle schwierig zu treffen ist.

Betrachtet man die einzelnen Kurven, so stellt man fest, dass – wie zu erwarten war – der komplexe Angriff stets eine höhere Erfolgsrate bietet. Der an dieser Stelle durch die Mix-Zonen realisierte Schutz der Teilnehmer (die Erfolgsrate bleibt ja stets unter 100 %) ist dabei jedoch als obere Grenze zu verstehen, da ein Angreifer noch deutlich raffiniertere Angriffsheuristiken entwickeln oder auch Kontextwissen in seine Verkettungsversuche einbeziehen kann.



(a) Mix-Zonen in Berlin in Abhängigkeit von der Anzahl der Fahrzeuge

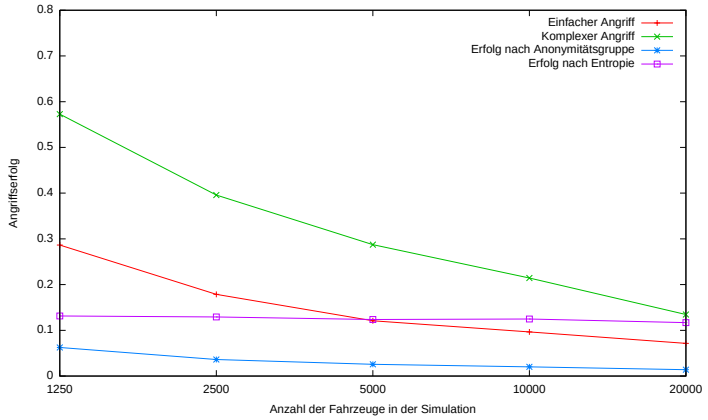


(b) Mix-Zonen in Berlin in Abhängigkeit von der Größe der Mix-Zonen

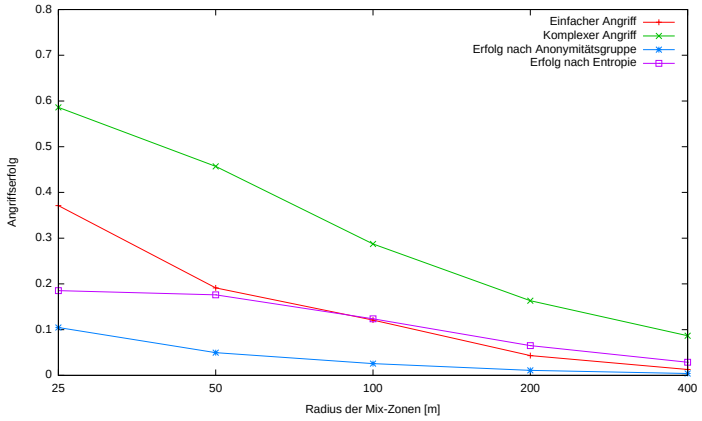
(c) Mix-Zonen in New York in Abhängigkeit von der Anzahl der Fahrzeuge

(d) Mix-Zonen in New York in Abhängigkeit von der Größe der Mix-Zonen

Abbildung 5.23.: Angriffserfolge und Metriken bei Mix-Zonen in Berlin und New York



(a) Mix-Zonen in Puebla in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Mix-Zonen in Puebla in Abhängigkeit von der Größe der Mix-Zonen

Abbildung 5.24.: Angriffserfolge und Metriken bei Mix-Zonen in Puebla

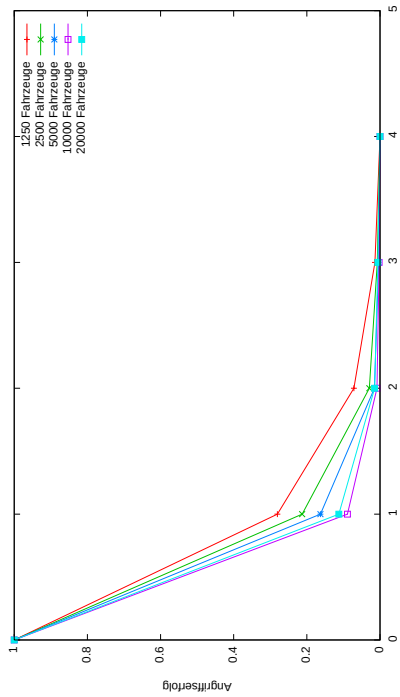
Die Betrachtungen der Anonymitätsgruppe und der Entropie führen zu einer Überschätzung der Effektivität der Mix-Zone und versprechen einen Schutz, der stets schlechter als der gegen den komplexen Angriff tatsächlich erreichte ist. Der einfache Angriff bewegt sich in etwa in der Größenordnung der Prognosen der beiden Metriken, unterschreitet sie jedoch nur sehr selten. Es ist also davon auszugehen, dass der mit Hilfe der Anonymitätsgruppe oder der Entropie abgeschätzte Schutz in der Praxis nicht erreichbar ist.

Verfolgung über mehrere Mix-Zonen hinweg Die Erkennungsraten nach der Durchfahrt einer einzelnen Mix-Zone sind zwar sehr hoch, doch das Konzept – wie Silent Periods und SLOW auch – vertraut darauf, dass dennoch ein ausreichendes Schutzniveau dadurch geboten werden kann, dass Teilnehmer eine Reihe von Mix-Zonen durchfahren. Theoretisch gilt, dass einem Angreifer mit einer Erfolgswahrscheinlichkeit der korrekten Verkettung der Pseudonyme eines Teilnehmers vor und nach dem Durchfahren einer Mix-Zone von p ($p < 1$) die Zuordnung nach n Zonen nur mit der Wahrscheinlichkeit p^n gelingt. Sie ist somit deutlich geringer als die zuvor erreichten Werte. Dieses Phänomen wurde ebenfalls in den Simulationen untersucht; die Ergebnisse für Berlin sind in Abbildung 5.25 und Abbildung 5.26 zusammengefasst. Die Ergebnisse der anderen Städte bieten ein sehr ähnliches Bild und sind im Anhang in Abschnitt C.1 zu finden.

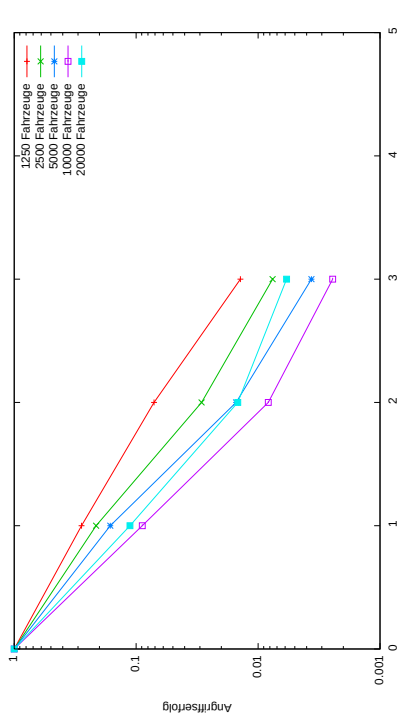
Der aus den theoretischen Überlegungen stammende logarithmische Verlauf der Erfolgsrate der Angriffe über mehrere Mix-Zonen hinweg zeigt sich in den Ergebnissen sehr deutlich. Daher sind alle Daten einmal linear und einmal logarithmisch skaliert dargestellt. Das Ende einer der Kurven bedeutet im linear skalierten Fall, dass in den Simulationen kein Fahrzeug eine höhere Anzahl an Mix-Zonen durchfahren hat. Das Erreichen der Nulllinie ist so zu interpretieren, dass von den noch vorhandenen Teilnehmern, die eine gewisse Anzahl an Zonen durchfahren haben, kein einziger erfolgreich zugeordnet werden konnte. In der logarithmisch skalierten Abbildung hingegen ist die Nulllinie nicht definiert, daher endet hier die Kurve auch in diesem zweiten Fall.

Der Verlauf der Kurve entspricht in jeder Hinsicht den Erwartungen und naheliegender Weise erzielt auch hier der komplexe Angriff deutlich bessere Angriffserfolge als der einfache. Bereits beim Durchfahren von drei Mix-Zonen konnten in allen Fällen die Erfolgsraten auf unter 10 % gedrückt werden, nach vier Zonen war eine korrekte Zuordnung kaum mehr möglich. Leider durchfuhren die Fahrzeuge aufgrund der wenig ausgereiften Positionierung keine wesentlich größere Anzahl an Zonen; hier könnte die bereits oben erwähnte gezielte Platzierung den Schutz noch etwas erhöhen.

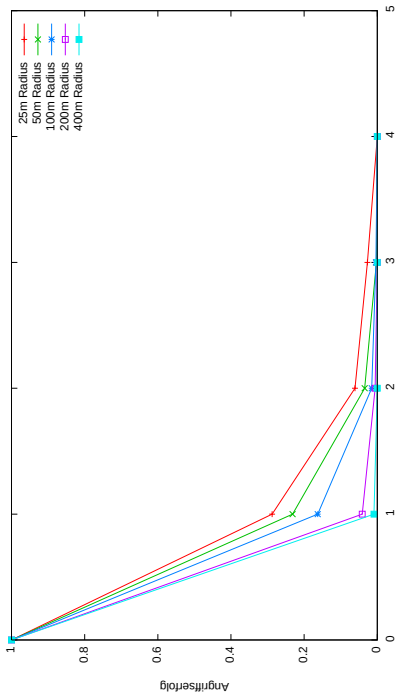
Bei einer größeren Anzahl an Mix-Zonen weisen einige Kurven überraschende Ver-



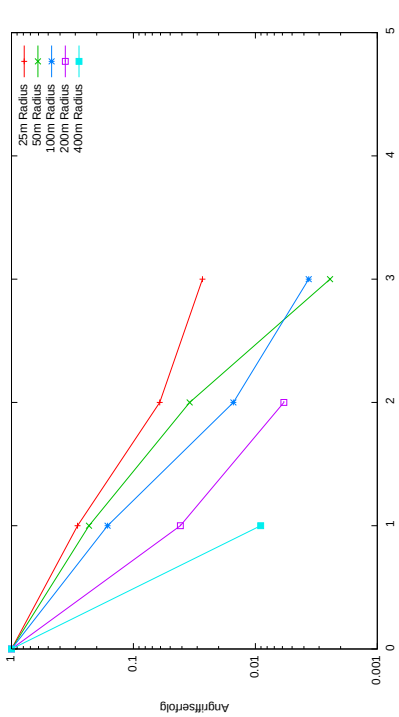
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

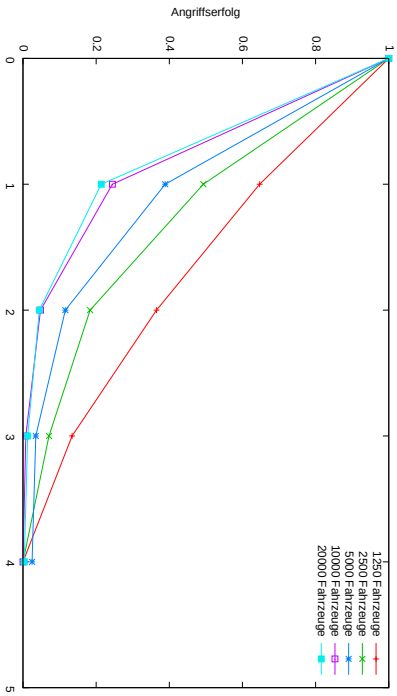


(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen

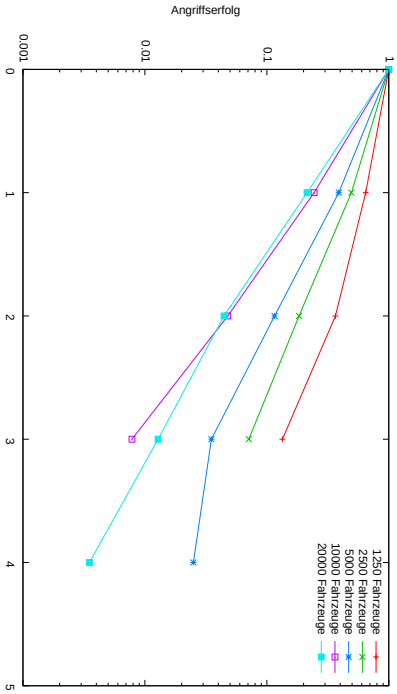


(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

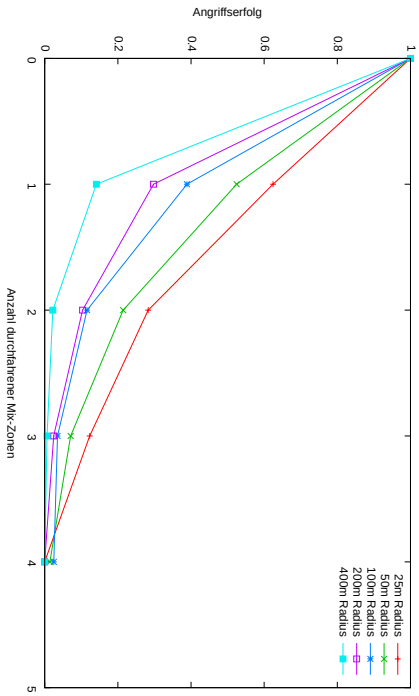
Abbildung 5.25: Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in Berlin



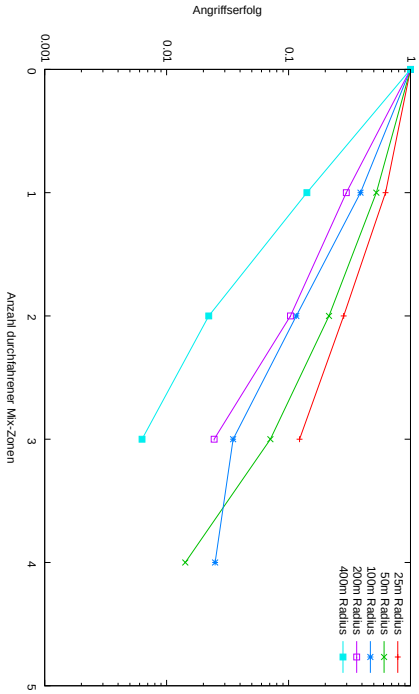
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen



(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

Abbildung 5.26.: Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in Berlin

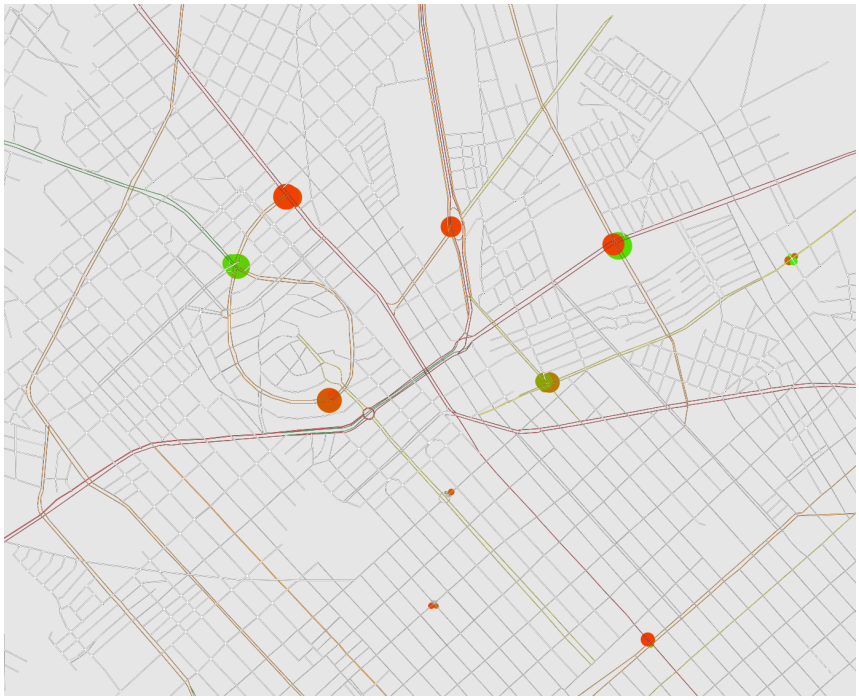
läufe auf. So ist beispielsweise in Abbildung 5.25d die Erfolgsrate bei Mix-Zonen der Größe 100 m höher als bei Zonen der Größe 50 m. Diese und ähnliche Ausreißer können dadurch entstehen, dass teilweise nur eine sehr geringe Anzahl an Teilnehmern überhaupt eine entsprechende Anzahl an Mix-Zonen durchfahren hat und somit die Aussagekraft der Daten an dieser Stelle leidet. Eine Interpretation dieser auffälligen Kurvenverläufe ist also nicht sinnvoll.

Positionierung von Mix-Zonen Betrachtet man die Mix-Zonen im Detail, so besteht die Möglichkeit, aus den Simulationsergebnissen Schlüsse für eine günstige Platzierung der Zonen zu ziehen. Generell ergibt sich aus den zuvor angestellten Betrachtungen, dass die Größe der Mix-Zone einen entscheidenden Einfluss auf den durch sie bereitgestellten Schutz hat. Die folgenden Betrachtungen greifen einige Aspekte der Ergebnisse in Puebla und Berlin auf; insgesamt zeigt sich jedoch in allen Städten ein sehr ähnliches Bild. Die hier nicht berücksichtigten Abbildungen sind im Anhang in Abschnitt C.2 zu finden.

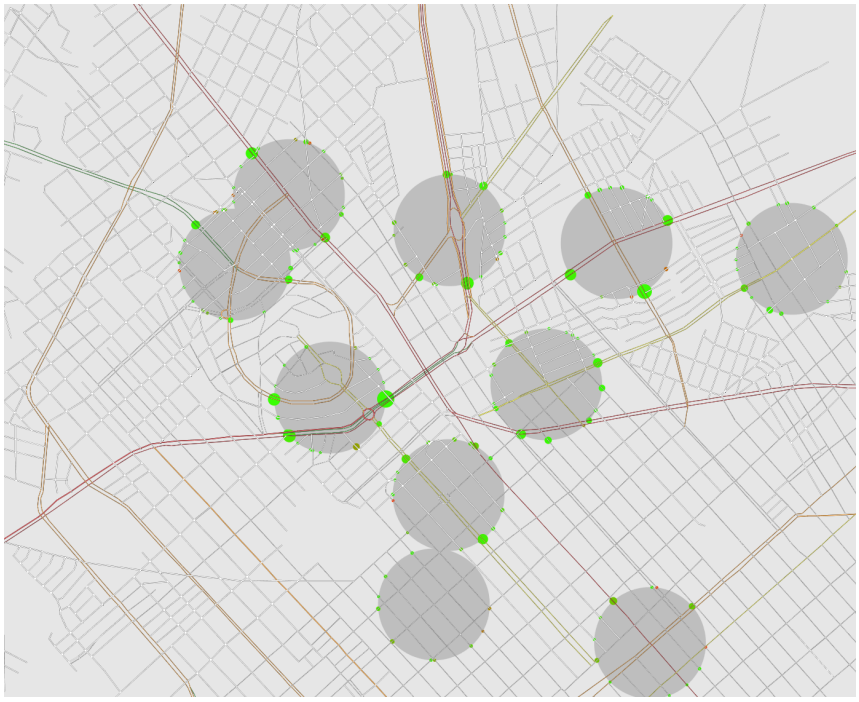
Abbildung 5.27 zeigt die in den Simulationen in Puebla platzierten Mix-Zonen (grau schattiert), einmal mit einem Radius von 25 m und einmal mit 400 m. Die Größe der farbigen Kreise zeigt den relativen Anteil der an dieser Position in die Zone eingefahrenen Fahrzeuge an; die Farbe liefert die Information, wie gut die dort eingefahrenen Teilnehmer von einem Angreifer wiedergefunden werden konnten. Ein stärkerer Grünanteil bedeutet dabei eine geringe Erfolgsrate des Angreifers, ein stärkerer Rotanteil eine hohe. Aus der Abbildung wird sehr schnell ersichtlich, dass neben dem hohen Schutzniveau großer Zonen dort auch der Eingangsverkehr (durch die größere Anzahl an Zufahrten) gleichmäßiger verteilt wird. Somit entsteht eine höhere Unsicherheit für den Angreifer und entsprechend ein besserer Schutz der Teilnehmer.

Besonders interessant ist noch die westlichste Zone, da sie auch bei sehr geringer Größe (siehe Abbildung 5.27a) ein sehr gutes Schutzniveau bietet. An ihrer Position kreuzen sich zwei große Straßenzüge, und eine weitere große Verbindungsstraße mündet in der Nähe ein. In diesem Bereich scheint es zu einer Vielzahl an Abbiegevorgängen zu kommen, zudem können an dicht befahrenen Knotenpunkten leicht Staus entstehen. Diese Aspekte führen dazu, dass Angriffe nach der beschriebenen Heuristik kaum noch gute Ergebnisse erzielen können, da die Unvorhersehbarkeit besonders groß wird. Hier ist deutlich ersichtlich, dass eine Platzierung von Mix-Zonen an verkehrsreichen Kreuzungen sehr vielversprechend ist.

Abbildung 5.28 zeigt die simulierten Mix-Zonen in Berlin in Abhängigkeit von der Verkehrsdichte. Wiederum wird auf den ersten Blick deutlich, was aus den vorhergehenden Betrachtungen hervorging: Je dichter der Verkehr, desto schwieriger wird es für einen Angreifer, korrekte Zuordnungen vorzunehmen. Dies wird besonders an der leicht nordöstlich des Zentrums gelegenen Mix-Zone deutlich, in der sich die

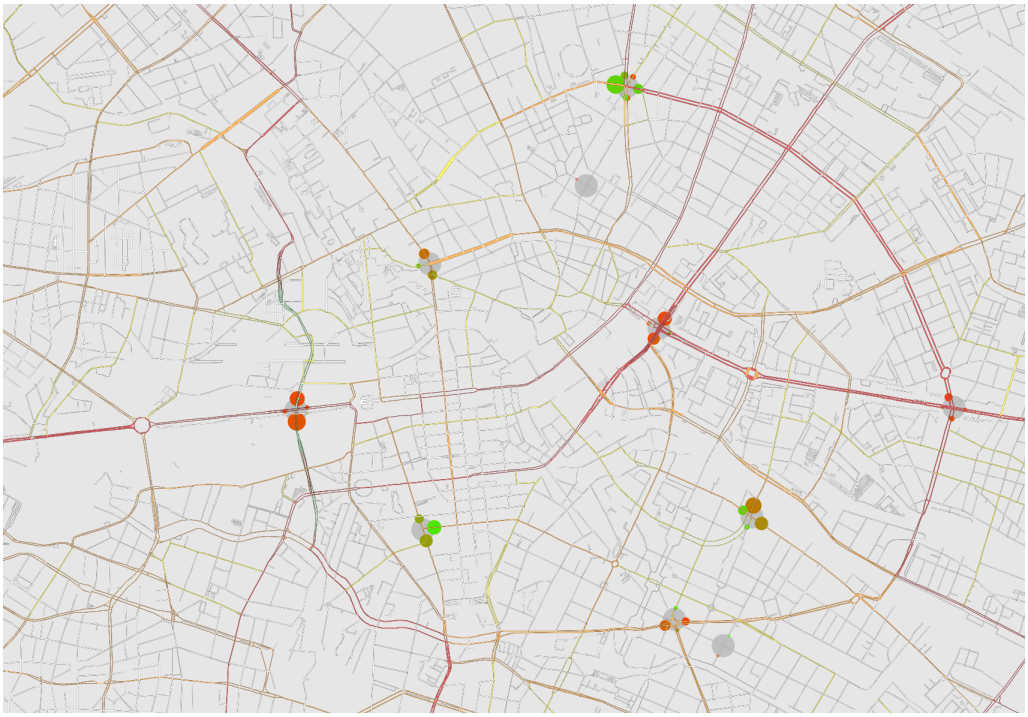


(a) Radius 25 m

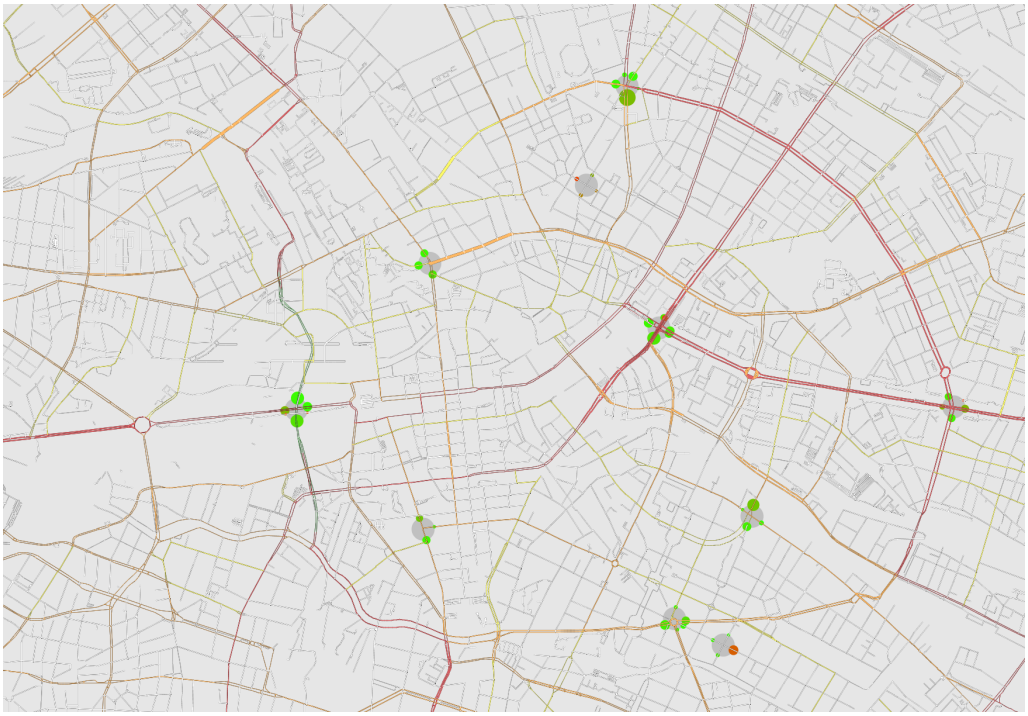


(b) Radius 400 m

Abbildung 5.27.: Überblick über die Mix-Zonen in Puebla in unterschiedlichen Größen

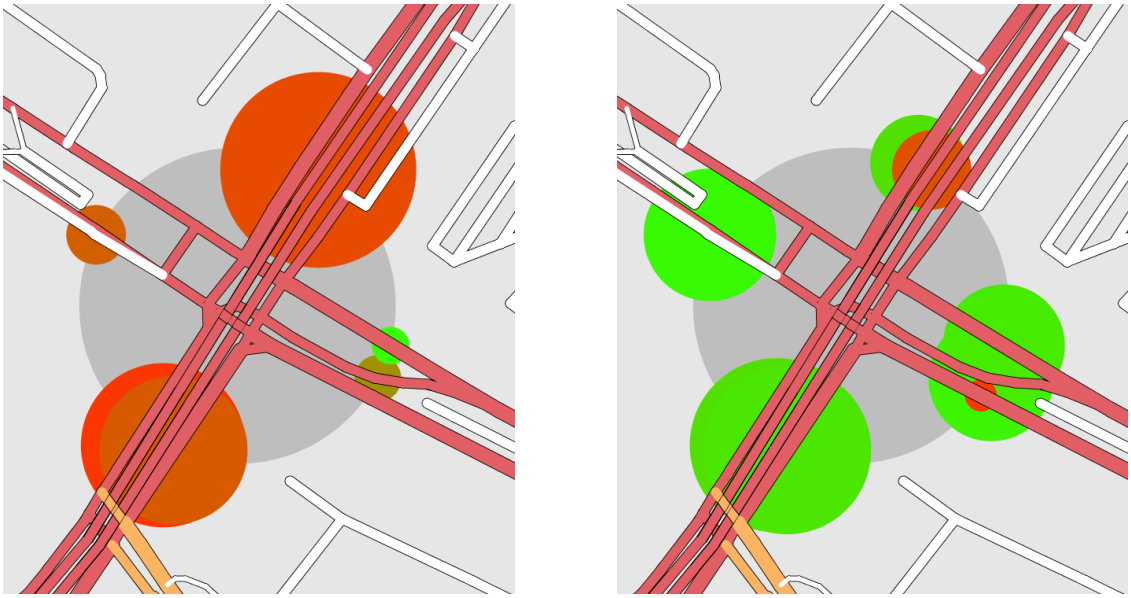


(a) 1250 Fahrzeuge in der Simulation



(b) 20 000 Fahrzeuge in der Simulation

Abbildung 5.28.: Überblick über die Mix-Zonen in Berlin bei unterschiedlichen Verkehrsdichten



(a) 1250 Fahrzeuge **(b)** 20 000 Fahrzeuge

Abbildung 5.29.: Detailansicht der Mix-Zone an der Otto-Braun-Straße in Berlin

Grunerstraße/Otto-Braun-Straße (südwest-nordost) mit der Karl-Marx-Allee/Alexanderstraße (südostost-nordwest) kreuzt. Dieser komplexe Verkehrsknotenpunkt ist im Detail in Abbildung 5.29 dargestellt.

Bei geringer Verkehrsdichte bewegt sich der größte Teil der Fahrzeuge entlang der Grunerstraße/Otto-Braun-Straße, und mangels Abbiegevorgängen oder Verzögerungen sind die Teilnehmer sehr leicht verfolgbar. Bei einer größeren Anzahl an Fahrzeugen in der Simulation verteilt sich der Verkehr gleichmäßiger auf die Einfahrten der Mix-Zone. Darüberhinaus kommt es zu mehr Abbiegevorgängen und den daraus entstehenden Verzögerungen durch Einfädelvorgänge. Somit steigt der Schutz der Zone deutlich an und die meisten Angriffe sind nicht von Erfolg gekrönt. Der große rote Punkt der erfolgreich angegriffenen Teilnehmer, die von Nordosten auf der zweiten Fahrbahn von links in die Zone einfahren, ist dadurch bedingt, dass es für die entsprechenden Fahrzeuge keine Abbiegemöglichkeiten gibt und bei gutem Verkehrsfluss somit die Zuordnung durch den Angreifer sehr einfach möglich ist.

Aus diesen Betrachtungen geht hervor, dass einerseits die Ausdehnung der Mix-Zone aus Datenschutzsicht möglichst groß gewählt werden sollte und andererseits die Positionierung für ihre Effektivität eine entscheidende Rolle spielt. Konkrete Vorschläge dazu können jedoch nicht gemacht werden, da empirische Untersuchungen über den Verkehrsfluss auf den in Frage kommenden Kreuzungen notwendig sind, um den möglichen Schutz durch eine Mix-Zone einschätzen zu können. Große Knoten-

punkte weisen tendenziell günstigere Eigenschaften auf als kleinere Kreuzungen an Nebenstraßen. Weiterhin sollte darauf geachtet werden, dass die Ein- und Ausfahrten der Kreuzung möglichst gleichmäßig genutzt werden, damit der Angreifer mit einer größeren Unsicherheit zu kämpfen hat.

5.4.4.2. Silent Periods

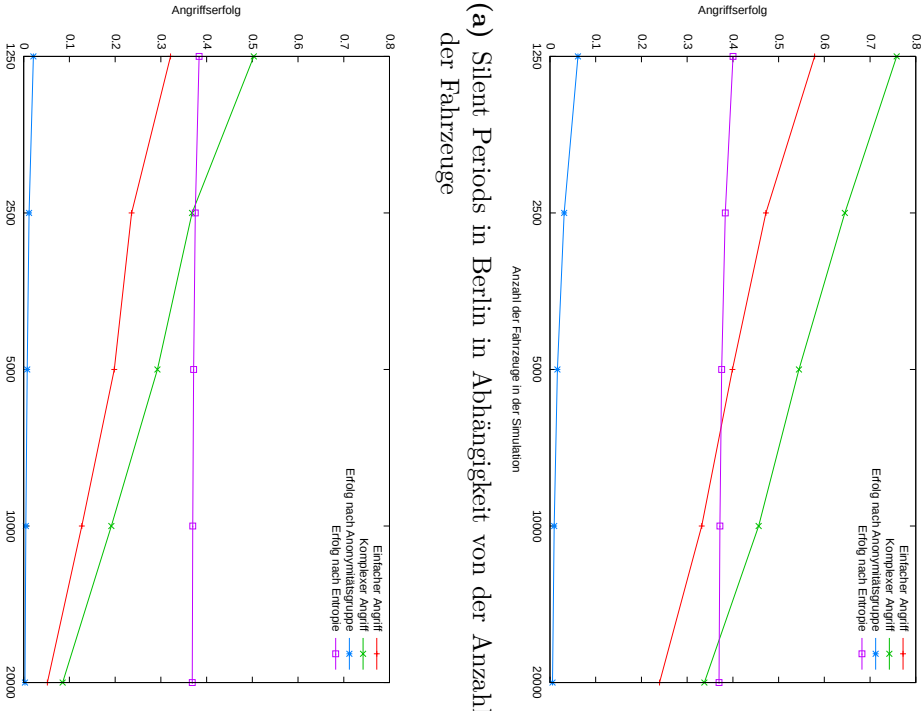
Überblick und Metriken Die Erfolgsraten der Angriffe auf Silent Periods (sowohl mit der einfachen als auch mit der komplexen Heuristik) sind in Abbildung 5.30 und Abbildung 5.31 zusammen mit den ermittelten Metriken *Anonymitätsgruppe* und *Entropie* in Abhängigkeit der untersuchten Einflussgrößen *Anzahl der Fahrzeuge* und *Dauer der Silent Periods* in den drei Städten dargestellt.

Wie bereits bei den Mix-Zonen zeigt sich auch bei den Silent Periods, dass eine größere Verkehrsdichte – also mehr Fahrzeuge in der Simulation – zu einer größeren Unsicherheit des Angreifers und somit zu einem besseren Schutz der Teilnehmer beiträgt. Auch hier bewegen sich im Rahmen der Heuristik des Angreifers mehr Fahrzeuge im Zielbereich, wodurch die Wahrscheinlichkeit steigt, dass eine falsche Zuordnung vorgenommen wird. Ebenfalls eine Rolle dürften wiederum die erhöhten Unwägbarkeiten während der Silent Period (und damit unsichtbar für den Angreifer) in Form von Verzögerungen durch Einschervorgänge und Ähnliches spielen.

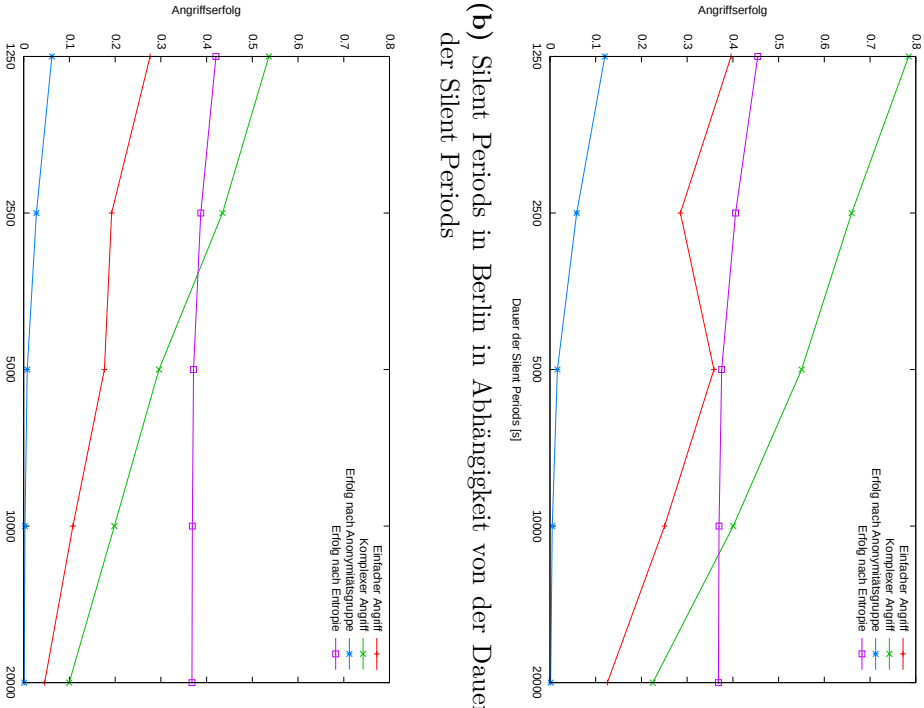
Die Dauer der Silent Periods weist ebenfalls den intuitiven Zusammenhang zu dem Angriffserfolg auf: je länger sich Fahrzeuge in der Funkstille unbeobachtbar durch den Angreifer bewegen, desto besser sind ihre Chancen, einer korrekten Verkettung ihrer Pseudonyme vor und nach der Silent Period zu entgehen. Allerdings weisen die Kurven an dieser Stelle einige Besonderheiten auf: Zum einen nimmt der Angriffserfolg nicht besonders stark bei einer Verlängerung der Silent Period ab und zum anderen fällt die Kurve der einfachen Angriffsheuristik nicht streng monoton.

Die erste Auffälligkeit lässt sich möglicherweise darauf zurückführen, dass nicht in allen Fällen eine Verlängerung der Funkstille zu einer höheren Unsicherheit des Angreifers führt, da sich viele Teilnehmer in dieser Zeit auf Straßenabschnitten bewegen können, die keinerlei Einmündungen oder Kreuzungen aufweisen. Ist die Straße dann noch einspurig, so ist eine Zuordnung in vielen Fällen trivial. Der Gesamttrend des Angriffserfolgs zeigt aber natürlich dennoch bei steigender Dauer der Funkstille nach unten, da ein größerer Anteil der Teilnehmer die Chance bekommt, sich unbeobachtbar durch eine Kreuzung zu bewegen und die dort bestehende Vermischung und Verzögerung zur Verschleierung der eigenen Position zu nutzen.

Die zweite Besonderheit, der relativ große Angriffserfolg bei einer Dauer der Silent Period von 5000 ms – der sogar größer ist als bei einer Dauer von nur 2500 ms –, ist



(a) Silent Periods in Berlin in Abhängigkeit von der Anzahl der Fahrzeuge

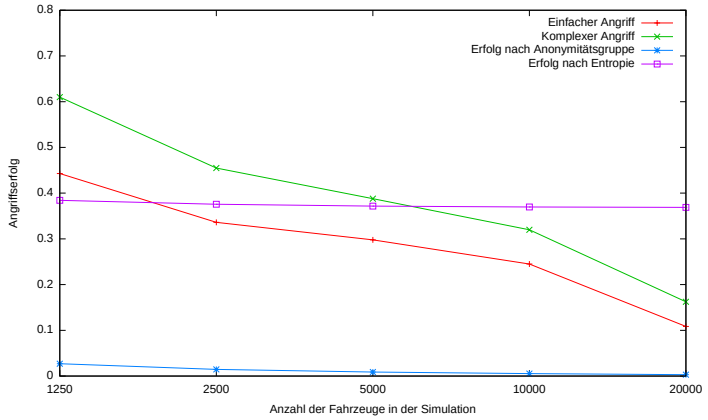


(b) Silent Periods in Berlin in Abhängigkeit von der Dauer der Silent Periods

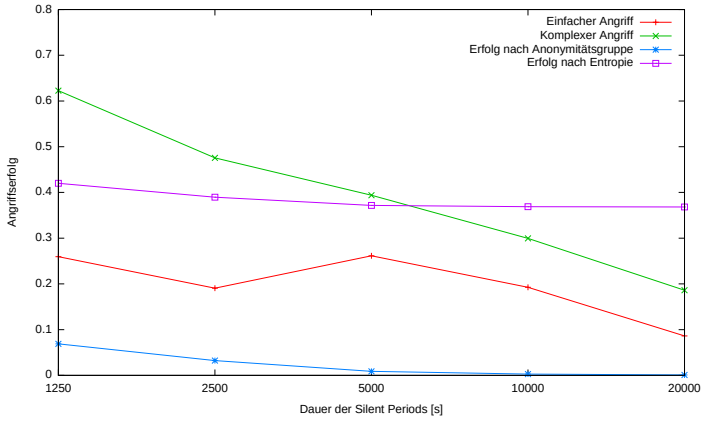
(c) Silent Periods in New York in Abhängigkeit von der Anzahl der Fahrzeuge

(d) Silent Periods in New York in Abhängigkeit von der Dauer der Silent Periods

Abbildung 5.30.: Angriffserfolge und Metriken bei Silent Periods in Berlin und New York



(a) Silent Periods in Puebla in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Silent Periods in Puebla in Abhängigkeit von der Dauer der Silent Periods

Abbildung 5.31.: Angriffserfolge und Metriken bei Silent Periods in Puebla

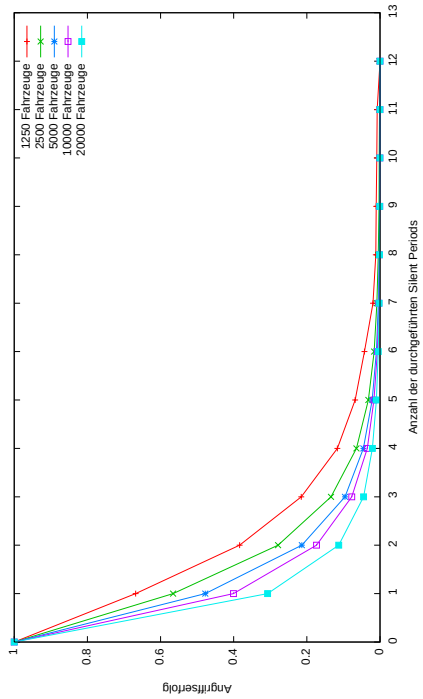
dagegen schwierig zu erklären. In den Simulationen hat sich dieser erhöhte Wert stets eingestellt und dürfte somit auf einer Besonderheit der Angriffsheuristik basieren. An dieser Stelle sind weitere Untersuchungen notwendig, um eine definitive Aussage treffen zu können.

Interessanterweise zeigen die Ergebnisse der einzelnen Städte gewisse Unterschiede. So weisen die Angriffe in Berlin deutlich höhere Erfolgsraten auf als in den anderen beiden Städten; die Ergebnisse in Puebla sind aus Angreifersicht etwas besser als die in New York. Möglicherweise zeigt sich hier ein Zusammenhang zwischen der Regelmäßigkeit des Straßennetzes und der Erfolgsrate bei der Verkettung von Pseudonymen vor und nach einer Silent Period. Dabei würde ein regelmäßiges Straßennetz zu einem besseren Schutz führen. Möglicherweise führt dieses zu einer gleichmäßigeren Verteilung des Verkehrs und somit zu einer größeren Zahl an Abbiegevorgängen an unterschiedlichen Orten, die es dem Angreifer schwer machen, korrekte Zuordnungen zu bilden. Um dies abschließend zu klären, müssen weitere Betrachtungen in anderen Straßensystemen durchgeführt werden.

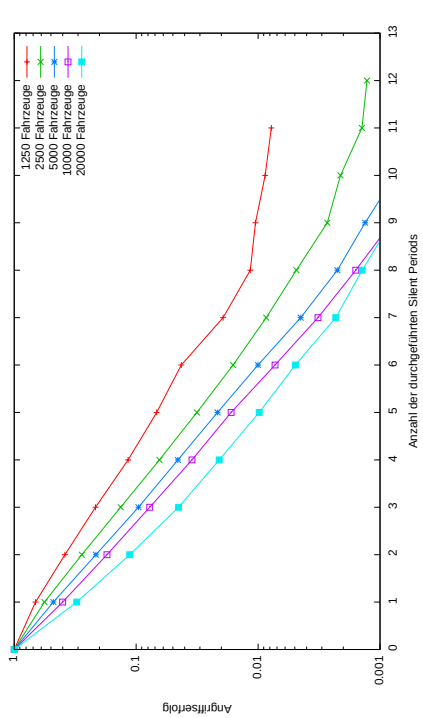
Der komplexe Angriff zeigt sich – wie zu erwarten – dem einfachen Angriff stets überlegen. Dies lässt darauf schließen, dass eine Berücksichtigung weiterer Aspekte der Bewegung der Teilnehmer die Erfolgsrate noch weiter vergrößert. Zudem kann der Angreifer möglicherweise seine Chancen verbessern, wenn er Kontextwissen (beispielsweise über die übliche Route seines Opfers) in die Betrachtungen mit einbezieht. Somit kann der durch die Kurve beschriebene Schutz – ebenso wie bei den Mix-Zonen – nur eine untere Grenze anzeigen.

Die oben beschriebenen Metriken *Anonymitätsgruppe* und *Entropie* stellen sich bei den Silent Periods als nicht besonders gut geeignet heraus. Die Anonymitätsgruppe überschätzt den möglichen Schutz dramatisch und bleibt somit hinter den tatsächlich ermittelten Angriffserfolgen deutlich zurück. Für eine aussagekräftigere Berechnung dieser Gruppe müssten also weitere Einschränkungen getroffen werden. Die Entropie dagegen bewegt sich mit ihren Werten im Bereich der Erfolgsraten der Angriffe, wird aber durch die Parameter in ihrer Größe kaum beeinflusst und bildet somit ebenfalls den Verlauf der Angriffe nur sehr unzureichend ab. Auch hier ist es notwendig, eine andere Definition zu finden, die die Simulationsergebnisse besser abbildet.

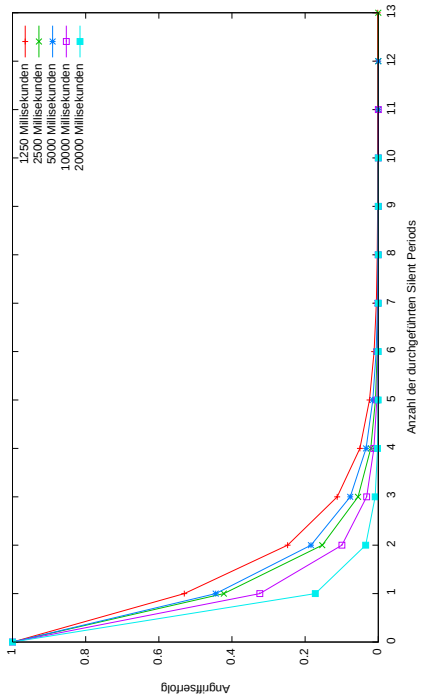
Verfolgung über mehrere Funkstillen hinweg Mehr noch als bei den Mix-Zonen vertraut man bei dem Konzept der Silent Periods darauf, dass Fahrzeuge eine Reihe von Funkstillen durchführen und es Angreifern somit nicht gelingt, eine Verfolgung über längere Zeit aufrechtzuerhalten. Die Untersuchungen über die korrekten Zuordnungen über mehrere Funkstillen hinweg sind in Abbildung 5.32 und Abbildung 5.33 für Berlin dargestellt. Da die Ergebnisse der anderen Städte eine große Ähnlichkeit aufweisen, sind die Diagramme im Anhang in Abschnitt C.3 zu finden.



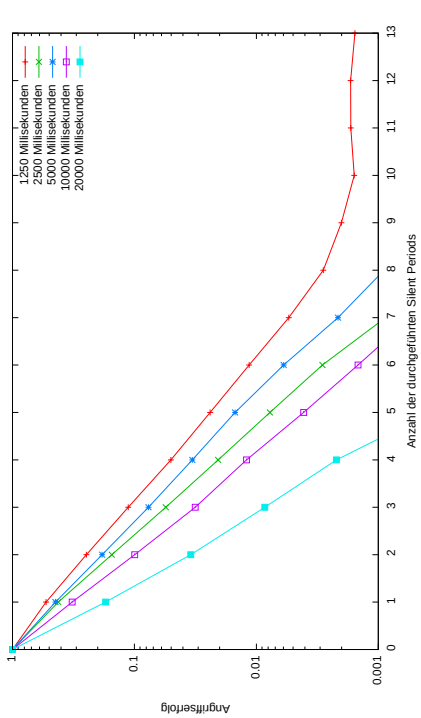
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

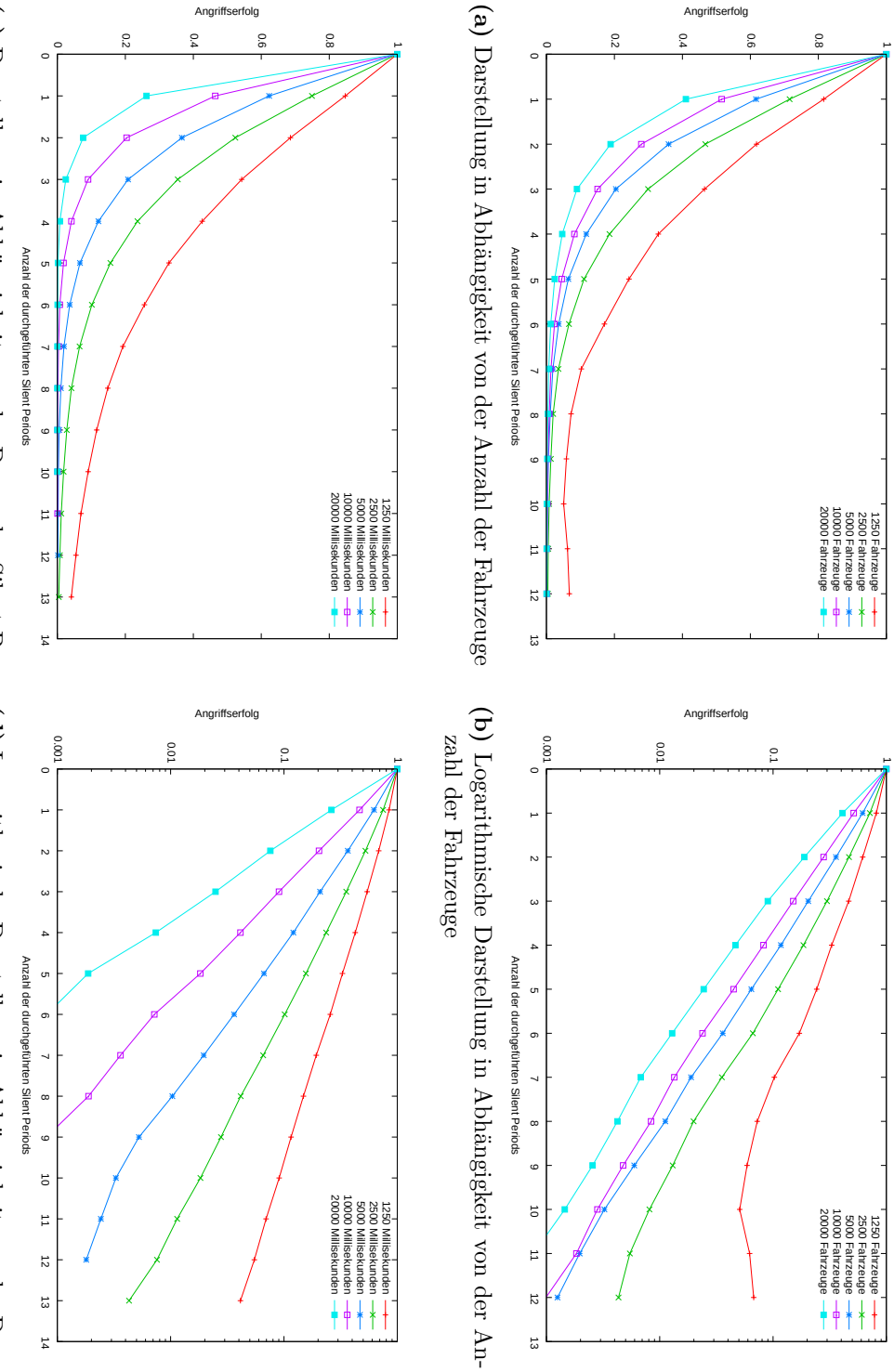


(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods



(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

Abbildung 5.32.: Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in Berlin



(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods

(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

Abbildung 5.33.: Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in Berlin

Der erwartete logarithmische Verlauf des Angriffserfolgs zeigt sich auch hier sehr deutlich. Wiederum sind die Diagramme einmal linear und einmal logarithmisch skaliert abgebildet, um einen Überblick über alle Informationen zu ermöglichen. Ein Wert von null im linear skalierten Diagramm bedeutet, dass kein Fahrzeug nach der entsprechenden Anzahl an Silent Periods korrekt zugeordnet werden konnte, das Ende einer Kurve zeigt an, dass kein einziges Fahrzeug eine größere Anzahl an Silent Periods durchlaufen hat. In dem logarithmisch skalierten Diagramm endet in beiden Fällen die Kurve, da eine Nulllinie hier nicht definiert ist.

Es fällt sofort auf, dass deutlich mehr Silent Periods durchlaufen als bei den anderen Simulationen Mix-Zonen durchfahren wurden. Dies liegt nahe, da bei Silent Periods keine Ortsgebundenheit vorliegt und die Funkstillen in regelmäßigen Abständen von allen Teilnehmern durchgeführt werden. Dennoch wirkt – auf Grund der bereits vielfach beschriebenen Einschränkungen in Hinblick auf den Straßenverlauf – die einzelne Silent Period nicht so effektiv wie eine Mix-Zone, da hier eine Zuordnung teilweise noch nach zehn Funkstillen absolut möglich war. Immerhin gelingt es in der Regel nach fünf bis sechs Funkstillen, die Erfolgsrate auf unter 10 % zu drücken. Besonders drastisch fällt der Unterschied zwischen dem einfachen und dem komplexen Angriff ins Auge, da mit letzterem eine Verfolgung über teilweise deutlich längere Distanzen möglich ist als in einem vergleichbaren Szenario des einfachen Angriffs. Da jedoch nicht vorhersagbar ist, wo sich ein Teilnehmer zum Zeitpunkt der Funkstille aufhält, ist eine Optimierung des Schutzes nur über die Häufigkeit und die Dauer der Silent Periods möglich.

Die Kurven weisen an einigen Stellen bei einer sehr hohen Anzahl an durchgeführten Silent Periods überraschende Verläufe auf: sie steigen an. Dies ist sehr einfach dadurch zu erklären, dass nur eine geringe Anzahl an Teilnehmern entsprechend viele Funkstillen durchgeführt hat und somit einige Ausreißer entstehen können. Die Daten sind daher an diesen Stellen nicht besonders aussagekräftig.

Auswirkungen des Straßenverlaufs auf die Unverkettbarkeit Die mehrfach geäußerte Vermutung, die Effektivität des Schutzes während einer Silent Period hänge sehr stark von der aktuellen Position des Teilnehmers ab, soll genauer untersucht werden. Dazu werden Karten der Städte herangezogen, in denen jeder Eintritt eines Teilnehmers in eine Funkstille mit einem kleinen Kreis markiert ist, wobei dieser in rot, wenn ein Angreifer das Pseudonym verketten konnte, oder in grün, wenn der Angriff auf das Fahrzeug misslungen ist, eingefärbt ist. Die folgenden Betrachtungen beziehen sich in einem Aspekt auf New York, in einem weiteren auf Puebla. Allerdings schränkt dies nicht die Allgemeingültigkeit der Aussagen ein, da alle Städte ein ähnliches Bild zeigen. Die nicht detailliert betrachteten Abbildungen sind im Anhang in Abschnitt C.4 zu finden.

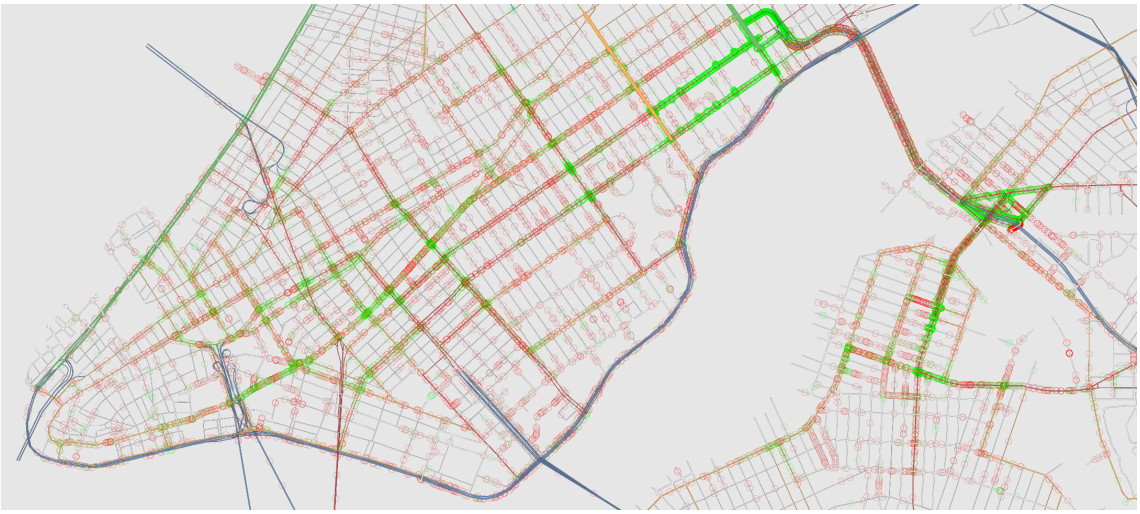
Abbildung 5.34 stellt den Erfolg und Misserfolg von Angriffen auf Teilnehmer bei unterschiedlichen Verkehrsdichten dar. Wiederum zeigt sich deutlich, dass eine höhere Verkehrsdichte – durch eine größere Anzahl an Fahrzeugen in der Simulation – zu einem besseren Schutz vor Angriffen führt. Interessant sind jedoch die wenigen grünen Bereiche in Abbildung 5.34a und die roten Bereiche in Abbildung 5.34b.

Bei den geringen Verkehrsdichten der ersten Abbildung zeigen sich überwiegend rote Bereiche, jedoch finden sich insbesondere zentral im nördlichen Bereich des Kartenausschnitts sowie in einem großen Kreuzungsbereich im Osten grüne Stellen nahezu ohne erfolgreiche Angriffe. Beobachtet man die Simulation mit der graphischen Benutzeroberfläche des *VANET-Simulators*, so stellt man fest, dass sich in diesen Bereichen sehr leicht Staus bilden. Die Angriffsheuristiken sind für diesen Fall jedoch nicht optimiert und eine korrekte Zuordnung gelingt daher dort nur äußerst selten. Ein intelligenterer Angreifer könnte den aktuellen Verkehrsfluss in diesen Abschnitten besser berücksichtigen und würde vermutlich auch dort mit größerem Erfolg Pseudonyme verketten können. Ansonsten erkennt man, dass nur in den Bereichen von Kreuzungen auf Hauptverkehrsstraßen (z. B. dem Broadway) ein ausreichender Schutz geboten wird. Das liegt vermutlich an den Abbiegevorgängen, die auf Grund der Heuristik (vgl. Abschnitt 5.4.2.2) schlecht erfasst werden können.

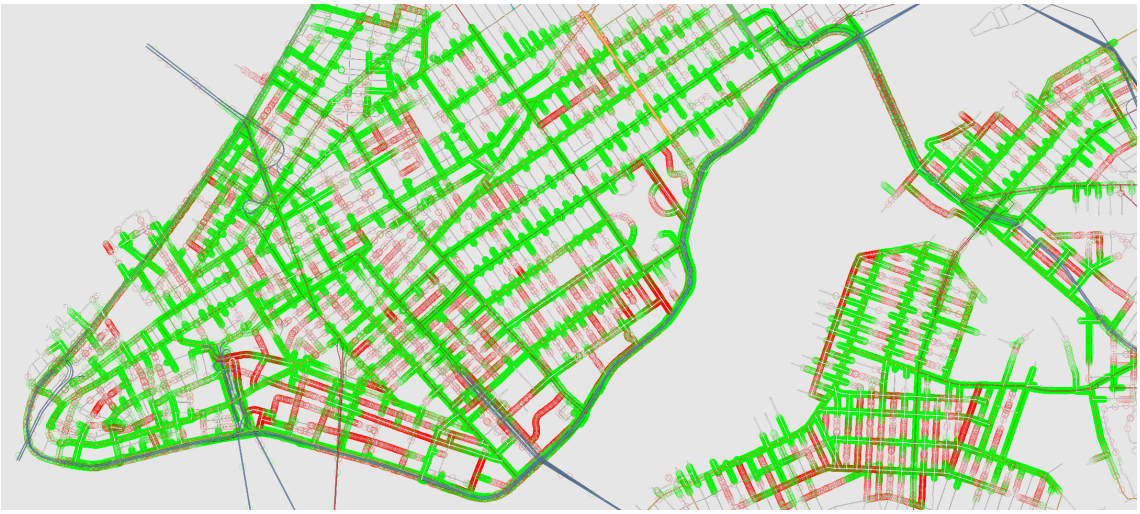
Abbildung 5.34b zeigt eine Situation mit recht hoher Verkehrsdichte, die insbesondere auf den Hauptverbindungsstrecken zu einem sehr guten Schutz vor Angriffen führt. Einige Nebenstraßen sind nur sehr schwach befahren, daher zeigt sich dort ein ähnliches Bild wie in weiten Teilen von Abbildung 5.34a. Die roten Stellen bei hoher Verkehrsdichte sind jedoch nicht einfach zu erklären. Die meisten betroffenen Straßenabschnitte stellen Ausweichmöglichkeiten der immer wieder mit Staus belasteten Hauptrouten dar. Fahrer, die dorthin ausweichen, kommen in der Regel recht gut voran und entsprechen damit den Erwartungen des Angreifers, was sich in dessen hohen Erfolgsraten widerspiegelt. Der Verkehr in diesen Nebenstraßen wird aber meist nicht dicht genug, um den Teilnehmern Schutz zu bieten.

Diese Einflussfaktoren des Straßennetzes auf die Effektivität der Silent Periods wird auch in Puebla deutlich (vgl. Abbildung 5.35). Bei diesen Simulationen bleibt die Verkehrsdichte konstant und stattdessen wird die Dauer der Silent Periods variiert. Wiederum zeigt sich, dass eine sehr kurze Funkstille nur einen geringen Schutz bietet und somit ein Großteil der Angriffe gelingt; bei länger andauernden Silent Periods dagegen verbessert sich der Schutz deutlich.

Die wenigen grünen Bereiche – also Regionen, in denen Angriffe überwiegend erfolglos verlaufen sind – zeigen sich fast ausschließlich in Kreuzungsbereichen. Hier sind die Teilnehmer, die nicht erfolgreich wiedergefunden werden konnten, schlichtweg abgebogen und die Heuristik hat sie, verstärkt durch die in Kreuzungssituationen vorkommende Verzögerung, nicht korrekt zuordnen können. Die drei geraden Ab-



(a) 1250 Fahrzeuge



(b) 20 000 Fahrzeuge

Abbildung 5.34.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in New York bei unterschiedlichen Verkehrsdichten

schnitte, die auch ein überwiegend grünes Bild abgeben, sind – ähnlich der Situation in New York – sehr stauanfällig und, wie oben beschrieben, versagt die Angrifssheuristik in diesem Fall sehr häufig. Diese beiden Aspekte zeigen jedoch, dass eine Verbesserung der Heuristik die Erfolgsrate der Angriffe höchstwahrscheinlich deutlich erhöhen könnte, da ein Angreifer lediglich die Abbiegevorgänge oder Staus besser berücksichtigen müsste. Insbesondere Letztere dürften durch das geringe Vorankommen und die seltene Durchmischung der Fahrzeuge sehr leicht abzudecken sein.

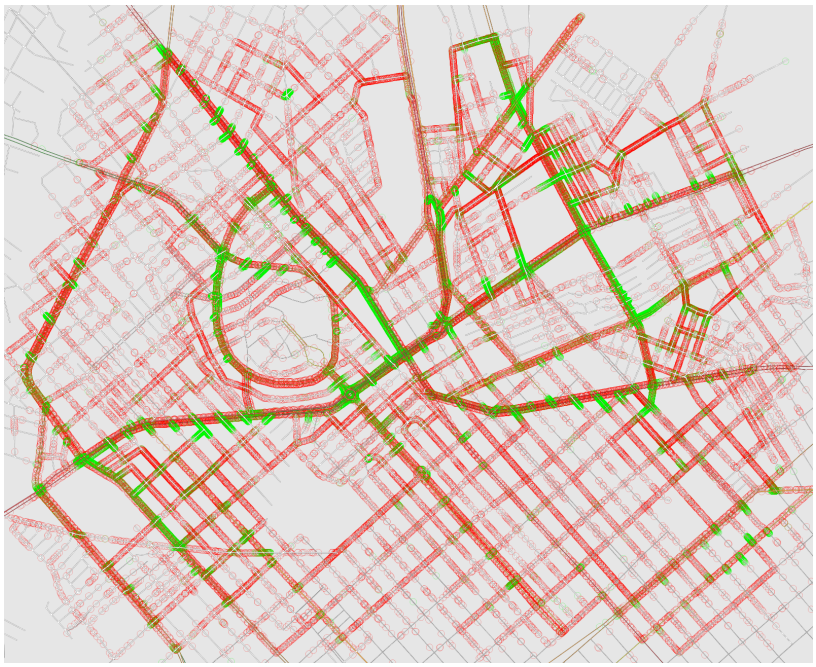
In Puebla neigen die Fahrzeuge in der Simulation dazu, die Nebenstraßen zu meiden und stattdessen die Hauptstraßen zu nutzen. Dies führt in Abbildung 5.35b dazu, dass selbst bei einer großen Anzahl an Teilnehmern auf vielen Nebenstraßen kein ausreichender Schutz geboten werden kann. Die großen Verkehrsadern erreichen dagegen eine sehr hohe Verkehrsdichte (die auch teilweise zu Staus führt), wodurch erfolgreiche Angriffe dort kaum durchgeführt werden.

5.4.4.3. SLOW

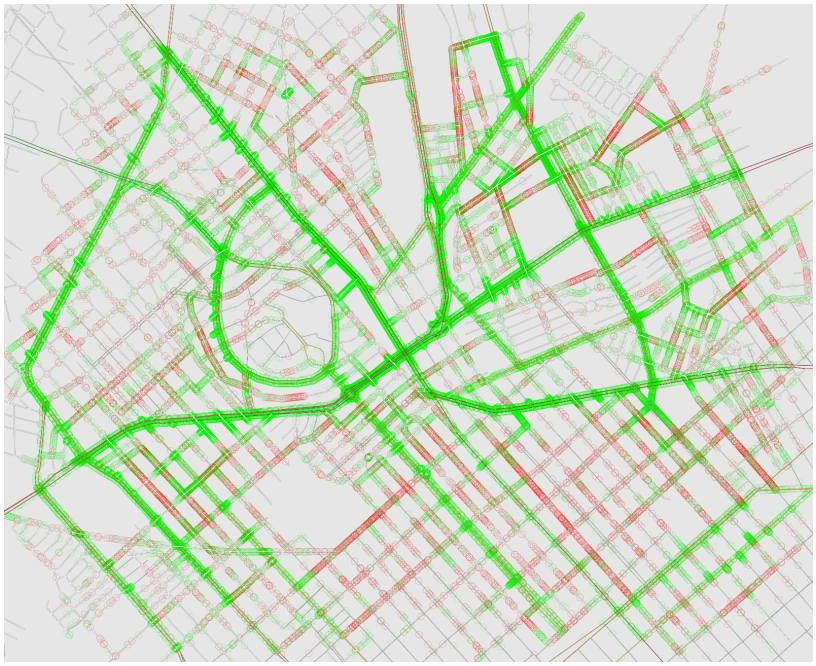
Überblick und Metriken Da bei SLOW die Metriken *Anonymitätsgruppe* und *Entropie* nicht ermittelt werden können (vgl. Abschnitt 5.4.3), stellen Abbildung 5.36 und Abbildung 5.37 lediglich die Erfolgsraten der Angriffe auf Funkstillen bei SLOW (sowohl mit der einfachen als auch mit der komplexen Heuristik) in Abhängigkeit der untersuchten Einflussgrößen *Anzahl der Fahrzeuge* und *Schwellengeschwindigkeit* in den drei Städten dar. In den folgenden Betrachtungen werden nur Funkstillen (und die darauf ausgeführten Angriffe) berücksichtigt, die lange genug angehalten haben, dass ein Pseudonymwechsel möglich war. Bei SLOW wechseln Teilnehmer erst dann ihr Pseudonym, wenn sie eine bestimmte Zeit lang die Schwellengeschwindigkeit unterschreiten und somit in Funkstille zubringen können (vgl. Abschnitt 4.4.2). Andernfalls wäre eine Zuordnung durch den Angreifer trivial. In den durchgeführten Simulationen betrug diese Zeitspanne eine Sekunde.

Bei der Betrachtung der Angriffserfolge bei SLOW zeigt sich zwar erneut der Zusammenhang zwischen der Verkehrsdichte und dem Schutz vor Angriffen, jedoch ist dieser Zusammenhang hier deutlich schwächer ausgeprägt als bei Mix-Zonen oder Silent Periods. Der durch SLOW bereitgestellte Schutz ist deutlich größer als bei den anderen Konzepten und wird bei einer Zunahme der Verkehrsdichte nur in geringem Maße vergrößert. Teilnehmer sind hier offenbar ohnehin kaum auffindbar, so dass sich der Einfluss der anderen Fahrzeuge in Grenzen hält.

Interessanter stellt sich der Sachverhalt jedoch bei der Betrachtung der Schwellengeschwindigkeit dar, bei deren Unterschreitung die Teilnehmer Funkstille halten. So liegt das Maximum bei dieser Kurve stets bei einer Geschwindigkeit von 20 $^{\mathrm{km}}/_{\mathrm{h}}$ und bietet somit in diesem Fall sogar einen schlechteren Schutz als selbst bei einer

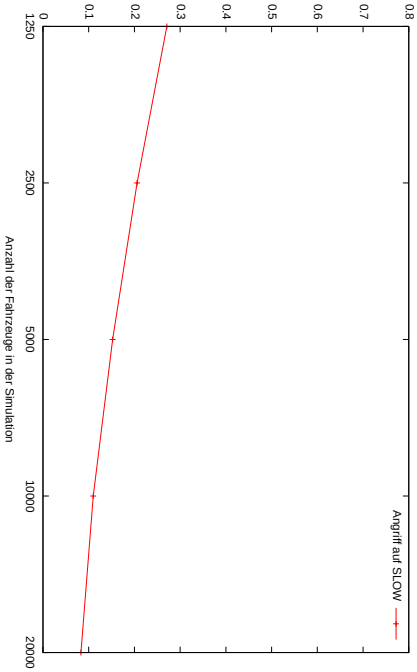


(a) 1250 ms

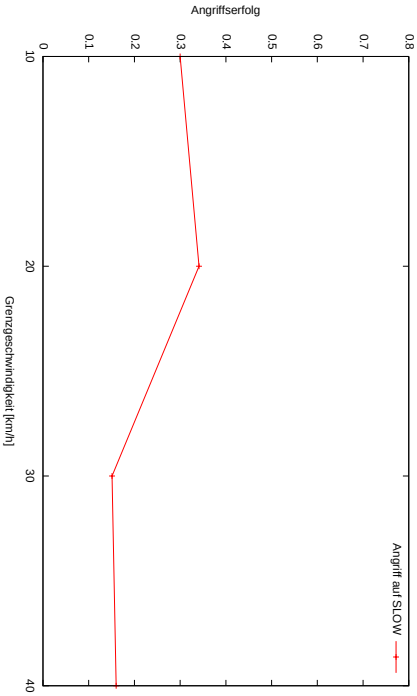


(b) 20 000 ms

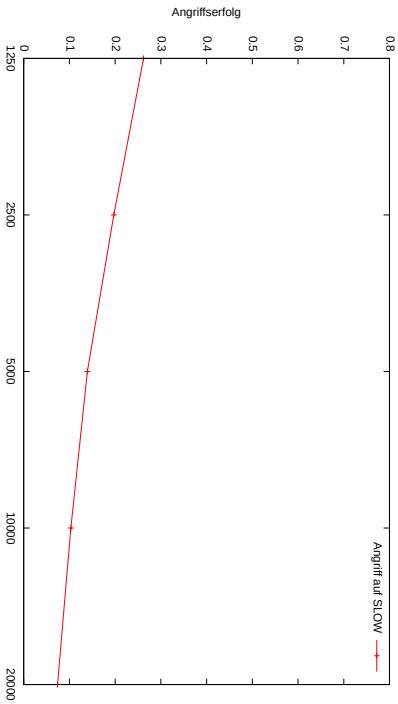
Abbildung 5.35.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in Puebla bei unterschiedlichen Längen der Silent Periods



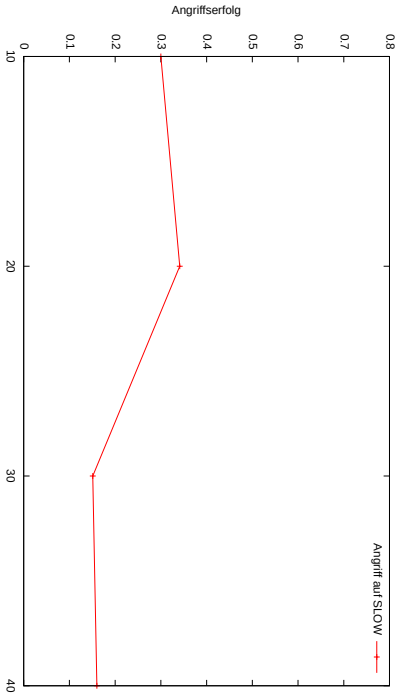
(a) SLOW in Berlin in Abhängigkeit von der Anzahl der Fahrzeuge



(b) SLOW in Berlin in Abhängigkeit von der Schwellengeschwindigkeit

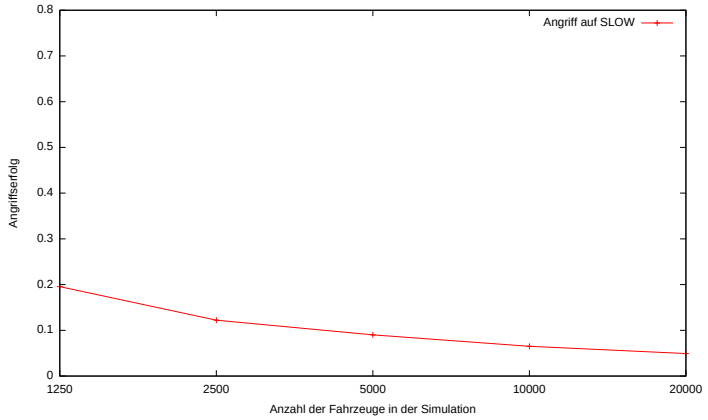


(c) SLOW in New York in Abhängigkeit von der Anzahl der Fahrzeuge

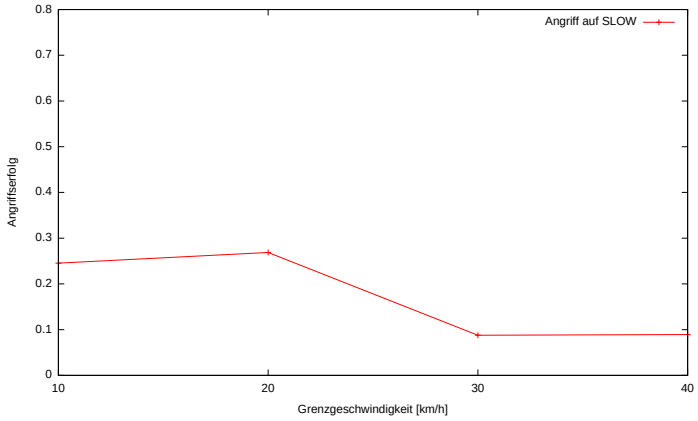


(d) SLOW in New York in Abhängigkeit von der Schwellengeschwindigkeit

Abbildung 5.36.: Angriffserfolge bei SLOW in Berlin und New York



(a) SLOW in Puebla in Abhängigkeit von der Anzahl der Fahrzeuge



(b) SLOW in Puebla in Abhängigkeit von der Schwellengeschwindigkeit

Abbildung 5.37.: Angriffserfolge bei SLOW in Puebla

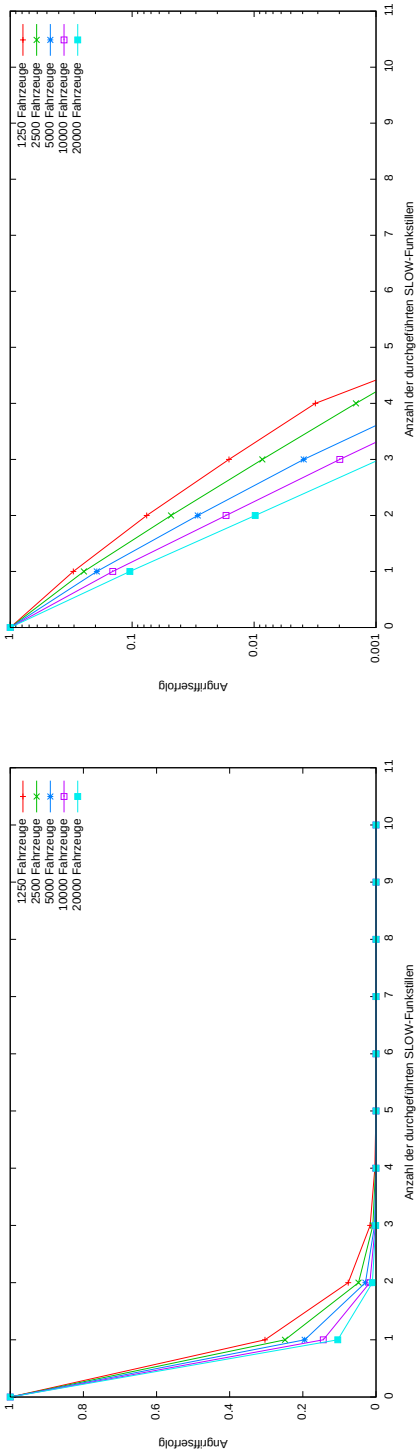
Grenzgeschwindigkeit von 10 km/h. Dies kann möglicherweise damit erklärt werden, dass Fahrzeuge nur in ganz seltenen Fällen (beispielsweise direkt beim Abbiegen oder wenn sie an einer Kreuzung Vorrang gewähren müssen) 10 km/h unterschreiten und es gerade in diesen Situationen einem Angreifer besonders schwerfällt, Verkettungen herzustellen. Somit wäre es denkbar, dass eine Geschwindigkeit von 20 km/h deutlich öfter und daher auch an Orten oder in Situationen, die keinen guten Schutz bieten, unterschritten wird, was dazu führt, dass hier der Anteil der erfolgreichen Angriffe steigt. Im ersten Fall würde ein Teilnehmer an diesen Stellen schlichtweg keine Funkstille einhalten und wäre ohnehin angreifbar – in die Statistik findet er jedoch keinen Eingang.

Das Erhöhen der Schwellengeschwindigkeit von 20 km/h auf 30 km/h verringert dagegen in allen drei Städten deutlich den Angriffserfolg (teilweise um ca. 50 %). Offenbar führt diese Reduktion zu einer deutlich häufigeren Unterschreitung der Grenzgeschwindigkeit – dies zudem in Situationen, in denen ein Angriff nicht einfach möglich ist. Ein Deutungsansatz besteht darin, dass einige Straßenabschnitte eine Geschwindigkeitsbeschränkung von 30 km/h aufweisen und die (in der Simulation regeltreuen) Fahrzeuge in diesen Bereichen teilweise eine sehr lange Zeit Funkstille halten, so dass es für den Angreifer beinahe unmöglich wird, den angegriffenen Teilnehmer wiederzufinden. Für diese Theorie spricht, dass eine weitere Erhöhung der Schwellengeschwindigkeit auf 40 km/h nahezu keine Auswirkungen auf den Angriffserfolg hat. Insbesondere in diesem Fall halten Fahrzeuge, die sich auf Straßenabschnitten mit einem Tempolimit von 30 km/h bewegen, über sehr lange Zeiträume Funkstille und sind damit kaum angreifbar.

Die drei untersuchten Städte weisen kaum Unterschiede auf, lediglich in Puebla scheinen die Angriffserfolge etwas geringer zu sein. Dies kann wiederum in der Eigenschaft des Straßensystems begründet liegen, das in den Simulationen dazu führt, dass sich viele Fahrzeuge auf den großen Verkehrsadern bewegen möchten und dort im Stau stehen. Daraus resultiert eine länger andauernden Unterschreitung der Schwellengeschwindigkeit und somit längere Funkstillen, die den Schutz der Privatsphäre begünstigen.

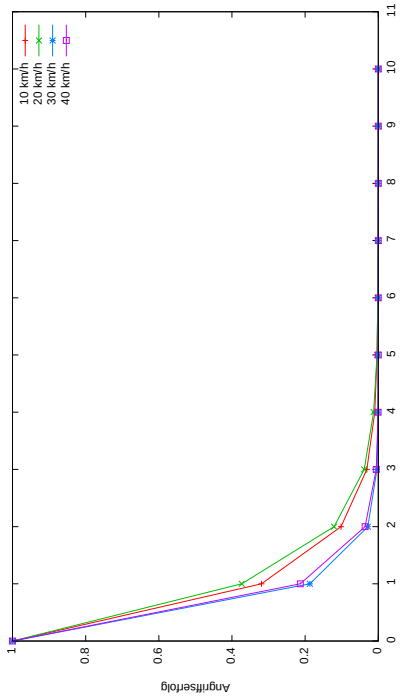
Verfolgung über mehrere Funkstillen hinweg Bei SLOW wird – wie bei den anderen Konzepten auch – im Laufe der Zeit eine Reihe von Funkstillen durchlaufen. Dies führt dazu, dass die ohnehin schon recht schlechte Chance eines Angreifers, einen Teilnehmer zu verfolgen, im Zeitverlauf weiter abnimmt. Abbildung 5.38 zeigt in Berlin den Angriffserfolg auf SLOW im Verlauf über mehrere Funkstillen hinweg. Die Ergebnisse der anderen Städte unterscheiden sich hiervon nicht wesentlich und sind im Anhang in Abschnitt C.5 zu finden.

Wiederum zeigt sich bei allen Städten der vermutete logarithmische Zusammenhang

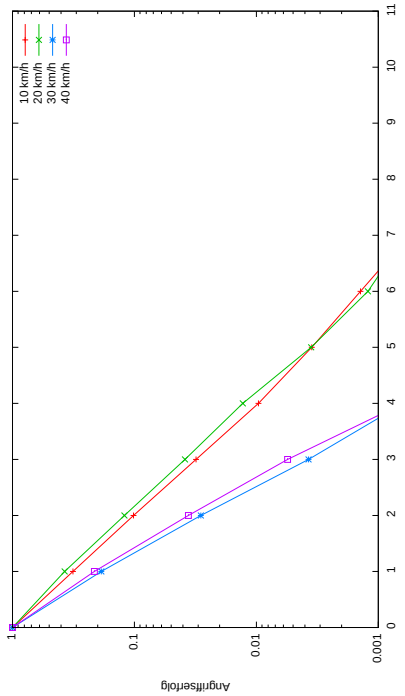


(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(c) Darstellung in Abhängigkeit von der Schwellengeschwindigkeit



(d) Logarithmische Darstellung in Abhängigkeit von der Schwellengeschwindigkeit

Abbildung 5.38.: Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in Berlin

zwischen der Erfolgsrate und der Anzahl an durchgeführten Funkstillen. Entsprechend sind auch hier die Daten jeweils in einem linear und einem logarithmisch skalierten Diagramm aufbereitet. Erreicht eine Kurve in der linearen Darstellung die Nulllinie, so bedeutet dies, dass kein Teilnehmer mehr nach der entsprechenden Anzahl an Funkstillen korrekt zugeordnet werden konnte; endet die Kurve, so hat kein Teilnehmer eine entsprechende Anzahl an Funkstillen mit Pseudonymwechsel (mindestens eine Sekunde Funkstille) durchgeführt. In den logarithmisch skalierten Diagrammen ist die Nulllinie nicht definiert. Aus diesem Grund endet die Kurve auch in dem Fall, in dem keine korrekte Zuordnung stattgefunden hat.

Die Betrachtung der Verkehrsdichte weist keine wesentlichen Besonderheiten auf, sondern spiegelt vielmehr die Erkenntnisse wider, die bereits aus den obigen Diagrammen hervorgegangen sind. Es zeigt sich jedoch, dass die Teilnehmer eine recht hohe Anzahl an Funkstillen durchlaufen, dabei jedoch nur über eine sehr geringe Anzahl hinweg verfolgt werden können.

Auch die Untersuchung der Schwellengeschwindigkeit liefert keine wesentlichen neuen Erkenntnisse. Interessant ist, dass in Abbildung 5.38d die oben festgestellten Besonderheiten bezüglich der Grenzgeschwindigkeit deutlich hervortreten: Die Kurven von 10 km/h und 20 km/h beziehungsweise 30 km/h und 40 km/h liegen sehr nahe beisammen und unterstreichen somit, dass durch eine Veränderung innerhalb dieser beiden Bereiche keine nennenswerten Verbesserungen erzielt werden können.

Insgesamt lässt sich festhalten, dass sich SLOW als ein sehr effektives Konzept präsentiert, da bereits nach zwei durchlaufenen Funkstillen die Erkennungsraten des Angreifers auf unter 10 % fallen. Somit erscheint es gar nicht notwendig, eine besonders hohe Schwellengeschwindigkeit festzulegen. Natürlich ist auch an dieser Stelle eine Verbesserung der Angriffsheuristik denkbar und kann wahrscheinlich zu besseren Erkennungsraten führen. Dies erfordert eine empirische Untersuchung des Verkehrs im gesamten Angriffsgebiet und vermutlich sehr große Rechenkapazitäten.

Einfluss des Straßennetzes auf die Effektivität von SLOW Im Folgenden soll genauer betrachtet werden, an welchen Orten Fahrzeuge die jeweilige Schwellengeschwindigkeit unterschreiten und somit eine Funkstille einhalten. Darüber hinaus soll geklärt werden, welchen Einfluss dies auf den Angriffserfolg hat. Zu diesen Betrachtungen dienen erneut Karten der simulierten Städte, in denen jeweils die Eintrittsposition eines Fahrzeugs in die Funkstille markiert wird. Eine rote Markierung bedeutet, dass es dem Angreifer gelungen ist, das Fahrzeug nach der Funkstille wiederzufinden und somit die Pseudonyme zu verketten; eine grüne Markierung zeigt an, dass dies fehlgeschlagen ist. Da jedoch bei SLOW auch so kurze Funkstillen (kürzer als eine Sekunde) stattfinden können, dass ein Pseudonymwechsel nicht sinnvoll ist, wurden diese nicht angegriffen (eine Zuordnung wäre in diesem Fall auch trivial

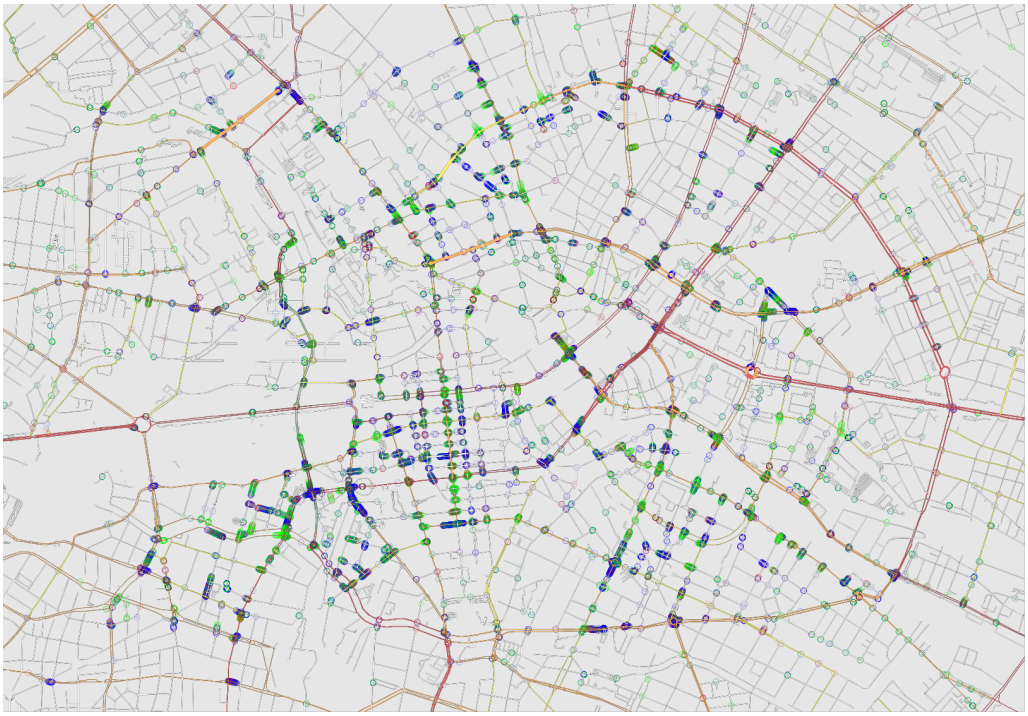
gewesen). Dennoch sollen die entsprechenden Fahrzeuge auf der Karte mittels einer blauen Markierung angezeigt werden. Die Betrachtungen beziehen sich explizit auf zwei Situationen in Berlin und Puebla, allerdings besitzen die daraus gezogenen Schlüsse auch für die anderen Szenarien Gültigkeit, da sich dort ein sehr ähnliches Bild ergibt. Die hier nicht explizit erläuterten Abbildungen befinden sich im Anhang in Abschnitt C.6.

Abbildung 5.39a zeigt die Situation bei SLOW in Berlin mit einer Schwellengeschwindigkeit von 10 km/h. Diese wird nur selten und im Wesentlichen in Kreuzungsbereichen unterschritten. Die betreffenden Orte sind jedoch breit über das Zentrum der Stadt verteilt. Es finden sich neben vielen blauen Markierungen, an denen die Grenzgeschwindigkeit nur für einen zu kurzen Zeitraum unterschritten wurde, als dass ein Pseudonymwechsel möglich gewesen wäre, einige grüne und nur relativ wenige rote Markierungen. Dies spiegelt die Erkenntnis wider, dass SLOW in den meisten Fällen einen passablen Schutz für die Teilnehmer bietet. Auf längeren mehrspurigen Streckenabschnitten (beispielsweise dem Innenstadtring), wo der Verkehr gut und zügig fließt, gibt es dagegen keine Funkstillen und somit sind alle Fahrzeuge, die sich dort bewegen, beobachtbar.

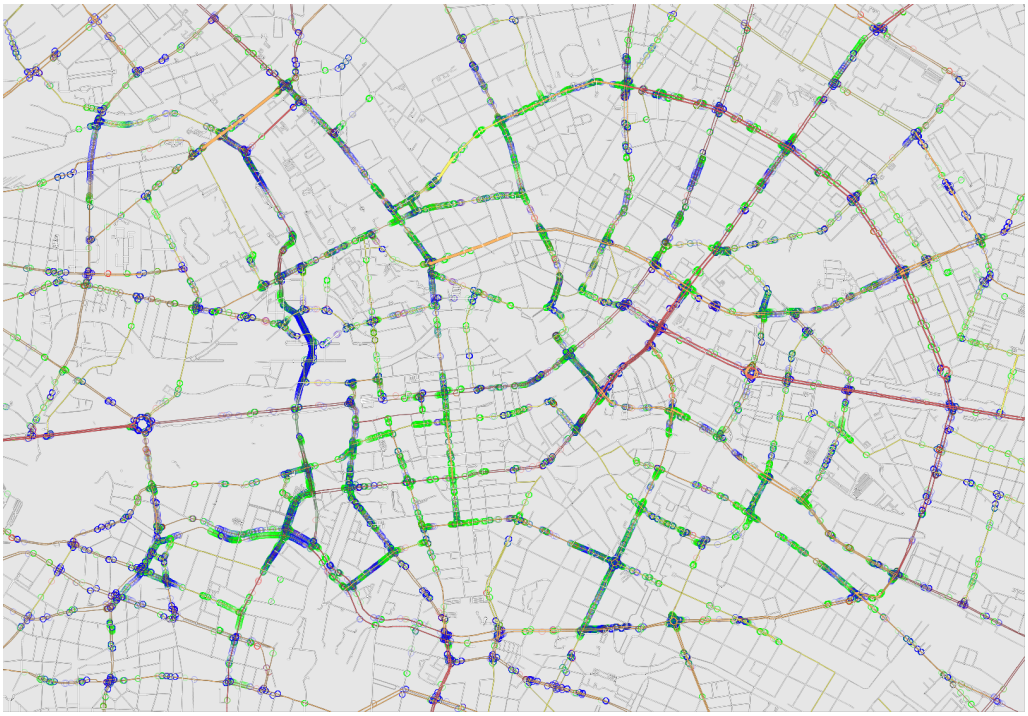
Erhöht man die Schwellengeschwindigkeit auf 40 km/h (vgl. Abbildung 5.39b), so verändert sich das Bild grundlegend. Zuerst fällt auf, dass es zu deutlich mehr Funkstillen kommt und diese auch zu einem sehr großen Teil einen Schutz vor Angriffen bieten (grüne Einfärbung). Weiterhin verlagert sich der Ort des Eintritts in eine Funkstille sehr deutlich auf die Hauptverkehrsadern, auf denen der Verkehr häufig mit einer höheren Geschwindigkeit fließt. Insbesondere auf kleinen Nebenstraßen finden nun keine Eintritte in die Funkstillen mehr statt, da die Grenzgeschwindigkeit nur sehr selten überschritten wird und die meiste Zeit über somit keinerlei Kommunikation stattfindet. Diese Beobachtung deckt sich mit den oben angestellten Überlegungen und erklärt erneut den deutlich größeren Anteil an fehlgeschlagenen Angriffen.

Der eben dargelegte Aspekt wird auch deutlich, wenn man die in Abbildung 5.40 dargestellten Angriffserfolge in Abhängigkeit der Fahrzeugdichte in Puebla betrachtet. Wie bereits oben festgestellt, zeigt sich, dass eine höhere Verkehrsdichte zwar einen Einfluss auf den Angriffserfolg hat (sichtbar durch die leichte anteilige Abnahme an roten Markierungen), dieser jedoch nicht so hoch ausfällt wie bei den anderen Konzepten. Bei der Verteilung der Eintrittspunkte in die Funkstille zeigt sich durch eine Veränderung der Verkehrsdichte kein Unterschied.

Diese dargestellten Untersuchungen wurden bei der von den Autoren in [BHWW09] vorgeschlagenen Schwellengeschwindigkeit von 30 km/h durchgeführt. Wie sich bereits im Beispiel von Berlin gezeigt hat, wird auch hier sehr deutlich, dass es nahezu keine Überschreitungen dieser Grenzgeschwindigkeit auf den Nebenstraßen gibt und sich

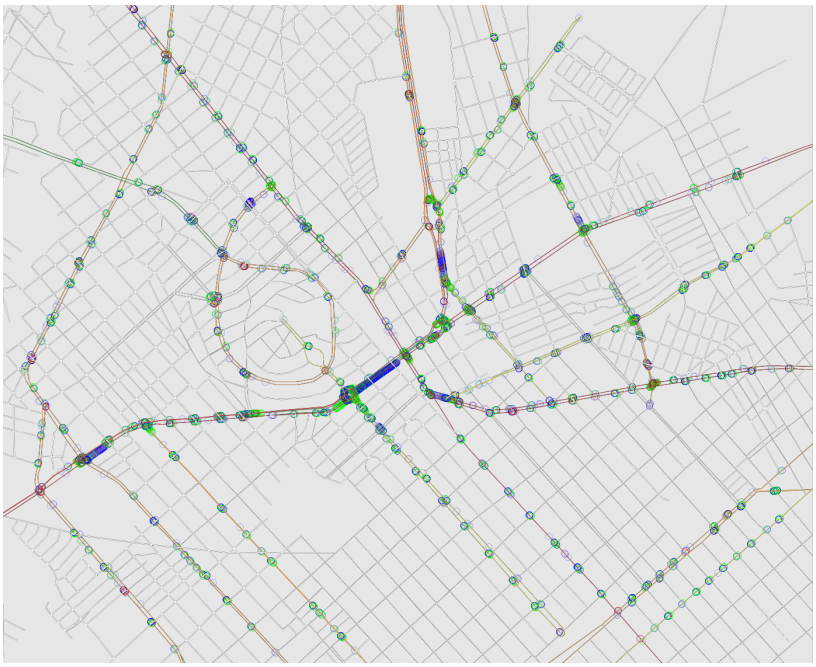


(a) 10 km/h

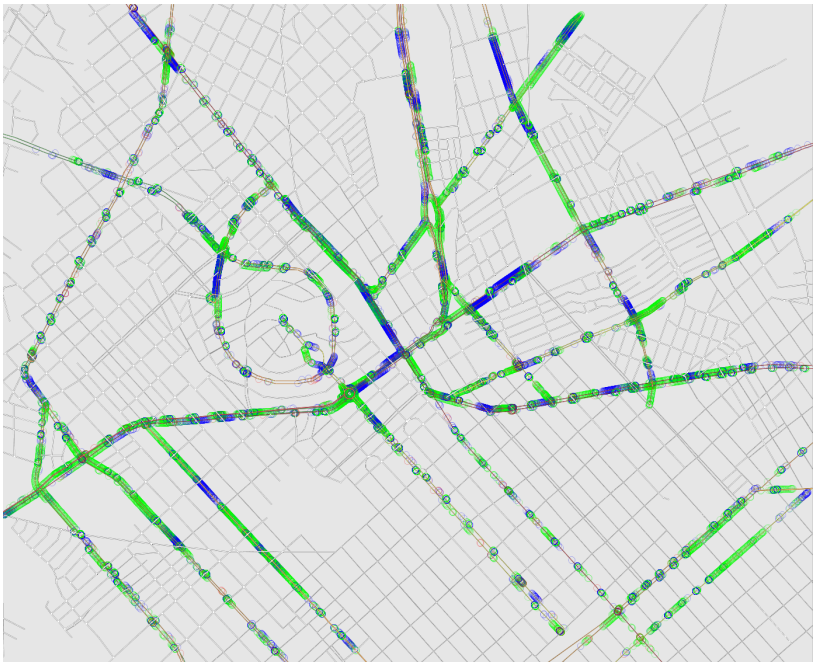


(b) 40 km/h

Abbildung 5.39.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Berlin bei unterschiedlichen Schwellengeschwindigkeiten



(a) 1250 Fahrzeuge



(b) 20 000 Fahrzeuge

Abbildung 5.40.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Puebla bei unterschiedlichen Verkehrsdichten

somit alle Eintrittsorte auf die Hauptverkehrsadern verlegen. Die geschwindigkeitsbegrenzten Gebiete daneben erlauben es den Teilnehmern nicht, die Grenzgeschwindigkeit zu überschreiten, wodurch sie dort über die gesamte Aufenthaltsdauer eine Funkstille wahren.

5.4.5. Fazit

Die Untersuchungen der Konzepte zum Schutz der Privatsphäre durch Realisierung eines unbeobachtbaren Pseudonymwechsels bei Funkstille haben eines sehr deutlich gezeigt: Insbesondere Mix-Zonen und Silent Periods, aber auch SLOW, weisen große Schwächen gegenüber einem selbst nur mäßig intelligenten Angreifer auf. Mit den beschriebenen relativ einfachen Angriffsheuristiken konnten bereits äußerst gute Erfolgsraten erzielt werden. Dies lässt nur den Schluss zu, dass ein effektiver Schutz vor einem Angreifer, der über eine große Rechenkapazität verfügt und zudem Kontextwissen über einzelne Teilnehmer in seine Kalkulationen einfließen lässt, mit den beschriebenen Ansätzen kaum realisierbar ist. Die Ergebnisse der einzelnen Untersuchungen sind in Tabelle 5.5 zusammengefasst.

Etwas relativieren lässt sich diese Aussage dadurch, dass der Angriffserfolg sehr deutlich abgenommen hat, wenn die Teilnehmer eine Reihe von Funkstillen eingehalten haben. So konnte nach dem Durchfahren von drei bis vier Mix-Zonen, sechs Silent Periods oder zwei Funkstillen durch SLOW die Erfolgsrate des Angreifers auf maximal 10 % gesenkt werden. Eine wichtige Erkenntnis dieser Untersuchungen ist somit, dass es unerlässlich ist, eine möglichst große Anzahl an Funkstillen zu durchlaufen.

Mix-Zonen in größerer Zahl scheinen Teilnehmern ein akzeptables Niveau an Schutz bieten zu können – insbesondere sobald sie eine gewisse Größe erreichen und somit eine Reihe von Ein- und Ausfahrten bieten. Allerdings sind sie räumlich sehr stark festgelegt, und Teilnehmer, auf deren Route sich nicht mehrere Mix-Zonen befinden, können von einem Angreifer äußerst einfach verfolgt werden. Ihre Eignung für den Einsatz innerorts ist somit eher als zweifelhaft einzuschätzen.

Der Ansatz der Silent Periods hat in den Untersuchungen die schlechtesten Ergebnisse gezeigt. Lediglich bei einer sehr langen Dauer der Funkstille oder einer hohen Verkehrsdichte kann ein ausreichender Schutz geboten werden. Dieser ist zudem erst nach einer Reihe von Funkstillen gegeben. Dies stellt sich jedoch als nicht besonders problematisch heraus, da Silent Periods nicht ortsgebunden stattfinden und somit im Laufe der Zeit automatisch durchgeführt werden. Die Unabhängigkeit von einem Aufenthaltsort führt jedoch auch dazu, dass es immer wieder Bereiche gibt, in denen die Funkstille (die zudem für eine gewisse Zeitspanne alle Assistenzsysteme außer Kraft setzt) fast keinen Schutz bietet, wenn sich ein Teilnehmer auf einem geraden

Konzept	Variierter Parameter	Stadt	Einfacher Angriff (Erfolgsrate)		Komplexer Angriff (Erfolgsrate)	
			min	max	min	max
Mix-Zone	Anzahl Fahrzeuge [1250; 20 000] (Radius Mix-Zonen: 100 m)	Berlin	8,41 %	26,79 %	21,03 %	63,24 %
		New York	12,08 %	28,58 %	21,60 %	50,40 %
		Puebla	7,15 %	28,65 %	13,46 %	57,27 %
	Radius der Mix-Zonen [25; 400] m (Anzahl Fahrzeuge: 5 000)	Berlin	0,77 %	28,29 %	14,48 %	62,03 %
		New York	2,11 %	39,58 %	10,88 %	61,14 %
		Puebla	1,29 %	37,13 %	8,66 %	58,60 %
Silent Periods	Anzahl Fahrzeuge [1250; 20 000] (Dauer der Funkstille: 5 000 ms)	Berlin	23,97 %	57,87 %	33,77 %	75,82 %
		New York	5,18 %	32,11 %	8,51 %	50,32 %
		Puebla	10,86 %	44,28 %	16,23 %	60,99 %
	Dauer der Funkstille [1250; 20 000] ms (Anzahl Fahrzeuge: 5 000)	Berlin	12,56 %	39,60 %	22,49 %	78,44 %
		New York	4,58 %	27,65 %	9,96 %	53,62 %
		Puebla	8,62 %	25,95 %	18,62 %	62,27 %
SLOW	Anzahl Fahrzeuge [1250; 20 000] (Schwellengeschwindigkeit: 30 km/h)	Berlin	-	-	8,30 %	27,09 %
		New York	-	-	7,40 %	26,21 %
		Puebla	-	-	4,92 %	19,55 %
	Schwellengeschwindigkeit [10; 40] km/h (Anzahl Fahrzeuge: 5 000)	Berlin	-	-	16,01 %	34,13 %
		New York	-	-	14,07 %	26,16 %
		Puebla	-	-	8,76 %	26,87 %

Tabelle 5.5.: Überblick über die Simulationsergebnisse

Abschnitt ohne Abzweigungen befindet und somit durch einen Angreifer sehr einfach wiederaufgefunden werden kann. Der mit diesem Ansatz mögliche Schutz ist also als eher gering einzuschätzen.

SLOW hat in den Betrachtungen am besten abgeschnitten und stellt auch die größten Anforderungen an die Entwicklung einer Heuristik und Durchführung von Angriffen. Allerdings ist es eine schwieriger Aufgabe, eine geeignete Grenzgeschwindigkeit zu finden. Wird diese zu niedrig angesetzt, wird sie von den Teilnehmern nur selten unterschritten und somit finden wenige Pseudonymwechsel statt. Wählt man sie dagegen zu hoch, so gibt es große Bereiche aus Straßenzügen, in denen es den Fahrzeugen nicht gelingt, diese Geschwindigkeit zu überschreiten, wodurch sie über den gesamten Bereich Funkstille wahren müssen. Dies hätte zur Folge, dass zwar Angriffe nicht möglich wären, aber auch sämtliche Verkehrssicherheitssysteme nicht zum Einsatz kommen könnten. Somit wäre dem *VANET* jeglicher Nutzen genommen.

Die von Buttyán et al. in ihrer Publikation über SLOW (vgl. [BHWW09]) vorgeschlagene Schwellengeschwindigkeit von 30 km/h hat sich als nicht besonders praktikabel herausgestellt, da es viele Bereiche gibt, in denen diese Geschwindigkeit nicht überschritten werden darf oder kann. Möglicherweise könnte sich eine Grenzgeschwindigkeit, die etwas darunter liegt – beispielsweise 25 km/h –, als viel geeigneter herausstellen. Dazu müssen weitere Untersuchungen durchgeführt werden.

Insgesamt stellt SLOW einen vielversprechenden Ansatz innerorts dar, der allerdings auf Grund der Festlegung einer (niedrigen) Schwellengeschwindigkeit (vgl. dazu auch die Argumentation in Abschnitt 4.4.2) nicht geeignet ist, Teilnehmer außerorts oder gar auf Schnellstraßen und Autobahnen Schutz vor einer Verfolgung durch einen Angreifer zu bieten. Möglicherweise könnten Mix-Zonen diese Lücke füllen und in regelmäßigen Abständen auf Knotenpunkten von Fernstraßen platziert werden, um die Fahrzeuge vor einer Verfolgung über weite Strecken zu schützen. Für eine definitive Aussage zur Effektivität von Mix-Zonen und weiteren Ansätzen außerhalb von städtischen Straßensystemen ist jedoch eine weitere Studienreihe nötig.

Kapitel 6

Fazit

6.1. Ergebnisse der Arbeit

Die vorliegende Arbeit leistet einen Beitrag dazu, Konzepte für den effektiven Schutz der Privatsphäre von Teilnehmern in Ad-hoc-Fahrzeugnetzen zu entwickeln und zu evaluieren und somit Wege aufzuzeigen, neue Technologien im Umfeld eines *VANET*s auf eine datenschutzfreundliche Art und Weise einzuführen. Weiterhin wird mit dem *VANET-Simulator* eine Möglichkeit geschaffen, auch ohne aufwändige Konfiguration von Prototypen unterschiedlichste Aspekte eines *VANET*s – seien es Nachrichtenaustausch, Sicherheitsfunktionen, Schutz der Privatsphäre, etc. – zu untersuchen und ihren Nutzen abzuschätzen. Insgesamt bestehen die Leistungen dieser Arbeit im Wesentlichen aus drei Teilen:

Datenschutzfreundliche Kommunikation zwischen Teilnehmern Unterschiedliche Ansätze für die Ausgestaltung einer datenschutzfreundlichen Kommunikation zwischen Teilnehmern eines Ad-hoc-Fahrzeugnetzes werden detailliert betrachtet, einander gegenübergestellt und kritisch bewertet. Die daraus gewonnenen Erkenntnisse fließen in einen eigenen Ansatz ein, der das größte Manko der in der Literatur beschriebenen Konzepte beseitigen kann. Ein Vertrauen in den Betreiber der Infrastruktur des Ad-hoc-Fahrzeugnetzes ist damit nicht mehr erforderlich. Erreicht wird dies mit Hilfe von Chaum'schen Mixen, die dafür sorgen, dass die Kommunikationswege verschleiert werden. Kombiniert mit ausschließlich verschlüsselter Speicherung von Aufenthaltsinformationen kann somit ein höheres Schutzniveau erreicht werden als dies mit den in der Literatur bereits bekannten Ansätzen möglich ist.

Sicherer Pseudonymwechsel in Ad-hoc-Fahrzeugnetzen Teilnehmer in Ad-hoc-Fahrzeugnetzen können sehr leicht anhand ihrer Beacons verfolgt werden. Dieses bekannte erstmals von Beresford und Stajano [BS03] beschriebene Problem wird eingehend betrachtet und durch eigene Analysen ergänzt. Mit den gewonnenen Erkenntnissen wird erstmals ein umfassender Katalog der in der Literatur bekannten Ansätze, dieser Bedrohung zu begegnen, zusammengestellt. Anschließend werden die Konzepte kategorisiert, analysiert und einander gegenübergestellt. Neben einigen wenig erfolgsversprechenden Ansätzen hinterlassen dabei drei Konzepte für bestimmte Verkehrs- und Straßensituationen einen vielversprechenden Eindruck. Diese werden in der Folge empirisch mit Hilfe des *VANET-Simulators* untersucht.

Einer der untersuchten Ansätze ist das in der Literatur am weitesten verbreitete Konzept – Mix-Zone. Diese bietet eine Reihe von Vorteilen, jedoch wird mit ihren bekannten Konfigurationen entweder die Funktionalität von *VANET*-Systemen (wie beispielsweise Fahrassistenzsysteme) massiv eingeschränkt oder es kann nur ein geringer Schutz vor Angreifern sichergestellt werden. Diesem Zwiespalt der konkurrierenden Schutzziele (Vertraulichkeit auf der einen und Verfügbarkeit auf der anderen Seite) soll der vorgestellte Ansatz der ProMix-Zone Rechnung tragen und eine Möglichkeit aufzeigen, wie in Form eines Trade-Offs eine beliebige Abstimmung zwischen diesen beiden Polen stattfinden kann. Zu diesem Zweck wird ein Proxy eingeführt, über den sämtliche Kommunikation stattfindet. Dieser entscheidet auch, welche Nachrichten welcher Teilnehmer erhalten soll. Je nach Konfiguration dieser Filterheuristik kann das Augenmerk mehr auf die Verkehrssicherheit oder den Schutz der Privatsphäre gelegt werden.

Entwicklung des *VANET-Simulators* und empirische Untersuchung von Konzepten zum Schutz der Privatsphäre Mangels einer Simulationslösung, die die beiden wesentlichen Aspekte eines Ad-hoc-Fahrzeugnetzes – Straßen- und Datenverkehr – simulieren kann und zudem eine einfache Erweiterbarkeit bereitstellt, wird ein eigener *VANET-Simulator* entwickelt. Dazu werden die erforderlichen Anforderungen beschrieben und umgesetzt, einschlägige Literatur berücksichtigt und auf Basis dessen eine plattformunabhängige Anwendung geschaffen, die auch auf herkömmlicher PC-Hardware selbst große Simulationen durchführen kann. Der Einsatz von realem Kartenmaterial ist durch die Bereitstellung einer Importfunktionalität von Daten aus dem OpenStreetMap-Projekt ebenso möglich wie das einfache und intuitive Erstellen von Verkehrsnetzen. Die Anwendung ist umfassend dokumentiert und kommentiert; auf die Einhaltung von Programmierrichtlinien wird geachtet, damit eine Einarbeitung in den Quellcode und die Ergänzung um neue Funktionalität einfach gelingen kann. Der Simulator verfügt bereits über Implementierungen von Techniken zum sicheren Pseudonymwechsel, deren Effektivität in einer umfassenden Studie überprüft werden.

Dazu wird jeweils eine Variante der Mix-Zonen, Silent Periods und des SLOW-Ansatzes definiert und umgesetzt. Mit Hilfe von 450 Simulationsläufen werden diese Ansätze in drei unterschiedlichen urbanen Szenarien mit diversen Parametereinstellungen untersucht. Mit Hilfe von entwickelten Heuristiken werden dazu Angriffe auf die Teilnehmer des Ad-hoc-Netzes durchgeführt. Weiterhin werden zwei weit verbreitete Metriken für Unbeobachtbarkeit – die Anonymitätsgruppe und die Entropie – bestimmt und den Ergebnissen gegenübergestellt.

Mit diesen Untersuchungen kann erstmals eine Betrachtung der Konzepte vorgenommen werden, die über eine kaum parametrisierte analytische Gegenüberstellung deutlich hinausgeht. Die Ergebnisse zeigen die Grenzen der Ansätze deutlich auf. Außerdem wird deutlich, welchen Einflussgrößen ihre Effektivität in welchem Umfang unterliegt. Der *SLOW*-Ansatz stellt sich dabei als vielversprechendstes Konzept zum Schutz der Privatsphäre in urbanen Regionen dar.

Publikationen Einige Teilergebnisse dieser Arbeit wurden bereits in Form von Papers [SPF08, SBF10, SFF11, TSF12] einem Fachpublikum vorgestellt und zur Veröffentlichung in den Tagungsbänden von Konferenzen angenommen.

6.2. Ausblick

Obwohl mit der vorliegenden Arbeit eine Vielzahl neuer Erkenntnisse über den Schutz der Privatsphäre in Ad-hoc-Fahrzeugnetzen gewonnen werden konnte, besteht darüberhinaus in diesem Feld noch großer Forschungsbedarf. Mit dem *VANET-Simulator* wurde eine Möglichkeit geschaffen, Ansätze sehr einfach und damit kostengünstig zu implementieren und zu testen, was die Entwicklung und Überprüfung neuer Ideen stark erleichtert.

Die im Rahmen dieser Arbeit durchgeführten Untersuchungen von Techniken für einen sicheren Pseudonymwechsel beschränken sich auf den urbanen Raum und können daher keine Aussage über ländliche Regionen oder die Wirkungsweise auf Fernstraßen treffen. An dieser Stelle gibt es noch Forschungsbedarf. Weiterhin sollten die durchgeführten Betrachtungen auf die bisher lediglich analytisch untersuchten Ansätze ausgedehnt werden, um diese ebenfalls in die Überlegungen einzubeziehen.

Auch in Bezug auf den *VANET-Simulator* selbst besteht Erweiterungspotential, das im Rahmen dieser Arbeit nicht ausgeschöpft werden konnte. So wäre ein echtes Plugin-Konzept wünschenswert, mit dem über klar definierte Schnittstellen beliebige Komponenten hinzugefügt werden könnten, ohne dass Programmcode der

Kernfunktionalität angepasst werden müsste. Zudem sind noch nicht alle in Ad-hoc-Fahrzeugnetzen denkbaren Kommunikationsformen umgesetzt, weshalb manche Untersuchungen bislang nicht möglich sind. Auch diese Umsetzung hätte den Rahmen der Arbeit gesprengt.

Insgesamt muss das Ziel sein, auf Grundlage der vorliegenden Arbeit sowie den vorangegangenen und darauf aufbauenden Forschungen den laufenden Standardisierungsprozess der Ad-hoc-Fahrzeugnetze dahingehend zu beeinflussen, dass datenschutzfreundliche Techniken und Schutzmaßnahmen gegen die Verfolgung von Teilnehmern bei der Umsetzung von Anfang an berücksichtigt und implementiert werden.

Literaturverzeichnis

[ABD+06] Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh, Tim Leinmüller. *Attacks on Inter-Vehicle Communication Systems - An Analysis*. In: Proceedings of Workshop on Intelligent Transportation (WIT 2006), 2006.

[ADA11] ADAC. *Geschichte der ADAC-Luftrettung*. `http://www.adac.de/infotestrat/adac-im-einsatz/luftrettung/wir-ueber-uns/chronik/default.aspx?tabid=tab1`, abgerufen am 25.10.2011, 2011.

[AK96] Ross Anderson, Markus Kuhn. *Tamper resistance: a cautionary note*. In: Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2, Berkeley, CA, USA, 1996. USENIX Association.

[AK00] Jeffery Archer, Iisakki Kosonen. *The Potential of microsimulation modelling in relation to traffic safety assessment*. In: Proceedings of the 12th European Simulation Symposium (ESS), Hamburg, September 2000.

[BAK98] Eli Biham, Ross J. Anderson, Lars R. Knudsen. *Serpent: A New Block Cipher Proposal*. In: Serge Vaudenay (Herausgeber), Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings, Volume 1372 of Lecture Notes in Computer Science, Seite 222–238. Springer, 1998.

[Bar05] Norbert Bartelme. *Geoinformatik: Modelle, Strukturen, Funktionen*. Springer Berlin Heidelberg, 4. Edition, 2005. ISBN 9783540202547

[BBEK11] Michael Behrisch, Laura Bieker, Jakob Erdmann, Daniel Krajzewicz. *SUMO - Simulation of Urban MObility - an Overview*. In: Proceedings of the Third International Conference on Advances in System Simulation, SIMUL 2011, Seite 63–68, Barcelona, Spain, Oktober 2011.

[BBSS01] Elmar Brockfeld, Robert Barlovic, Andreas Schadschneider, Michael

Schreckenberg. *Optimizing traffic lights in a cellular automaton model for city traffic.* Physical Review E, 64(5): 056132, Oct 2001.

[BCSW98] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, Barry A. Woodward. *A distance routing effect algorithm for mobility (DREAM).* In: MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking, Seite 76–84, New York, NY, USA, 1998. ACM.

[BDD+04] Rainer Böhme, George Danezis, Claudia Díaz, Stefan Köpsell, Andreas Pfitzmann. *On the PET Workshop Panel "Mix Cascades Versus Peer-to-Peer: Is One Concept Superior?".* In: David Martin, Andrei Serjantov (Herausgeber), Privacy Enhancing Technologies, 4th International Workshop, PET 2004, Toronto, Canada, May 26-28, 2004, Revised Selected Papers, Volume 3424 of Lecture Notes in Computer Science, Seite 243–255. Springer, 2004.

[Ben07] Ralf Bendrath. *Der ‚gläserne Mensch'und der vorsorgliche Staat. Zum Verhältnis von Privatheit und Sicherheit in der Informationsgesellschaft.* kommunikation@gesellschaft, 8, 2007. *(http://www.soz.uni-frankfurt.de/K.G/E1_2007_Bendrath.pdf)*

[Ber05] Alastair R. Beresford. *Location privacy in ubiquitous computing.* Technical Report 612, University of Cambridge, Cambridge, United Kingdom, Januar 2005.

[BFK00] Oliver Berthold, Hannes Federrath, Stefan Köpsell. *Web MIXes: A System for Anonymous and Unobservable Internet Access.* In: Hannes Federrath (Herausgeber), Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability, Volume 2009 of Lecture Notes in Computer Science, Seite 115–129. Springer, 2000.

[BHV07] Levente Buttyán, Tamás Holczer, István Vajda. *On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs.* In: Proceedings of the Fourth European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2007), Juli 2007.

[BHWW09] Levente Buttyán, Tamás Holczer, André Weimerskirch, William Whyte. *SLOW: A Practical Pseudonym Changing Scheme for Location Privacy in VANETs.* In: Proceedings of the IEEE Vehicular Networking Conference (VNC), Tokyo, Japan, 2009. IEEE, October 2009.

[BJH96] Simon C. Benjamin, Neil F. Johnson, P. M. Hui. *Cellular automata models of traffic flow along a highway containing a junction.* Journal of Physics A: Mathematical and General, 29(12): 3119, 1996.

[BNS05] Albrecht Beutelspacher, Heike B. Neumann, Thomas Schwarzpaul. *Kryptografie in Theorie und Praxis.* Vieweg+Teubner, 1. Edition, 2005. ISBN 3528031689

[BS03] Alastair R. Beresford, Frank Stajano. *Location Privacy in Pervasive Computing.* IEEE Pervasive Computing, 2, 2003.

[BS04] Alastair R. Beresford, Frank Stajano. *Mix Zones: User Privacy in Location-aware Services.* In: 2nd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2004 Workshops), 14-17 March 2004, Orlando, FL, USA, Seite 127–131. IEEE Computer Society, 2004.

[BSH03] Fan Bai, Narayanan Sadagopan, Ahmed Helmy. *IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks.* In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE, April 2003.

[BT98] A. Baruya, Transport Research Laboratory, U.K. *Speed-Accident Relationship on Different Kinds of European Roads.* VTT Communities & Infrastructure (Finland), September 1998.

[Bun10] Bundesministerium für Verkehr, Bau und Stadtentwicklung (Herausgeber). *Verkehr in Zahlen 2009/2010*, Volume 38. Deutscher Verkehrs-Verlag, Hamburg, 2010.

[Bun11] Bundesnetzagentur, Presse und Öffentlichkeitsarbeit (Herausgeber). *Jahresbericht 2011.* Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, Bonn, März 2011.

[BW04] Elmar Brockfeld, Peter Wagner. *Testing and Benchmarking of Microscopic Traffic Flow Models.* In: Proceedings of the 10th World Conference on Transport Research, Volume abstract book I, Istanbul, Türkei, Juli 2004.

[CAR07] CAR 2 CAR Communication Consortium. *Car 2 Car Communication Consortium Manifesto - Overview of the C2C-CC System.* Technical Report Version 1.1, Car2Car Consortium, August 2007.

[CBW02] Tracy Camp, Jeff Boleng, Lucas Wilcox. *Location information services in mobile ad hoc networks.* In: Proceedings of the IEEE International Conference on Communications (ICC) 2002, August 2002.

[CH06] Srdjan Capkun, Jean-Pierre Hubaux. *Secure positioning in wireless networks.* IEEE Journal on Selected Areas in Communications, 24(2): 221–232, 2006.

[Cha81] David L. Chaum. *Untraceable electronic mail, return addresses, and digital pseudonyms.* Communications of the ACM, 24(2), 1981.

[Cha82] David Chaum. *Blind Signatures for Untraceable Payments.* In: David Chaum, Ronald L. Rivest, Alan T. Sherman (Herausgeber), CRYPTO: Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, Seite 199–203, New York, NY, USA, 1982. Plenum Press.

[Cha83] David Chaum. *Blind Signature System.* In: David Chaum (Herausgeber), CRYPTO: Advances in Cryptology, Proceedings of CRYPTO '83, Santa Barbara, California, USA, August 21-24, Seite 153, New York, NY, USA, 1983. Plenum Press.

[CKQ03] Julien Cathalo, François Koeune, Jean-Jacques Quisquater. *A New Type of Timing Attack: Application to GPS.* In: Colin D. Walter, Çetin Kaya Koç, Christof Paar (Herausgeber), Proceedings of the Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, Seite 291–303, September 2003.

[CM99] S. Corson, J. Macker. *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations.* RFC 2501 (Informational), Januar 1999. *(`http://www.ietf.org/rfc/rfc2501.txt`)*

[CPHL07] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, Antonio Lioy. *Efficient and robust pseudonymous authentication in VANET.* In: Wieland Holfelder, Paolo Santi, Yih-Chun Hu, Jean-Pierre Hubaux (Herausgeber), Proceedings of the Fourth International Workshop on Vehicular Ad Hoc Networks, VANET 2007, Montréal, Québec, Canada, September 10, 2007, Seite 19–28. ACM, 2007.

[CRCS08] Srdjan Capkun, Kasper Bonne Rasmussen, Mario Cagalj, Mani B. Srivastava. *Secure Location Verification with Hidden and Mobile Base Stations.* IEEE Transactions on Mobile Compututing, 7(4): 470–483, April 2008.

[CSF+08] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* RFC 5280 (Proposed Standard), 2008. *(`http://www.ietf.org/rfc/rfc5280.txt`)*

[CVTA09] Brijesh Kumar Chaurasia, Shekhar Verma, G. S. Tomar, Ajith Abraham. *Optimizing Pseudonym Updation in Vehicular Ad-Hoc Networks.* Transactions on Computational Science IV, Special Issue on Security in Computing, 4: 136–148, 2009.

[DFNI07] Marios Dikaiakos, Andreas Florides, Tamer Nadeem, Liviu Iftode. *Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication.* IEEE Journal on Selected Areas in Communications, 25(8): 1590–1602, Oktober 2007.

[DH76] Whitfield Diffie, Martin E. Hellman. *New Directions in Cryptography.* IEEE Transactions on Information Theory, 22(6): 644–654, November 1976.

[Dij59] Edsger W. Dijkstra. *A Note on Two Problems in Connexion with Graphs.* Numerische Mathematik, 1: 269–271, 1959.

[DKKS05] Florian Dötzer, Florian Kohlmayer, Timo Kosch, Markus Strassberger. *Secure Communication for Intersection Assistance.* In: Proceedings of the 2nd International Workshop on Intelligent Transportation, März 2005.

[Döt05] Florian Dötzer. *Privacy Issues in Vehicular Ad Hoc Networks.* In: George Danezis, David Martin (Herausgeber), Proceedings of the Privacy Enhancing Technologies, 5th International Workshop, PET 2005, Seite 197–209, 2005.

[DS03] Claudia Díaz, Andrei Serjantov. *Generalising Mixes.* In: Roger Dingledine (Herausgeber), Privacy Enhancing Technologies, Third International Workshop, PET 2003, Dresden, Germany, March 26-28, 2003, Revised Papers, Volume 2760 of Lecture Notes in Computer Science, Seite 18–31. Springer, 2003.

[Eic07] Stephan Eichler. *Strategies for Pseudonym Changes in Vehicular Ad Hoc Networks depending on Node Mobility.* In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Seite 541–546, Istanbul, Turkey, Juni 2007. IEEE.

[Elg85] Taher Elgamal. *A public key cryptosystem and a signature scheme based on discrete logarithms.* IEEE Transactions on Information Theory, 31(4): 469–472, 1985.

[Eur05] European Space Agency. *Galileo – The European Programme for Global Navigation Services.* Technical report, European Commision, Januar 2005.

[Eur08] European GNSS Supervisory Authority (GSA). *GALILEO and EGNOS – Europe's Satellite Navigation Programmes.* Office for Official Publications of the European Communities, Luxemburg, 2008. ISBN 978-929206001-5

[FC06] Daniel B. Faria, David R. Cheriton. *Detecting identity-based attacks in wireless networks using signalprints.* In: Proceedings of the 5th ACM workshop on Wireless security, WiSe '06, Seite 43–52, New York, NY, USA, 2006. ACM.

[Fed99a] Hannes Federrath. *Protection in Mobile Communications.* In: Günter Müller, Kai Rannenberg (Herausgeber), Multilateral security in communications, Information security, Seite 349–364. Addison-Wesley, München, 1999.

[Fed99b] Hannes Federrath. *Sicherheit mobiler Kommunikation - Schutz in GSM-Netzen, Mobilitätsmanagement und mehrseitige Sicherheit.* DuD-Fachbeiträge. Vieweg, Wiesbaden, 1999.

[Fed01] Federal Processing Standards Publication. *Announcing the Advanced Encryption Standard (AES).* Technical Report FIPS Publication 197, National Institute of Standards and Technology, 2001.

[FFBA07] Emanuel Fonseca, Andreas Festag, Roberto Baldessari, Rui Aguiar. *Support of Anonymity in VANETs – Putting Pseudonymity into Practice.* In: In Proceedings WCNC, Hong Kong, März 2007.

[FGF06] Nicola Fricke, Charlotte Glaser, Monica De Filippis. *Passive und Aktive Sicherheitsmaßnahmen im Kraftfahrzeug.* MMI-Interaktiv, 8: 43–47, August 2006.

[FHFB07] Marco Fiore, Jérôme Härri, Fethi Filali, Christian Bonnet. *Vehicular Mobility Simulation for VANETs.* In: Proceedings 40th Annual Simulation Symposium (ANSS-40 2007), Seite 301–309, Norfolk, Virginia, USA, 2007. IEEE Computer Society.

[FJP96] Hannes Federrath, Anja Jerichow, Andreas Pfitzmann. *Mixes in mobile communication systems: Location management with privacy.* In: Ross Anderson (Herausgeber), Information hiding: first international workshop, Volume 1174. Springer Berlin / Heidelberg, 1996.

[FMH+02] Holger Füßler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, Dieter Vollmer. *A Comparison of Routing Strategies for Vehicular Ad Hoc Networks.* Technical Report TR-02-003, Department of Computer Science, University of Mannheim, 2002.

[FP97] Hannes Federrath, Andreas Pfitzmann. *Bausteine zur Realisierung mehrseitiger Sicherheit*, Seite 83–104. Addison-Wesley-Longman, 1997.

[FP00] Hannes Federrath, Andreas Pfitzmann. *Gliederung und Systematisierung von Schutzzielen in IT-Systemen.* Datenschutz und Datensicherheit, 24(12), 2000.

[FRF+07] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, Jean-Pierre Hubaux. *Mix-Zones for Location Privacy in Vehicular Networks*. In: ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS), Vancouver, 2007.

[FSH09] Julien Freudiger, Reza Shokri, Jean-Pierre Hubaux. *On the Optimal Placement of Mix Zones*. In: Privacy Enhancing Technologies Symposium (PETS), Seite 216–234, Seattle, 2009.

[Gal03] Galilei Project. *The Galilei Project – GALILEO Design Consolidation*. Technical report, European Commision, 2003.

[GBW07] Jinhua Guo, John P. Baugh, Shengquan Wang. *A Group Signature Based Secure and Privacy-Preserving Vehicular Communication Framework*. In: 2007 Mobile Networking for Vehicular Environments, Seite 103 – 108, Anchorage, AK, Mai 2007.

[GDA95] *GDA - Geocentric Datum of Australia: Technical Manual*. Technical report, Intergovernmental Committee on Surveying & Mapping, 1995.

[Ger06] Matthias Gerlach. *Assessing and Improving Privacy in VANETs*. In: Embedded Security in Cars (ESCAR), 2006.

[GFL+07] Matthias Gerlach, Andreas Festag, Tim Leinmüller, Gabriele Goldacker, Charles Harsch. *Security Architecture for Vehicular Communication*. In: Fourth International Workshop on Intelligent Transportation (WIT 2007), 2007.

[GG03] Marco Gruteser, Dirk 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, Seite 31–42, New York, NY, USA, 2003. ACM.

[GGS04] Philippe Golle, Dan Greene, Jessica Staddon. *Detecting and correcting malicious data in VANETs*. In: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, VANET '04, Seite 29–37, New York, NY, USA, 2004. ACM.

[GH00] Silvia Giordano, M. Hamdi. *Mobility Management: The Virtual Home Region*. Technical report, EPFL, Oktober 2000.

[Gip81] Peter G. Gipps. *A behavioural car-following model for computer simulation*. Transportation Research Part B: Methodological, 15(2): 105 – 111, 1981.

[Gip86] Peter G. Gipps. *A model for the structure of lane-changing decisions*. Transportation Research Part B: Methodological, 20(5): 403 – 414, 1986.

[GL05] Buğra Gedik, Ling Liu. *A Customizable k-Anonymity Model for Protecting Location Privacy.* In: Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS 2005), Seite 620–629. IEEE Computer Society, 2005.

[GMP+08] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, David Wetherall. *Improving wireless privacy with an identifier-free link layer protocol.* In: Proceedings of the 6th international conference on Mobile systems, applications, and services, MobiSys '08, Seite 40–53, New York, NY, USA, 2008. ACM.

[Gör10] Barbara Görres. *Vom Globalen Bezugssystem bis zur Umsetzung für die Praxis.* In: GNSS 2010 - Vermessung und Navigation im 21. Jahrhundert, Volume 63 of DVW-Schriftenreihe, Seite 39–57. Wißner-Verlag, Augsburg, Oktober 2010.

[GP09] Philippe Golle, Kurt Partridge. *On the Anonymity of Home/Work Location Pairs.* In: Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing, Seite 390–397, Berlin, Heidelberg, 2009. Springer-Verlag.

[Gru08] Bernhard Gruber. *Entwicklung eines agentenbasierten Simulators für VANETs.* Diplomarbeit, Universität Regensburg, Regensburg, November 2008.

[Har08] Hartwig Harm. *Umrechnung zwischen geographischen Koordinaten (Länge und Breite) und dem UTM-Format.* `http://www.mydarc.de/dh2mic/files/utm.pdf`, abgerufen am 28.03.2012, 2008.

[HCL04] Jean-Pierre Hubaux, Srdjan Capkun, Jun Luo. *The Security and Privacy of Smart Vehicles.* IEEE Security & Privacy, 2(3): 49–55, 2004.

[HD06] R. Hinden, S. Deering. *IP Version 6 Addressing Architecture.* RFC 4291 (Draft Standard), 2006. *(`http://www.ietf.org/rfc/rfc4291.txt`)*

[Hei05] Heise Online. *Die Auseinandersetzung um Ausweise mit digitalisierten biometrischen Merkmalen.* `http://heise.de/-302400`, abgerufen am 24.09.2012, November 2005.

[Hei06] Heise Online. *Auto zahlt Parkticket selbst.* `http://heise.de/-145508`, abgerufen am 28.10.2010, Juli 2006.

[Hei08] Heise Online. *Big Brother kommt an.* `http://heise.de/-213477`, abgerufen am 28.10.2010, Juni 2008.

[Hei10a] Heise Online. *Google Street View sorgt weiter für Erregung.* `http://heise.de/-1054466`, abgerufen am 24.09.2012, August 2010.

[Hei10b] Heise Online. *Google Street View spaltet die Gemüter.* `http://heise.de/-1059127`, abgerufen am 24.09.2012, August 2010.

[Hei10c] Heise Online. *Klagen gegen die elektronische Gesundheitskarte.* `http://heise.de/-1068127`, abgerufen am 24.09.2012, August 2010.

[HFBF06] Jérôme Härri, Fethi Filali, Christian Bonnet, Marco Fiore. *VanetMobiSim: generating realistic mobility patterns for VANETs.* In: Wieland Holfelder, David B. Johnson, Hannes Hartenstein, Victor Bahl (Herausgeber), Proceedings of the Third International Workshop on Vehicular Ad Hoc Networks, VANET 2006, Seite 96–97, Los Angeles, CA, USA, September 2006. ACM.

[HFFB11] Jérôme Härri, Marco Fiore, Fethi Filali, Christian Bonnet. *Vehicular mobility simulation with VanetMobiSim.* SIMULATION: Transactions of The Society for Modeling and Simulation International, 87(4), April 2011.

[HKO09] Rasheed Hussain, Sangjin Kim, Heekuck Oh. *Towards Privacy Aware Pseudonymless Strategy for Avoiding Profile Generation in VANET.* In: Heung Youl Youm, Moti Yung (Herausgeber), Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers, Volume 5932 of Lecture Notes in Computer Science, Seite 268–280. Springer, 2009.

[HL91] Sauli Häkkinen, Juha Luoma. *Liikennepsykologia.* Otatieto, 1991.

[HMYS05] Leping Huang, Kanta Matsuura, Hiroshi Yamane, Kaoru Sezaki. *Enhancing wireless location privacy using silent period.* In: Proceedings of the 2005 IEEE Wireless Communications and Networking Conference (WCNC), Seite 1187–1192, New Orleans, LA, USA, März 2005.

[HP01] Zygmunt J. Haas, Marc R. Pearlman. *The performance of query control schemes for the zone routing protocol.* IEEE/ACM Transactions on Networking (TON), 9(4): 427–438, 2001.

[HW08] Mordechai (Muki) Haklay, Patrick Weber. *OpenStreetMap: User-Generated Street Maps.* IEEE Pervasive Computing, 7(4): 12–18, 2008.

[HYMS05] Leping Huang, Hiroshi Yamane, Kanta Matsuura, Kaoru Sezaki. *Towards Modeling Wireless Location Privacy.* In: George Danezis, David Martin (Herausgeber), Privacy Enhancing Technologies, Volume 3856 of Lecture Notes in Computer Science, Seite 59–77. Springer, 2005.

[HYMS06] Leping Huang, Hiroshi Yamane, Kanta Matsuura, Kaoru Sezaki. *Silent Cascade: Enhancing Location Privacy Without Communication QoS*

Degradation. In: John A. Clark, Richard F. Paige, Fiona Polack, Phillip J. Brooke (Herausgeber), Security in Pervasive Computing, Third International Conference, SPC 2006, York, UK, April 18-21, 2006, Proceedings, Volume 3934 of Lecture Notes in Computer Science, Seite 165–180. Springer, 2006.

[IEE10] IEEE Computer Society. *IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*, Juli 2010.

[IL10] Muhammad Usman Iqbal, Samsung Lim. *Privacy Implications of Automated GPS Tracking and Profiling.* IEEE Technology and Society Magazine, 29(2): 39–46, Juni 2010.

[JBH11] Murtuza Jadliwala, Igor Bilogrevic, Jean-Pierre Hubaux. *Optimizing Mixing in Pervasive Networks: A Graph-Theoretic Perspective.* In: Vijay Atluri, Claudia Díaz (Herausgeber), Proceedings of the 16th European Symposium on Research in Computer Security, ESORICS 2011, Volume 6879 of Lecture Notes in Computer Science, Seite 548–567, Berlin / Heidelberg, 2011. Springer.

[JM96] David B. Johnson, David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, Volume 353, Seite 153–181. Kluwer Academic Publishers, 1996.

[KB11] Kraftfahrt-Bundesamt. *Fahrzeugklassen und Aufbauarten – Zeitreihe 1955 bis 2011.* `http://www.kba.de/cln_033/nn_124394/sid_9B80E80C303B74B33A8117CCBF236DDE/nsc_true/DE/Statistik/Fahrzeuge/Bestand/FahrzeugklassenAufbauarten/b__fzkl__zeitreihe.html?__nnn=true`, abgerufen am 25.10.2011, 2011.

[KEB98] Doğan Kesdoğan, Jan Egner, Roland Büschkes. *Stop-and-Go-MIXes Providing Probabilistic Anonymity in an Open System.* In: David Aucsmith (Herausgeber), Information Hiding, Second International Workshop, Portland, Oregon, USA, April 14-17, 1998, Proceedings, Volume 1525 of Lecture Notes in Computer Science, Seite 83–98. Springer, 1998.

[Ken94] Melita Kennedy. *Understanding Map Projections.* Technical report, ETSI - Environmental Systems Research Institute Inc., 1994.

[KGP+09] Manuel Koschuch, Johann Großschädl, Dan Page, Philipp Grabher, Matthias Hudler, Michael Krüger. *Hardware/Software Co-design of Public-Key Cryptography for SSL Protocol Execution in Embedded Systems.* In: Sihan Qing, Chris J. Mitchell, Guilin Wang (Herausgeber),

Information and Communications Security, 11th International Conference, ICICS 2009, Beijing, China, December 14-17, 2009. Proceedings, Volume 5927 of Lecture Notes in Computer Science, Seite 63–79. Springer, 2009.

[KK00] Brad Karp, H. T. Kung. *GPSR: greedy perimeter stateless routing for wireless networks.* In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, Seite 243–254, 2000.

[KMMP97] C.N. Kloeden, A.J. McLean, V.M. Moore, G. Ponte. *Travelling Speed and the Risk of Crash Involvment.* Technical report, NHMRC Road Accident Research Unit, University of Adelaide, November 1997.

[KR08] James F. Kurose, Keith W. Ross. *Computernetzwerke: Der Top-Down-Ansatz.* Pearson Studium. Pearson Studium, 2008. ISBN 9783827373304

[Kra98] Stefan Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics.* Dissertation, Universität Köln, Köln, April 1998.

[Kru07] John Krumm. *Inference Attacks on Location Tracks.* In: Anthony LaMarca, Marc Langheinrich, Khai Truong (Herausgeber), Pervasive Computing, Volume 4480 of Lecture Notes in Computer Science, Seite 127–143. Springer, Berlin / Heidelberg, 2007.

[KSW05] Frank Kargl, Stefan Schlott, Michael Weber. *Identitäten in mobilen Ad hoc Netzwerken.* In: Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Herausgeber), INFORMATIK 2005 - Informatik LIVE! Band 2, Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e.V. (GI), Bonn, 19. bis 22. September 2005, Volume 68 of LNI, Seite 349–353. Gesellschaft für Informatik, 2005.

[Kun03] Christian P. Kunze. *Digitale Identität und Identitäts-Management.* In: GI Gesellschaft für Informatik e.V. (Herausgeber), Informatiktage 2003, Seite 197–201, November 2003.

[KV99] Young-Bae Ko, Nitin H. Vaidya. *Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms.* In: Proceedings of the 2nd Workshop on Mobile Computing Systems and Applications (WMCSA '99), Seite 101–110. IEEE Computer Society, Februar 1999.

[Lan03] Klaus Langwieder. *Perspektiven der aktiven und passiven Fahrzeugsicherheit.* Verkehrsunfall und Fahrzeugtechnik, 1: 11–18, 2003.

[LB06] Christine Laurendeau, Michel Barbeau. *Threats to Security in DSRC/WAVE.* In: Thomas Kunz, S. S. Ravi (Herausgeber), Proceedings of the Ad-Hoc, Mobile, and Wireless Networks, 5th International Conference, ADHOC-NOW 2006, Ottawa, Canada, Volume 4104 of Lecture Notes in Computer Science, Seite 266–279. Springer, 2006.

[LBH+06] Tim Leinmüller, Levente Buttyán, Jean-Pierre Hubaux, Frank Kargl, Rainer Kroh, Panos Papadimitratos, Maxim Raya, Elmar Schoch. *SEVECOM - Secure Vehicle Communication.* In: Proceedings of IST Mobile Summit 2006, 2006.

[LG03] Gustav Lindström, Giovanni Gasparini. *The Galileo satellite system and its security implications.* Occasional Papers, 44, April 2003.

[LHV07] Ming Lei, Xiaoyan Hong, Susan V. Vrbsky. *Protecting Location Privacy with Dynamic Mac Address Exchanging in Wireless Networks.* In: Proceedings of the Global Communications Conference, 2007. GLOBECOM '07, Seite 49–53, Washington, DC, USA, November 2007. IEEE.

[LJDC+00] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, Robert Morris. *A scalable location service for geographic ad hoc routing.* In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, Seite 120–130, New York, NY, USA, 2000. ACM.

[LLL+11] Rongxing Lu, Xiaodong Lin, Tom H. Luan, Xiaohui Liang, Xuemin Shen. *Anonymity Analysis on Social Spot Based Pseudonym Changing for Location Privacy in VANETs.* In: Proceedings of IEEE International Conference on Communications, ICC 2011, Seite 1–5, Kyoto, Japan, Juni 2011. IEEE.

[LLZ+08] Xiadong Lin, Rongxing Lu, Chenxi Zhang, Haojin Zhu, Pin-Han Ho, Xuemin Shen. *Security in vehicular ad hoc networks.* IEEE Communications Magazine, 46(4): 88–95, April 2008.

[LP99] W.A. Leaf, David F. Preusser. *Literature review on vehicle travel speeds and pedestrian injuries.* NHTSA technical report. U.S. Dept. of Transportation, National Highway Traffic Safety Administration, Oktober 1999.

[LP06] Loukas Lazos, Radha Poovendran. *HiRLoc: high-resolution robust localization for wireless sensor networks.* IEEE Journal on Selected Areas in Communications, 24(2): 233–246, 2006.

[LSHP06] Mingyan Li, Krishna Sampigethaya, Leping Huang, Radha Poovendran. *Swing & swap: user-centric approaches towards maximizing location*

privacy. In: Ari Juels, Marianne Winslett (Herausgeber), Proceedings of the 2006 ACM Workshop on Privacy in the Electronic Society, WPES 2006, Alexandria, VA, USA, October 30, 2006, Seite 19–28. ACM, 2006.

[Mai04] Christian Maihöfer. *A Survey of Geocast Routing Protocols.* IEEE Communications, 6(2): 32–42, Q2 2004.

[Mas07] Sergio Mascetti. *Privacy Protection through Anonymity in Location-based Services.* PhD thesis, Università degli Studi di Milano / University of Milan, Milan, Italy, 2007.

[MC98] Joseph P. Macker, M. Scott Corson. *Mobile ad hoc networking and the IETF.* SIGMOBILE Mob. Comput. Commun. Rev., 2(1): 9–14, 1998.

[MCA06] Mohamed F. Mokbel, Chi-Yin Chow, Walid G. Aref. *The New Casper: Query Processing for Location Services without Compromising Privacy.* In: Proceedings of the 32nd international conference on Very large data bases, VLDB '06, Seite 763–774. VLDB Endowment, 2006.

[MES04] Christian Maihöfer, Reinhold Eberhardt, Elmar Schoch. *CGGC: Cached Greedy Geocast.* In: Peter Langendoerfer, Mingyan Liu, Ibrahim Matta, Vassilis Tsaoussidis (Herausgeber), Wired/Wireless Internet Communications, Volume 2957, Seite 171–182. Springer Berlin / Heidelberg, 2004.

[MFA07] Guoqiang Mao, BarIs Fidan, Brian Anderson. *Wireless sensor network localization techniques.* Computer Networks, 51(10): 2529 – 2553, 2007.

[Mil85] Victor S. Miller. *Use of Elliptic Curves in Cryptography.* In: Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings, Volume 218 of Lecture Notes in Computer Science, Seite 417–426. Springer, 1985.

[MJK+00] Robert Morris, John Jannotti, Frans Kaashoek, Jinyang Li, Douglas Decouto. *CarNet: a scalable ad hoc wireless network system.* In: Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system, EW 9, Seite 61–65, New York, NY, USA, 2000. ACM.

[MKW08] Zhendong Ma, Frank Kargl, Michael Weber. *Pseudonym-on-demand: a new pseudonym refill strategy for vehicular communications.* In: Proceedings of the 68th IEEE Vehicular Technology Conference, VTC Fall 2008, 21-24 September 2008, Seite 1–5. IEEE, Sep 2008.

[MWH01] Martin Mauve, Jörg Widmer, Hannes Hartenstein. *A Survey on Position-Based Routing in Mobile Ad Hoc Networks.* IEEE Network, 15(6): 30–39, November/Dezember 2001.

[MWR+06] Rahul Mangharam, Daniel Weller, Raj Rajkumar, Priyantha Mudalige, Fan Bai. *GrooveNet: A Hybrid Simulator for Vehicle-to-Vehicle Networks*. Annual International Conference on Mobile and Ubiquitous Systems, Seite 1–8, 2006.

[MWS+05] Rahul Mangharam, Daniel S. Weller, Daniel D. Stancil, Ragunathan Rajkumar, Jayendra S. Parikh. *GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks*. In: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, VANET '05, VANET '05, Seite 59–68, New York, NY, USA, 2005. ACM.

[Nag93] Takashi Nagatani. *Jamming transition in the traffic-flow model with two-level crossings*. Physical Review E, 48(5): 3290–3294, November 1993.

[Nat77] National Bureau of Standards. *Announcing the Data Encryption Standard*. Technical Report FIPS Publication 46, National Bureau of Standards, Januar 1977.

[Nat01] National Institute of Standards and Technology, NIST. *Federal Informaion Processsing Standards Publication (FIPS) 140-2: Security Requirements for Cryptographic Modules*. Technical Report FIPS Publication 140, U.S. Department of Commerce, 2001.

[NH99] Harry Niemann, Armin Hermann. *Geschichte der Strassenverkehrssicherheit im Wechselspiel zwischen Fahrzeug, Fahrbahn und Mensch*. Wissenschaftliche Schriftenreihe des DaimlerChrysler Konzernarchivs. Delius Klasing, Bielefeld, 1999. ISBN 9783768811682

[Nil82] Nils J. Nilsson. *Principles of Artificial Intelligence*. Symbolic Computation: Artificial Intelligence. Springer, Mai 1982. ISBN 9783540113409

[NNS03] G.P. Nikishkov, Yu.G. Nikishkov, V.V. Savchenko. *Comparison of C and Java performance in finite element computations*. Computers & Structures, 81(24–25): 2401 – 2408, September 2003.

[NS92] Kai Nagel, Michael Schreckenberg. *A cellular automaton model for freeway traffic*. J. Phys. I France, 2(12): 2221–2229, September 1992.

[NS10] Amiya Nayak, Ivan Stojmenović. *Wireless Sensor and Actuator Networks – Algorithms and Protocols for Scalable Coordination and Data Communication*. John Wiley & Sons, Ottawa, 2010.

[NZZ12] Pascal Neis, Dennis Zielstra, Alexander Zipf. *The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007-2011*. Future Internet, 4(1): 1–21, 2012.

[Opp12] Marvin Oppong. *Süddeutsche.de: Widerstand gegen elektronische Gesundheitskarte.* `http://www.sueddeutsche.de/geld/x-1.1443420`, abgerufen am 24.09.2012, August 2012.

[Pat11] Amit Patel. *Introduction to A*.* `http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html`, abgerufen am 26.04.2012, September 2011.

[PB94] Charles E. Perkins, Pravin Bhagwat. *Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers.* SIGCOMM Comput. Commun. Rev., 24(4): 234–244, Oktober 1994.

[PBH+07] Panos Papadimitratos, Levente Buttyán, Jean-Pierre Hubaux, Frank Kargl, Antonio Kung, Maxim Raya. *Architecture for Secure and Private Vehicular Communications.* In: Proceedings of the 7th International Conference on ITS Telecommunications, Juni 2007.

[PBH+08] Panos Papadimitratos, Levente Buttyán, Tamás Holczer, Elmar Schoch, Zhendong Ma, Frank Kargl, Antonio Kung, Jean-Pierre Hubaux. *Secure vehicular communication systems: design and architecture.* IEEE Wireless Communications Magazine, Seite 100–109, November 2008.

[Pea84] Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984. ISBN 0-201-05594-5

[PF07] Klaus Plößl, Hannes Federrath. *A Privacy Aware and Efficient Security Infrastructure for Vehicular Ad Hoc Networks.* In: Security in Information Systems: Proceedings of the 5th International Workshop on Security in Information Systems – WOSIS 2007, 2007.

[Pfi93] Andreas Pfitzmann. *Technischer Datenschutz in öffentlichen Funknetzen.* Datenschutz und Datensicherheit - DuD, 8: 451–463, August 1993.

[PGG+07] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, David Wetherall. *802.11 user fingerprinting.* In: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, MobiCom '07, Seite 99–110, New York, NY, USA, 2007. ACM.

[Plö09] Klaus Plößl. *Mehrseitig sichere Ad-hoc-Vernetzung von Fahrzeugen.* Gabler, Wiesbaden, DuD-Fachbeiträge. Edition, 2009.

[Pon93] György Pongor. *OMNeT: Objective Modular Network Testbed.* In: Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, MASCOTS

'93, Seite 323–326, San Diego, CA, USA, 1993. The Society for Computer Simulation International.

[PP89] Birgit Pfitzmann, Andreas Pfitzmann. *How to Break the Direct RSA-Implementation of Mixes.* In: Jean-Jacques Quisquater, Joos Vandewalle (Herausgeber), Proceedings of Advances in Cryptology - EUROCRYPT '89, Workshop on the Theory and Application of of Cryptographic Techniques, Volume 434 of Lecture Notes in Computer Science, Seite 373–381. Springer, 1989.

[PP05] Bryan Parno, Adrian Perrig. *Challenges in Securing Vehicular Networks.* In: Proceedings of Workshop on Hot Topics in Networks (HotNets-IV), November 2005.

[PPW88] Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner. *Datenschutz garantierende offene Kommunikationsnetze.* Informatik Spektrum, 11(3): 118–142, 1988.

[PPW91] Andreas Pfitzmann, Birgit Pfitzmann, Michael Waidner. *ISDN-MIXes: Untraceable Communication with Very Small Bandwidth Overhead.* In: In Proceedings of the GI/ITG Conference on Communication in Distributed Systems, Seite 451–463. Springer-Verlag, 1991.

[PR99] Charles E. Perkins, Elizabeth M. Royer. *Ad-hoc On-Demand Distance Vector Routing.* Second IEEE Workshop on Mobile Computer Systems and Applications, Seite 90, 1999.

[PRL+08] M. Piórkowski, M. Raya, A. Lezama Lugo, P. Papadimitratos, M. Grossglauser, J.-P. Hubaux. *TraNS: realistic joint traffic and network simulator for VANETs.* ACM SIGMOBILE Mobile Computing and Communications Review, 12(1): 31–33, 2008.

[PVC01] Michael Paleczny, Christopher A. Vick, Cliff Click. *The Java HotSpot Server Compiler.* In: Proceedings of the 1st Java Virtual Machine Research and Technology Symposium, Monterey, CA, USA, April 2001. USENIX.

[PWK05] Oscar Pozzobon, Chris Wullems, Kurt Kubic. *Secure Tracking using Trusted GNSS Receivers and Galileo Authentication Services.* Journal of Global Positioning Systems, 3(1-2), 2005.

[RCC07] Kasper Bonne Rasmussen, Srdjan Capkun, Mario Cagalj. *SecNav: secure broadcast localization and time synchronization in wireless networks.* In: Evangelos Kranakis, Jennifer C. Hou, Ram Ramanathan (Herausgeber), Proceedings of the 13th Annual International Conference on Mobile Computing and Networking, MOBICOM 2007, Montréal, Québec, Canada, Seite 310–313. ACM, 2007.

[Rei10] Konrad Reif. *Sicherheitssysteme.* In: Konrad Reif (Herausgeber), Fahrstabilisierungssysteme und Fahrerassistenzsysteme, Seite 176–191. Vieweg+Teubner, 2010.

[Res99] E. Rescorla. *Diffie-Hellman Key Agreement Method.* RFC 2631 (Proposed Standard), Juni 1999. *(`http://www.ietf.org/rfc/rfc2631.txt`)*

[RFSK08] Stefan Rass, Simone Fuchs, Martin Schaffer, Kyandoghere Kyamakya. *How to Protect Privacy in Floating Car Data Systems.* In: Varsha K. Sadekar, Paolo Santi, Yih-Chun Hu, Martin Mauve (Herausgeber), Proceedings of the Fifth International Workshop on Vehicular Ad Hoc Networks, VANET 2008, Seite 17–22, San Francisco, California, USA, 2008. ACM.

[RG05] K. Rüdiger, M. Gersch. *Embedded Security in Cars*, Kapitel In-Vehicle M-Commerce: Business Models for Navigation Systems and Location-based Servies. Springer-Verlag, 2005.

[RH05] Maxim Raya, Jean-Pierre Hubaux. *The Security of Vehicular Ad Hoc Networks.* In: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN), Seite 11–21. ACM, November 2005.

[RH07] Maxim Raya, Jean-Pierre Hubaux. *Securing vehicular ad hoc networks.* Journal of Computer Security, 15(1): 39–68, Januar 2007.

[RJP+06] Maxim Raya, Daniel Jungels, Panos Papadimitratos, Imad Aad, Jean pierre Hubaux. *Certificate revocation in vehicular networks.* Technical report, 2006.

[RNSL96] Marcus Rickert, Kai Nagel, Michael Schreckenberg, Andreas Latour. *Two lane traffic simulations using cellular automata.* Physica A: Statistical Mechanics and its Applications, 231(4): 534 – 550, 1996.

[Ros03] Martin Rose. *Modellbildung und Simulation von Autobahnverkehr.* Dissertation, Universität Hannover, Juni 2003.

[RPA+07] Maxim Raya, Panos Papadimitratos, Imad Aad, Daniel Jungels, Jean-Pierre Hubaux. *Eviction of Misbehaving and Faulty Nodes in Vehicular Networks.* IEEE Journal on Selected Areas in Communications, 2007.

[RPH06] Maxim Raya, Panos Papadimitratos, Jean-Pierre Hubaux. *Securing Vehicular Communications.* IEEE Wireless Communications, 13(5): 8–15, October 2006.

[RSA78] Ronald L. Rivest, Adi Shamir, Leonard M. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.* Communications of the ACM, 21(2): 120–126, 1978.

[Sam89] Hanan Samet. *Hierarchical Spatial Data Structures.* In: Alejandro P. Buchmann, Oliver Günther, Terence R. Smith, Y.-F. Wang (Herausgeber), Design and Implementation of Large Spatial Databases, First Symposium SSD'89, Santa Barbara, California, July 17/18, 1989, Proceedings, Volume 409 of Lecture Notes in Computer Science, Seite 193–212. Springer, 1989.

[Sam90] Hanan Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.

[Sar00] Palash Sarkar. *A Brief History of Cellular Automata.* ACM Computing Surveys (CSUR), 32(1): 80–107, März 2000.

[SBF10] Florian Scheuer, Matthias Brecht, Hannes Federrath. *A privacy-aware location service for VANETs using Chaum's mixes.* In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2010), Niagara Falls, Canada, 2010.

[Sch08] Florian Scheuer. *Privatsphäre und Sicherheit bei VANETs.* Diplomarbeit, Universität Regensburg, Februar 2008.

[SD08] Christoph Sommer, Falko Dressler. *Progressing Towards Realistic Mobility Models in VANET Simulations.* IEEE Communications Magazine, 46(11): 132–137, November 2008.

[SE90] Jeffrey Star, John Estes. *Geographic information systems: an introduction.* Prentice Hall series in geographic information science. Prentice Hall, 1990.

[SFF11] Florian Scheuer, Karl-Peter Fuchs, Hannes Federrath. *A Safety-Preserving Mix Zone for VANETs.* In: Steven Furnell, Costas Lambrinoudakis, Günther Pernul (Herausgeber), Proceedings of Trust, Privacy and Security in Digital Business - 8th International Conference, TrustBus 2011, Toulouse, France, Volume 6863 of Lecture Notes in Computer Science, Seite 37–48. Springer Berlin / Heidelberg, 2011.

[SGD11] Christoph Sommer, Reinhard German, Falko Dressler. *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis.* IEEE Transactions on Mobile Computing, 10(1): 3–15, Januar 2011.

[Sha48] Claude Elwood Shannon. *A Mathematical Theory of Communication.* Bell System Technical Journal, 27(3): 379–423, Juli 1948.

[SHL+05] Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, Kaoru Sezaki. *CARAVAN: Providing location privacy for VANET.* In: Embedded Security in Cars (ESCAR), 2005.

[SKL+06] Elmar Schoch, Frank Kargl, Tim Leinmüller, Stefan Schlott, Panos Papadimitratos. *Impact of Pseudonym Changes on Geographic Routing in VANETs.* In: Security and Privacy in Ad-Hoc and Sensor Networks, Third European Workshop, ESAS 2006, Hamburg, Germany, September 20-21, 2006, Revised Selected Papers, Lecture Notes in Computer Science, Seite 43–57. Springer, 2006.

[SKW+98] Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson. *Twofish: A 128-Bit Block Cipher.* In: in First Advanced Encryption Standard (AES) Conference, Juni 1998.

[SLHP07] Krishna Sampigethaya, Mingyan Li, Leping Huang, Radha Poovendran. *AMOEBA: Robust Location Privacy Scheme for VANET.* IEEE Journal on Selected Areas in Communications, 25(8): 1569–1589, 2007.

[SPF08] Florian Scheuer, Klaus Plößl, Hannes Federrath. *Preventing Profile Generation in Vehicular Networks.* In: WIMOB '08: Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication, Seite 520–525, Washington, DC, USA, 2008. IEEE Computer Society.

[SS98] Pierangela Samarati, Latanya Sweeney. *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression.* Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International, 1998.

[SSBP09] Ahren Studer, Elaine Shi, Fan Bai, Adrian Perrig. *TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs.* In: Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2009, Seite 1–9. IEEE Computer Society, 2009.

[Sta11] Statistisches Bundesamt. *Verkehrsunfälle – Polizeilich erfasste Unfälle – lange Reihe.* `https://www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/Verkehrsunfaelle/Tabellen_/Strassenverkehrsunfaelle.html`, abgerufen am 25.10.2011, 2011.

[Ste94] Anthony Stentz. *Optimal and Efficient Path Planning for Partially-Known Environments.* In: Proceedings of the International Conference on Robotics and Automation, ICRA94, Seite 3310–3317, 1994.

[Sto99] Ivan Stojmenovic. *Home agent based location update and destination search schemes in ad hoc wireless networks.* Technical Report TR-99-10, University of Ottawa, 1999.

[Swe02] Latanya Sweeney. *k-Anonymity: A Model for Protecting Privacy.* International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10(5): 557–570, 2002.

[SWL10] Joo-Han Song, Vincent W. S. Wong, Victor C. M. Leung. *Wireless Location Privacy Protection in Vehicular Ad-Hoc Networks.* Mobile Networks and Applications, 15(1): 160–171, Februar 2010.

[SYGD08] Christoph Sommer, Zheng Yao, Reinhard German, Falko Dressler. *On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation.* In: 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2008): 1st ACM International Workshop on Mobility Models for Networking Research (MobilityModels 2008), Seite 41–48, Hong Kong, China, Mai 2008. ACM.

[Tan03] Andrew S. Tanenbaum. *Computernetzwerke.* Pearson Studium, 4. Edition, 2003.

[TC03] Jing Tian, Luca Coletti. *Routing approach in CarTALK 2000 project.* In: Proceedings of the IST Mobile & Wireless Communications Summit 2003. Vol. 2, Seite 473–477, Aveiro - Portugal, Juni 2003. Instituto de Telecomunica cões (IT).

[TH02] Martin Treiber, Dirk Helbing. *Realistische Mikrosimulation von Straßenverkehr mit einem einfachen Modell.* In: 16. Symposium Simulationstechnik ASIM, 2002.

[THH00] Martin Treiber, Ansgar Hennecke, Dirk Helbing. *Congested traffic states in empirical observations and microscopic simulations.* Physical Review E, 62: 1805–1824, August 2000.

[TK84] Hideaki Takagi, Leonard Kleinrock. *Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals.* IEEE Transactions on Communications, 32(3): 246–257, März 1984.

[Tom09] Andreas Tomandl. *Entwicklung einer Komponente des VANET-Simulators zur Erstellung von Szenarien.* Bachelorarbeit, Universität Regensburg, Regensburg, August 2009.

[Tom11] Andreas Tomandl. *Verkehrsmodelle in Mikrosimulationen.* Theoreitsches Seminar, Universität Regensburg, Regensburg, Februar 2011.

[Tom12] Andreas Tomandl. *Evaluation von Konzepten zum Schutz der Privatsphäre in VANETs.* Masterarbeit, Universität Regensburg, Regensburg, Januar 2012.

[TP75] Steven Tanimoto, Theodosios Pavlidis. *A hierarchical data structure for picture processing.* Computer Graphics and Image Processing, 4(2): 104 – 119, 1975.

[TSF12] Andreas Tomandl, Florian Scheuer, Hannes Federrath. *Simulation-based evaluation of techniques for privacy protection in VANETs.* In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2012), 2012.

[TWLC01] Yu-Chee Tseng, Shih-Lin Wu, Wen-Hwa Liao, Chih-Min Chao. *Location Awareness in Ad Hoc Wireless Mobile Networks.* Computer, 34: 46–52, 2001.

[U.S09] U.S. Census Bureau. *Population Estimates – National Totals: Vintage 2009.* `http://www.census.gov/popest/data/historical/2000s/vintage_2009/index.html`, abgerufen am 27.01.2012, 2009.

[Var01] András Varga. *The OMNeT++ Discrete Event Simulation System.* Proceedings of the European Simulation Multiconference (ESM'2001), Juni 2001.

[Ver09] Verkehrsclub Deutschland. *Vision Zero – VCD-Masterplan (Kurzfassung).* `http://www.vcd.org/index.php?eID=tx_nawsecuredl&u=0&file=fileadmin/user_upload/redakteure_2010/themen/verkehrssicherheit/vision_zero/masterplan_vision_zero_090619.pdf&t=1319575697&hash=1f6a1e943f757940cce4a80285df52fbd7846f3e`, abgerufen am 19.10.2011, März 2009.

[Vol01] John A. Volpe. *Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning System.* Technical report, National Transportation Systems Center, August 2001.

[Waß04] René Waßmer (Herausgeber). *Vision Zero – Null Verkehrstote.* Verkehrsclub Deutschland e.V., Bonn, 2004.

[WB90] Samuel D. Warren, Louis D. Brandeis. *The Right to Privacy.* Harvard Law Review, 4(5): 193–220, 1890.

[WGS00] *Department of Defense (DoD) World Geodetic System 1984 (WGS84).* Technical Report 8350.2, 3rd Edition, National Imagery and Mapping Agency, Januar 2000.

[Whi12] Zack Whittaker. *UK government staff caught snooping on citizen data.* `http://www.zdnet.com/blog/london/`

uk-government-staff-caught-snooping-on-citizen-data/4716, abgerufen am 18.06.2012, Mai 2012.

[Wir75] Niklaus Wirth. *Algorithmen und Datenstrukturen.* Teubner Verlag, 1975.

[WJH97] A. Ward, A. Jones, A. Hopper. *A new location technique for the active office.* IEEE Personal Communications, 4(5): 42–47, 1997.

[WNR05] Xiaoxin Wu, Cristina Nita-Rotaru. *On the Security of Distributed Position Services.* In: SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, Seite 35–46, Washington, DC, USA, 2005. IEEE Computer Society.

[WPR+08] Axel Wegener, Michal Piorkowski, Maxim Raya, Horst Hellbrück, Stefan Fischer, Jean-Pierre Hubaux. *TraCI: An Interface for Coupling Road Traffic and Network Simulators.* In: 11th Communications and Networking Simulation Symposium (CNS), 2008.

[WS10] Albert Wasef, Xuemin (Sherman) Shen. *REP: Location Privacy for VANETs Using Random Encryption Periods.* Mobile Networks and Applications, 15(1): 172–185, Februar 2010.

[YLN03] Jungkeun Yoon, Mingyan Liu, Brian Noble. *Random Waypoint Considered Harmful.* In: INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE, April 2003.

[Zim01] Alf Zimmer. *Wie intelligent darf/muss ein Auto sein? Anmerkungen aus ingenieurpsychologischer Sicht*, Seite 39–55. Springer-Verlag, Heidelberg, 2001.

[ZY05] Zhi Zhou, Kin Choong Yow. *Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy.* In: ICDCSW '05: Proceedings of the Third International Workshop on Mobile Distributed Computing, Seite 646–651, Washington, DC, USA, 2005. IEEE Computer Society.

Anhang A

Statistische Verkehrsdaten

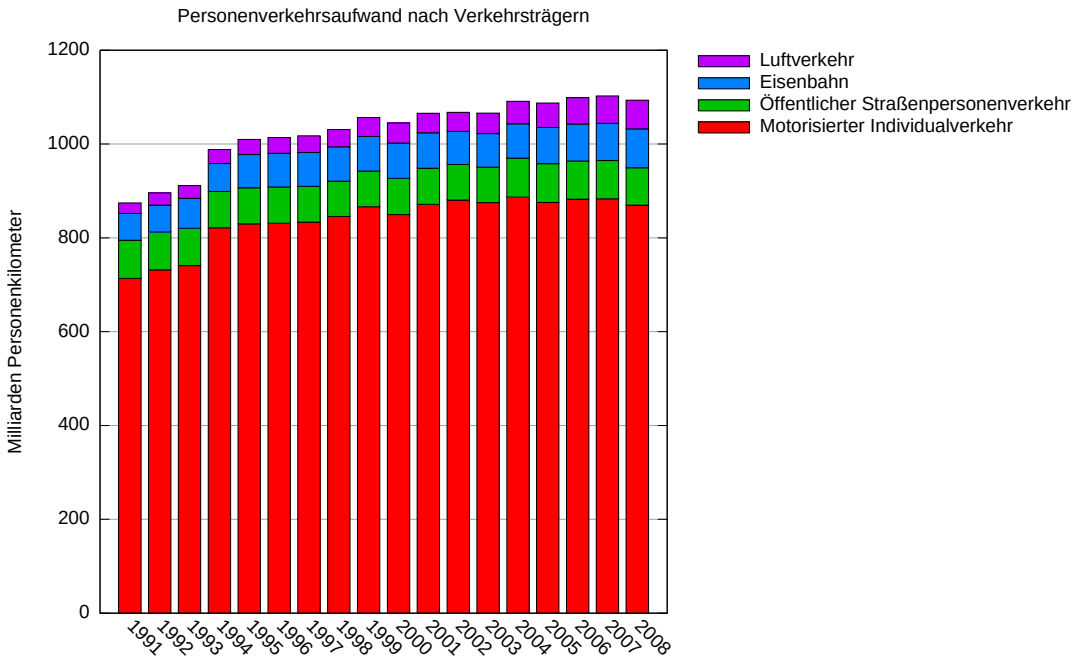


Abbildung A.1.: Personenverkehrsaufwand nach Verkehrsträgern in Deutschland [Bun10]

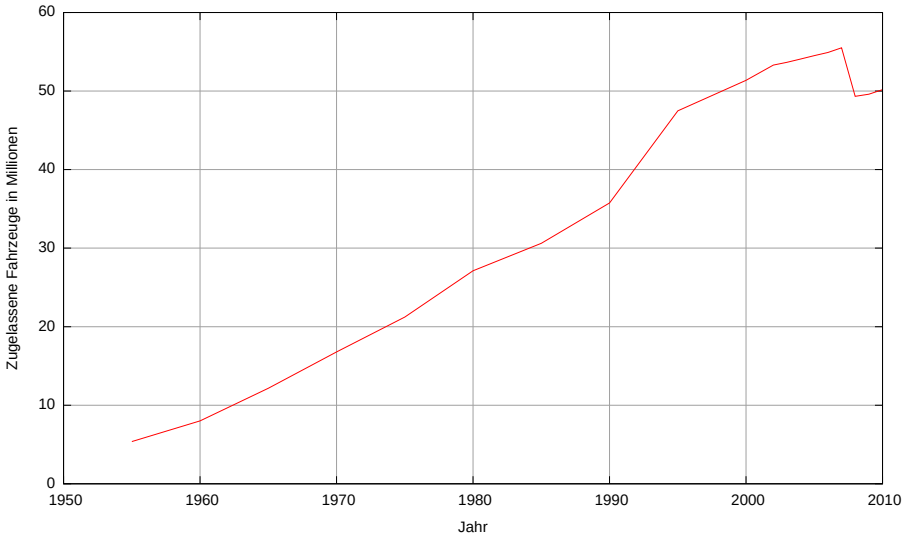


Abbildung A.2.: Fahrzeugbestand in Deutschland [KB11] (seit 2008 nur noch Fahrzeuge ohne vorübergehende Stilllegung)

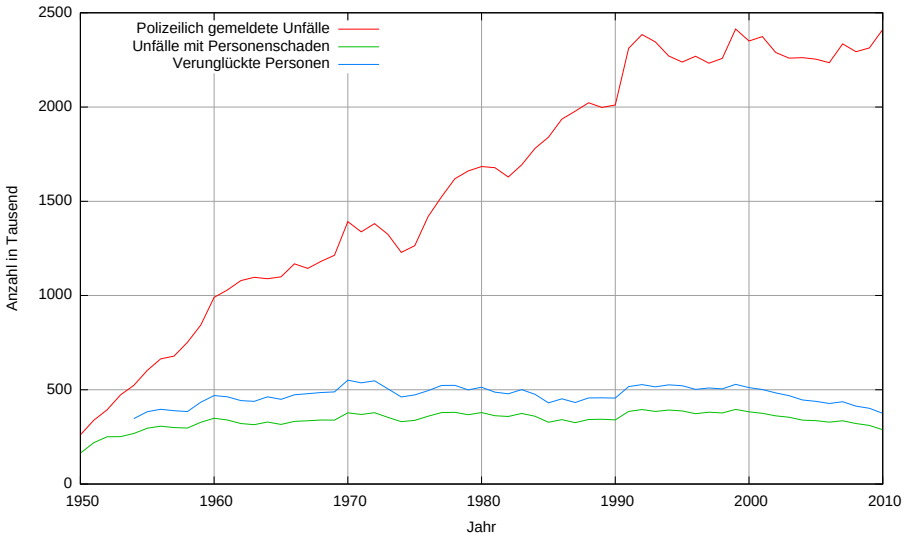


Abbildung A.3.: Polizeilich gemeldete Verkehrsunfälle in Deutschland [Sta11]

Anhang B

VANETsim XML-Schemata

B.1. XML-Schema der VANETsim-Karte

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
4   <xs:element name="Map">
5    <xs:complexType>
6     <xs:sequence>
7      <xs:element name="Settings" minOccurs="1" maxOccurs="1">
8       <xs:complexType>
9        <xs:sequence>
10        <xs:element name="Map_height" type="xs:unsignedInt"
              minOccurs="1" maxOccurs="1"/>
11        <xs:element name="Map_width" type="xs:unsignedInt"
              minOccurs="1" maxOccurs="1"/>
12        <xs:element name="Region_height" type="xs:unsignedInt"
              minOccurs="1" maxOccurs="1"/>
13        <xs:element name="Region_width" type="xs:unsignedInt"
              minOccurs="1" maxOccurs="1"/>
14       </xs:sequence>
15      </xs:complexType>
16     </xs:element>
17     <xs:element name="Streets" minOccurs="0" maxOccurs="1">
18      <xs:complexType>
19       <xs:sequence>
20        <xs:element name="Street" minOccurs="0" maxOccurs="unbounded">
21         <xs:complexType>
22          <xs:sequence>
23           <xs:element name="Name" minOccurs="1" maxOccurs="1"
                type="xs:string"/>
24           <xs:element name="StartNode" minOccurs="1" maxOccurs="1">
```

```
25            <xs:complexType>
26             <xs:sequence>
27              <xs:element name="x" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
28              <xs:element name="y" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
29              <xs:element name="trafficSignal" minOccurs="1"
                    maxOccurs="1" type="xs:boolean"/>
30             </xs:sequence>
31            </xs:complexType>
32           </xs:element>
33           <xs:element name="EndNode" minOccurs="1" maxOccurs="1">
34            <xs:complexType>
35             <xs:sequence>
36              <xs:element name="x" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
37              <xs:element name="y" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
38              <xs:element name="trafficSignal" minOccurs="1"
                    maxOccurs="1" type="xs:boolean"/>
39             </xs:sequence>
40            </xs:complexType>
41           </xs:element>
42           <xs:element name="Oneway" minOccurs="1" maxOccurs="1"
                type="xs:boolean"/>
43           <xs:element name="SteetType" minOccurs="1" maxOccurs="1"
                type="xs:string"/>
44           <xs:element name="Lanes" minOccurs="1" maxOccurs="1">
45            <xs:simpleType>
46             <xs:restriction base="xs:unsignedInt">
47              <xs:minInclusive value="1"/>
48             </xs:restriction>
49            </xs:simpleType>
50           </xs:element>
51           <xs:element name="Speed" minOccurs="1" maxOccurs="1"
                type="xs:unsignedInt"/>
52           <xs:element name="Color" minOccurs="1" maxOccurs="1">
53            <xs:simpleType>
54             <xs:restriction base="xs:int">
55              <xs:maxInclusive value="-1"/>
56             </xs:restriction>
57            </xs:simpleType>
58           </xs:element>
59          </xs:sequence>
60         </xs:complexType>
61        </xs:element>
62       </xs:sequence>
63      </xs:complexType>
64     </xs:element>
```

```
65      </xs:sequence>
66    </xs:complexType>
67  </xs:element>
68 </xs:schema>
```

map.xsd

B.2. XML-Schema des VANETsim-Szenarios

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <xs:schema attributeFormDefault="unqualified"
       elementFormDefault="unqualified"
       xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="Scenario">
4    <xs:complexType>
5     <xs:sequence>
6      <xs:element name="Settings" minOccurs="1" maxOccurs="1">
7       <xs:complexType>
8        <xs:sequence>
9         <xs:element name="CommunicationEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
10        <xs:element name="BeaconsEnabled" minOccurs="1" maxOccurs="1"
              type="xs:boolean"/>
11        <xs:element name="GlobalInfrastructureEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
12        <xs:element name="CommunicationInterval" minOccurs="1"
              maxOccurs="1" type="xs:unsignedInt"/>
13        <xs:element name="BeaconsInterval" minOccurs="1" maxOccurs="1"
              type="xs:unsignedInt"/>
14        <xs:element name="MixZonesEnabled" minOccurs="1" maxOccurs="1"
              type="xs:boolean"/>
15        <xs:element name="MixZoneRadius" minOccurs="1" maxOccurs="1"
              type="xs:unsignedInt"/>
16        <xs:element name="RoutingMode" minOccurs="1" maxOccurs="1">
17         <xs:simpleType>
18          <xs:restriction base="xs:unsignedInt">
19           <xs:minInclusive value="0"/>
20           <xs:maxInclusive value="1"/>
21          </xs:restriction>
22         </xs:simpleType>
23        </xs:element>
24        <xs:element name="VehicleRecyclingEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
25        <xs:element name="FallBackInMixZonesEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
26        <xs:element name="FallBackInMixZonesFloodingOnly" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
27        <xs:element name="MinTravelTimeForRecycling" minOccurs="1"
              maxOccurs="1" type="xs:unsignedInt"/>
28        <xs:element name="ARSULoggingEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
29        <xs:element name="ARSULog" minOccurs="1" maxOccurs="1"
              type="xs:string"/>
30        <xs:element name="ARSUEncryptedLoggingEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
```

```
31        <xs:element name="privacyLoggingEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
32        <xs:element name="SilentPeriodsEnabled" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
33        <xs:element name="SilentPeriodDuration" minOccurs="1"
              maxOccurs="1" type="xs:unsignedInt"/>
34        <xs:element name="SilentPeriodFrequency" minOccurs="1"
              maxOccurs="1" type="xs:unsignedInt"/>
35        <xs:element name="SlowEnabled" minOccurs="1" maxOccurs="1"
              type="xs:boolean"/>
36        <xs:element name="SlowTimeToChangePseudonym" minOccurs="1"
              maxOccurs="1" type="xs:unsignedInt"/>
37        <xs:element name="SlowSpeedLimit" minOccurs="1" maxOccurs="1"
              type="xs:unsignedInt"/>
38        <xs:element name="showEncryptedCommInMix" minOccurs="1"
              maxOccurs="1" type="xs:boolean"/>
39      </xs:sequence>
40    </xs:complexType>
41  </xs:element>
42  <xs:element name="Vehicles" minOccurs="1" maxOccurs="1">
43    <xs:complexType>
44      <xs:sequence>
45        <xs:element name="Vehicle" minOccurs="0" maxOccurs="unbounded">
46          <xs:complexType>
47            <xs:sequence>
48              <xs:element name="VehicleLength" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
49              <xs:element name="MaxSpeed" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
50              <xs:element name="MaxCommDist" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
51              <xs:element name="Wifi" minOccurs="1" maxOccurs="1"
                    type="xs:boolean"/>
52              <xs:element name="emergencyVehicle" minOccurs="1"
                    maxOccurs="1" type="xs:boolean"/>
53              <xs:element name="braking_rate" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
54              <xs:element name="acceleration_rate" minOccurs="1"
                    maxOccurs="1" type="xs:unsignedInt"/>
55              <xs:element name="timeDistance" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
56              <xs:element name="politeness" minOccurs="1" maxOccurs="1"
                    type="xs:unsignedInt"/>
57              <xs:element name="Color" minOccurs="1" maxOccurs="1"
                    type="xs:negativeInteger"/>
58              <xs:element name="isAttacker" minOccurs="1" maxOccurs="1"
                    type="xs:boolean"/>
59              <xs:element name="isAttacked" minOccurs="1" maxOccurs="1"
                    type="xs:boolean"/>
```

```
60          <xs:element name="Destinations" minOccurs="1" maxOccurs="1">
61           <xs:complexType>
62            <xs:sequence>
63             <xs:element name="WayPoint" minOccurs="2"
                   maxOccurs="unbounded">
64              <xs:complexType>
65               <xs:sequence>
66                <xs:element name="x" minOccurs="1" maxOccurs="1"
                      type="xs:unsignedInt"/>
67                <xs:element name="y" minOccurs="1" maxOccurs="1"
                      type="xs:unsignedInt"/>
68                <xs:element name="wait" minOccurs="1" maxOccurs="1"
                      type="xs:unsignedInt"/>
69               </xs:sequence>
70              </xs:complexType>
71             </xs:element>
72            </xs:sequence>
73           </xs:complexType>
74          </xs:element>
75         </xs:sequence>
76        </xs:complexType>
77       </xs:element>
78      </xs:sequence>
79     </xs:complexType>
80    </xs:element>
81    <xs:element name="MixZones" minOccurs="1" maxOccurs="1">
82     <xs:complexType>
83      <xs:sequence>
84       <xs:element name="MixNode" minOccurs="0" maxOccurs="unbounded">
85        <xs:complexType>
86         <xs:sequence>
87          <xs:element name="x" minOccurs="1" maxOccurs="1"
                type="xs:unsignedInt"/>
88          <xs:element name="y" minOccurs="1" maxOccurs="1"
                type="xs:unsignedInt"/>
89          <xs:element name="radius" minOccurs="1" maxOccurs="1"
                type="xs:unsignedInt"/>
90          <xs:element name="hasRSU" minOccurs="1" maxOccurs="1"
                type="xs:boolean"/>
91         </xs:sequence>
92        </xs:complexType>
93       </xs:element>
94      </xs:sequence>
95     </xs:complexType>
96    </xs:element>
97    <xs:element name="RSUs" minOccurs="1" maxOccurs="1">
98     <xs:complexType>
99      <xs:sequence>
100      <xs:element name="RSU" minOccurs="0" maxOccurs="unbounded">
```

```
101          <xs:complexType>
102           <xs:sequence>
103            <xs:element name="x" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
104            <xs:element name="y" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
105            <xs:element name="radius" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
106            <xs:element name="isEncrypted" minOccurs="1" maxOccurs="1"
                 type="xs:boolean"/>
107           </xs:sequence>
108          </xs:complexType>
109         </xs:element>
110        </xs:sequence>
111       </xs:complexType>
112      </xs:element>
113      <xs:element name="ARSUs" minOccurs="1" maxOccurs="1">
114       <xs:complexType>
115        <xs:sequence>
116         <xs:element name="RSU" minOccurs="0" maxOccurs="unbounded">
117          <xs:complexType>
118           <xs:sequence>
119            <xs:element name="x" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
120            <xs:element name="y" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
121            <xs:element name="radius" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
122           </xs:sequence>
123          </xs:complexType>
124         </xs:element>
125        </xs:sequence>
126       </xs:complexType>
127      </xs:element>
128      <xs:element name="Events" minOccurs="0" maxOccurs="1">
129       <xs:complexType>
130        <xs:sequence>
131         <xs:element name="Event" minOccurs="0" maxOccurs="unbounded">
132          <xs:complexType>
133           <xs:sequence>
134            <xs:element name="Time" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
135            <xs:element name="Type" minOccurs="1" maxOccurs="1"
                 type="xs:string"/>
136            <xs:element name="x" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
137            <xs:element name="y" minOccurs="1" maxOccurs="1"
                 type="xs:unsignedInt"/>
138            <xs:element name="Direction" minOccurs="0" maxOccurs="1">
```

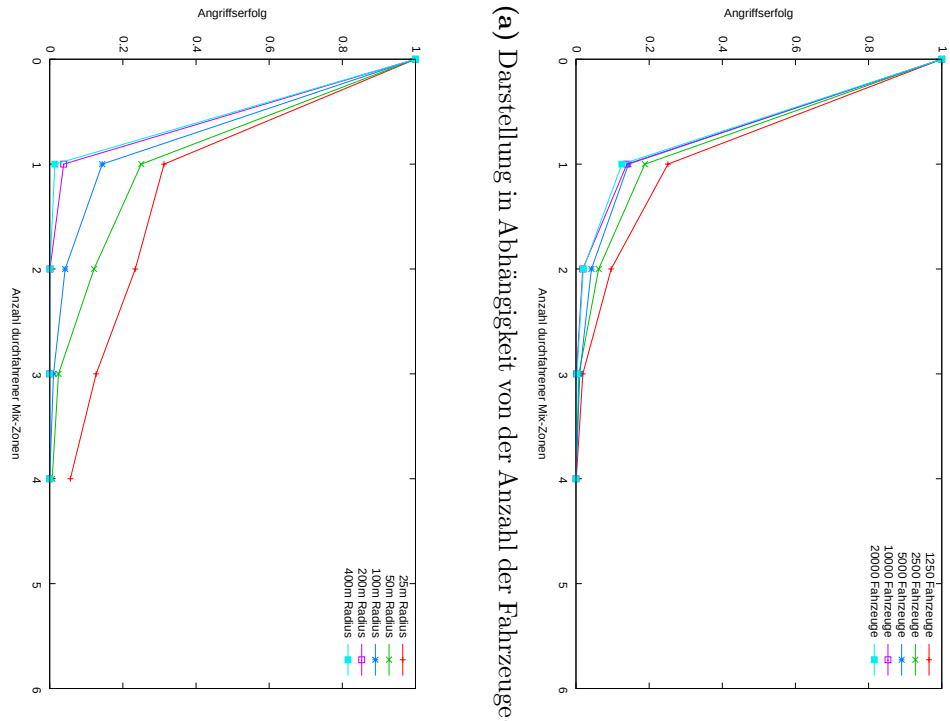
```
139             <xs:simpleType>
140              <xs:restriction base="xs:int">
141               <xs:minInclusive value="-1"/>
142               <xs:maxInclusive value="1"/>
143              </xs:restriction>
144             </xs:simpleType>
145            </xs:element>
146            <xs:element name="Lanes" minOccurs="0" maxOccurs="1"
                 type="xs:unsignedInt"/>
147           </xs:sequence>
148          </xs:complexType>
149         </xs:element>
150        </xs:sequence>
151       </xs:complexType>
152      </xs:element>
153     </xs:sequence>
154    </xs:complexType>
155   </xs:element>
156 </xs:schema>
```

scenario.xsd

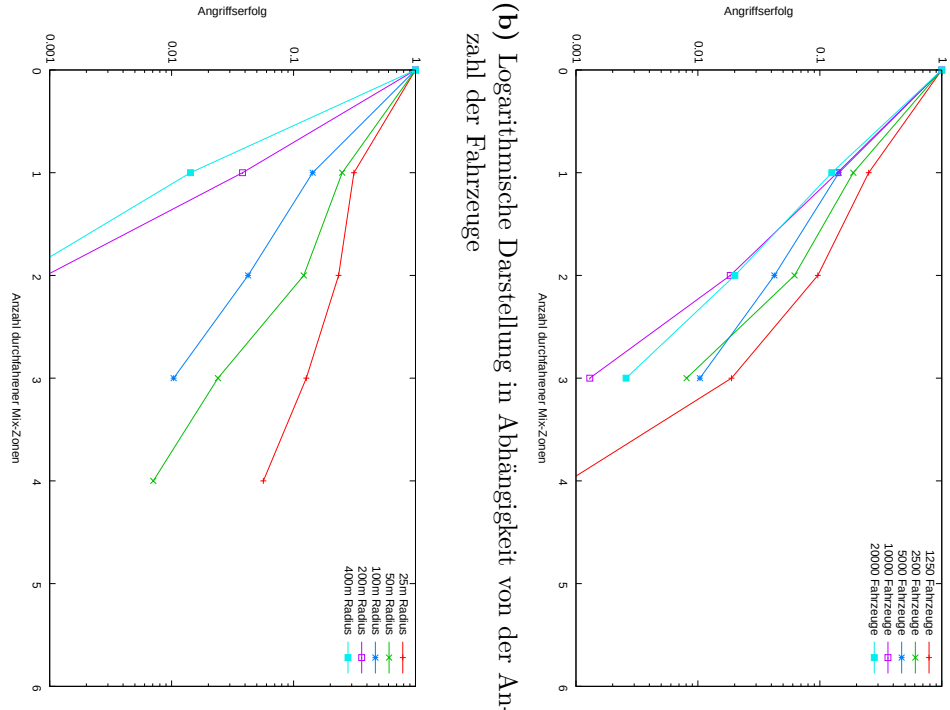
Anhang C

Simulationsergebnisse

C.1. Verfolgung über mehrere Mix-Zonen hinweg



(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

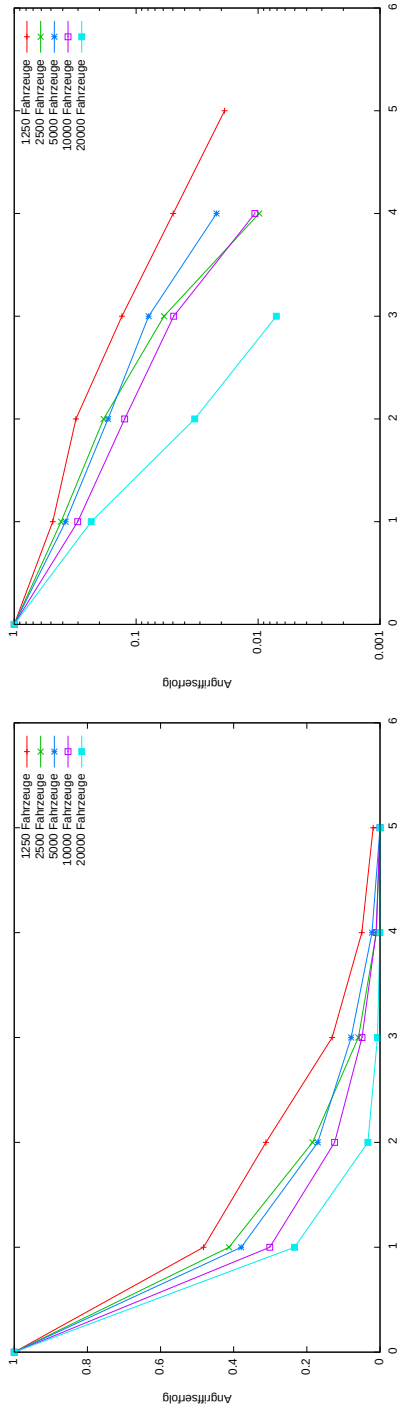


(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen

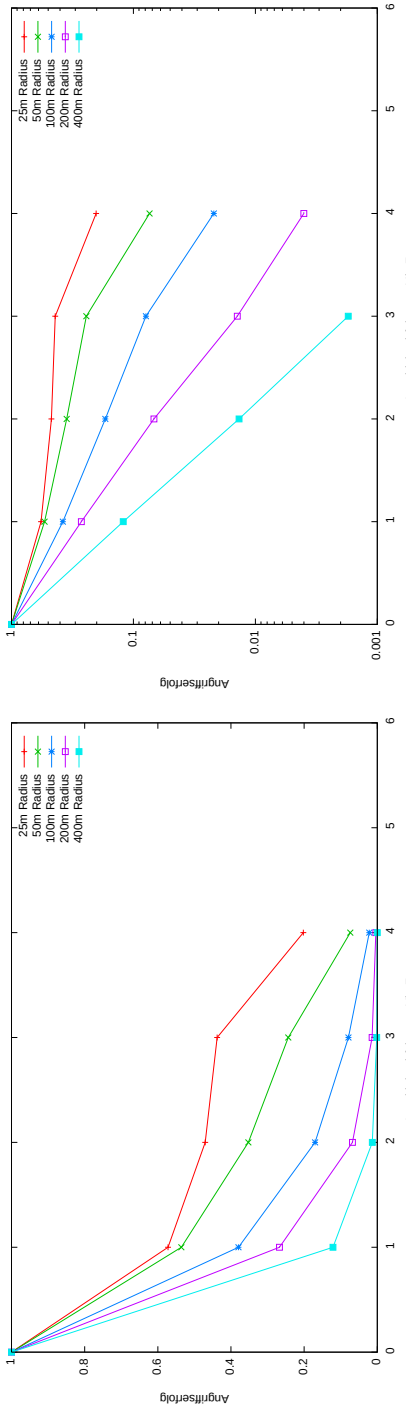
(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

Abbildung C.1.: Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in New York



(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

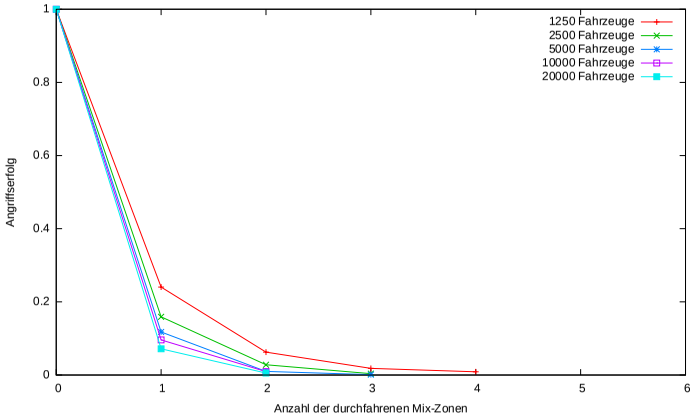
(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



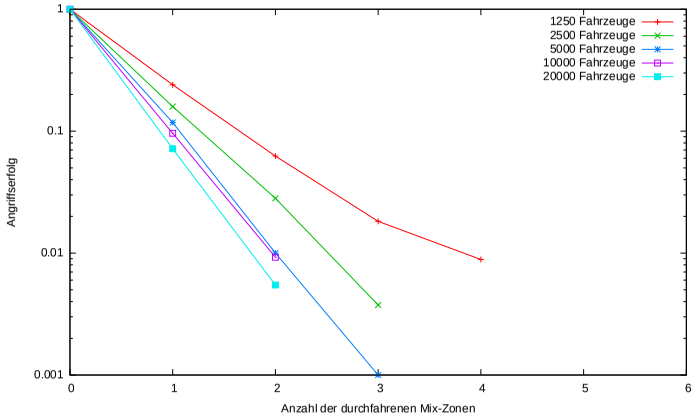
(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen

(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

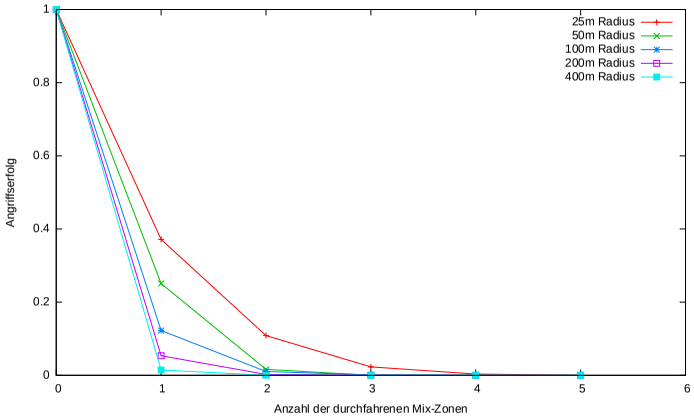
Abbildung C.2.: Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in New York



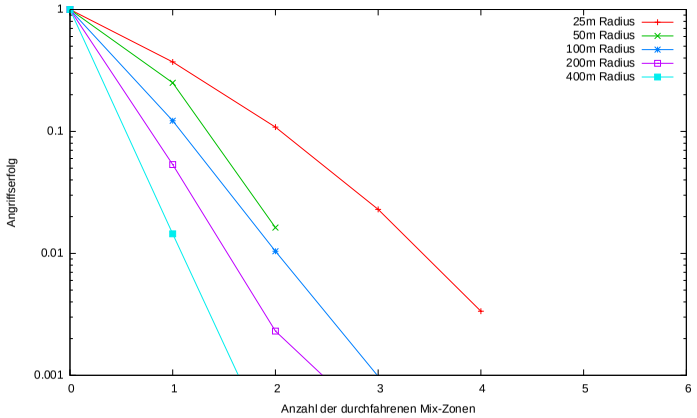
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

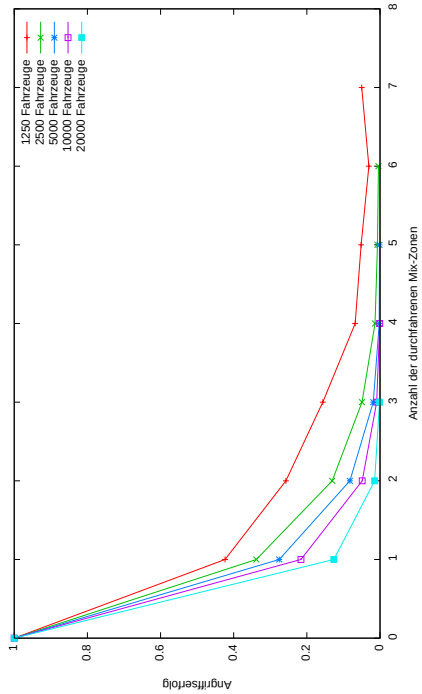


(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen

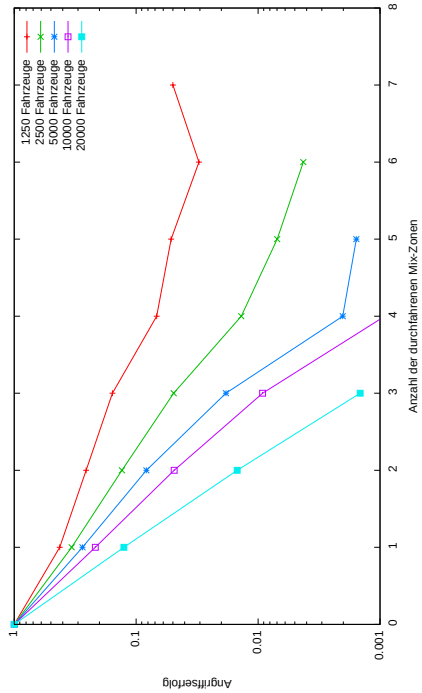


(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

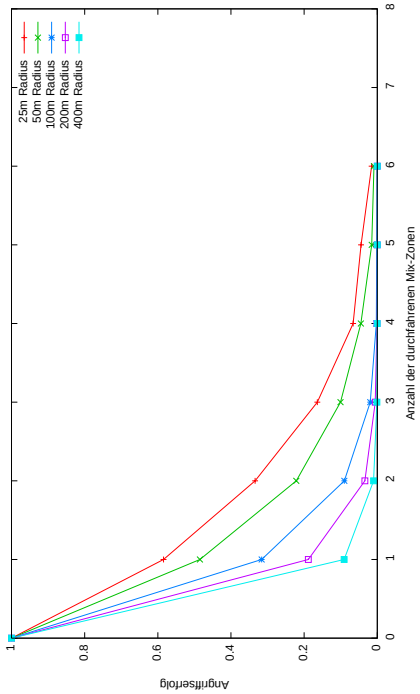
Abbildung C.3.: Angriffserfolge des einfachen Angriffs über mehrere Mix-Zonen hinweg in Puebla



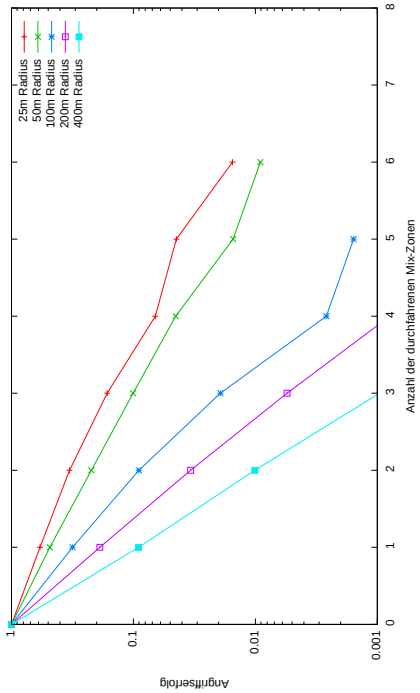
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



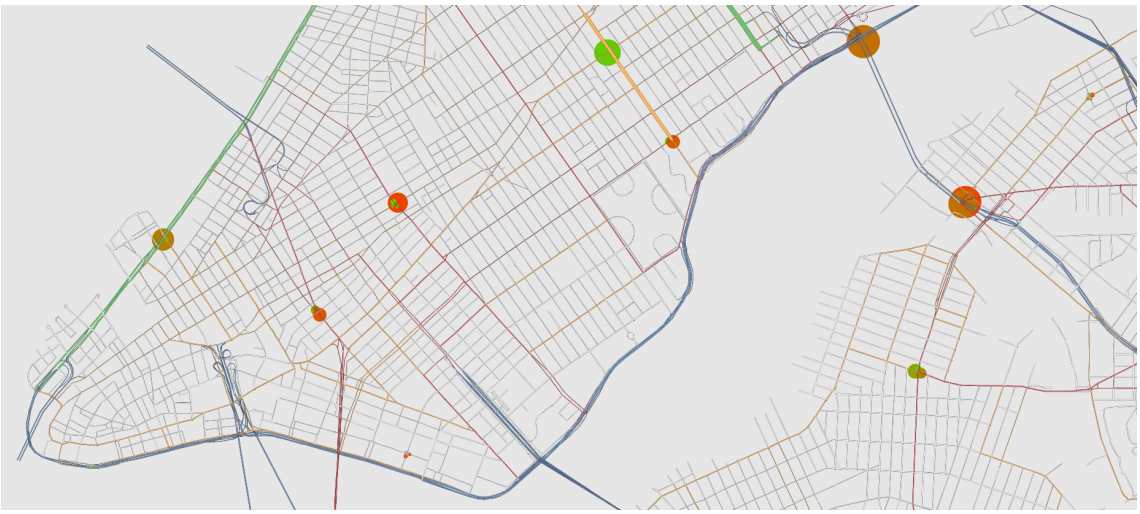
(c) Darstellung in Abhängigkeit von der Größe der Mix-Zonen



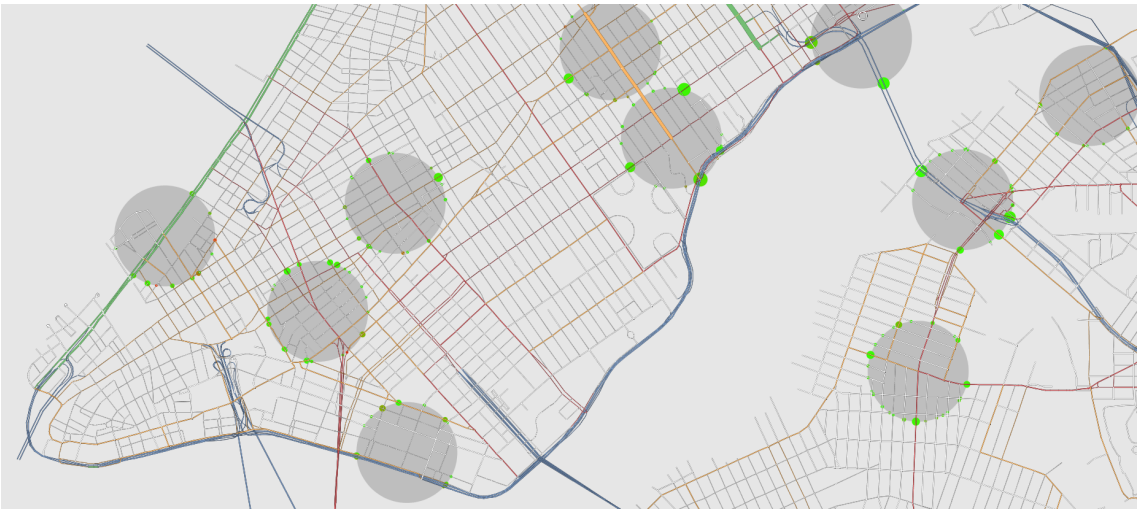
(d) Logarithmische Darstellung in Abhängigkeit von der Größe der Mix-Zonen

Abbildung C.4.: Angriffserfolge des komplexen Angriffs über mehrere Mix-Zonen hinweg in Puebla

C.2. Mix-Zonen bei unterschiedlichen Konfigurationen

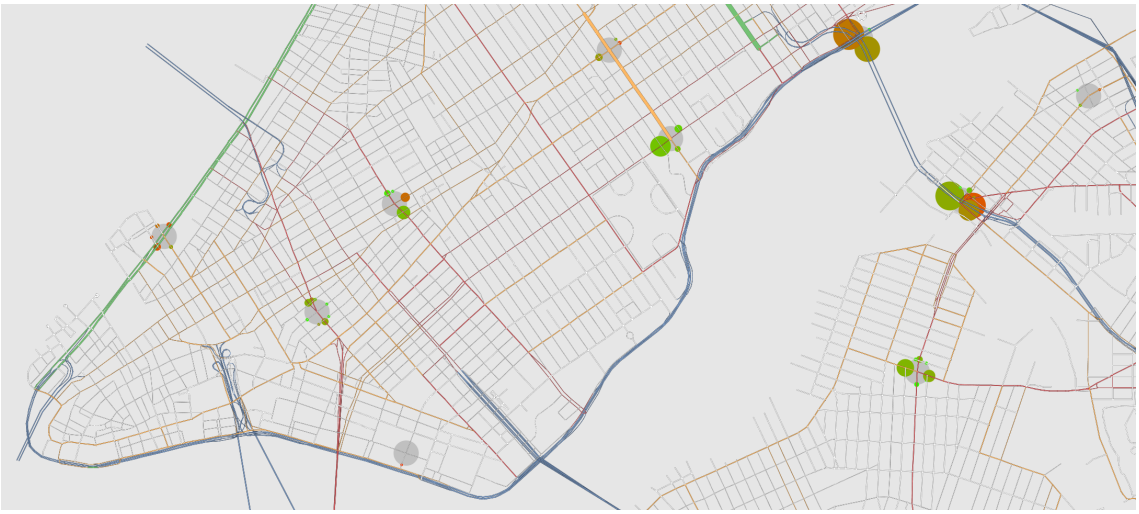


(a) Radius 25 m

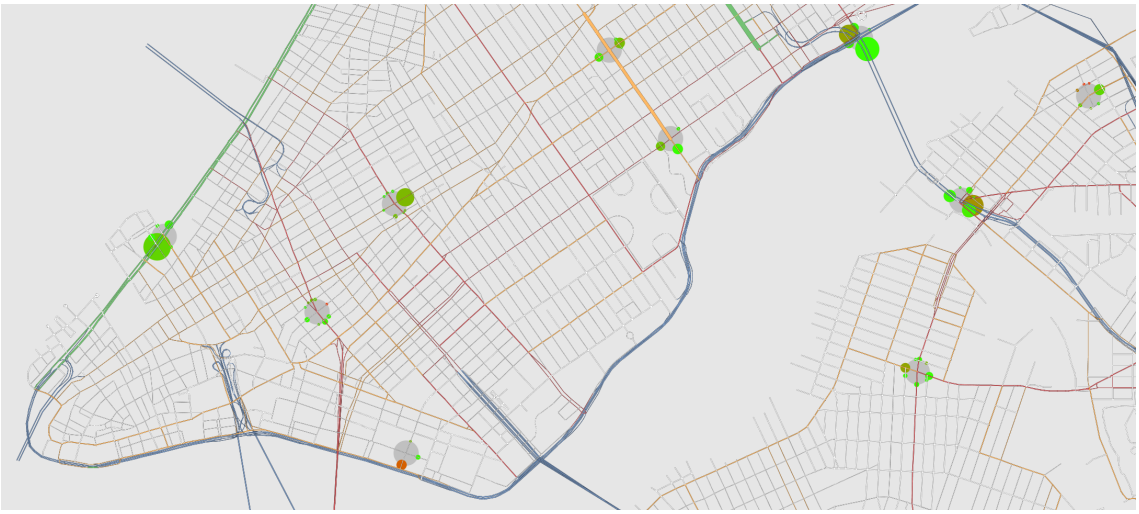


(b) Radius 400 m

Abbildung C.5.: Überblick über die Mix-Zonen in New York in unterschiedlichen Größen



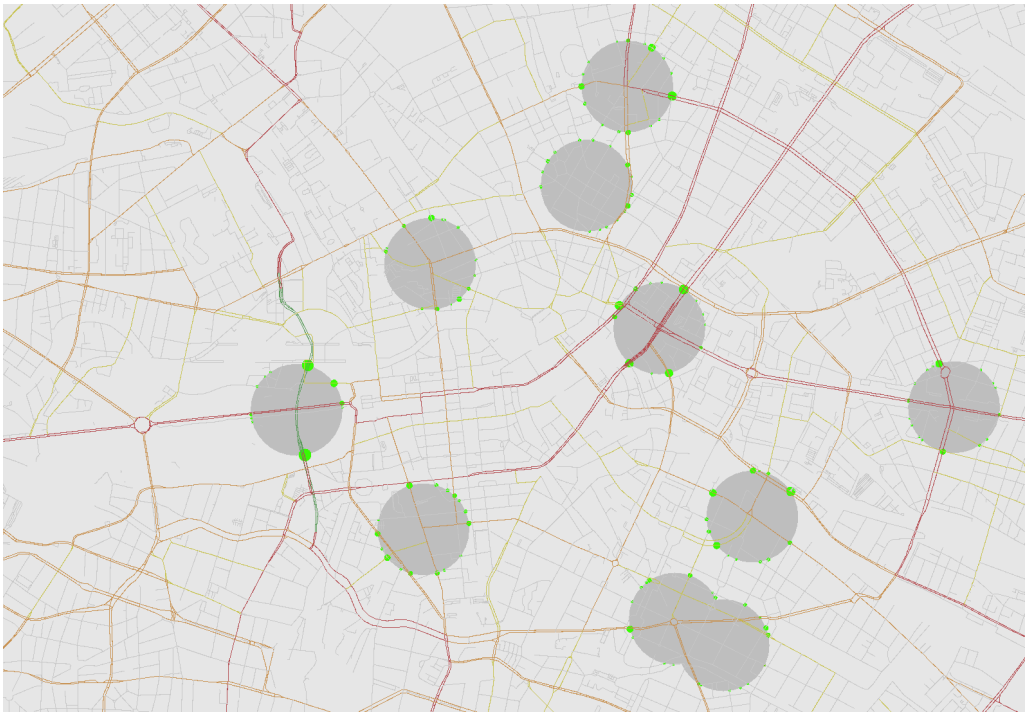
(a) 1250 Fahrzeuge in der Simulation



(b) 20 000 Fahrzeuge in der Simulation

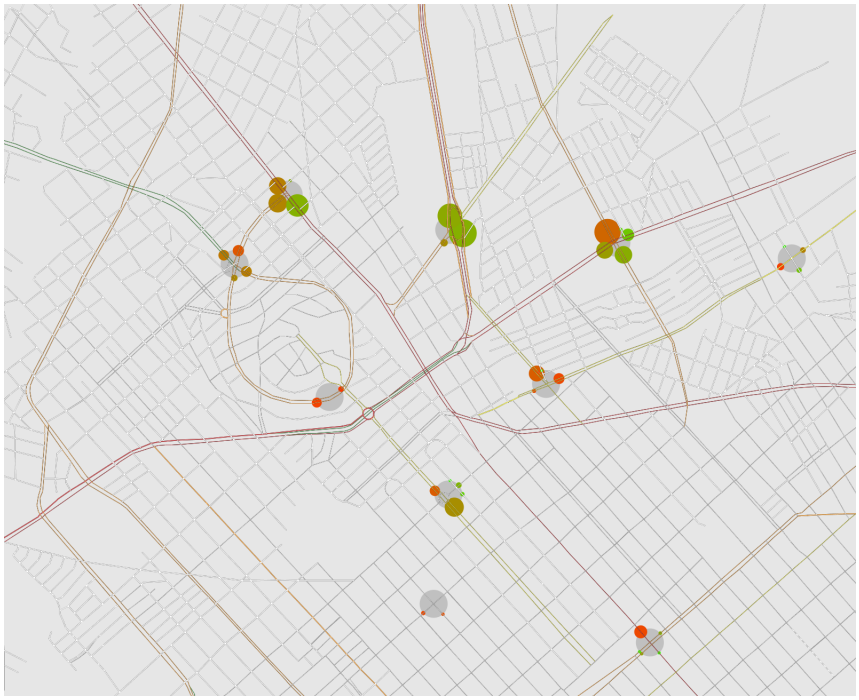
Abbildung C.6.: Überblick über die Mix-Zonen in New York bei unterschiedlichen Verkehrsdichten

(a) Radius 25 m

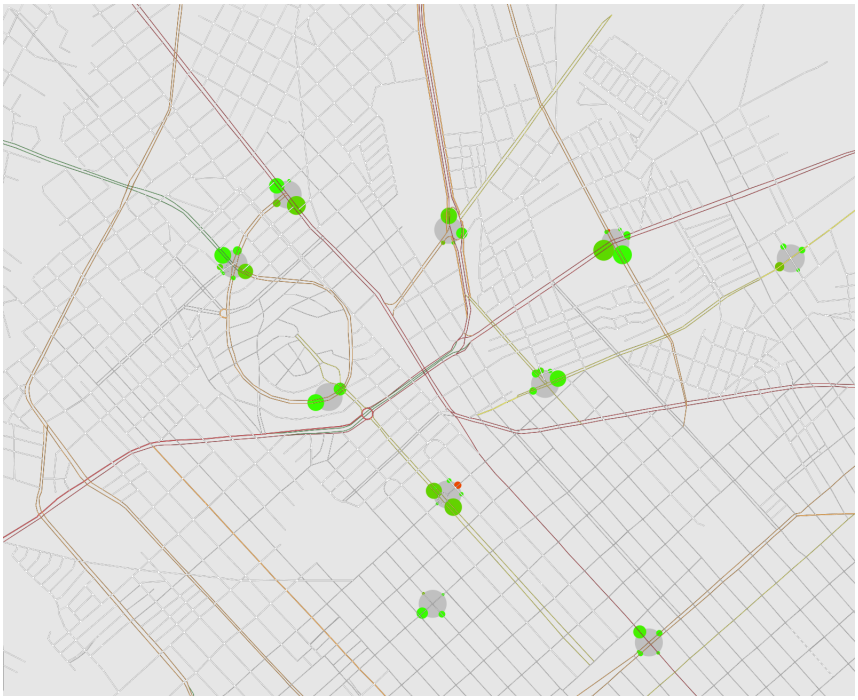


(b) Radius 400 m

Abbildung C.7.: Überblick über die Mix-Zonen in Berlin in unterschiedlichen Größen



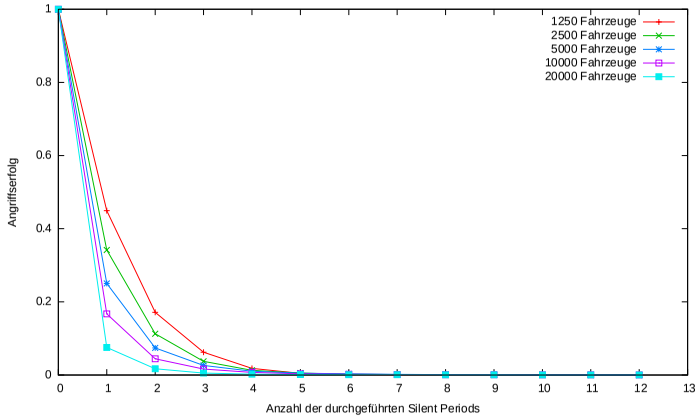
(a) 1250 Fahrzeuge in der Simulation



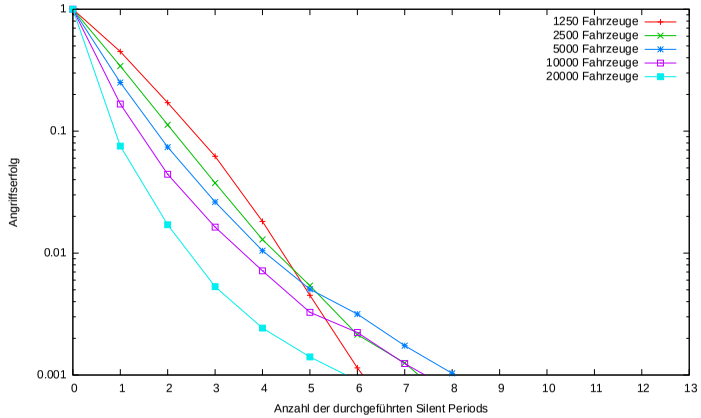
(b) 20 000 Fahrzeuge in der Simulation

Abbildung C.8.: Überblick über die Mix-Zonen in Puebla bei unterschiedlichen Verkehrsdichten

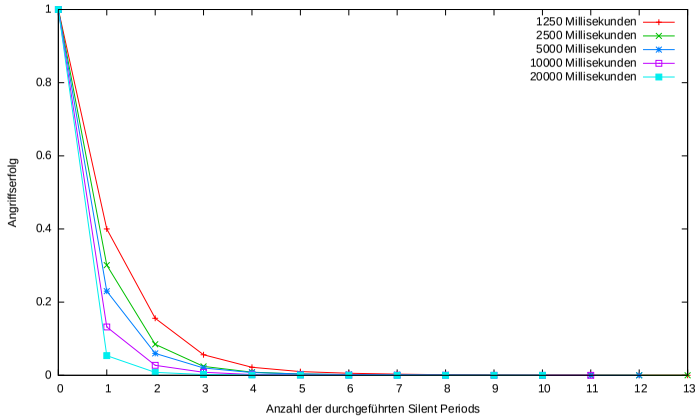
C.3. Verfolgung über mehrere Silent Periods hinweg



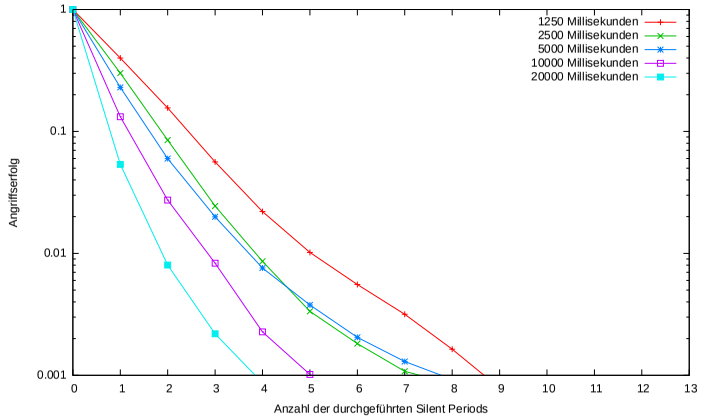
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

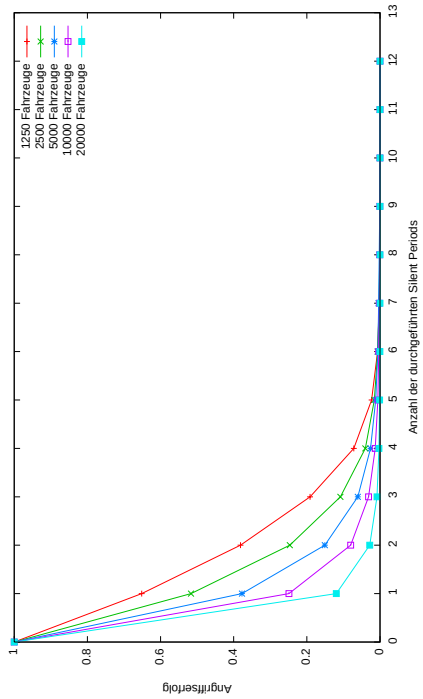


(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods

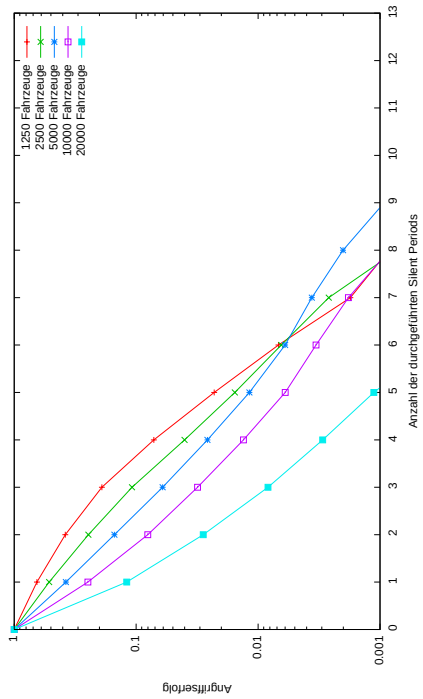


(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

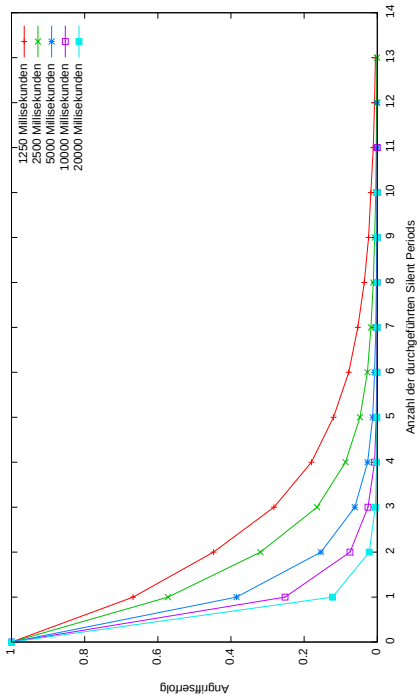
Abbildung C.9.: Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in New York



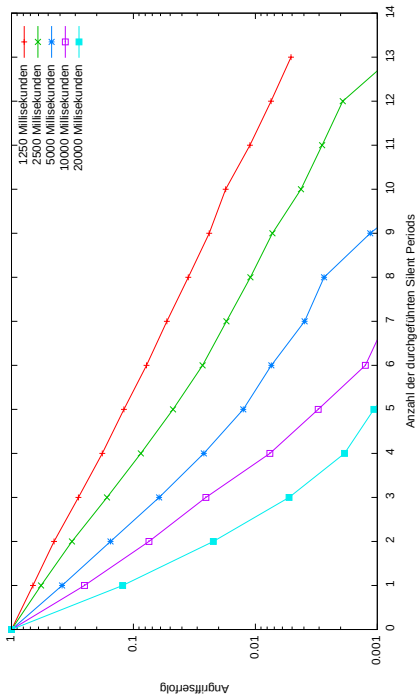
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

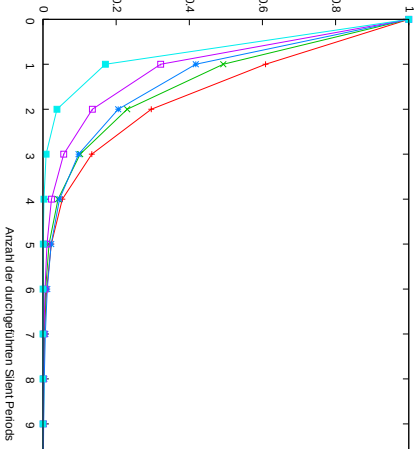


(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods

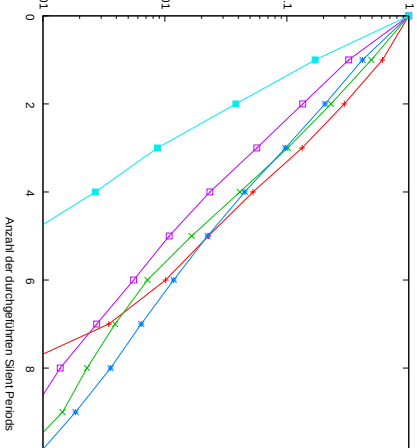


(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

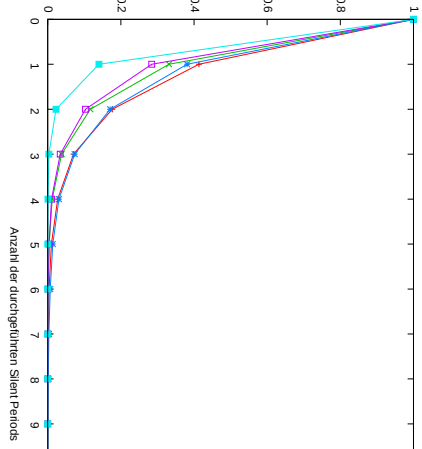
Abbildung C.10: Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in New York



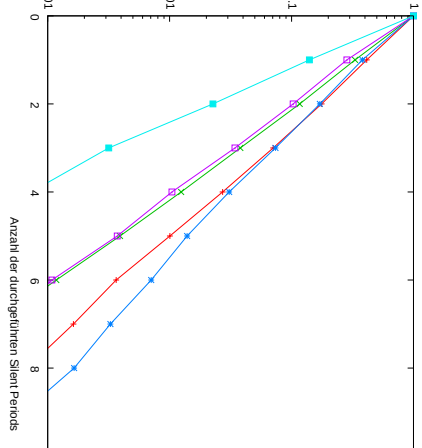
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

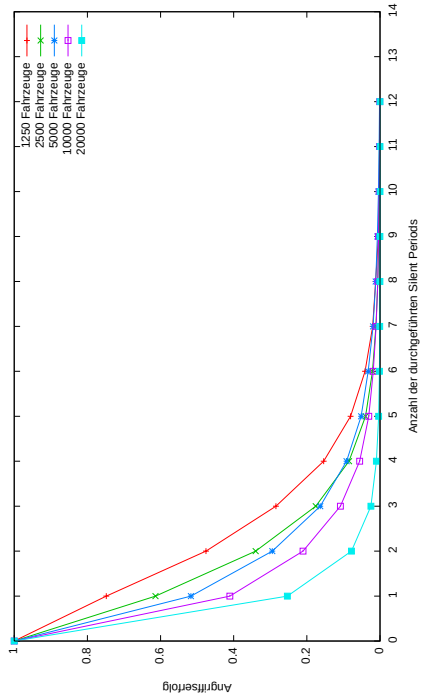


(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods

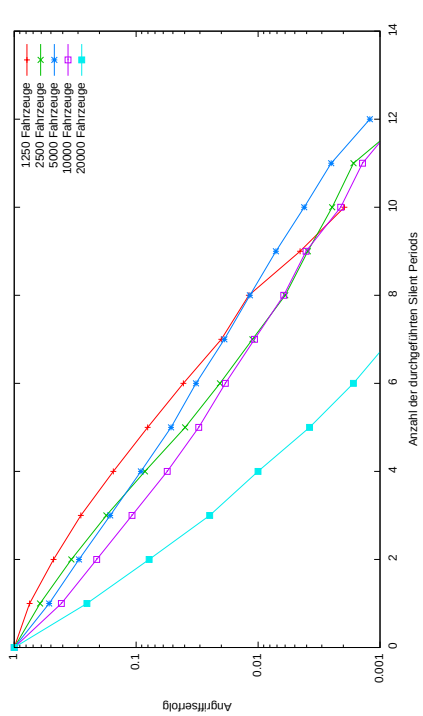


(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

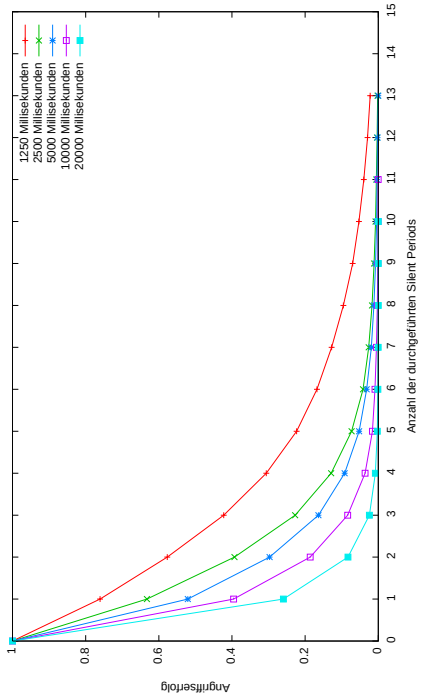
Abbildung C.11.: Angriffserfolge des einfachen Angriffs über mehrere Silent Periods hinweg in Puebla



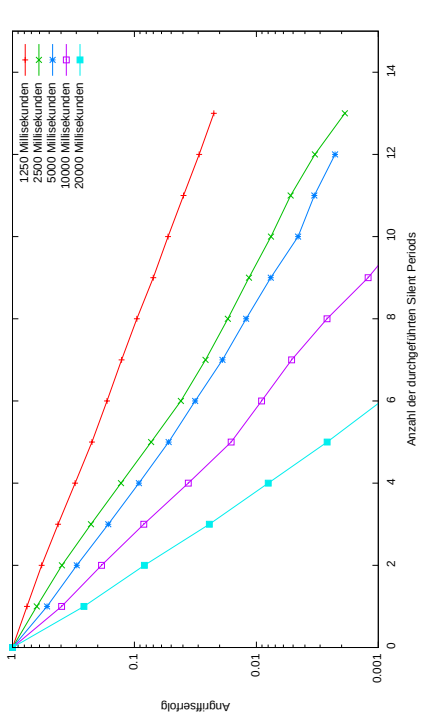
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



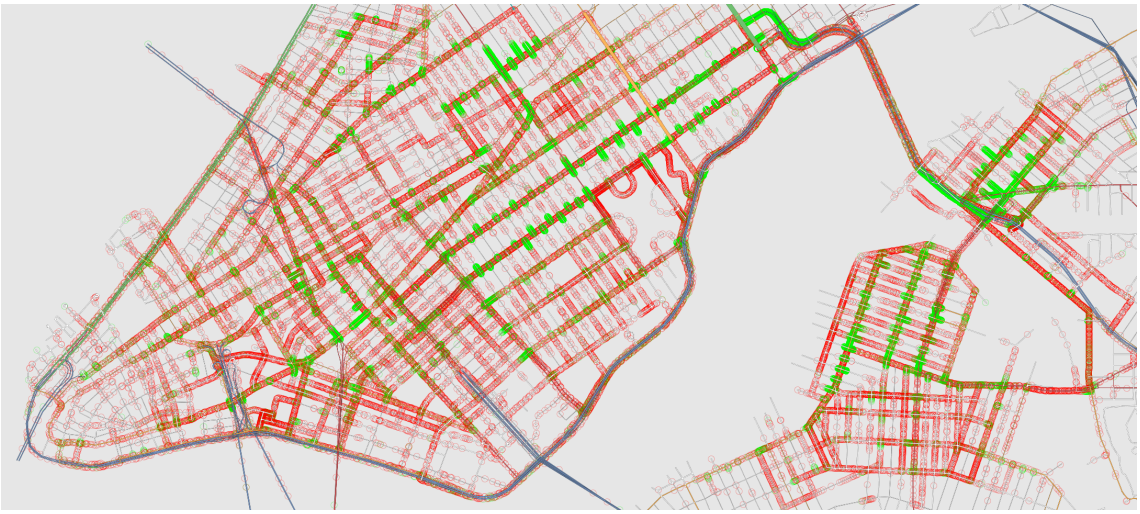
(c) Darstellung in Abhängigkeit von der Dauer der Silent Periods



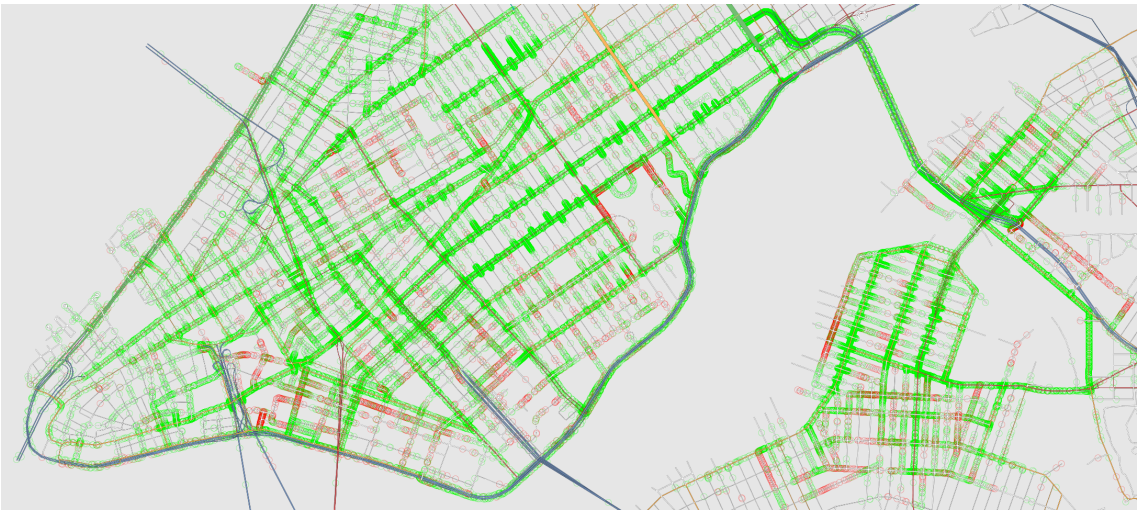
(d) Logarithmische Darstellung in Abhängigkeit von der Dauer der Silent Periods

Abbildung C.12: Angriffserfolge des komplexen Angriffs über mehrere Silent Periods hinweg in Puebla

C.4. Effektivität der Silent Periods an unterschiedlichen Orten

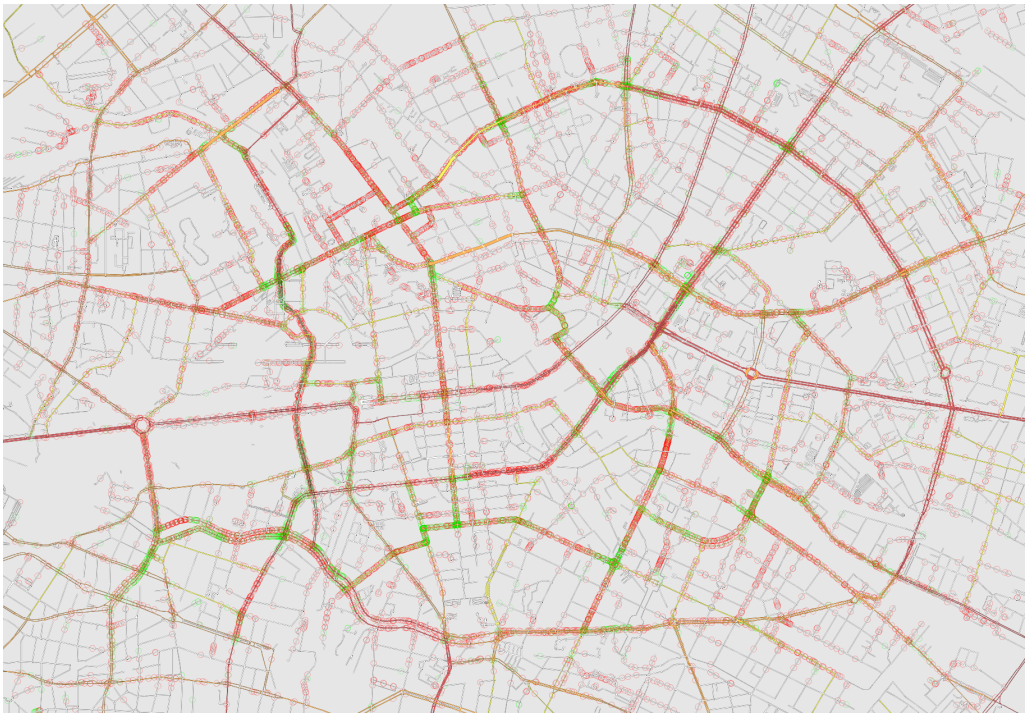


(a) 1250 ms



(b) 20 000 ms

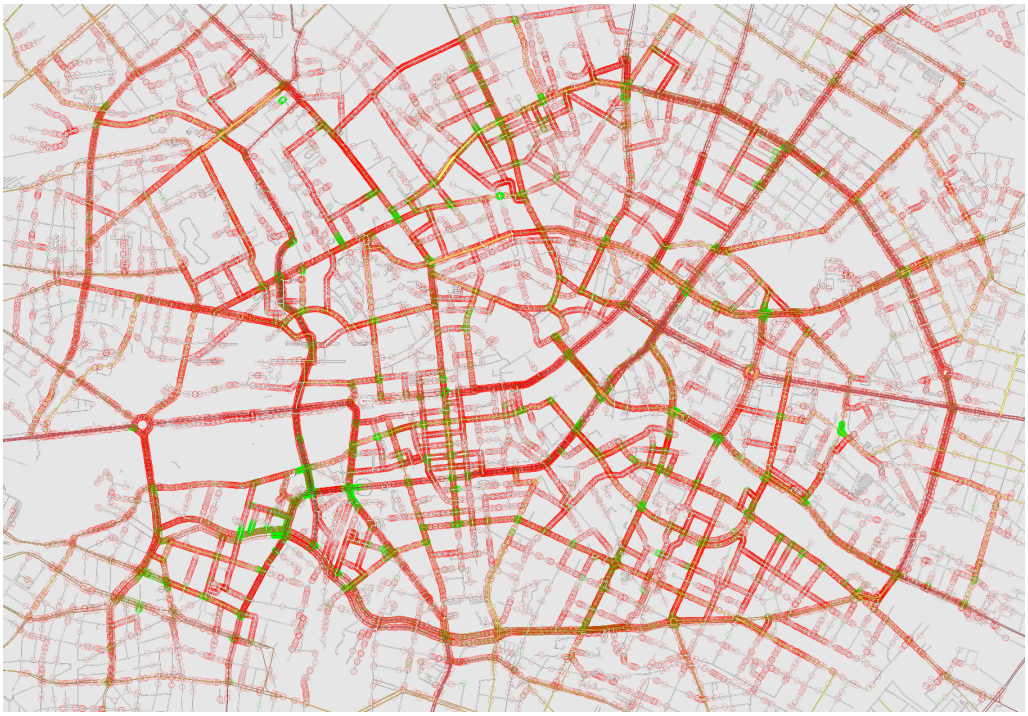
Abbildung C.13.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in New York bei unterschiedlichen Längen der Silent Periods



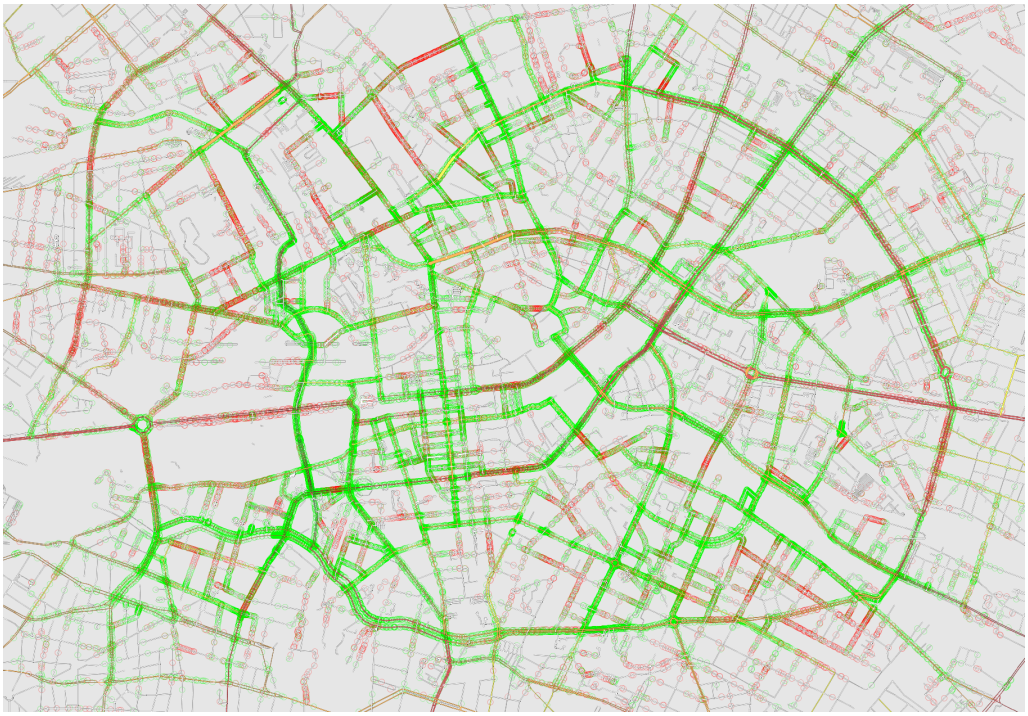
(a) 1250 Fahrzeuge

(b) 20 000 Fahrzeuge

Abbildung C.14.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in Berlin bei unterschiedlichen Verkehrsdichten

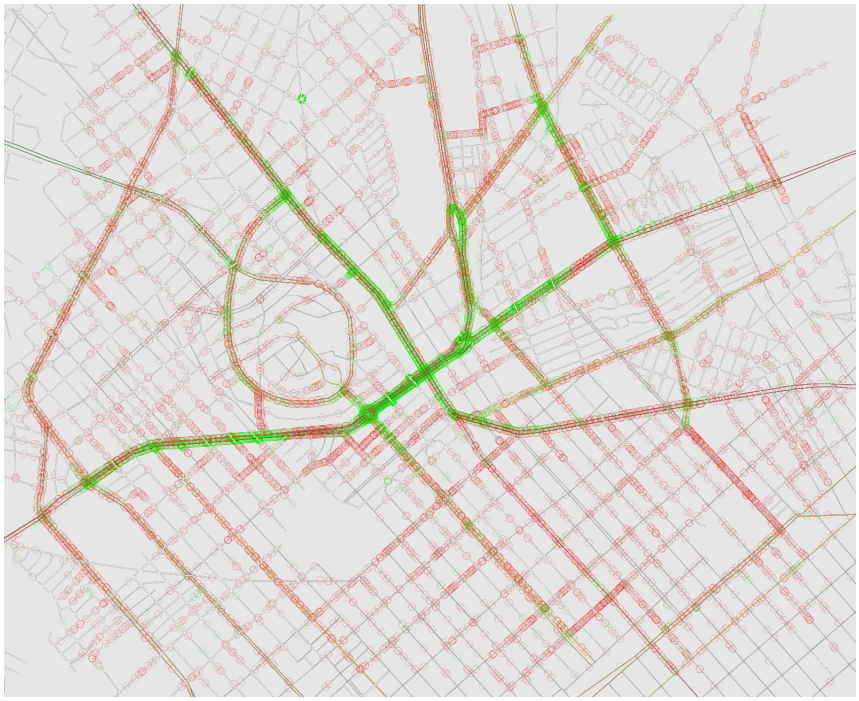


(a) 1250 ms

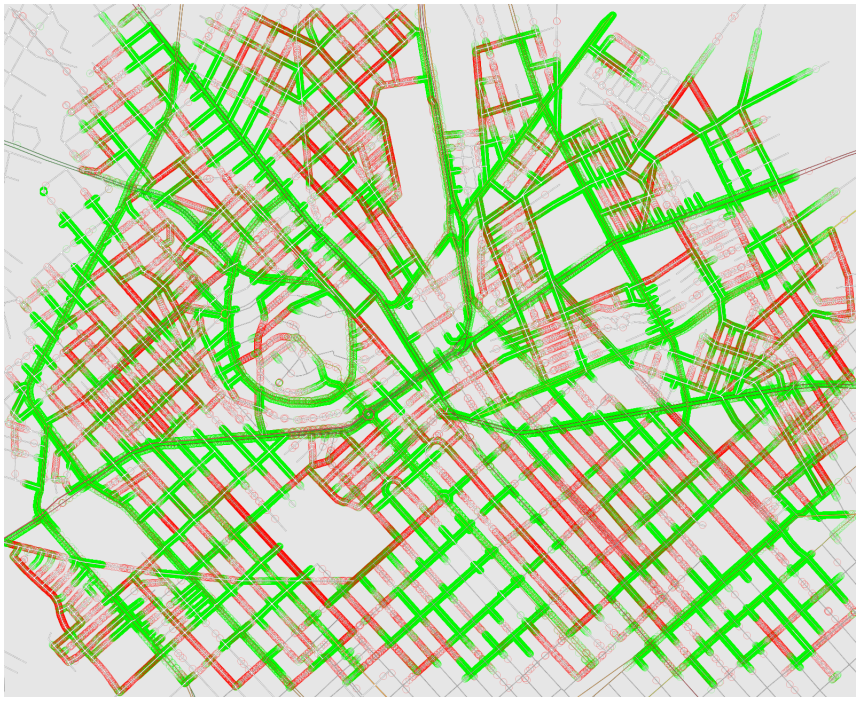


(b) 20 000 ms

Abbildung C.15.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Funkstille in Berlin bei unterschiedlichen Längen der Silent Periods



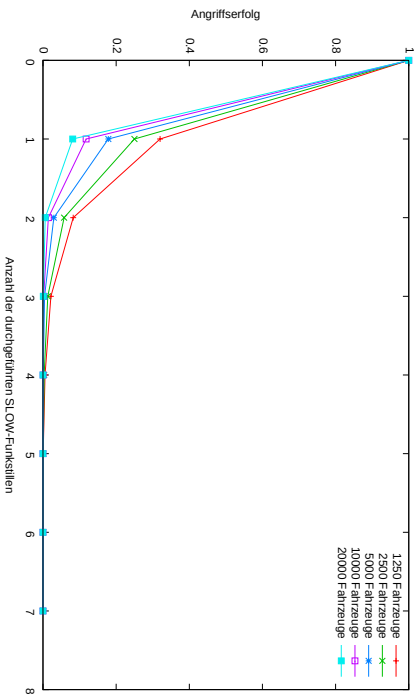
(a) 1250 Fahrzeuge



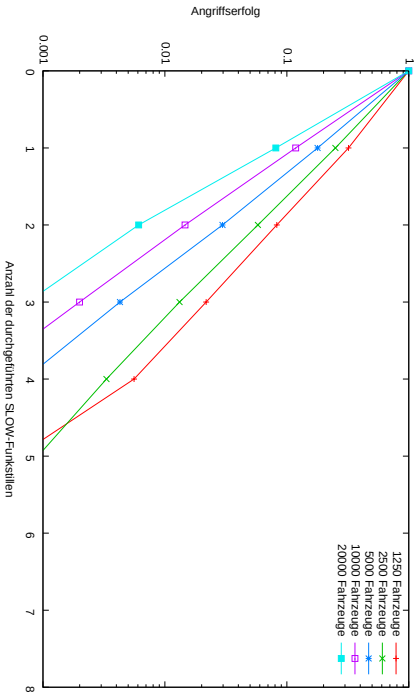
(b) 20 000 Fahrzeuge

Abbildung C.16.: Überblick über die Angriffserfolge auf Teilnehmer nach einer Silent Period in Puebla bei unterschiedlichen Verkehrsdichten

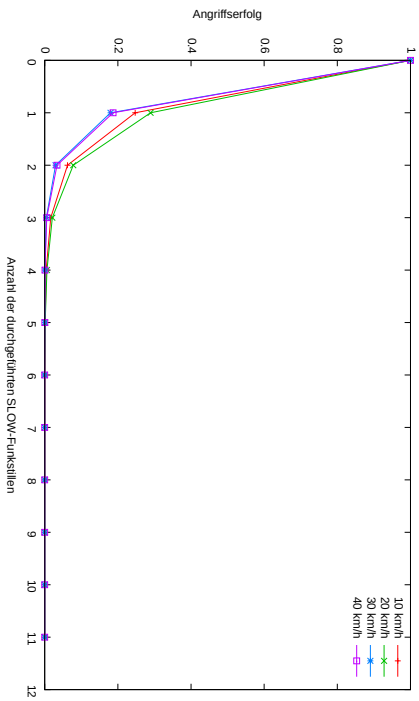
C.5. Verfolgung über mehrere SLOW-induzierte Funkstillen hinweg



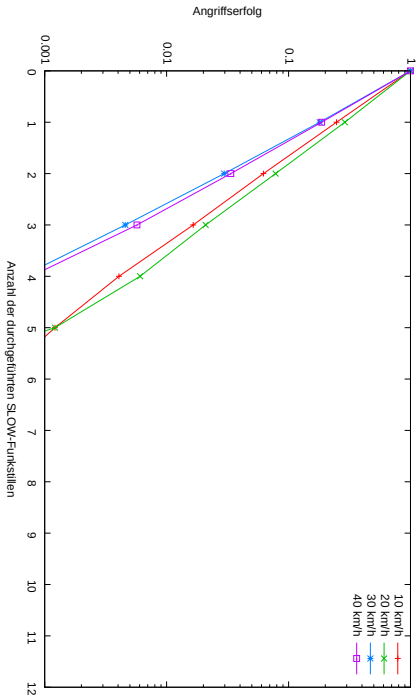
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge

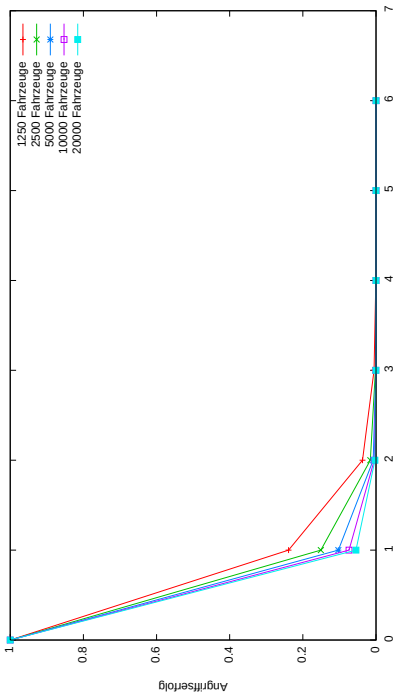


(c) Darstellung in Abhängigkeit von der Schwellengeschwindigkeit

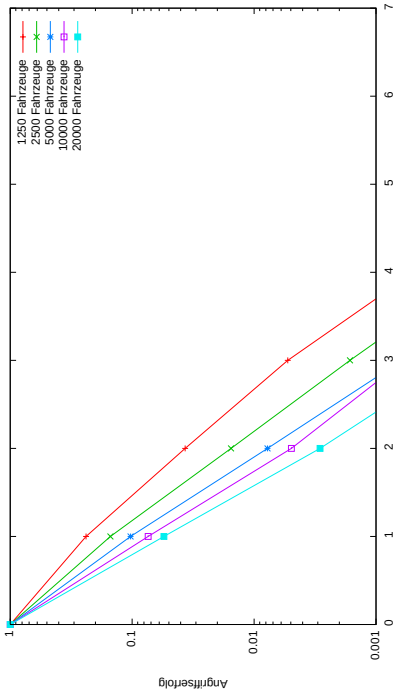


(d) Logarithmische Darstellung in Abhängigkeit von der Schwellengeschwindigkeit

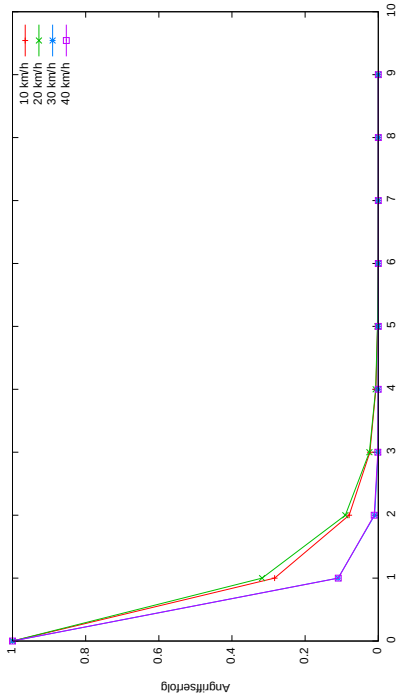
Abbildung C.17.: Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in New York



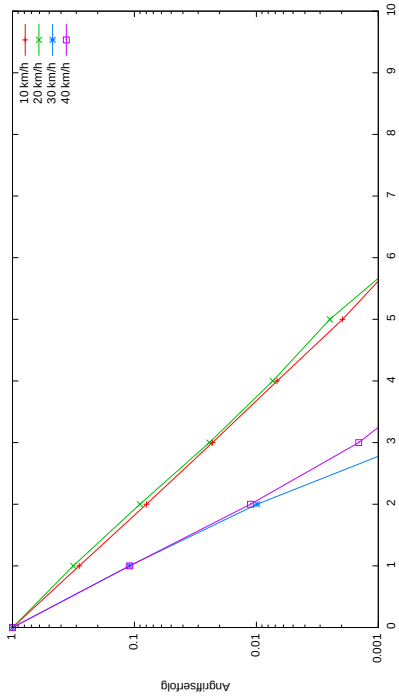
(a) Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



(b) Logarithmische Darstellung in Abhängigkeit von der Anzahl der Fahrzeuge



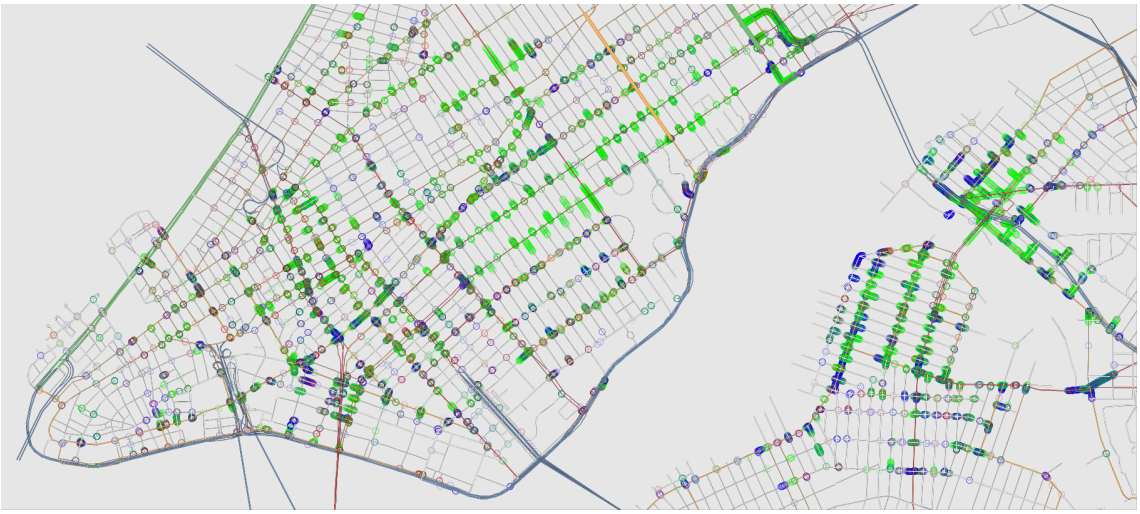
(c) Darstellung in Abhängigkeit von der Schwellengeschwindigkeit



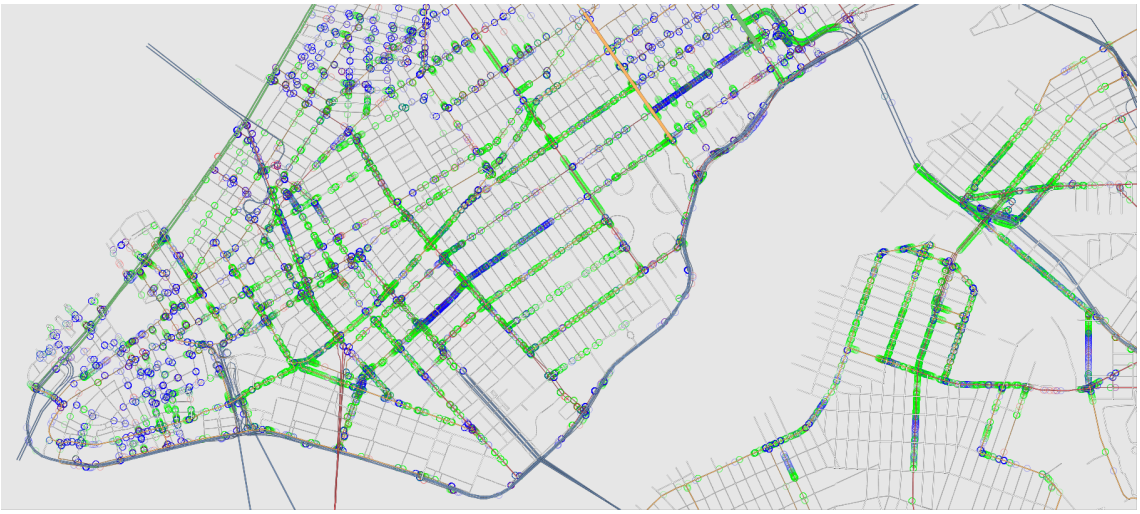
(d) Logarithmische Darstellung in Abhängigkeit von der Schwellengeschwindigkeit

Abbildung C.18: Angriffserfolge über mehrere Funkstillen bei SLOW hinweg in Puebla

C.6. Einfluss des Straßennetzes auf die Effektivität von SLOW

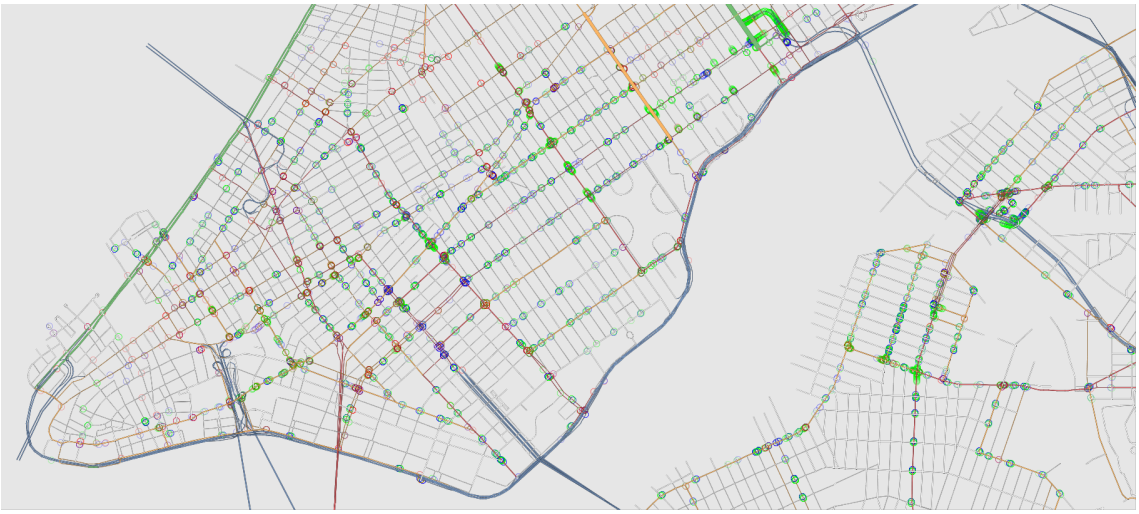


(a) 10 km/h

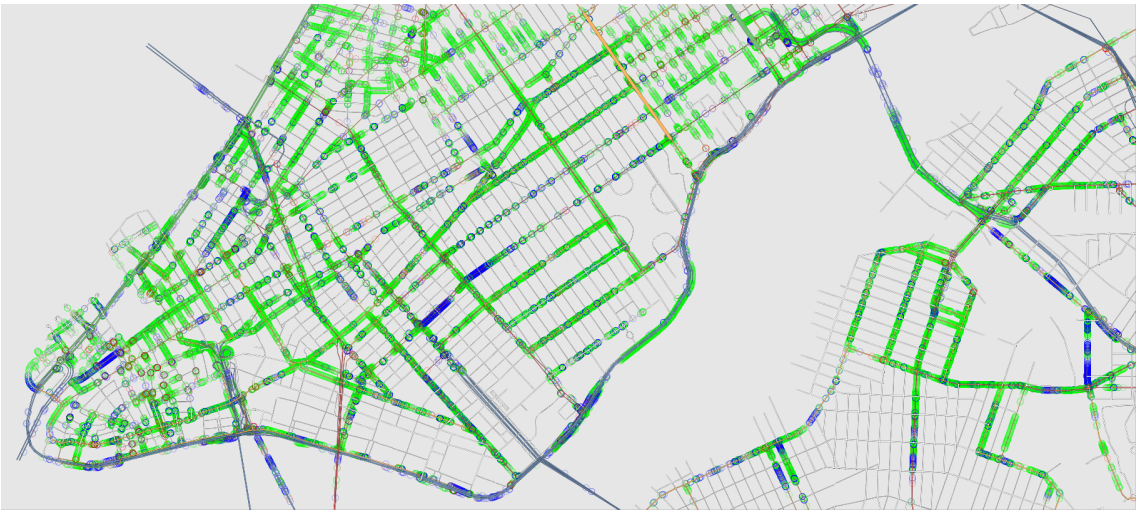


(b) 40 km/h

Abbildung C.19.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in New York bei unterschiedlichen Schwellengeschwindigkeiten

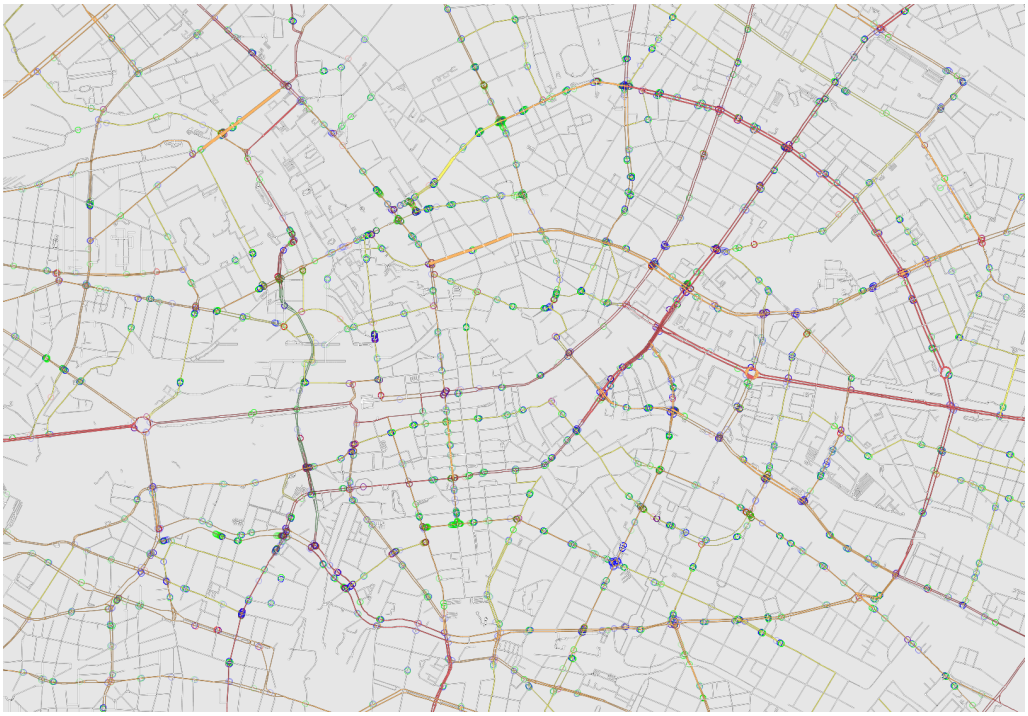


(a) 1250 Fahrzeuge



(b) 20 000 Fahrzeuge

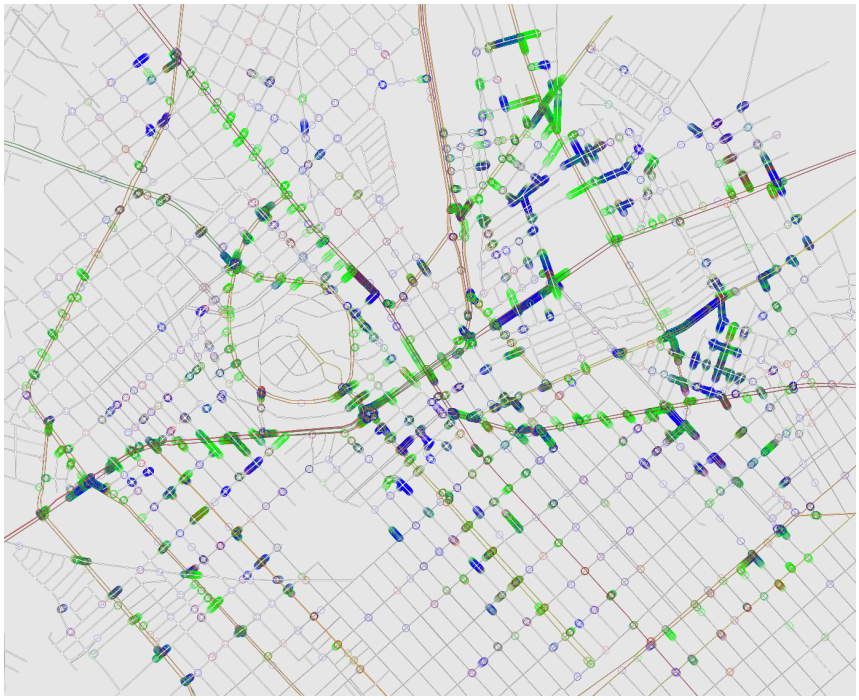
Abbildung C.20.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in New York bei unterschiedlichen Verkehrsdichten



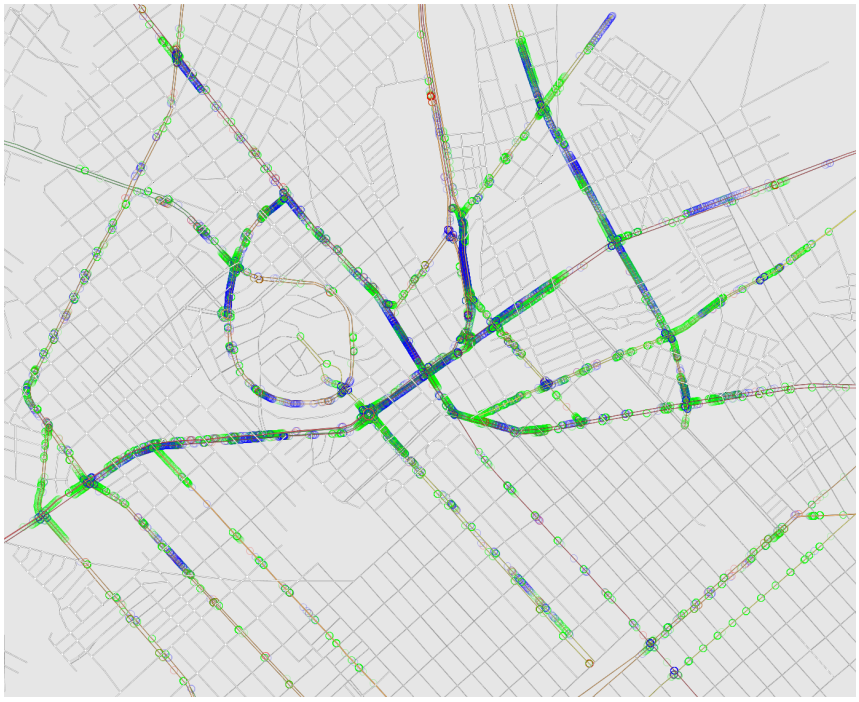
(a) 1250 Fahrzeuge

(b) 20 000 Fahrzeuge

Abbildung C.21.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Berlin bei unterschiedlichen Verkehrsdichten



(a) 10 km/h



(b) 40 km/h

Abbildung C.22.: Überblick über die Angriffserfolge auf Teilnehmer bei SLOW in Puebla bei unterschiedlichen Schwellengeschwindigkeiten

Aus der Dissertation hervorgegangene Publikationen

- Florian Scheuer, Matthias Brecht, Hannes Federrath. *A privacy-aware location service for VANETs using Chaum's mixes.* In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2010), Niagara Falls, Canada, 2010.
- Florian Scheuer, Karl-Peter Fuchs, Hannes Federrath. *A Safety- Preserving Mix Zone for VANETs.* In: Steven Furnell, Costas Lambrinoudakis, Günther Pernul (Herausgeber), Proceedings of Trust, Privacy and Security in Digital Business - 8th International Conference, TrustBus 2011, Toulouse, France, Volume 6863 of Lecture Notes in Computer Science, Seite 37–48. Springer Berlin / Heidelberg, 2011.
- Andreas Tomandl, Florian Scheuer, Hannes Federrath. *Simulation-based evaluation of techniques for privacy protection in VANETs.* In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2012), 2012.