UNIVERSITÄT REGENSBURG
WIRTSCHAFTSWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR WIRTSCHAFTSINFORMATIK

Diplomarbeit

Entwurf einer Sicherheitsinfrastruktur für Vehicular Ad-hoc Networks (VANETs)

MANUEL REIL

geb. am 3. März 1980 in Cham
Matrikelnummer 1064070

Eingereicht am 13. März 2006

Betreuer:
Dipl.-Wirtschaftsinformatiker Klaus Plößl
Prof. Dr.-Ing. Hannes Federrath

Alles sollte so einfach wie möglich gemacht werden, aber nicht einfacher.
ALBERT EINSTEIN (zugeschrieben)

Aufgabenstellung

Thema: Entwurf einer Sicherheitsinfrastruktur für Vehicular Ad-hoc Networks (VANETs)

Zielsetzung: Mobile Ad-hoc-Netze bestehend aus Fahrzeugen (so genannte Vehicular Ad-hoc Networks, VANETs) können zur Steigerung von Sicherheit und Komfort im Straßenverkehr beitragen. In dieser Diplomarbeit soll eruiert werden, wie eine auf WLAN basierende drahtlose Übertragung abgesichert werden kann. Dazu sind nach Identifizierung der Schutzziele und Erstellung geeigneter Angreifermodelle Vorschläge für eine Sicherheitsinfrastruktur zu erarbeiten und zu bewerten. Dabei soll unter anderem auch darauf eingegangen werden, wer eine solche Sicherheitsinfrastruktur betreiben sollte und wie die genauen Rahmenbedingungen aussehen sollten.

Die Bewertung der Ansätze soll aufgrund ihrer Einsetzbarkeit erfolgen, d.h. es sind z. B. die Verarbeitungskapazitäten der beteiligten Geräte und die zur Verfügung stehenden Bandbreiten und Kommunikationszeitfenster zu berücksichtigen, um sicherzustellen, dass unter Umständen bestehende Echtzeitanforderungen noch erfüllbar sind.

Weiterhin sind aus den im Fahrzeug zur Verfügung stehenden Daten die zu bestimmen, welche für andere Verkehrsteilnehmer sinnvoll und nötig sind. Dabei ist zu beachten, dass die Privatsphäre der einzelnen Verkehrsteilnehmer nach Möglichkeit gewahrt bleibt, d.h. bei der Auswahl ist nach dem Grundsatz der Datensparsamkeit zu verfahren. Da die Einsetzbarkeit auch wesentlich von den damit verbundenen Kosten abhängt, soll eine wirtschaftliche Betrachtung der einzelnen Ansätze gemacht werden.

Zusammenfassung

Zukünftigen automobilen Ad-hoc-Netzwerken stehen viele Bedrohungen gegenüber, zu denen Wissenschaft und Forschung derzeit noch keine ganzheitlichen Konzepte anbieten. Der zentrale Spagat, jedem Teilnehmer Datenschutz zu gewährleisten, aber ihre Aktionen im Bedarfsfall dennoch zurechenbar zu machen, bildet zusammen mit anderen Schutzzielen die Basis der Sicherheitsbetrachtungen dieser Arbeit. Die Charakterisierung dieser VANETs und die Erstellung von Anwendungsklassen helfen, Einblicke in die Funktionsweise und die Besonderheiten dieser speziellen Netze zu erlangen.

Es wird deutlich, dass neben den Schutzzielen auch die strengen Echtzeitanforderungen und wirtschaftliche Aspekte Herausforderungen darstellen und ein Konzept in Form einer Sicherheitsinfrastruktur erfordern.

Im Kern der Arbeit werden durch die Betrachtung erprobter Sicherheitsmechanismen und eigener Beiträge die Basiskonzepte einer solchen Infrastruktur bestimmt. Diese umfassen neben dem Punkt privacy vs. auditability auch die Frage nach dem Betreiber einer solchen Sicherheitsinfrastruktur und die Wahl geeigneter Kryptoalgorithmen, die durch eine in Java implementierte Performance-Analyse untermauert wird.

Im nächsten Schritt analysiert und erweitert die Arbeit bestehende Vorschläge der aktuellen Forschungsliteratur und entscheidet sich letztendlich für zwei Konzepte, in die die vorher ermittelten Basiskonzepte eingebettet werden. Neben einer *PKI*-basierten Lösung – VPKI – ist dies das Verfahren 2MAC, das weithin auf symmetrische Kryptographie vertraut.

Inhaltsverzeichnis

KAPITEL 1

Einleitung

1.1. Motivation und Problematik

Wenn Autos miteinander reden, kann es durchaus vorkommen, dass eines lügt, sich als ein anderes ausgibt, belauscht, oder gar laut schreit und andere übertönt. In den *VDI Nachrichten* vom 16. September 2005 wurde den automobilen Ad-hoc Netzwerken der Artikel *Wenn Autos miteinander reden* veröffentlicht, der im Telematik-Bereich eine Fülle von Anwendungsmöglichkeiten ab dem Jahr 2010 prophezeit. Nicht erwähnt wurden hingegen die eingangs beschriebenen, vielfältigen Bedrohungen der Vertraulichkeit, Integrität, Zurechenbarkeit und Verfügbarkeit für diese neue Massentechnologie.

Auch die aktuelle Forschung bietet keine ganzheitlichen Konzepte an, um diese Schutzziele unter Berücksichtigung der mehrseitigen Sicherheit, der Echtzeitanforderungen von VANETs und der Praktikabilität zu realisieren.

Vor dem Hintergrund einer drohenden Öffnung der Mautdaten zur Verbrecherverfolgung[1] macht es sich diese Arbeit also zum Ziel, konkrete Sicherheitsinfrastrukturen für VANETs zu entwerfen und zu diesem Zweck erprobte Sicherheitsmechanismen mit eigenen Erkenntnissen zu kombinieren.

1.2. Vorgehensweise der Arbeit

In der Einleitung wird kurz berichtet, wie in den Medien allmählich eine Markteinführung von VANETs vorbereitet wird und welche aktuellen Entwicklungen und welche Umstände dazu motivieren, eine Sicherheitsinfrastruktur für automobile Ad-hoc-Netzwerke zu entwerfen.

Diese Motivation wird auch mit der Situation in der aktuellen Forschungsliteratur begründet, die ganzheitliche Lösungen missen lässt. Einem Top-down-Prinzip folgend, werden in den nächsten Abschnitten die speziellen Charakteristika und Ausprägungen von VANETs erarbeitet, die

[1] Nach Wolfgang Schäuble wagte am 44. Verkehrsgerichtstag, 25. Januar 2006, Generalbundesanwalt Kay Nehm einen erneuten Vorstoß in diese Richtung.

die Herausforderungen für einen geeigneten Entwurf darstellen. Im Anschluss liefert die Betrachtung zukünftiger Anwendungen wichtige Einblicke in die Brisanz dieses Szenarios und die Grundlage für die Ermittlung der Schutzziele und Angreifermodelle im folgenden Kapitel.

Zusammen mit den Echtzeitanforderungen und wirtschaftlichen Fragestellungen bilden die beiden letztgenannten Punkte die Anforderungen an eine geeignete Sicherheitsinfrastruktur. Noch in diesem Grundlagenkapitel wird in angemessener Form auf die Sicherheitsmechanismen einer *PKI* eingegangen, die einen der beiden Vorschläge dieser Arbeit stützen.

Eine Ebene tiefer versucht das Kapitel *Basiskonzepte*, Kernelemente eines Konzeptes aus erprobten Maßnahmen leitungsgebundener Netze, von Ad-hoc-Netzen und eigenen Beiträgen zu kombinieren, ohne sich auf asymmetrische oder symmetrische Kryptographie festzulegen. Hierbei hilft die definierte Struktur und Funktionsweise einer *PKI*, für diese Variante bereits konkretere Formen zu diskutieren.

Innerhalb dieser Basiskonzepte werden die Themen Identitäten und Rollen, auditability vs. privacy, Schlüsselmanagement, die Betreiberfrage und Kryptographie eingehend diskutiert. Für die Wahl geeigneter Kryptoalgorithmen wurde in Java ein Performance-Vergleich für Verschlüsselungs-, Signatur- und *MAC*-Funktionen erstellt, der zumindest eine qualitative Einschätzung ermöglicht.

Auf der untersten Ebene will diese Arbeit möglichst breit diskutieren, welche Konzepte die Forschung bereitstellt und welche ihrer Konzepte auch diesem Thema zuträglich sind. Letztendlich fließen diese Erkenntnisse in zwei konkrete Vorschläge ein – eine Variante, die auf asymmetrische Kryptographie setzt, und eine, die hauptsächlich symmetrischer Kryptographie vertraut.

KAPITEL 2

VANETs

Beginnend mit dem derzeitigen Stand in der Forschung und Entwicklung werden in diesem Kapitel die technischen Gegebenheiten und Ausprägungen von VANETs aufgezeigt, die die Basis für sämtliche Sicherheitsbetrachtungen in dieser Arbeit bilden. Zudem werden hier die globalen Annahmen über die Netzcharakteristika und die Hardwareausstattung der Fahrzeuge grundgelegt.

Als nächster Schritt erfolgt eine Klassifizierung der angedachten Anwendungen, die mit Beispielen erläutert werden.

2.1. Stand in Forschung und Entwicklung

Die früheren Projekte im Umfeld automobiler Ad-hoc-Netzwerke hatten nur einige ausgewählte Aspekte zum Ziel, z. B. die Übertragung von Warnnachrichten (Inter-Vehicle Hazard Warning)[1], das Platooning von Lkws (CHAUFFEUR 1/2)[2] oder die Steigerung des Verkehrsflusses (INVENT VLA)[3].

Hier, wie auch in dem im Jahre 2003 abgeschlossenen Projekt *Fleetnet - Internet on the Road*[4], spielte die Informationssicherheit, vor allem Datenschutz, keine oder eine stark untergeordnete Rolle, erst in neueren Forschungsprojekten wie CarTALK2000[5] wurde die Thematik am Rande aufgegriffen.

Aktuell ist ein gegenläufiger Trend zu beobachten: In den laufenden Projekten wie PReVENT[6] oder GST[7] erfährt Informationssicherheit stärkere Beachtung; dennoch leisten nur wenige Bei-

[1] http://www.deufrako.org/pdf/flyer_a.pdf. Kurzbeschreibung des Projekts Inter-Vehicle Hazard Warning
[2] Platooning: Durch die Übertragung von Fahrdaten vom führenden Fahrzeug zu den folgenden können diese ohne Einwirken des Fahrers dem ersten Fahrzeug folgen. http://www.chauffeur2.net
[3] http://www.invent-online.de/downloads/VLAhandout-D.pdf. Projektbroschüre des Projekts INVENT VLA
[4] http://www.fleetnet.de
[5] http://www.chartalk2000.net
[6] http://www.prevent-ip.org
[7] http://www.gstforum.org, http://www.gstproject.org/sec

träge zum Datenschutz, darunter NOW - Network on Wheels[8]. Doch gerade im Zuge von Überlegungen der Wirtschaftlichkeit und einer erfolgreichen Markteinführung von VANETs wird der Teilaspekt der *privacy* immer mehr als ein Enabler, ein grundlegender Baustein für den Erfolg, gesehen ([Doe05], S. 3).

Abbildung 2.1.: Bilder vom Prototypen des SOTIS-Projektes

Weitaus den meisten der hier genannten Projekte ist eine enge Verzahnung von führenden (Automobil-)Unternehmen und Hochschulen gemein. Die TU Harburg, TU München, Universität Köln (Institut für Verkehrswissenschaften), u. a. tragen vor allem in technischer Hinsicht bei, während einige wenige Forscher, darunter Prof. Dr. Christoph Paar (Ruhr-Universität Bochum)[9], Prof. Dr. Frank Kargl (Universität Ulm) und Prof. Dr. Jean-Pierre Hubaux (EPF Lausanne)[10] durchaus gute Ansätze und Vorschläge auf einzelnen Gebieten leisten, aber Gesamtkonzepte wie eine konkrete Sicherheitsinfrastruktur noch missen lassen.

Auch Standardisierungsgremien, Arbeitsgruppen und (Industrie-)Konsortien treiben verstärkt ihre Arbeit in der Standardisierung der Informationssicherheit in VANETs voran: Die P1556-Arbeitsgruppe der IEEE[11] legt ihren Schwerpunkt auf die Entwicklung essentieller Sicherheitsmechanismen für vehicle-to-vehicle- und vehicle-to-roadside-Kommunikation, die Ermittlung der zu schützenden Entitäten, der Schutz der Privatsphäre und der Informationsintegrität (siehe [Kur03], S.3 f.). In Europa beteiligen sich viele große Automobilhersteller (Audi, BMW,

[8] http://www.network-on-wheels.org
[9] http://www.crypto.rub.de/news.html
[10] http://icawww.epfl.ch/hubaux/
[11] Institute of Electrical and Electronics Engineers

DaimlerChrysler, Fiat, Renault, Volkswagen) am Car-to-Car Communications Consortium[12], das sich als non-profit organisation zum Ziel gesetzt hat, die Verkehrssicherheit und -effizienz durch Nutzung von Fahrzeug-zu-Fahrzeug-Kommunikation zu verbessern. Vor dem Hintergrund des weiterhin steigenden Verkehrsaufkommens und des eSafety-Programm der Europäischen Kommission[13] definiert dieses Konsortium selbst seine Aufgaben folgendermaßen:

- Schaffen und Etablieren eines EU-weiten offenen Industriestandards auf WLAN-Basis
- Spezifizieren und Entwickeln von Prototypen und Demonstrationssystemen
- Fördern einer gebührenfreien europaweiten Frequenzzuweisung
- Vorantreiben der weltweiten Standardisierungsbemühungen
- Entwickeln von Markteinführungsstrategien und Geschäftsmodellen.

(vgl. [Fra04])

2.2. Begriffe, Charakteristika und Technik

Die Charakteristika, die einzusetzende Technik und die Standardisierung bilden die Rahmenbedingungen und finden daher in den nächsten drei Abschnitten Beachtung. Vor allem die speziellen Eigenschaften dieser Ad-hoc-Netze bringen Herausforderungen und Einschränkungen mit sich, die in der Entwicklung von Sicherheitskonzepten zu berücksichtigen sind.

2.2.1. Die Begriffe VANET und Sicherheitsinfrastruktur

> *„A mobile ad hoc network (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links – the union of which form an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet.“ ([Int02])*

Im Grunde ist diese Definition eines MANETs auch auf den Begriff VANET zutreffend, einige Aussagen sind jedoch noch anzupassen: Ein VANET besteht neben mobilen auch aus stationären Knoten. Die mobilen Knoten sind in ihrem Bewegungsmuster an das Straßennetz, den Verkehr und die Straßenverkehrsordnung gebunden.

Das Thema dieser Arbeit besteht in dem Entwurf einer Sicherheitsinfrastruktur für solche Netze, die wie folgt definiert wird:

[12] http://www.car-2-car.org

[13] http://europa.eu.int/information_society/activities/esafety/index_en.htm Das e-Safety-Programm wurde initiiert, die Zahl der Verkehrstoten bis zum Jahre 2010 um die Hälfte zu verringern.

Eine Sicherheitsinfrastruktur umfasst alle technischen und organisatorischen Maßnahmen und Einrichtungen, die zum Erreichen der Schutzziele notwendig sind.

2.2.2. Charakteristika

Wie aus den Definitionen in 2.2.1 auf der vorherigen Seite hervorgeht, stellen VANETs eine Unterklasse von MANETs dar. Daher erben sie viele Eigenschaften, die im Detail aber Unterschiede aufweisen:

- VANETs sind autonome Systeme mobiler Knoten, von denen möglichst jeder Beiträge (z. B. Routingaufgaben) leisten muss. Von der Kooperationsbereitschaft dieser Knoten – im Wesentlichen sind dies Fahrzeuge mit mehr als zwei Rädern – hängt wesentlich die „Güte“ eines VANETs ab.

- Die Mobilität der Knoten ist im Gegensatz zu MANETs auf den Straßenverlauf ([RH05a], S. 2) beschränkt und daher in gewissem Maße vorhersehbar. Dies begünstigt bzw. ermöglicht zwar einerseits einige Anwendungen und die intelligente Verteilung von Nachrichten, erschwert aber andererseits den Schutz der Privatsphäre der Verkehrsteilnehmer (siehe auch 3.2.1 auf Seite 22).

- Um die Struktur und Mitglieder seiner Netzwerknachbarschaft zu ermitteln, tauschen die Fahrzeuge kleine Nachrichten, *beacons* genannt, aus, die neben der aktuellen Zeit, Position, Richtung und Geschwindigkeit auch relevante Sensordaten enthalten. Dadurch kann nicht nur ein Telematiksystem (siehe Kapitel 2.3.1 auf Seite 13) realisiert werden, sondern auch die Aufgabe des Routings.

  Spatially Aware Routing ([TC03]) setzt *aktives beaconing* voraus, bei dem jedes Fahrzeug periodisch *beacons* sendet. Die Angaben zum Zeitintervall schwanken in der Literatur stark. Zwischen 10 ([TMN+03], S. 25) und 300 ms ([RH05a], S. 3) werden veranschlagt. Dem steht *passives beaconing* gegenüber, das nur ein Senden von Nachrichten veranlasst, wenn ein konkretes Problem vorliegt.[14]

- Neben den mobilen sind fest installierte Knoten vorgesehen. Beide Typen können privater (im Eigentum von Privatpersonen oder -unternehmen) oder öffentlicher Natur (Busse, Einsatzfahrzeuge von Polizei, Feuerwehr, etc.) sein (siehe [RH05b], S. 2).

- Die höhere und stark schwankende Geschwindigkeit der Knoten schafft vor allem Herausforderungen funktechnischer Art (siehe auch 2.2.5 auf Seite 11), da die Verbindungszeiten u. U. sehr kurz ausfallen. Die untere Grenze dieses Geschwindigkeitsintervalls markieren parkende oder im Stau stehende Fahrzeuge, die obere bestimmt die Straßenverkehrsordnung. Während in den meisten europäischen Ländern zwischen 100 und 130 km/h erlaubt sind, existiert in Deutschland auf Autobahnen kein Tempolimit (siehe auch [RH05a], S.

[14] TIMO KOSCH bettet in [Kos05], S. 5 f., den Begriff des *beaconing* in das Netzwerkmodell des Car-to-Car Communications Consortium ein.

2); daher wird in dieser Arbeit von einer Spitzengeschwindigkeit von 250 km/h ausgegangen[15].

- Obwohl VANETs die größten mobilen Ad-hoc-Netzwerke werden, was die Knotenanzahl und die geographische Ausbreitung anbelangt, wird die Kommunikation in den meisten Fällen in geographischem Sinne lokal beschränkt sein.
- Die Datenübertragung vollzieht sich in der Regel drahtlos.

2.2.3. Ausprägungen

Die Definition (Kapitel 2.2.1 auf Seite 5) und die Charakteristika (Kapitel 2.2.2 auf der vorherigen Seite) eines VANETs führen zu einer Reihe von Ausprägungen, die ein solches Ad-hoc-Netz annehmen kann. Diese Ausprägungen können (funk-)technischer, anwendungsabhängiger und/oder geographischer Art sein und müssen in die Konzeption einer Sicherheitsinfrastruktur einbezogen werden.

self-organized vs. hybrid

Automobile Ad-hoc-Netze heißen hybrid, wenn Teilnehmern zumindest zeitweise Verbindungen zu fest installierten Stationen möglich sind. Der straßennetzweite Ausbau stellt dabei eine hohe finanzielle und organisatorische Herausforderung dar, so dass mit Einführung der VANETs nicht allerorts mit Infrastrukturanbindung zu rechnen ist. Die Vorteile einer solchen Infrastruktur liegen allerdings auf der Hand; nur so sind zentrale, rechen- und speicherintensive Serverdienste und eine Internet-Anbindung möglich. Unter den Aspekten der Informationssicherheit werden auf der einen Seite altbewährte Sicherheitsmechanismen anwendbar, auch wenn Anpassungen notwendig sind. Auf der anderen Seite ergibt sich zusätzliches Gefahrenpotential (siehe Kapitel 3.2.1 auf Seite 20).

Diesen Gegebenheiten folgend muss man auch von zeitweilig sich selbst organisierenden VANETs ausgehen, in deren Reichweite keine Station zur Verfügung steht. Diese self-organized VANETs dürfen in ihrer Grundfunktionalität und vor allem ihrem Sicherheitsniveau keine gravierenden Einbußen erfahren.

v2v vs. v2i

Die Kommunikationsbeziehungen unter Fahrzeugen werden kurz v2v (für vehicle to vehicle), die zwischen Fahrzeugen und Stationen v2i (für vehicle to infrastructure) genannt.

[15] Bei diesem Wert regeln viele Automobilbauer (z. B. Audi, BMW, Mercedes-Benz) die Geschwindigkeit elektronisch ab.

singlehop vs. multihop

Laut der Definition in 2.2.1 auf Seite 5 haben Ad-hoc-Netze multihop-Charakter. Die Kooperationsbereitschaft der beteiligten Knoten vorausgesetzt, werden hier Pakete über mehrere Knoten bis zum Ziel weitergeleitet. Dem steht singlehop gegenüber: Die Nachrichtenreichweite, die die maximale Zahl an Weiterleitungen meint, ist in diesem Fall auf 1 limitiert. Das *beaconing* (siehe 2.2.2 auf Seite 6) fällt genau in diese Kategorie.

unicast vs. geocast vs. broadcast

Wie auch in anderen WLAN-basierten Netzen werden alle Informationen technisch via broadcast gesendet, auch wenn sie ausschließlich an einen Empfänger (unicast) oder an eine ausgewählte Gruppe adressiert sind. Wenn eine solche Empfängergruppe sich über einen geographischen Zusammenhang[16] definiert, spricht man im Rahmen der VANETs von geocast (vgl. [MWH01], S. 2).

highway vs. city

Das highway-Szenario (Autobahnen, Landstraßen) unterscheidet sich vom city-Szenario (innerstädtischer Verkehr) in folgenden Punkten:

- Auf Autobahnen und Landstraßen kann die Fahrzeuggeschwindigkeit im ein Vielfaches höher sein als im Stadtbereich. VANET-Teilnehmer haben also sehr kurzlebige Kontakte zueinander.
- In der Stadt sind die Fahrtrichtungen aufgrund von Kreuzungen, Einmündungen, etc. weniger limitiert wie im highway-Bereich, in dem die Fahrzeugbewegungen leicht vorhersehbar sind.
- Die Fahrzeugdichte an verkehrsneuralgischen Punkten des city-Szenarios (Kreuzungen, etc.) bedingt auch ein hohes Aufkommen von *beacons*, die von jedem Teilnehmer zeitnah verarbeitet werden müssen. U.U. ist es sogar notwendig, Aktionen zur Vermeidung von Unfällen einzuleiten.
- Die Menge der zu verarbeitenden *beacons* hängt von der Sendefrequenz ab, die in der gegenwärtigen Literatur kontrovers gesehen wird. In [RH05b], S. 3f., wird DSRC-konform von einem Abstand von 300 Millisekunden zwischen zwei *beacons* im highway-Szenario und von 100 Millisekunden im city-Szenario ausgegangen.

  Dagegen fordern Projektmitarbeiter von CARTALK 2000 eine gegenteilige Anpassung der Sendefrequenz: Ihrer Implementation nach werden auf Autobahnen aufgrund der höheren

[16] Bespiele hierfür: alle Teilnehmer hinter dem Senderfahrzeug innerhalb von 100 Metern; Fahrzeuge auf der Gegenfahrbahn; Fahrzeuge im „toten Winkel“ .

Fahrzeuggeschwindigkeit auch die *beacons* in kürzeren Abständen (alle 10 Millisekunden) gesendet (vgl. [TMN+03], S. 24f.).

In dieser Arbeit wird dem ersten Vorschlag zugesprochen, da hier eher der Intention von VANETs, sprich der Verbesserung der Verkehrssicherheit, gefolgt wird. Die Argumentation von CARTALK 2000 beruht vor allem auf der zu erreichenden Aktualität des *neighborhood table*, also schließlich der zu erwartenden Qualität des Routings.

2.2.4. Annahmen

Da die Standardisierungsbemühungen derzeit noch im Gange sind, müssen für VANETs gewisse Annahmen getroffen werden, auf die die Betrachtungen dieser Arbeit gegründet sind. Es handelt sich dabei in der Masse um technische Grundgegebenheiten wie die Fahrzeugausstattung und die Netzeigenschaften künftiger automobiler Ad-hoc-Netze.

Sichere Hardware

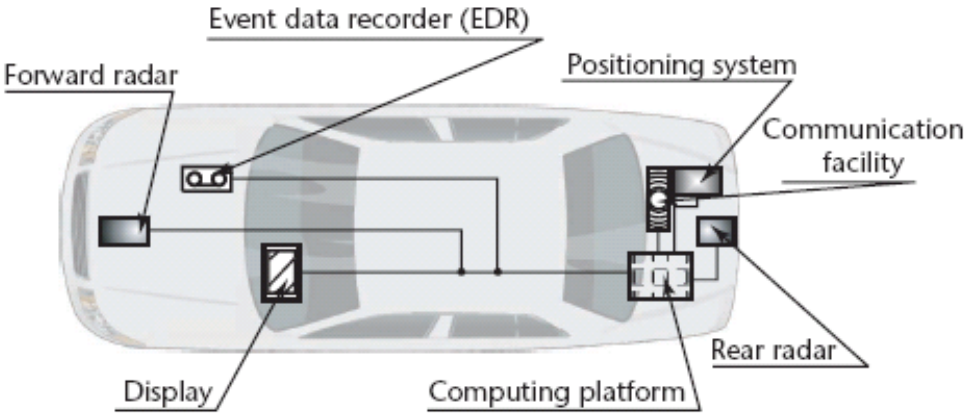

Abbildung 2.2.: Annahmen über die Hardwareausstattung teilnehmender Fahrzeuge

Um an VANETs teilnehmen zu können, müssen Fahrzeuge bestimmte technische Voraussetzungen erfüllen. Zwingend erforderlich sind Mechanismen, die seine Position und die aktuelle Zeit möglichst sicher bestimmen. Da bereits mit den Navigationssystemen dementsprechende Einrichtungen zur Ort- und Zeitbestimmung zum Einsatz kommen, bieten sich Satelliten-gestützte Systeme wie *GPS*[17] und ab 2008 *Galileo* an. *Galileo* bietet neben der Unabhängigkeit vom amerikanischen und militärisch geprägten GPS (vgl. [JJ04]) auch technologischen Fortschritt, der die Integritätssicherung und die Echtheitsprüfung[18] der Daten verspricht ([Eur03], S. 8 ff.). In Abbildung 2.2 aus [HCL04], S. 50, entspricht dies dem Punkt *Positioning system*.

[17] Global Positioning System

[18] Satelliten können sich gegenüber zertifizierten Teilnehmern über eine Public-Key-Infrastruktur authentifizieren und so vor Spoofing schützen.

Des Weiteren muss ein Teilnehmer die uneingeschränkte Möglichkeit haben, Nachrichten in alle Richtungen (omnidirektional) zu senden und zu empfangen (*Communication facility*).

Zur Verarbeitung der Nachrichten ist eine Rechnereinheit vorgesehen, die die Rechen- und Speicherkapazität aktueller Notebooks erreicht oder auch leicht übersteigt; ein Fahrzeug bietet dafür ausreichend Raum und Energie. Neben der Datenaggregation, -aufbereitung und -darstellung gehören die Maßnahmen zu einer sicheren und integeren Kommunikation[19] zu ihrem Aufgabenbereich, sofern sie nicht auf externe *TPM*[20] oder SmartCards[21] ausgelagert werden (vgl. [RH05b], S. 5).

Context awareness heißt ein stark propagiertes Schlagwort der Automobilindustrie (siehe z. B. [HCL04], S. 49) und beschreibt die fortschreitende Integration von Fahrzeugsensoren, die dem Fahrzeug einen Überblick über die Verkehrssituation und die Umgebung verschaffen. All diese Sensoren wie ABS, ESP, ASR, Reifendruckmesser, Abstandsradar, Nachtsichtkameras, etc. liefern kontinuierlich Daten, die auch anderen VANET-Teilnehmern von Nutzen sein können.[22]

Eine optionale Komponente stellt der *EDR, Event Data Recorder* dar, der in einigen Entwürfen von Sicherheitsinfrastrukturen verkehrskritische Meldungen ähnlich einer Blackbox in Flugzeugen konserviert. Damit sollen später z. B. Unfallhergänge schneller nachvollzogen werden können, auch wenn das Fahrzeug stark in Mitleidenschaft gezogen wurde. Außerdem liegen Informationen vor, die flüchtige Unfallbeteiligte greifbar machen.

All diese Komponenten dürfen nur von autorisierten Fachkräften verbaut und gewartet werden, um Manipulationen an der Hardware zu minimieren.

Nachrichtenübertragung

Die Übertragung aller Nachrichten geschieht über die Luftschnittstelle, es wird dabei von *DSRC*-gestützten Systemen (siehe auch 2.2.5 auf der nächsten Seite) ausgegangen, die eine maximale Reichweite von 1000 Metern aufweisen. Die natürlicherweise auftretenden Probleme in physikalischer oder technischer Hinsicht finden hier aber keine Betrachtung.

Sicheres Routing

Seit Beginn der VANET-Forschung wurde ein starker Focus auf die Entwicklung von Routing-Protokollen gelegt. Mittlerweile gibt es eine Vielzahl an wissenschaftlichen Veröffentlichungen, die sich zumeist in topologie- oder positionsbasierte Verfahren einteilen lassen. Letztere haben

[19] Dazu zählen alle Aufgaben, die mit Einführung einer Sicherheitsinfrastruktur zusätzlich von jedem Teilnehmer zu erbringen sind: Ver- und Entschlüsseln, Signieren und Signaturprüfungen von Nachrichten, Bilden und Prüfen von *message authentication codes*, Speichern von Schlüsseln.

[20] *Tamper Proof Modules*: manipulationssichere Hardwarekomponenten

[21] SmartCards besitzen gegenüber einer gewöhnlichen Chipkarte einen Mikroprozessor und einen programmierbaren Speicher, z. B. JavaCard. Siehe dazu [Eck03], S. 387 ff.

[22] Die Absicherung der Sensordaten im Fahrzeug ist Thema weitergehender Forschung und wird daher hier nicht weiter verfolgt.

den Vorteil in ihrer Unterstützung von geocast (siehe Kapitel 2.2.3 auf Seite 8). Als praktikables Beispiel ist *SAR* – Spatially Aware Routing ([TC03]) – zu nennen.

Teilnehmerstruktur

Als Teilnehmer eines VANETs kommen nur Fahrzeuge in Frage, die die Anforderungen hinsichtlich Platz-, Speicher- und Rechenkapazität erfüllen, dies sind in erster Linie PKWs und LKWs, auch Züge oder Helikopter sind nicht auszuschließen.

Teilnehmerverhalten

Generell wird in dieser Arbeit ein sehr defensiver Ansatz zur Einschätzung des Teilnehmerverhaltens gewählt, d.h. man geht davon aus, dass jeder Teilnehmer, jede Institution ein potentieller Angreifer ist. Kooperationen unter Teilnehmern, wie sie in Kapitel 3.2.1 auf Seite 26 kurz behandelt werden, bieten jedoch Möglichkeiten, Fehlinformationen, die konform zum Sicherheitskonzept versandt wurden, dennoch als solche zu entlarven. Besondere Bedeutung kommt auch dem Schaffen von Anreizen zu, sich als VANET-Teilnehmer korrekt zu verhalten. So erfahren die rein technischen Sicherheitsmaßnahmen eine Unterstützung.

2.2.5. 802.11p – DSRC

> *Unter DSRC[23] versteht man die funkbasierte bidirektionale Kommunikation zwischen Fahrzeugen (v2v – vehicle to vehicle) oder Fahrzeugen und stationären Einrichtungen (v2r – vehicle to roadside bzw. v2r – vehicle to infrastructure). ([Gor], S. 2)*

Die Organisationen ASTM[24] und die IEEE Standard Group treiben die Standardisierungsbemühungen hierin an, auf die sich die in Kapitel 2 auf Seite 3 vorgestellten Gremien, vor allem das Car-to-Car Communications Consortium, und die Automobilhersteller stützen. Bereits 1999 wurde durch die FCC[25] ein 75 MHz breites Frequenzband (5,850 - 5,924 GHz) für die USA reserviert, für Europa und Japan dagegen den Frequenzbereich um die 5,8 GHz (vgl. [Gor], S. 1 ff., und [Why05], S. 12).

Für diesen neuen Standard 802.11p[26] fließen vor allem Entwicklungen des 802.11a- und des 802.11g-Standards ein, diese werden mit bewährten Algorithmen der Kommunikationstechnik kombiniert. Bei einer Reichweite von 300 - 1000 Metern wird eine Datenrate von 6 - 27 Megabit pro Sekunde angestrebt.

[23] Dedicated Short Range Communication
[24] American Society for Testing and Materials
[25] Federal Communication Commission
[26] früher WAVE: Wireless Access in Vehicular Environments

802.11p bringt laut [Gor], [CAM05], S. 139, [Sto04], [YEY+04] und [Kos05] folgende Vorteile mit sich:

- 802.11 ist eine erprobte und international verbreitete Technologie, die das Vertrauen der Staaten und das der Automobilunternehmen genießt.
- Die Kompatibilität mit existierenden Strukturen ist gegeben, durch Frequenzwechsel in den Geräten ist der Zugang zu öffentlichen oder privaten Hotspots[27] möglich. Zusätzlich kann aufgrund existierender Standards kosteneffizient entwickelt werden.
- 802.11p ermöglicht sowohl private also auch öffentliche Übertragungen (broadcast), was einen erheblichen Vorteil gegenüber zellulären Techniken wie GSM oder UMTS darstellt.
- Der Straßenverkehr als Einsatzgebiet fordert eine Priorisierung bestimmter sicherheitsrelevanter Nachrichten gegenüber gewöhnlichen. Dies leistet bei DSRC ein Kontrollkanal, der z. B. den Transfer von Motormanagementdaten zugunsten einer aktuellen Unfallmeldung unterbricht.
- Nach einigen Praxistests unter realen Bedingungen scheint DSRC Latenzzeiten unter 100 ms zu leisten, wie sie für einige zeitkritischen Anwendungen Voraussetzung sind. Dies ist einer der Hauptvorteile gegenüber anderen drahtlosen Kommunikationstechnologien.

2.3. Anwendungen

Das Spektrum möglicher Anwendungsgebiete und Einsatzszenarien wird zunehmend größer und vielfältiger. Im März 2005 veröffentlichte das Vehicle Safety Communications Consortium (VSCC) eine umfassende Liste (siehe [CAM05], S. 4 - 138) an möglichen Anwendungen für automobile Ad-hoc-Netzwerke – über 75 Einsatzszenarien unterteilt in safety- und non-safety-related applications – wurden identifiziert. Diese wurden auch unter dem Gesichtspunkt der Markteinführung und der Realisierbarkeit mit DSRC-Systemen [28] analysiert. Für die Ermittlung von Schutzzielen, Angreifermodellen und schließlich einer geeigneten Sicherheitsinfrastruktur, wie sie diese Arbeit vorsieht, ist es jedoch nicht vorteilhaft, die Anwendungen und damit die Nachrichten, die zwischen den Knoten eines VANETs ausgetauscht werden, einer thematischen Einteilung zu unterwerfen. Vielmehr wird der einfache Ansatz ([RH05a], Seite 3) von MAXIM RAYA und JEAN-PIERRE HUBAUX, nur zwischen safety-related messages und other messages zu unterscheiden, um die Gruppe der Alarmsignale erweitert. Eine solche Klassifizierung ist überhaupt notwendig, um den Sicherheitsanforderungen, die sich in den genannten drei Gruppen durchaus unterscheiden können, gerecht zu werden.

Im Folgenden werden die vorgenommene Einteilung der Anwendungen und die dafür zur Verfügung stehenden Daten erläutert und durch Beispiele (Quellen: [CAM05], S. 6-31, [Why05], S. 28-37) veranschaulicht.

[27] Hotspots: Geographisches Gebiet, in dem Zugang zu drahtlosen Netzwerken verfügbar ist.
[28] DSRC: Dedicated Short Range Communications, siehe Kapitel 2.2.5 auf der vorherigen Seite

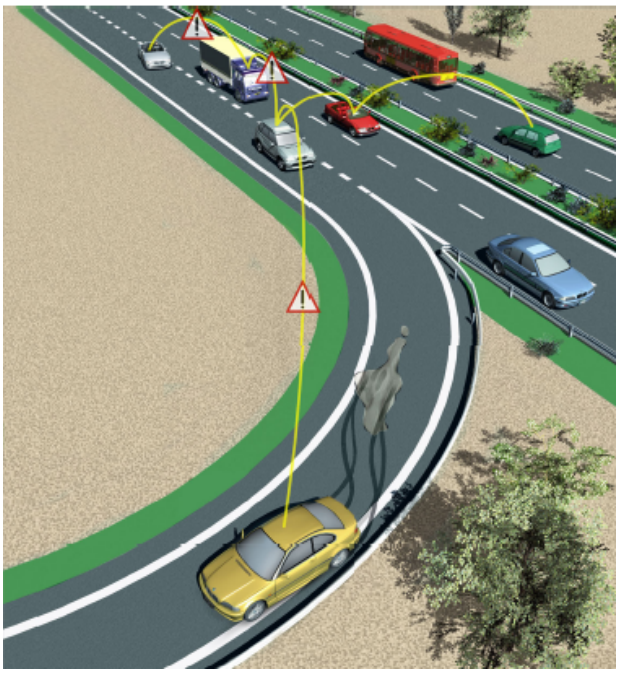

Abbildung 2.3.: Telematik-Anwendungen: Ein Fahrzeug ermittelt über seine ESP-Sensoren eine Gefahrenstelle und warnt nachfolgende Fahrzeuge.

2.3.1. Telematik-Nachrichten und Warnungen – A1

Die erste Klasse von VANET-Anwendungen, die *Telematik-Nachrichten* und *Warnungen*, bildet die Schnittmenge aller in Kapitel 2.1 auf Seite 3 genannten Projekte, da sie den Ursprung aller VANET-Entwicklungen darstellen. Hier subsumieren sich alle Applikationen, die den Erfassungshorizont eines Fahrzeugführers (eines VANET-Knotens) bezüglich der Verkehrslage erweitern; dazu tauschen die Fahrzeuge selbsttätig Informationen aus, die von den Knoten selbst oder von Basisstationen aggregiert und an alle relevanten Teilnehmer verschickt werden. In dieser Klasse werden damit weitgehend alle Nachrichten über broad-, multi- oder geocast-Mechanismen (siehe 2.2.5 auf Seite 11) verteilt (vgl. [Why05], S. 38). DÖTZER nimmt in [DKS05] eine detaillierte Klassifizierung rein verkehrsdatenrelevanter Nachrichten vor; diese zeigt zwar in anschaulicher Weise die Diversität dieser Anwendungsklasse, ist aber nicht auf eine Sicherheitsbetrachtung ausgerichtet.

Die *Telematik-Nachrichten* bestehen u. a. aus den Daten, die aus den Fahrzeugsensoren Airbag, ABS, ESP, ASR usw. gewonnen werden, der aktuellen Position, Geschwindigkeit, Beschleunigung und Sendezeit.

Durch geschicktes Aggregieren der empfangenen Nachrichten können einerseits die mobilen Knoten komplexere Verkehrszusammenhänge wie Staubildung früh erkennen und zeitnah Entscheidungen treffen, andererseits können die stationären Knoten gelenkt von zentralen Instanzen den Verkehrsfluss in einem größeren geographischen Raum verbessern. Für kritische Verkehrsabschnitte bietet es sich an, solche Basisstationen gezielt zur Information über Gefahrenstellen

einzusetzen.

1. An Kreuzungen werden Informationen über die Präsenz, Position und Richtung anderer Fahrzeuge verwendet, um die Verkehrsregelung dynamisch zu optimieren. (v2i, city)

   a) Ein grüner Linkspfeil einer Ampel wird nicht mehr zeitgesteuert, sondern abhängig von der aktuellen Verkehrssituation signalisiert.

   b) Selbige Daten assistieren auch bei der Geschwindigkeitsanpassung innerhalb einer „grünen Welle“.

2. Mehrere Beacons, die eine für den Streckenabschnitt untypisch niedrige Fahrzeuggeschwindigkeit beinhalten, lassen auf eine beginnende oder bereits andauernde Stausituation schließen. Diese Information kann wiederum an nachfolgende Fahrzeuge weitergeleitet werden und die Routenfindung und Streckenplanung entscheidend unterstützen. Um die Fahrzeugführer zusätzlich vorzubereiten, werden Informationen über die Stauursache (Baustellen, gesperrte Fahrbahnen, Fahrzeuge mit Übermaßen, etc.) an die betroffenen Knoten verteilt. (v2v, highway, city)

3. Platooning bietet Möglichkeiten, den Verkehrsfluss zu optimieren. Dabei verschmelzen durch Positions- und Geschwindigkeitskontrolle eine Reihe von Fahrzeugen zu einer Einheit (Platoon) und lassen so kürzere Fahrzeugabstände zu. (v2v, highway)

4. Damit verwandt ist die Cooperative Adaptive Cruise Control. Bereits bestehende Anwendungen der Abstandsmessungen und daraus resultierenden Geschwindigkeitsanpassungen erfahren durch Fahrzeug-zu-Fahrzeug-Kommunikation eine Verbesserung. (v2v, highway)

5. Stationäre Knoten leisten Beiträge zur Parkplatzsuche in Städten. (v2i, city)

Wesentlich zeitkritischer gestalten sich die *Warnungen*, die vor konkreten Bedrohungen der Verkehrssicherheit schützen wollen. Dazu zählen vor allem Unfälle, Hindernisse, schlechte Fahrbahnbeschaffenheit, usw. Mittlerweile ermöglichen neuere Entwicklungen im Sensorik-Bereich genauere Einschätzungen und Abstufungen , so kann z. B. zwischen Aquaplaning, Eis-, Schneeglätte oder einer Ölspur unterschieden werden. Wegen der Fähigkeit, Fahrzeugführer und Fahrzeug u. U. vor Schaden zu bewahren, werden den *Warnungen* absolute Priorität zugesichert.

1. Verkehrsteilnehmer werden vor einer beginnenden oder anhaltenden Rotphase einer Ampel oder vor einem Stoppschild gewarnt; sogar eine Bremsassistenz basierend auf der Fahrbahnbeschaffenheit, der derzeitigen Geschwindigkeit des Fahrzeugs, der Position des Verkehrszeichens und ggf. den Schaltintervallen der Ampel ist vorgesehen. (v2i,city).

2. Mobile Knoten werden vor unmittelbar bevorstehenden Kollisionen an Kreuzungen, Bahnübergängen, etc. gewarnt; dies kann nicht nur auf Grundlage der Fahrzeugnachrichten, sondern auch auf Nachrichten von stationären Sendern initiiert werden. Ziel ist es,

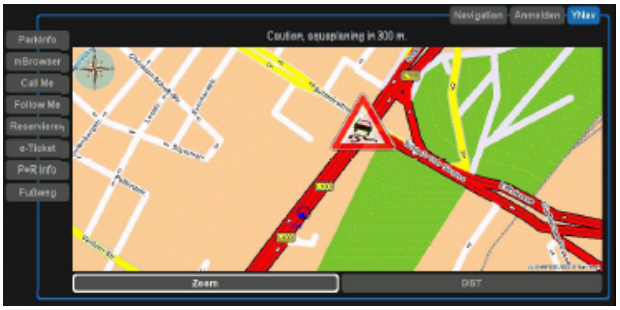

Abbildung 2.4.: Anzeige einer Telematikwarnung im Display des Navigationssystems (BMW)

Fahrer und Fahrzeug[29] zu einer eventuellen Vermeidung oder Abschwächung eines Verkehrsunfalls zu befähigen oder auf den bevorstehenden Aufprall bestmöglich vorzubereiten. (v2v, v2i, city, highway)

3. Vorausfahrende Fahrzeuge übermitteln ihre erworbenen Kenntnisse über Fahrbahnbeschaffenheit (Glätte, Aquaplaning, Ölspuren, etc.). (v2v, city, highway)

4. Nachdem sich ein Unfall ereignet hat[30], werden alle nachfahrenden Knoten darüber in Kenntnis gesetzt und können so Folgeunfälle womöglich vermeiden und darüber hinaus Hilfe leisten. (v2v, city, highway)

5. Das Unfallfahrzeug leistet - wenn technisch noch möglich - selbst einen Beitrag zur Initiierung der Rettungskette, die genaue Position der Unfallstelle, die Zahl der involvierten Fahrzeuge, die Unfallschwere (siehe vorheriges Beispiel), etc. liefern den Einsatzkräften wertvolle Hinweise für ein rasches und adäquates Eingreifen. (v2v, v2i, city, highway)

2.3.2. Einsatzsignale – A2

Einsatzfahrzeugen der Polizei, Feuerwehr, Rettungswagen, THW, etc. werden aufgrund ihres Aufgabenbereichs besondere Privilegien in einem VANET eingeräumt. Wie in A1 herrscht geocast als Verteilungsmodus für Nachrichten vor.

1. Im Einsatz signalisieren sie ihr Eintreffen am Ort des Geschehens schon aus weiter Entfernung und regen damit Gassenbildung gemäß der übermittelten Zahl und Dimensionen der Einsatzfahrzeuge an.

[29] Die Effizienz der Sicherheitseinrichtungen eines Fahrzeugs (Airbags, etc.) wird durch zusätzliche Informationen wie Aufprallwinkel, Geschwindigkeit, Fahrbahnart und -verlauf gesteigert. Langfristig wird in Gefahrsituationen sogar ein autonomes Handeln des Fahrzeugs selbst erwogen.

[30] Dies wird aufgrund von ausgelösten Airbags oder einer neuartigen akustischen Messung der Unfallschwere (siehe http://www.all4engineers.com/index.php;do=show/lng=de/alloc=34/id=758/sid=13) ermittelt werden.

2. Die Mitglieder einer Rettungskolonne koordinieren sich gegenseitig in schwierigen Einsatzszenarien. Wie auch im ersten Fall können diese auch von zentralen Leitstellen über Basisstationen Unterstützung erfahren.

3. Wenn es notwendig ist, sind Einsatzfahrzeuge autorisiert, den Verkehr zu ihren Gunsten zu regeln, z. B. durch Schalten einer Ampelkaskade. Aktuell wird das Wenden auf Autobahnen in einer Stausituation diskutiert; dies soll aber erst durch die Polizei vor Ort erlaubt und gelenkt werden, die hier besprochenen Einsatzsignale geben den Polizeikräften ein zusätzliches Instrument mit kurzer Reaktionszeit an die Hand.

Ampeln sind ebenfalls vorgesehen, über Einsatzsignale ihren aktuellen Status und ihre Schaltzeiten an die Verkehrsteilnehmer weiterzugeben. Des Weiteren sollen diese gewarnt werden, wenn sie kritische Verkehrszeichen (rote Ampeln, Einbahnstraßen) nicht beachten. (vgl. [DKKS05])

2.3.3. Andere Dienste – A3

Unter der dritten Klasse verbergen sich hauptsächlich Anwendungen mit wirtschaftlichen Hintergrund. Obwohl diese durch Missbrauch in der Regel keine Menschenleben bedrohen, dürfen hier Sicherheitsaspekte keineswegs außer Acht gelassen werden; vielmehr sind es gerade diese Anwendungen, die erst für eine hohe Verbreitung von ausgestatteten Fahrzeugen sorgen werden.

Trotz ihrer unterschiedlichen Ausprägungen ist den Diensten dieser Klasse der Transaktionscharakter gemein, d.h. die Nachrichten werden i. A. nicht per broadcast verteilt. (vgl. [Why05], S. 38)

1. Über vehicle-to-roadside-Kommunikation wird mobilen Knoten eine Internetanbindung ermöglicht.(v2i, city, highway)

2. Bereits in einigen europäischen Staaten realisiert (z. B. in Österreich[31]), bietet der Mautbereich (toll collection) großes Potential in wirtschaftlicher Hinsicht. Zudem leistet er einen der größten Beiträge zur Netzabdeckung. (v2i, city, highway)

3. Drive thru payment an Raststätten, Fast-Food-Restaurants oder Parkhäusern gestaltet nicht nur alltägliche Abfertigungsprozesse bequemer und schneller, sondern fördert – bei erfolgreicher Konzeption und Implementation – die Benutzerakzeptanz von VANETs. In diesen Bereich fallen auch Abwicklungen in der Fahrzeugvermietung. (v2i, city, highway)

2.3.4. Die Daten in VANETs

Leider bestehen in der derzeitigen Forschungliteratur nur sehr wenig Hinweise darauf, welche Fahrzeugdaten für welche Anwendungsbereiche zu Rate gezogen werden. In der Forschergruppe

[31] http://www.go-maut.at/go

um TIMO KOSCH, MARKUS STRASSBERGER, u. a. ist aber eine derartige Auflistung in Bearbeitung und damit in naher Zukunft verfügbar. Exemplarisch sollen in diesem kurzen Abschnitt nur ein Eindruck von der Datenvielfalt gegeben werden, der in heutigen Fahrzeugen zur Verfügung steht. Zudem werden in weiteren Klassen kurz die Daten, die eine Blackbox (*EDR*) speichern könnte, und mögliche Benutzereingaben vorgestellt.

Fahrzeugsensordaten

Ein Fahrzeug neuerer Generation verfügt über ein Netzwerk an Sensoren (vgl. [Rob02], S. 351), die fahrzeugweit ihre Datenmessungen zur Verfügung stellen. Diese werden vor allem im Anwendungsgebiet AM1 verwendet, um andere Verkehrsteilnehmer über die Verkehrslage, u. ä. zu informieren. Folgende Sensoren werden wohl in verkehrssichernde Nachrichten und *Warnungen* einfließen:

- Abstandsradar (*ACC*, Precrash)
- Hochdrucksensor (*ESP*)
- Lenkradwinkelsensor (*ESP*)
- Beschleunigungssensor (Airbag)
- Sitzbelegungssensor (Airbag)
- Drehratesensor (*ESP*)
- Neigungssensor (Sicherungssysteme)
- Drehratesensor (Überrollsensierung)
- Drehzahlsensor (*ABS*)
- Neigungssensor (Scheinwerferverstellung)
- Regensensor (Scheibenwischersteuerung)
- Abstandssensor Ultraschall (Rückraumüberwachung)

Zum Beispiel könnte aus den Daten des Beschleunigungssensors des Airbags und des Sitzbelegungssensors eine *Warnung* und ein Notruf abgesetzt werden, der über einen Frontalaufprall eines Fahrzeugs mit vier Insassen informiert.

Weiterhin könnte der Drehratesensor zusammen mit Lenkradwinkelsensor einen Abbiegevorgang erkennen und hier Hilfestellung für alle betroffenen Fahrzeuge bieten. Schlupfregelsysteme wie *ESP* und *ASR* erkennen Glatteis, Aquaplaning, etc.: Diese Informationen sind für nachfahrende Fahrzeuge von Vorteil. Zum Beispiel sind auch ein automatisches Abblenden des Lichts oder eine *Warnung* bei überfrierender Nässe denkbar.

Daten der Blackbox

In der Blackbox oder *EDR* werden nach dem FIFO-Prinzip in begrenztem Umfang empfangene Nachrichten gespeichert, was folgenden Einsatzzwecken dienlich ist:

- Ermittlung der Beteiligten und der Rekonstruktion eines Unfalls
- Vergleich mehrerer Nachrichten unterschiedlicher Sender, um Falschinformationen auszusortieren
- Aggregieren einer Vielzahl von Nachrichten zu einer Information über die Verkehrslage.

Benutzereingaben für Dienstenutzung

Nur in den Anwendungsgebieten A2 und A3 sind Benutzereingaben zu erwarten, die über ein einfaches Bestätigen oder Abbrechen hinausgehen. Sie werden in aller Regel getätigt, wenn das Fahrzeug steht oder ein Beifahrer dies übernehmen kann.

KAPITEL 3

Grundlagen

3.1. Motivation

In Kapitel 2 auf Seite 3 wurde gezeigt, wie vielfältig die Ausprägungen, die Einsatzszenarien und damit die Anwendungen von VANETs sein werden. Auch weil in Forschung und Entwicklung die Standardisierung noch nicht abgeschlossen ist, ist die Schar zukünftiger Anwendungen kaum in Gänze vorhersehbar. Deshalb werden die Anforderungen an eine Sicherheitsinfrastruktur nicht bis ins Detail konkretisiert, sondern offen für neue Entwicklungen und im Sinne aller VANET-Beteiligten spezifiziert.

Dieser umfangreiche Bedarf an Schutzmaßnahmen und die Brisanz des Einsatzszenarios, des Straßenverkehrs, fordern eine ganzheitliche Sicht und Konzeption der Sicherheit in VANETs, die sich auch in den ermittelten Angreifermodellen in Kapitel 3.3 auf Seite 31 widerspiegelt.

In drahtgebundenen Netzen wie dem Internet oder Intranets von Unternehmen, Organisationen, Ämtern, etc. werden bereits Sicherheitsinfrastrukturen wie *PKI*[1] oder *web of trust* (siehe Kapitel 3.4.2 auf Seite 41) eingesetzt, um die generischen Schutzziele Vertraulichkeit und Integrität/-Zurechenbarkeit zu gewährleisten.

Die Überlegungen, inwieweit solch bestehende Ansätze auf automobile Ad-hoc-Netze adaptiert werden können, erfolgen zwar erst in Kapitel 4 auf Seite 45, die Grundlagen dafür werden aber in diesem Kapitel gelegt.

3.2. Anforderungen

Alle im Laufe dieser Arbeit vorgestellten Ansätze zu Sicherheitsinfrastrukturen werden an den Anforderungen gemessen, die in diesem Kapitel aufgestellt werden. Neben den reinen Sicherheitsfragen werden die Echtzeitanforderungen von VANETs berücksichtigt, die von den zusätzlichen Maßnahmen einer Sicherheitsinfrastruktur bedroht werden. Auch wirtschaftliche Aspekte werden in gewissem Umfang betrachtet.

[1] *Public Key Infrastructure*

3.2.1. Schutzziele - AN1

Bevor man Sicherheitskonzepte entwirft, ist es nötig, sich im Detail zu vergegenwärtigen, welche Güter eines Systems gegen welche Angreifer (vgl. Kapitel 3.3 auf Seite 31) zu schützen sind. Bei den zu betrachtenden Gütern handelt es sich um Daten und Informationen, die in den VANETs zwischen den Knoten – gleichwohl mobilen wie stationären – ausgetauscht werden. In den drei folgenden Abschnitten werden die Schutzziele in Abhängigkeit von den in 2.3 auf Seite 12 klassifizierten Anwendungen betrachtet.

Mehrseitige Sicherheit

[Fed99] untersucht auf den Seiten 22 - 26 die Sicherheitsanforderungen mobiler Kommunikation in der Literatur. Zusammenfassend lässt sich feststellen, dass die hier zitierten Autoren Schutzziele unter unterschiedlichen Standpunkten betrachten und damit jeweils die Schutzinteressen einer oder mehrerer Parteien stärker in den Vordergrund rücken. Indem diese Interessen explizit formuliert und diskutiert werden, treten nicht erkannte Konflikte und Unvereinbarkeiten zu Tage.

> „Mehrseitige Sicherheit *bedeutet die Einbeziehung der Schutzinteressen aller Beteiligten sowie das Austragen daraus resultierender Schutzkonflikte, etwa beim Entstehen einer Kommunikationsverbindung.“ ([Pfi00], S. 17)*

Gerade in VANETs, in denen Knoten eine Vielzahl von möglichen Rollen (siehe 3.3.1 auf Seite 31), teilweise auch gleichzeitig, verkörpern können, ist der Ansatz, alle Beteiligten per se als potentielle Angreifer zu betrachten, hilfreich. Damit wird auch der Forderung von FEDERRATH in [Fed99], S. 20, entsprochen, die Sicherheit unabhängig von explizitem „Vertrauen in fremde Systemteile oder andere Teilnehmer“ postuliert. Da im Folgenden Teilaspekte der allgemeinen Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit aus den Blickwinkeln unterschiedlicher Teilnehmerkategorien besprochen werden, kann nach deren Erreichen von *mehrseitiger Sicherheit* gesprochen werden (analog zu [Fed99], S. 21).

Vertraulichkeit

> „Vertraulichkeit *bedeutet, dass Daten und Informationen nur Berechtigten bekannt werden.“ (nach [Pfi00], S. 7)*

Die Informationsvertraulichkeit im engeren Sinne (engl. *confidentiality, privacy*) ist im Datenschutzgesetz Deutschlands fest verankert.

Bundesdatenschutzgesetz (BDSG), § 1 Absatz 1[2]:

> *Zweck dieses Gesetzes ist es, den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.*

Teledienstedatenschutzgesetz (TDDSG), § 4 Absatz 6:

> *„Der Diensteanbieter hat dem Nutzer die Inanspruchnahme von Telediensten und ihre Bezahlung anonym oder unter Pseudonym zu ermöglichen, soweit dies technisch möglich und zumutbar ist. Der Nutzer ist über diese Möglichkeit zu informieren.“*

Mit fortschreitender Entwicklung in der maschinellen Informationsspeicherung und -verarbeitung, rückt der Schutz der Privatsphäre stärker in das Bewusstsein der Menschen. Offen sichtbare Überwachungskameras in den Innenstädten, Diskussionen über den polizeilichen Lauschangriff oder die Big-Brother-Phänomene[3] tragen dazu bei. So stehen die Menschen neuen technischen Entwicklungen skeptisch gegenüber, vor allem wenn sie wie im Fall der VANETs in Bereiche des Lebens vordringen, die bisher – vermeintlich – frei von informationsverarbeitenden Systemen waren.

Um der Gesetzeslage und dem Anspruch der zukünftigen Benutzer gerecht zu werden, ist die Vertraulichkeit der Identität der sendenden und empfangenden Knoten in jedem Anwendungsfall Schutzziel. Der hier verwendete Begriff der Identität ist dabei in diesem Teil der Arbeit noch unabhängig von einer Zuordnung zu einer Person, einem Fahrzeug oder einer Verbindung beider Entitäten zu sehen.

V1 Da eine Nachricht immer in gewisser Weise Aufschluss über Quelle und Ziel gibt, sind diese durch geeignete Maßnahmen bestmöglich zu schützen. MAC-Adresse[4], IP-Adresse[5] und bei signierten Nachrichten Zertifikate[6] können hier als Beispiele für solche „identifying marks“ ([Why05], S. 50) angeführt werden.

[2] Im *Volkszählungsurteil* wurde vom Bundesverfassungsgericht das Recht auf informationelle Selbstbestimmung als Ausprägung des allgemeinen Persönlichkeitsrechts ausgelegt; vgl. [Sch00].

[3] vgl. auch [RH05b], S. 5

[4] Die MAC- oder Adapteradresse identifiziert eindeutig die Sendehardware und dient auf der untersten Schicht des ISO/OSI-Referenzmodells der Adressierung von Paketen.

[5] IP-Adressen erlauben eine logische Adressierung von Geräten (Hosts) in IP-Netzwerken wie beispielsweise dem Internet.

[6] Zertifikate ordnen einen öffentlichen Schlüssel einer Identität (einer Person, einer Organisation, etc.) überprüfbar zu. So kann eindeutig festgestellt werden, ob die Signatur tatsächlich von der vorgebenden Identität erzeugt wurde. Zertifikate werden in Kapitel 3.4.1 auf Seite 39 ausführlicher behandelt.

V2 Selbst wenn nun den Verbindungsdaten im Sicherheitskonzept ausreichend Beachtung geschenkt wurde, ist es für Angreifer immer noch möglich, über ungeschützte Daten und deren Aggregation über einen Zeitraum die Identität eines Knotens zu enthüllen. Bei allen Nachrichten, die durch eine broadcast-Technik explizit an mehrere, in den meisten Fällen dem jeweiligen Sender unbekannten Teilnehmer verteilt werden (vor allem A1), ist es ausdrücklich erwünscht, sie möglichst schnell an den relevanten Teilnehmerkreis zu übermitteln, also Dateninhalte nicht geheim zu halten.

Um vor der Erstellung von Bewegungs-, Verhaltens- und Dienstnutzungsprofilen, die für die gewerbliche Nutzung oder die Exekutive ohne richterlichen Beschluss von Interesse sein können, zu schützen, ist die Unverkettbarkeit von Nachrichten zu gewährleisten. Folgende „Angriffe“ sind ansonsten denkbar:

- Verfolgung von Verdächtigen durch Polizei, Zoll, usw.
- Verkehrsüberwachung, insbesondere Geschwindigkeitskontrollen durch Polizei
- Platzierung von Werbung
- Überwachung von Arbeitnehmern

Im Einzelnen handelt es sich hier neben den „identifying marks“ aus V1 um Daten, die einen Teilnehmer aufgrund ihrer Natur aus einer Anonymitätsgruppe herausheben. Dies können zum Beispiel Fahrleistungen sein, die eindeutig einem Sportwagen oder einem Schwerlasttransport zugeordnet werden können. Durch statistische Methoden und einer ausreichend großen Datenmenge können aufgezeichnete Zeit-, Positions- und Dienstnutzungsdaten und geographischen Merkmalen (Straßenverlauf, etc.) korreliert werden und engen so den Teilnehmerkreis immer weiter ein.

Die Gebundenheit der Fahrzeuge an den Straßenverlauf liefert Angreifern situationsabhängig Informationen . Auf wenig befahrenen Strecken, die zudem kaum Kreuzungen, Einmündungen, o.ä. bieten, dürfte es leichter sein, gelauschten Informationen Identitäten zuzuordnen, als z. B. an großen Autobahnkreuzen mit einer sehr viel höheren Anzahl an Fahrzeugen, die sich schnell und richtungsändernd bewegen. Die Vorteile einer größeren Anonymitätsgruppe in letzterer Situation könnte in Sicherheitskonzepten durchaus Berücksichtigung finden.

Im Gegensatz zu Anwendungen in A1 sind in allen anderen transaktionsbasierte Anwendungsgebieten, hauptsächlich A3 und auch Teile von A2, die Daten vor dem Abhören durch Dritte zu schützen.

V3 Unabhängig von den Nutzdaten sind im Rahmen der Möglichkeiten auch alle administrativ ausgetauschten Nachrichten zu schützen. Darunter fallen zum Beispiel Nachrichten des Routingprotokolls[7] und (falls eingesetzt) des Managements kryptographischer Schlüssel, Zertifikate, etc.

[7] Bereits 1999 formulierten LIDONG ZHOU und ZYGMUNT J. HAAS diese Forderung in [ZH99], S. 2.

V4 Zuletzt stellt die Sicherheit vor unbefugtem Gerätezugriff eine wichtige Säule der Vertraulichkeit dar, da Fahrzeuge und stationäre Knoten wesentlich leichter zugänglich sind also andere Geräte mobiler Ad-hoc-Netze und oft in periodischen oder für Angreifer vorhersehbaren Zeitabständen den persönlichen Sicherheitsbereich verlassen (z. B. Bau eines Fahrzeugs, Reparaturen, Kundendienste, Polizeikontrollen, Wechseln von Firmenfahrzeugen unter den Mitarbeitern eines Unternehmens, usw.).

Maßnahmen zum Erreichen von Vertraulichkeit

Anonymisierung[8] und Pseudonymisierung[9] sind wissenschaftlich und praktisch erprobte Methoden, um Beziehungen zwischen Subjekten, Nachrichten, Ereignissen, Aktionen, etc. zu verschleiern. Nach [PH04], S. 3 definiert sich Anonymität in allgemeiner Form wie folgt:

> *„Anonymity is the state of being not identifiable within a set of subjects, the anonymity set.“*

Anonymität wird also dadurch erreicht, dass ein einzelnes Subjekt in einer Gruppe von Subjekten nicht mehr zu unterscheiden ist. Konkret gewährleisten MIX- oder DC-Netze Empfänger- und Senderanonymität, broadcast-Techniken sichern zumindest Empfängeranonymität. PFITZMANN erläutert in [PH04], S. 8, zudem, dass *dummy traffic* [10] allgemein ein probates Mittel darstellt, Unbeobachtbarkeit herzustellen.

> *„Being pseudonymous is the state of using a pseudonym as ID.“ ([PH04], S. 9)*

Anders als bei der Anonymität geht beim Pseudonym der Schutz der Unverkettbarkeit verloren, zu Gunsten der Möglichkeit, sich zu authentisieren (vgl. [PH04], S.11). Da in 3.2.1 auf der nächsten Seite festgestellt wird, dass die (zumindest nachträgliche) Zurechenbarkeit von Teilnehmeraktionen Schutzziel ist, gibt es nach der Matrix der Schutzzielabhängigkeiten ([Fed05c], S. 12, vgl. Tabelle 3.1 auf der nächsten Seite) keine Möglichkeit, Anonymität zu realisieren; vielmehr wird Pseudonymität angestrebt.

Bezüglich der Inhaltsvertraulichkeit gibt es keine direkten Abhängigkeiten zur Zurechenbarkeit und Inhaltsintegrität, sie kann also weitgehend unabhängig realisiert werden. Um Vertraulichkeit auf Ebene der Daten[11] zu etablieren, müssen ihnen durch bewusste Auswahl, Transformation und eventuell durch Verschlüsselung[12] möglichst viele bzw. alle identifizierenden Merkmale entzogen werden, ohne ihren Nutzwert einzuschränken.

Dies kann auf dreierlei Arten vollzogen werden:

[8] Das ANON-Projekt: http://anon.inf.tu-dresden.de

[9] Bei GSM werden temporäre Pseudonyme, TMSIs, eingesetzt.[Fed99], S. 49 ff.

[10] *dummy traffic*: Ein Nutzer sendet ständig Daten. Wenn er keine (verschlüsselten) Nachrichten zu senden hat, sendet er Zufallszahlen, die nicht unterscheidbar sind von echten verschlüsselten Nachrichten.([Fed05a], S. 39)

[11] Dazu zählen z. B. Nachrichteninhalte, Positionsdaten und Zeitdaten.

[12] Eine Auswahl gängiger symmetrischer und asymmetrischer Verschlüsselungsverfahren und deren Performance sind in Tabelle 4.2 auf Seite 68 zu finden.

	Inhaltsintegrität	**Zurechenbarkeit**
Inhaltsvertraulichkeit	unabhängig realisierbar	weitgehend unabhängig realisierbar
Anonymität	unabhängig realisierbar	Pseudonymität
Unbeobachtbarkeit	unabhängig realisierbar	*TTP* decken ggf. Kommunikationsbeziehungen auf.

Tabelle 3.1.: Matrix der Schutzzielabhängigkeiten

- Daten werden grundsätzlich sparsam und gerade noch für den Einsatzzweck ausreichend ausgewählt.
- Daten werden nicht mit ihrem absoluten Messwert weitergeben, sondern auf eine systemweit gültige, wesentlich gröber auflösende Skala abgebildet, um damit wiederum Anonymitätsgruppen zu schaffen. Zum Beispiel ist es vorstellbar, für die Fahrzeuggeschwindigkeit eine alternative Skala einzuführen, die nach oben hin von der gerade erlaubten Höchstgeschwindigkeit begrenzt wird. Auf einer Landstraße würde ein LKW statt des Werts 80 km/h nun 8 übertragen, Fahrzeuge, die 100 km/h bzw. 120 km/h fahren, den Wert 10. Solche Maßnahmen oder das Beifügen von Unschärfe schränken aber nicht nur die Profilbildung, sondern auch die Genauigkeit der Assistenzsysteme und damit u. U. die Verkehrssicherheit ein. Deshalb muss hier im Einzelfall und in Prototypen getestet werden, inwieweit derartige Einschränkungen toleriert werden können.
- Daten werden aggregiert, wenn Details für ihre Verwendungsweise nicht entscheidend sind, z. B. um sehr weit entfernten Fahrzeugen ein Überblick über die Verkehrslage zu geben.

Weitere Überlegungen zu diesem Themengebiet werden in [SHL+05] angestellt.

Integrität

> „Integrität *bedeutet, dass Informationen richtig, vollständig und aktuell sind oder dies erkennbar nicht der Fall ist.“ (nach [Pfi00], S. 7)*

In VANETs können absichtlich gefälschte oder unverschuldet inkorrekte Daten weitreichende Folgen nach sich ziehen, das Spektrum reicht von Verwirrung und Irreführung bis zu lebensgefährlichen Verkehrssituationen. Darum kommt gerade der nachprüfbaren Korrektheit der Daten und der ausgewiesenen Knotenidentität besondere Bedeutung zu. Analog zu den Ausführungen des vorherigen Abschnitts sind daher auch auf dem Gebiet der Integrität sowohl die Kommunikationspartner, als auch die Daten zu betrachten, da an sich korrekte Daten von einem nicht als vertrauenswürdig erachteten Knoten genauso wenig vertraut werden kann wie Daten, die zwar von einem vertrauenswürdigen Sender stammen, aber absichtlich oder unabsichtlich verfälscht wurden.

I1 Die Authentizität von Sendern und Empfängern – sofern sie in den jeweiligen Nachrichten enthalten sind – muss dabei in allen Anwendungen gewährleistet sein:[13]

- Angreifer dürfen sich nicht unerlaubt als höher privilegierte Einsatzfahrzeuge oder stationäre Knoten ausgeben, um Vorteile im city- (Schalten von Ampeln) oder im highway-Szenario (Gassenbildung) zu erlangen (Anwendungsgebiet A2).
- Knoten darf es nicht möglich sein, Dienste unter der Identität eines anderen bekannten oder fiktiven Knotens in Anspruch nehmen. Unter dem Gesichtspunkt der mehrseitigen Sicherheit sind zum Beispiel in A3 die Dienstbetreiber (Mauterheber, Banken, sonstige gebührenpflichtige Dienste) hinsichtlich der Abrechnungsintegrität zu schützen, d.h. das Versenden dieser Nachrichten muss eindeutig zurechenbar und darf nicht abstreitbar sein.
- Dies gilt ebenso für Knoten, die falsche Daten in das Netz eingespielt haben. Diese müssen rechtlich korrekt zur Rechenschaft gezogen werden können (A1). Zum Beispiel könnte eine Raststätte Vorteile daraus ziehen, Meldungen über Staus oder Unfälle in ihrem Einzugsgebiet zu verbreiten. Ein nicht-rationaler Angreifer (siehe 3.3.1 auf Seite 31) könnte Störungen und eine inkonsistente Informationslage im Netz provozieren, ohne konkrete (wirtschaftliche) Vorteile daraus ziehen zu wollen.
- Wie im vorherigen Punkt könnte das erforderliche Schutzziel der Nichtabstreitbarkeit von Nachrichten auch zur Aufklärung von Straftaten, Unfällen, etc. genutzt werden, sofern
  - eine gesetzliche Grundlage geschaffen wird und diese Aufklärung nur autorisierten Personenkreisen möglich ist,
  - dies nicht gegen die Schutzziele V1-4 verstößt.

I2 Fahrzeughalter und -führer werden einem VANET-System nur dann Vertrauen entgegenbringen, wenn die Daten- und Übertragungsintegrität garantiert werden kann. Daten und Informationen sind nur dann hilfreich, wenn man davon ausgehen kann, dass diese nicht (nachträglich) manipuliert wurden. Ansonsten würden die meisten Nutzer innerhalb kurzer Zeit ihre Geräte abschalten und so den Nutzen für die verbliebenen den Nutzen minimieren. Auch schwerwiegende Folgen im Straßenverkehr sind nicht auszuschließen, sollte man sich auf Informationen stützen, die von (terroristischen) Angreifern bewusst eingespeist oder manipuliert wurden.

- Zum Einen muss sichergestellt werden, dass im Fall von A1 die Nutzdaten unverfälscht von den Fahrzeugsensoren[14] über den fahrzeuginternen Rechner zu anderen Verkehrsteilnehmern fließen.
- Zum Anderen dürfen korrekte Daten nicht mit einer inkorrekten Erhebungs- oder Sendezeit verknüpft werden[15]. Gleiches gilt für den Erhebungs- oder Sendeort.

[13] vgl. auch [RH05b], S. 5

[14] Dies ist nicht Bestandteil dieser Arbeit.

[15] Bei einer solchen Replay-Attacke spielen Angreifer gültige Datenpakete anderer Teilnehmer in hoher Zahl wieder

Maßnahmen zum Erreichen von Integrität

Völlig unabhängig von den Aspekten der Vertraulichkeit kann die Inhaltsintegrität umgesetzt werden. Es ist also keineswegs notwendig, die Identität eines Senders zu preiszugeben, um eine Nachricht als authentisch ansehen zu können.

Nach derzeitigem Wissensstand bieten digitale Signaturen (asymmetrische Kryptographie)[16] und *Message Authentication Codes* (symmetrische Kryptographie)[17] ausreichend Sicherheit, die Authentizität und Integrität einer Nachricht eindeutig zu verifizieren. Eine Auflistung verschiedener Algorithmen zum Schutz von Integrität und deren Performance spiegelt Tabelle 4.3 auf Seite 69 wider.

Kooperation stellt in VANETs nicht nur für die multihop-Kommunikation eine wichtige Säule dar, Knoten können so auch auf Datenebene die Nachrichtenqualität beurteilen, indem sie Nachrichteninhalte verschiedener Absender vergleichen, die zu ähnlichen Zeiten und Positionen geschickt wurden. Stark abweichende Nachrichten werden im *EDR* gespeichert und bei wiederholter Auffälligkeit eines Senders an die TTP gemeldet. Diese kann dann prüfen, ob auch andere Knoten den beschuldigten Knoten ebenfalls als verdächtig eingestuft haben. Als Konsequenzen kann die Instanz REV Verwarnungen, befristete oder entgültige Sperrungen durchsetzen.

Es muss aber dafür Sorge getragen werden, dass die einen Knoten andere nicht unrechtmäßig denunzieren können. Zum Einen mindern die vorher vorgestellten Maßnahmen der Pseudonymisierung die Wahrscheinlichkeit, eine konkrete Identität gezielt anzuschwärzen, da man unter der Bedingung einer korrekten Implementierung einem Pseudonym keine Identität zuordnen kann. Zum anderen werden Nachrichten in jedem Fall Integritätsmerkmale hinzugefügt, die kein angenommener Angreifer fälschen kann. Einem Angreifer ist es also nicht möglich, eine qualitativ gute Nachricht zu verfälschen und den *MAC* oder die Signatur entsprechend anzupassen.

Zusätzlich bietet die Kooperation der VANET-Teilnehmer auch die Möglichkeit, selbst über die Korrektheit der eigenen Sensordaten im Klaren zu sein; denn im Fall, dass alle umliegenden Knoten ähnlicher Position abweichende Informationen senden, kann von einer Fehlfunktion oder Manipulation der eigenen Daten ausgegangen werden.

Weitere Ansätze hierzu liefern [DKS05], [DFM05], [BSBHJ06] u. a.

Um darüber hinaus die Zurechenbarkeit bzw. Nichtabstreitbarkeit einer Nachricht zu erreichen, muss in die Signatur oder in den *MAC* zusätzlich die Dimension der Zeit einfließen[18], die von einem unabhängigen zuverlässigen Zeitgeber bereitgestellt wird. In Kapitel 2.2.4 auf Seite 9 wurde

in das Netz ein, um z. B. bei einem Server bestimmte Reaktionen auszulösen, die den Angreifern in ihrem Vorhaben zuträglich sind.

[16] Ein Teilnehmer besitzt einen privaten, geheimzuhaltenden und einen öffentlichen Schlüssel. Um eine Nachricht zu signieren, „verschlüsselt“ er sie mit seinem privaten Schlüssel. Andere Teilnehmer können die Signatur mit dem öffentlichen Schlüssel des Teilnehmers prüfen. Eine detaillierte Ausführung ist unter [Eck03], S. 373 ff., zu finden.

[17] Um einen Message Authentication Code zu prüfen, wird derselbe geheime Schlüssel benötigt, mit dieser MAC erzeugt wurde. vgl. [Eck03], S. 365 ff.

[18] Dies ist auch im Deutschen Signaturgesetz vom 22.05.2001 verankert, siehe dazu [Eck03], S. 308 ff.

Galileo als zukünftiger zuverlässiger Zeitgeber kurz vorgestellt, dessen Zeitdaten auf Authentizität geprüft werden können.

Fehlt jedoch in der Signatur oder im *MAC* die Zeitangabe, ist ein Teilnehmer in der Lage, nachträglich seinen Schlüssel zu zerstören und zu behaupten, ein Angreifer hätte ihm seinen Schlüssel gestohlen und damit die Signatur/den *MAC* erzeugt (vgl. [NDJB02], S. 75).

Zusätzlich können bei Existenz eines *EDR* (siehe Kapitel 2.2.4 auf Seite 9) wichtige Nachrichten, z. B. die *Warnungen* aus Anwendungsgebiet A1 und die Nachrichten aus A2, darin gespeichert werden. Einen rechtlich einwandfreien Rahmen vorausgesetzt, können diese manipulationssicheren Daten helfen, Straftaten und ihren Hergang aufzuklären.

Welche Methoden sinnvoll in VANETs eingesetzt werden können, hängen nicht unwesentlich von der Berechnungsgeschwindigkeit, von der Performance der Verfahren ab und stellen eine fundamentale Entscheidung im Infrastrukturdesign dar (siehe 3.2 auf Seite 19).

Verfügbarkeit

> „Verfügbarkeit *bedeutet, dass Daten und Informationen dort und dann zugänglich sind, wo und wann sie von Berechtigten gebraucht werden.“ (nach [Pfi00], S. 7)*

„Gebraucht werden“ erfährt auf dem Feld der VANETs eine völlig neue Bedeutung: Wenn Dienste und Informationen, auf die sich Fahrer gelernt haben zu verlassen, nicht verfügbar sind, kann dies nicht nur den Ruf dieses speziellen Dienstes und des gesamten VANET-Systems langhaltig schädigen, sondern bei sicherheitskritischen Anwendungen Fahrzeug und Leben eines Verkehrsteilnehmers gefährden.

R1 FEDERRATH unterscheidet beim Terminus *Verfügbarkeit* zwischen Informationssicherheit und technischer Sicherheit: In ersterem Fall, dem „Ermöglichen von Kommunikation“ stellen die Bedrohung Teilnehmer dar, die ihre Geräte abgeschaltet haben, sich egoistisch verhalten[19], jedwede Kommunikation über ihre eigene Sendeeinheit „schlucken“ [20] oder den Kanal fluten ([RH05a], S. 7). Auch die Verfügbarkeit der sichernden Maßnahmen, die von einer geeigneten Sicherheitsinfrastruktur bereitgestellt werden, können davon bedroht werden.

R2 Im zweiten Fall, der technischen Verfügbarkeit, soll gewährleistet werden, dass überhaupt Kommunikation ermöglicht wird; dieses Schutzziel kann z. B. durch Störsender, Abschalten oder Zerstören von vitalen Basisstationen, o. ä. bedroht werden.

[19] selfish nodes
[20] black-hole-Attacke: siehe [WS02], S. 4 ff.

Sensor network layers and denial-of-service defenses		
Network layer	**Attacks**	**Defenses**
Physical	Jamming	Spread-spectrum, priority messages,
		lower duty cycle, region mapping, mode change
	Tampering	Tamper-proofing, hiding
Link	Collision	Error-correcting code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network and routing	Neglect and greed	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authorization, monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client puzzles
	Desynchronization	Authentication

Tabelle 3.2.: Sensor network layers and denial-of-service defenses

Maßnahmen, Verfügbarkeit zu sichern

Verfügbarkeit zu sichern stellt im Allgemeinen eine große technische Herausforderung dar. Tabelle 3.2 aus [WS02], S. 2, zeigt mögliche Denial-of-Service-Attacken und Gegenmaßnahmen gegliedert nach ausgewählten Netzwerkschichten des ISO-OSI-Referenzmodells. In der derzeitigen Literatur, die sich mit der Sicherung der Verfügbarkeit in VANETs beschäftigt, gibt es zwar viele Untersuchungen, wie solche Ad-hoc-Netze auf DoS-Angriffe reagieren[21][22], konkrete Lösungen gestalten sich jedoch generell bei diesem Schutzziel als schwierig. In [RH05a], S. 7, wird vorgeschlagen, auf alternative Funktechnologien wie UTRA-TDD oder Bluetooth auszuweichen, wenn das Übertragungsmedium durch Angriffe nicht oder nicht ausreichend verfügbar ist.

Eine andere Möglichkeit, Verfügbarkeit zu erhalten, besteht darin, Anreize für Benutzer zu schaffen, sich korrekt und netzfördernd zu verhalten[23].

3.2.2. Performance - AN2

In VANETs kommt der Verarbeitungsgeschwindigkeit beim Senden und Empfangen von Nachrichten eine kritische Rolle zu. In den Anwendungsgebieten A1 und A2 in Kapitel 2.3 auf Seite 12

[21] [AHK] veranschaulicht die Auswirkungen von Jellyfish- und Blackhole-Attacken in einem VANET.

[22] Angriffe auf die Verfügbarkeit in Sensornetzwerken erläutert [WS02]. Eine kurze Zusammenfassung liefert Tabelle 3.2

[23] [Ref05] untersucht Möglichkeiten, solche Anreize technisch umzusetzen.

herrschen strikte Echtzeitanforderungen, Verzögerungen von einigen Sekunden sind nicht tragbar. Dabei muss folgenden Gegebenheiten begegnet werden:

- Im highway-Szenario haben Fahrzeuge trotz bis zu 1000 Metern Funkreichweite ein äußerst kurzes Zeitfenster, Nachrichten auszutauschen[24]. Natürlich verkürzt sich in diesem Szenario auch die Verbindungszeit zu Basisstationen erheblich.
- Im city-Szenario hingegen müssen u. U. eine große Anzahl von Nachrichten unterschiedlicher Herkunft[25] schnell verarbeitet werden, vor allem an Kreuzungen und anderen Straßenabschnitten, in denen Unfälle wahrscheinlicher[26] sind.

Um die in dieser Arbeit geforderten Schutzziele zu erfüllen, müssen von der Rechnereinheit zusätzlich kryptographische Operationen durchgeführt werden, die das Aufbereiten von Nachrichten zeitlich erheblich verlängern.

Die Maßnahmen der Integritätssicherung (Signatur, *MAC*) vergrößern die Nachrichtenlänge. Dieser Umstand ist in die Bewertung verschiedener Ansätze für Sicherheitsinfrastrukturen einzubeziehen.

In allen Protokollen sollten bei gegebenem Sicherheitsniveau möglichst wenig Daten verschickt werden, um den Kommunikationsoverhead gering zu halten. Aufgrund der Mobilität, der Fahrzeuggeschwindigkeit und dem Einsatz von WLAN-Technologie sind erhöhte Paketverlustraten zu erwarten, die mit umfangreichen Protokollen eher noch zunehmen.

Zuletzt ist – mit weniger Gewicht – Wert auf die Verarbeitungskapazität bzw. den anfallenden Arbeitsaufwand der zentralen Verwaltungsinstanzen der Sicherheitsinfrastruktur zu legen. Natürlich kann mit load balancing-Mechanismen und Hard- und Softwareredundanz den Leistungsanforderungen begegnet werden, jedoch steigen damit auch die Kosten für Hard-, Software und Personal.

In Kapitel 4.1.5 auf Seite 67 werden Kryptoalgorithmen gegenübergestellt und auf ihre Eignung in VANETs geprüft.

3.2.3. Wirtschaftliche Aspekte - AN3

Als letzte Komponente fließen in die Bewertung später vorgestellter Ansätze und Vorschläge die zu erwartenden Kosten der Sicherheitsinfrastruktur ein. Dies ist ein nicht zu unterschätzender Faktor, da dies sich vor allem in der Einführungszeit von VANETs deutlich auf die Ausstattungsrate von Fahrzeugen mit VANET-Technologie und damit auf den Nutzwert dieses Netzes

[24] Im Extremfall – zwei Fahrzeuge mit 250 km/h fahren in entgegengesetzte Richtungen – haben sie eine rechnerische Kontaktzeit von etwa 14 Sekunden, bei Funkreichweite von 300 Metern nur noch etwa vier Sekunden.

[25] Nachrichten von anderen Fahrzeugen, von stationären Einrichtungen, von Einsatzkräften, etc.

[26] Laut dem Statistischen Bundesamt machen „Nichtbeachten der Vorfahrt, Fehler beim Abbiegen“ und „Falsches Verhalten gegenüber Fußgängern“ den Großteil der Unfallursachen aus; vgl. http://www.destatis.de/basis/d/verk/verktab9.php

auswirken wird. In die Bewertung hinsichtlich wirtschaftlicher Aspekte werden folgende Kostenfaktoren in Betracht gezogen:

- Kosten der benötigten Fahrzeughardware (Nutzung bestehender Technologien, etc.)
- Aufwand der Registrierung neuer VANET-Teilnehmer bei den zentralen Instanzen der Infrastruktur
- Initiale Kosten beim Aufbau des stationären Netzes
- Unterhaltskosten der zentralen Instanzen

3.2.4. Herausforderungen

Die Notwendigkeit, Nachrichten vertraulich zu senden, wurde in Kapitel 3.2.1 auf Seite 20 festgestellt, als Maßnahmen wurden neben Datensparsamkeit die Verschlüsselung genannt. Diese kann jedoch nur sinnvoll angewandt werden, wenn die Schlüssel nachprüfbar ihren Besitzern zugeordnet sind. Nur unter dieser Bedingung erhalten auch die Mechanismen zur Integrationsicherung (Prüfung auf Authentizität und Unversehrtheit einer Nachricht) ein verlässliches Fundament. Die Forderung nach dieser starken Zuordnungseigenschaft ist ein großer Motivationsgrund, eine Sicherheitsinfrastruktur für VANETs einzusetzen.

Folgende Fragen sind neben der Erfüllung der geforderten Schutzziele von einer geeigneten Sicherheitsinfrastruktur zu beantworten:

1. Wer betreibt diese Infrastruktur, so dass sie die Akzeptanz aller VANET-Beteiligten genießt?
2. Wie werden folgende Aufgaben organisatorisch in der Infrastruktur verteilt und bewältigt (vgl. [NDJB02], S. 106):
   - Sichere Erstellung gültiger Schlüssel (für die Verschlüsselung und Integritätssicherung von Nachrichten)
   - Gültigkeitsprüfung der ursprünglichen Identität bei der Registrierung eines neuen VANET-Teilnehmers
   - Ausgabe, Erneuerung und Beendigung von Zertifikaten[27] (nur bei *PKI*s)
   - Gültigkeitsprüfungen von Zertifikaten (nur bei *PKI*s)
   - Verteilung von kryptographischem Material (öffentliche, private, geheime Schlüssel, Zertifikate)
   - Sichere Archivierung und sicheres Wiederfinden von Schlüsseln

[27] Zertifikate ordnen einen öffentlichen Schlüssel einer Identität nachprüfbar zu; sie werden in Kapitel 3.4.1 auf Seite 39 eingehender behandelt.

- Generierung von Signaturen/*MACs* und Zeitstempeln
- Aufbau und Verwaltung von Vertrauensstellungen

3. Welche Form haben die Identitäten[28], also die Schlüsselinhaber?
4. Wie gestaltet sich der Lebenszyklus von Identitäten und ihrem kryptographischen Material?
5. Wie kann einer (bei der VANET-Einführung) geringen Netzabdeckung mit Basisstationen begegnet werden, wenn diese ein unabdingbarer Teil der Sicherheitsinfrastruktur ist?
6. Wie wird die große Mobilität und Anzahl von potentiellen Teilnehmern[29] organisatorisch und hinsichtlich der Echtzeitanforderungen bewältigt?
7. Wie wird die Sicherheitsinfrastruktur selbst vor Angriffen geschützt?

3.3. Angreifermodelle

Schutz vor einem allmächtigen Angreifer ist weder möglich noch bezahlbar. Nachdem in den vorherigen Kapitel ermittelt wurde, was (Anwendungen, 2.3 auf Seite 12) in welcher Hinsicht (Schutzziele, 3.2.1 auf Seite 20) zu schützen ist, werden die möglichen Angreifer einer Betrachtung unterworfen. Angreifermodelle bilden die maximale Stärke eines Angreifers hinsichtlich vier Dimensionen ab und eignen sich damit sowohl als Grundlage für den Entwurf sicherer Systeme, als auch für die Bewertung von existierenden Entwürfen oder Systemen ([Pfi00], S. 13 ff.). MLETZKO stellt in seiner Diplomarbeit ([Mle05], S. 8 ff.) die Angreiferdimensionen und konkrete Angreifermodelle in VANETs ausführlich dar.

3.3.1. Die vier Dimensionen eines Angreifers

Die in diesem Abschnitt vorgestellten Dimensionen eines Angreifers spiegeln die Ausprägungen und Fähigkeiten eines potentiellen Angreifers wider, wobei Naturgewalten als Aspekte technischer Sicherheit[30] in dieser Arbeit außen vor bleiben.

[28] Die Identität von VANET-Knoten wird gesondert in Kapitel 4.1.1 auf Seite 46 erörtert.

[29] Laut den Angaben des Statistischen Bundesamtes Deutschland vom 20. Juli 2005 wurden im Jahr 2004 3266800 PKW neu zugelassen. Der Bestand erhöhte sich 2005 damit inklusive der LKWs auf 54519700 Fahrzeuge. Mit einer erfolgreichen Einführung und Akzeptanz bei der Bevölkerung ist mit einer hohen Ausstattungsrate mit VANET-Hardware zu rechnen.

[30] Dazu sind auch Funktionssicherheit, Schutz vor Überspannung und Temperaturschwankungen, Schutz vor Spannungsausfall, usw. zu zählen ([Fed05b], S.3).

Rollen eines Angreifers

Zunächst unterscheidet MLETZKO zwischen zehn verschiedenen Rollen, die ein potentieller Angreifer annehmen kann:

1. Fahrzeugführer
2. Fahrzeughalter
3. Andere Teilnehmer
4. Konstrukteure und Produzenten von VANET-Komponenten
5. Netzbetreiber
6. Dienstbetreiber
7. Wartungspersonal von Fahrzeugen
8. Konkurrenten (unter Netz- und Dienstbetreibern, Fahrzeugherstellern)
9. Exekutive
10. Außenstehende

Verbreitung eines Angreifers

Die vorgestellten Rollen eines Angreifers implizieren stark seine Verbreitung, d.h. seine Möglichkeiten, auf „Kommunikationsverbindungen, (Sub-) Systeme und Komponenten“ ([Mle05], S. 9) Einfluss (Kontrolle, Manipulation) auszuüben.

- Fahrzeuge und (Netz-) Komponenten: Außer der Exekutive, die physikalischen Zugriff im Zuge der Strafverfolgung erwirken kann, haben nur die Rollen 1, 2 und 7 Hardwarezugang.
- Funkschnittstelle: Begrenzt von den technischen Möglichkeiten der einzelnen Rollen können sie allesamt Datenübertragungen dieser Schnittstelle empfangen und eigene Nachrichten absetzen, angefangen von der Exekutive, der per Gesetz Protokollierungs- und Überwachungsmechanismen zur Verfügung stehen[31], über die Netzbetreiber, die Teile oder die Gesamtheit des Straßennetzes überwachen kann, zu allen anderen vorher genannten Rollen, deren Zugriff auf ihre lokale Reichweite beschränkt ist.

[31] Übergabepunkte im Sinne von TKÜV (in Kraft getr. am 29.01.2002) Teil 2 §8 bei Netzbetreibern und eigene Einrichtungen, sofern ein VANET als eine Telekommunikation nach TKÜV (Telekommunikations-Überwachungsverordnung) erachtet wird.

Verhalten eines Angreifers

Das Verhalten eines Angreifers hängt wiederum von seinen anderen Dimensionen ab: „Je höher Kompetenz, Ressource und Verbreitung eines Angreifers sind, desto wirkungsvoller werden aktive Kontrolle und Eingriffe.“ ([Mle05], S. 10).

Ressourcen und Kompetenzen eines Angreifers

Die Ressourcen, also Rechen- und Speicherkapazitäten, und die Kompetenzen eines Angreifers unterscheiden sich stark, teilweise auch innerhalb von Angreiferrollen.

Angreifer	Ressourcen	Kompetenzen
Fahrzeugführer, -halter, ander Teilnehmer	gering	stark schwankend
Außenstehende	gering - hoch	
Konkurrenten	gering - hoch	
Wartungspersonal	mittel	
Netz- und Dienstbetreiber	hoch	
Exekutive	sehr hoch	

Tabelle 3.3.: Ressourcen und Kompetenzen von Angreifern

3.3.2. Konkrete Angreifermodelle

Im Folgenden werden die starken, aber möglichst realistischen Angreifermodelle (AM1 - AM5) präsentiert, die MLETZKO aus den Dimensionen eines möglichen Angreifers (3.3.1 auf Seite 31) kombiniert. In jedem hier vorgestellten Modell wird dem Angreifer unterstellt, kryptographische Verfahren nicht brechen oder sichere Hardware nicht manipulieren zu können.

Angreifermodell AM1

- Rollen: Fahrzeugführer, -halter, andere Teilnehmer, Außenstehende
- Verbreitung: sehr klein, auf seine Reichweite beschränkt
- Verhalten: physische Kontrolle eigener Knoten oder Fahrzeuge, rein passiv und beobachtend
- Ressourcen und Kompetenzen: gering, kein Brechen kryptographischer Verfahren oder Manipulation sicherer Hardware

Angreifermodell AM2

- Rollen: Fahrzeugführer, -halter, andere Teilnehmer, Wartungspersonal, Konkurrenten, Außenstehende
- Verbreitung: gleich AM1
- Verhalten: aktiv und passiv: unerkannt beobachten, Komponenten ändern, Protokolle stören, Informationen aggregieren und auswerten
- Ressourcen und Kompetenzen: mittlere Kompetenzen, geringe bis mittlere Ressourcen

Angreifermodell AM3

- Rollen: Automobilhersteller, Konkurrenten, Außenstehende
- Verbreitung: gleich AM2
- Verhalten: gleich AM2
- Ressourcen und Kompetenzen: mittlere bis hohe Ressourcen, hohe Kompetenzen

Angreifermodell AM4

- Rollen: Netz- und Dienstbetreiber
- Verbreitung: physischer Zugriff auf eigene Komponenten, weitreichende Kontrolle der Netzinfrastruktur
- Verhalten: gleich AM2
- Ressourcen und Kompetenzen: mittlere bis hohe Ressourcen, hohe Kompetenzen

Angreifermodell AM5

- Rollen: Exekutive
- Verbreitung: größtmöglich, auch physischer Zugriff auf fremde Komponenten bei begründetem Verdacht
- Verhalten: flächendeckend aktiv und passiv,
- Ressourcen und Kompetenzen: hohe bis sehr hohe Ressourcen und Kompetenzen

3.3.3. Beispiele für Angriffe

In diesem Abschnitt soll zur Verdeutlichung noch eine (unvollständige) Auflistung möglicher Angriffe den vorher bestimmten Angreifermodellen zugeordnet werden:

- *eavesdropping:*
  Abhören von ungeschützer Datenübertragung (ab AM1)
- *bogus information:*
  Versenden von Falschinformationen, um Verhalten anderen Verkehrsteilnehmer zu beeinflussen. (ab AM2)
- *spoofing/masquerading:*
  Vorgeben einer anderen Identität, Hardware-Adresse, etc. (ab AM2)
- *replay attack:*
  Erneutes, oft massives Einspielen von vorher abgefangenen Nachrichten (ab AM2)
- *cheating with positioning information:*
  Fälschen der eigenen Positionsangaben, z. B. um sich der Strafverfolgung zu entziehen (ab AM2)
- *movement patterns:*
  Erstellen von Bewegungsprofilen, z. B. zu Überwachungszwecken (ab AM4)
- *denial of service:*
  Massiver Netzeingriff (Störsender, dummy messages), um Kommunikation zum Erliegen zu bringen. (ab AM3)
- *selective forwarding:*
  „dropping“ von einzelnen Paketen (ab AM2)
- *sinkhole attackes:*
  Umleiten des Netzverkehrs auf Angreiferknoten durch Einflussnahme auf die routing-Protokolle, eventuell „dropping“ der Nachrichten (ab AM3)
- *sybil attacks:*
  Simulieren von mehreren Teilnehmerknoten durch einen einzigen Angreifer, z. B. um den Mechanismus der Teilnehmerkooperation (vgl. Kapitel 3.2.1 auf Seite 23) zu untergraben (ab AM3)
- *wormholes:*
  Umleiten des Netzverkehrs von einem Teil des Netzes zu einem anderen unter der Kooperation von zwei Angreifern (ab AM3)

(aus [PM04], S. 5 f., [SE04], S. 1 ff., [RH05a], S. 1 ff.)

Weitere Analysen von möglichen Angriffen skizziert [ABD+05] mit Hilfe von Angriffsbäumen.

3.4. Arten von Sicherheitsinfrastrukturen

Den Kern dieser Arbeit – Kapitel 4.1 auf Seite 45, 4.3 auf Seite 81 und 4.2 auf Seite 70 – nehmen die Vorstellung, Bewertung und Weiterentwicklung von Sicherheitsinfrastrukturen für VANETs ein. Diese basieren zum Einen auf klassischen *Public Key Infrastructures* und deren Anpassung an die Eigenschaften von VANETs, zum Anderen auf neuen Ansätzen, die von vornherein den speziellen Charakter dieser automobilen Netze berücksichtigen. In diesem Kapitel werden nun diese zugrunde liegenden Konzepte, deren Probleme und mögliche Umsetzungsschwierigkeiten in der notwendigen Breite erörtert.

3.4.1. *Public Key Infrastructures*

Asymmetrische Kryptographie leistet – isoliert betrachtet – die in VANETs wie auch anderen Anwendungsbereichen[32] erwünschte Funktionalität des Verschlüsselns und Signierens von Nachrichten. Ohne weitere Maßnahmen und Einrichtungen ist es jedoch nicht möglich, folgende Probleme und Aufgaben zu bewältigen (vgl. [Sch01], S. 280 f., [NDJB02], S. 90 ff.):

- Ein öffentlicher Schlüssel trägt keinerlei Authentizitätsmerkmale, d.h. es ist nicht möglich, ihn zweifelsfrei einem Besitzer zuzuordnen. Dies ermöglicht man-in-the-middle-Attacken der Form, dass Angreifer AM2-5 die Übertragung eines öffentlichen Schlüssels (von einer Schlüsseldatenbank oder dem rechtmäßigen Besitzer) abfangen und ihn gegen den eigenen austauschen können. Werden nun Nachrichten mit diesem Schlüssel verschlüsselt, kann der Angreifer die Nachricht mit dem passenden privaten Schlüssel entschlüsseln und für den eigentlichen Adressaten transparent wieder verschlüsseln.

- Ein Schlüsselpaar (ein öffentlicher und ein dazu passender privater Schlüssel) besitzt keinerlei Gültigkeitsmerkmale. Wird ein privater Schlüssel von einem Angreifer gestohlen oder auf andere Weise ermittelt, gibt es keine Möglichkeit, dieses kompromittierte Schlüsselpaar derart zu markieren (sperren), dass davon erzeugte Signaturen und Verschlüsselungen sofort als Werk eines Angreifers und damit als ungültig erkannt werden.

- Es ist nicht beweisbar, wem ein Schlüssel gehört. Ein Angreifer kann z. B. abstreiten, eine vorliegende Signatur erstellt zu haben, und zwar mit der Begründung, der verwendete Schlüssel sei nicht seiner.

- In einem großem System, wie ein VANET es darstellt, sind zentrale Forderungen und Regeln[33] schwierig in konsistenter Weise durchzusetzen.

[32] Prominentester Vertreter dieser Anwendungsbereiche ist derzeit sicherlich das Verschlüsseln und Signieren von E-Mails.

[33] Dazu zählen z. B. vorgeschriebene (Mindest-)Schlüssellängen, periodischer Schlüsselwechsel, zentrale Registrierung von öffentlichen Schlüsseln, automatische Sperrung von ausscheidenden Mitgliedern, etc.

Wie wir sehen, muss durch einen zusätzlichen Mechanismus in allen Netzteilnehmern gleichermaßen das Vertrauen in die Zuordnung Schlüssel–Besitzer geschaffen werden. Im Gegensatz zum *web of trust* (siehe Kapitel 3.4.2 auf Seite 41) leistet dies im Fall der *PKI* eine unabhängige, dritte Instanz, die das Vertrauen aller Teilnehmer genießt (*trust center*).

Komponenten einer *PKI*

Dieses *trust center* als zentrale Verwaltungsinstanz besteht gemäß [NDJB02], S. 111 ff., und [Sch01], S. 298 ff., aus folgenden zentralen Komponenten:

- **Die Zertifizierungsinstanz CA**.

  Als zentraler, vertrauenswürdiger Instanz obliegt es der CA, Identitäten zu zertifizieren. Dazu werden von ihr alle Informationen eines Zertifikats entgegengenommen und mit ihrem privaten Schlüssel digital signiert. Das Schlüsselpaar des Antragstellers kann dabei von ihm selbst, von der CA oder von einem Schlüsselgenerierungsserver GEN erzeugt werden[34]. Die CA muss allerdings bereits eine vertrauenswürdige Stellung genießen, die sie durch Tradition (z. B. Banken, Kreditinstitute, Post, staatliche Behörden, etc.) oder durch Prestige erlangt hat. Der Umstand, dass die CA Zertifikate signiert, reicht nicht aus, das Vertrauen in diese Zertifikate zu schüren. So ist es von höchster Notwendigkeit, den passenden Betreiber einer CA zu ermitteln (siehe Kapitel 4.1.4 auf Seite 61).

  Diese muss zudem höchste Sicherheitsanforderungen erfüllen; sie stellt für einen Angreifer (mindestens AM3) einer *PKI* das primäre Ziel dar: Wenn es ihm gelingt, den privaten Schlüssel der CA zu stehlen oder anderweitig herauszufinden, müssen alle herausgegebenen Zertifikate ab diesem Zeitpunkt als ungültig betrachtet werden, da der Angreifer nun selbst in der Lage ist, Zertifikate ununterscheidbar von den „echten“ zu erstellen.

  Dementsprechend hoch sind die Sicherheitsvorkehrungen zu treffen: Zugriffs- und Zugangskontrollmaßnahmen, bauliche Maßnahmen, Abschottung vom Internet, Evaluierung der Software nach ITSEC[35] etc.[36]

  In großen *PKIs* ist es sinnvoll, nicht eine einzige CA, sondern eine CA-Hierarchie einzusetzen. Eine Zertifizierungsinstanz kann nämlich nicht nur die Identität von Benutzern, sondern auch die anderer CAs bestätigen. Die erste, oberste CA, die Wurzelinstanz, stellt dabei nur Zertifikate für die untergeordneten CAs aus und kann dabei mit positiver Wirkung auf die Sicherheit der *PKI* meist offline bleiben (vgl. [NDJB02], S. 140 ff., Abbildung 3.1 auf der nächsten Seite).

[34] Die Generierung kann dabei in Soft- (z. B. Webbrowser) oder Hardware (z. B. SmartCards) erfolgen.

[35] Information Technology Security Evaluation Criteria: „Die ITSEC-Kriterien legen analog zu den deutschen IT-Kriterien Funktionsklassen mit Sicherheitsanforderungen für spezifische Anwendungsklassen fest.“ ([Eck03], S. 171)

[36] Dies ist im Signaturgesetz und den dazugehörenden Bestimmungen verankert, vgl. [Eck03], S. 310 ff.

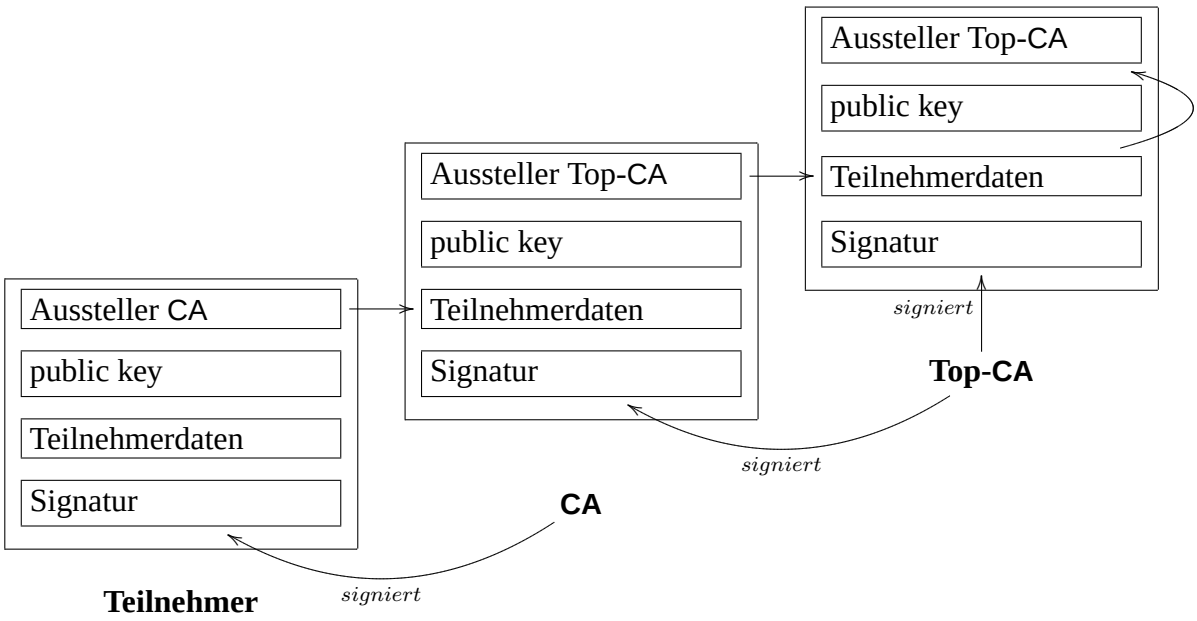

Abbildung 3.1.: Eine Top-CA signiert das Zertifikat einer CA, die wiederum ein Teilnehmerzertifikat ausstellt: Eine Zertifikathierarchie entsteht.

- **Die Registrierungsinstanz RA**. Die RA ist die Anlaufstelle für einen Zertifikatsantrag, sie leitet nach eingehender Prüfung die Daten eines Bewerbers an die CA weiter, die schließlich durch ihre Signatur das Zertifikat generiert. Die Registrierungsinstanz nimmt in ihrer vorgelagerten Stellung Last von der Zertifizierungsinstanz und erhöht zugleich die Sicherheit dieser, da sie Bewerber von der CA abschirmt.

  Die RA kann dabei unterschiedlich implementiert sein: Die Datenprüfung kann sowohl rein maschinell als auch manuell (i. d. R. höhere Sicherheit) geschehen. Sie kann Teil der CA sein, aber auch eine eigenständige Komponente darstellen. Bei größeren Netzen, wie sie VANETs darstellen, sind mehrere lokale Registrierungsinstanzen (LRAs) von Vorteil.

- **Der Zertifikatsspeicher DIR**.

  Im Gegensatz zu den privaten Schlüsseln müssen die public keys für alle Netzteilnehmer jederzeit abrufbar sein. Üblicherweise werden für die Speicherung dieser Zertifikate Verzeichnisdienste wie LDAP[37] eingesetzt, die auch die gezielte Suche nach Schlüsseln erlauben.

- **Die Sperrinstanz REV**.

  In engem Kontakt mit oben genannten Zertifikate-Server steht die REV, die im Auftrag des Schlüsselinhabers oder einer administrativen Funktion ein Schlüsselpaar als ungültig kennzeichnet und damit sperrt. Dies wird z. B. dann nötig, wenn der private Schlüssel dieser

[37] Lightweight Directory Access Protocol

Identität oder gar der CA von Angreifern kompromittiert worden ist, sich Zertifikatinhalte ändern oder die Identität die *PKI* nicht mehr nutzt.

- **Der Zeitstempeldienst TSS**.

  Wie schon in 3.2.1 auf Seite 26 erläutert, benötigt das Schutzziel der Nichtabstreitbarkeit (I2) eine verlässliche, überprüfbare Zeitangabe als Bestandteil einer Signatur. Solche digital signierte Zeitstempel liefert der TSS, der wiederum vom Signaturgesetz als Bestandteil eines *trust centers* ausdrücklich vorgeschrieben ist.

- **Die Schlüsselwiederherstellungsinstanz REC**.

  Gehen private Schlüssel verloren oder müssen sie auf Druck von Gesetzeshütern hin herausgegeben werden, ist das die Aufgabe der REC. Fällt diese Datenbank jedoch Angreifern in die Hände, ist die Sicherheit der gesamten *PKI* nicht mehr gegeben; deshalb muss diese Instanz kritisch beäugt werden.

- **Endeinheit EE**.

  Schließlich stellen Endeinheiten – in unserem Fall die VANET-Teilnehmer – die Anwender einer *PKI* dar.

Zertifikate und ihr Management

> *„Ein digitales Zertifikat ist eine überprüfbare Verknüpfung zwischen einer Identität und dem public-/private-Schlüsselpaar, das sich im Besitz des Inhabers der Identität befindet.“ ([NDJB02], S. 96)*

Diese recht allgemein gehaltene Definition eines digitalen Zertifikats benötigt eine Standardisierung, um die Kommunikation zwischen den EEs untereinander und zum *trust center* genau festzulegen. Der etablierte Standard X.509 konnte erst in seiner dritten Auflage anfängliche Mängel beheben. Wie Tabelle 3.4 auf der nächsten Seite zu entnehmen ist, werden in einem Zertifikat neben eindeutigen Teilnehmerinformationen und seinem öffentlichen Schlüssel Informationen über die Zertifizierungsstelle eingebettet. Durch die enthaltene Signatur der ausgewiesenen Zertifizierungstelle wird die Zuordnung Teilnehmer – öffentlicher Schlüssel für alle anderen Teilnehmer beweisbar.

X.509v3 lässt zudem Erweiterungen zu, mit denen anwendungsabhängig weitere Informationen im Zertifikat gespeichert werden können; dies kann in Anbetracht der verschiedenen Rollen und Identitäten eines VANETs (siehe Kapitel 4.1.1 auf Seite 46) von Vorteil sein.

In X.509v3 sind Standarderweiterungen festgehalten, die nähere Informationen zu den bereits vorhandenen Zertifikatsdaten zur Verfügung stellen sollen, z. B. Einsatzzweck des öffentlichen Schlüssels (Verschlüsseln, Signieren, etc.), Richtlinienerweiterungen, erweiterte Teilnehmer- und Ausstellerinformationen, etc.

Versionsnummer	des Zertifikat-Standards: z. B. 2 für X.509v3, 1 für X.509v2
Seriennummer	eindeutiges Identifikationsmerkmal eines Zertifikats
Signatur	Hash-Funktion, Signaturalgorithmus und die Signatur
Aussteller	identifiziert die signierende Zertifizierungsstelle (CA)
Gültigkeitsdauer	Zeitraum, in dem das Zertifikat gültig ist
Teilnehmer	Name des Teilnehmers, der mit dem public key verknüpft wird
public key-Informationen	public key des Teilnehmers, einzusetzender Algorithmus
Kennung des Ausstellers	eindeutiges Identifikationsmerkmal des Ausstellers
Kennung des Teilnehmers	eindeutiges Identifikationsmerkmal des Teilnehmers
Erweiterungen	Möglichkeit zur Einbettung zusätzlicher Informationen

Tabelle 3.4.: Das generische Schema des X.509-Zertifikats

Jedes Erweiterungsfeld muss zusätzlich als kritisch oder unkritisch beschriftet werden. Im Gegensatz zu einem unkritischen Feld muss ein kritisches Feld bei der Zertifikatsprüfung bearbeitet werden. Ist dies nicht möglich oder wird diese Erweiterung nicht erkannt, wird das Zertifikat als ungültig betrachtet.

In manchen Fällen mag es sinnvoll sein, teilnehmerbezogene Attribute, Rechte oder Privilegien nicht direkt in Erweiterungsfeldern zu speichern, sondern diese in eine separate Datenstruktur auszulagern. Diese werden Attribut-Zertifikate genannt und gleichen in ihrem Aufbau – bis auf das Fehlen des öffentlichen Schlüssels – herkömmlichen (Schlüssel-)Zertifikaten. (vgl. [NDJB02], S. 493 ff., [Sch01], S. 316 f.)

Der Lebenslauf von Zertifikaten umfasst i. A. folgende Stationen:

1. Der Begriff *enrollment* bezeichnet die Erstanmeldung eines Teilnehmers, verbunden mit der Generierung eines Zertifikats. In diesem Schritt nimmt die initiale Feststellung der Identität des Antragstellers durch die RA eine zentrale Stellung ein, der durchaus unterschiedlich implementiert sein kann. Der neue Teilnehmer kann sich zum Einen persönlich und durch Dokumente wie Personalausweis, o. ä.[38], zum Anderen über ein Geheimnis (Passwort, o. ä.), das vorab persönlich oder auf dem Postweg übermittelt wurde, authentifizieren. Der nächste Schritt umfasst die Erzeugung eines Schlüsselpaares unter Einhaltung der zentralen Schutzziele und ggf. dessen sichere Übermittlung an den Teilnehmer. Hier gibt es mehrere Varianten, die in Kapitel 4.1.4 auf Seite 61 in Hinblick auf ihren Einsatz in VANETs diskutiert werden.

   (vgl. [Sch01], S. 308 f.)

2. Sind das Schlüsselpaar oder das Zertifikat selbst am Ende ihrer Lebensdauer angelangt, müssen sie erneuert werden. In diesem Fall kann die Identität auf einfacheren Weg als

[38] Diese Variante schreibt das Signaturgesetz vor.

beim *enrollment* geprüft werden: Es genügt eine Signatur mit dem alten Schlüssel, um die *Zertifikatserneuerung* durchzuführen. (vgl. [NDJB02], S. 198 ff.)

3. Wie bereits angesprochen, haben Zertifikate eine begrenzte Lebensdauer, die durch eine bekannt gewordene Kompromittierung des privaten Schlüssels abrupt verkürzt wird. Dann nämlich muss dieses Zertifikat auf schnellstem Wege gesperrt (*revocation*) werden und dieser Umstand allen Teilnehmern bekannt gemacht werden. Auch hier bieten sich mehrere – allgemeine wie spezielle für Ad-hoc-Netzwerke angepasste – Möglichkeiten an, die im Kapitel 4.1.3 auf Seite 54 erörtert werden.

3.4.2. Andere Sicherheitsinfrastrukturen

web of trust

Das *web of trust* stellt die dezentrale Variante einer *PKI* dar. Die Authentizität der digitalen Schlüssel wird nicht durch eine zentrale Certification Authority beglaubigt (Zertifikate), sondern durch ein Netz gegenseitiger Bestätigungen (Signaturen) gesichert.[39] Zentrale keyserver halten öffentliche Schlüssel vor, die zuvor von ihrem Inhaber selbst signiert und hochgeladen wurden. Über Suchfunktionen können Kommunikationspartner Schlüssel herunterladen. Zu diesem Zeitpunkt ist durch die Selbstsignatur nur gewiss, dass der angebliche Schlüsselbesitzer auch über den privaten Gegenpart verfügt.

Es wird also ein sicherer Kanal (z. B. key-signing-Parties) benötigt, mit dem sich die Kommunikationspartner der Relation Schlüssel – Inhaber vergewissern.

Die Schlüsselspeicherung erfolgt in *public key rings* und in *private key rings*. In ersterem trägt jeder Benutzer den Grad des Vertrauens für jeden fremden öffentlichen Schlüssels ein, zwischen „unknown“ und „ultimate“ . Es liegt also in der Hand des Benutzers, die Vertrauenswürdigkeit anderer Teilnehmer einzustufen – eine sehr kostengünstige Lösung. Die Nachteile bestehen im Fehlen jeglicher juristischen Bindung – es handelt sich nicht um eine qualifizierte elektronische Signatur ([Eck03], S. 308 ff.) – und der geringen Aktualität der Sperrinformationen.[40]

Verfahren basierend auf symmetrischer Kryptographie

Obwohl Verfahren, die auf asymmetrischer Kryptographie beruhen, zunächst viele der Anforderungen (Kapitel 3.2 auf Seite 19) zu erfüllen scheinen, schlagen symmetrische Verfahren sowohl bei den vertrauens- als auch integritätssichernden Maßnahmen ihre asymmetrischen Pendants um Längen in der Berechnungszeit (siehe Abschnitt 3.2.2 auf Seite 28) und der benötigen Schlüssellänge[41].

[39] PGP (Pretty Good Privacy, http://www.pgp.com) und GnuPG (Gnu Privacy Guard, http://www.gnupg.org sind wohl die bekanntesten Anwendungen im Bereich des *web of trust*.

[40] Weitere Informationen sind unter http://de.wikipedia.org/wiki/Web_of_trust zu finden.

[41] Um in asymmetrischer Kryptographie das gleiche Sicherheitsniveau zu erreichen, muss in der Regel die Länge der verwendeten Schlüssel wesentlich größer ausfallen, Details in Kapitel 4.1.5 auf Seite 67.

Aufgrund der geheimen Natur der Schlüssel und der Möglichkeit der Zurechenbarkeit und des Schlüsselrückrufs kann hier aber nicht auf eine *Trusted Third Party* verzichtet werden: Im Allgemeinen wird sie in diesem Fall *KDC – Key Distribution Center* genannt, da sie notwendigerweise die Organisation des Schlüsselaustauschs oder -erzeugens übernimmt.

In manchen Modellen kann diese *TTP* nur bei der Netzinitialisierung erforderlich sein und im regulären Betrieb durchaus in den Hintergrund treten. Diese zentrale Instanz ist meist jedoch das einzige Instrument, mit dem das Problem der Verteilung symmetrischer Schlüssel in großen Netzen gelöst werden kann: Bei n Knoten müssten $\frac{n*(n-1)}{2}$ im Netz verteilt werden, bei 54 Millionen Knoten allein in Deutschland sind dies ungefähr $1,458 * 10^{15}$ Schlüssel (vgl. [MVO96], S. 546).

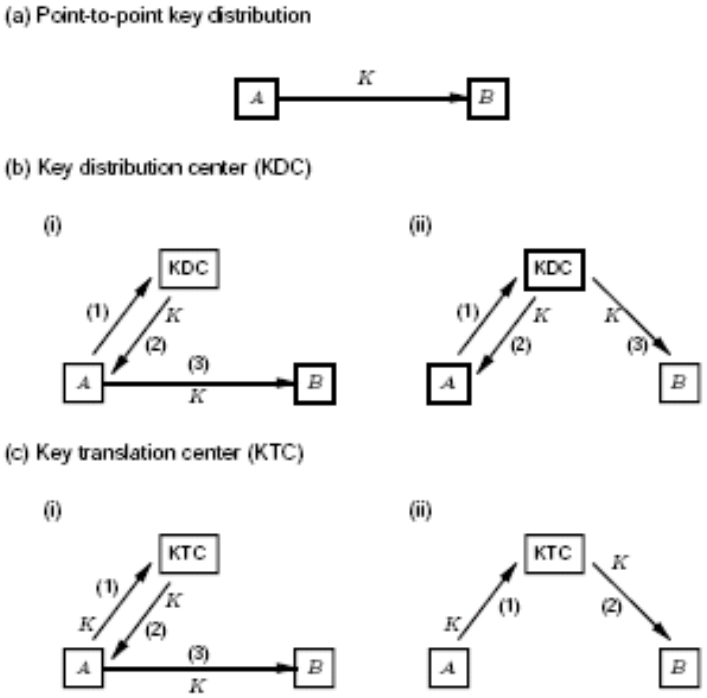

Abbildung 3.2.: a) zeigt den Austausch zwischen zwei Kommunikationspartner ohne *TTP*, bei b) gibt das *KDC* den zu verwendenden gemeinsamen, aber geheimen Schlüssel vor und bei c) schlägt ein Partner diesen vor.

Mit den Elementen

- *Message Authentication Codes* zur Integritätssicherung
- Verschlüsselungsalgorithmen zum Schutz von Vertraulichkeit
- Schlüsselverteilungs- und -erzeugungsprotokolle[42] (siehe Abbildung 3.2)

[42][MVO96], S. 497 ff., erläutert einige dieser Verfahren, u. a. *Kerberos*, *Needham-Schroeder*, *Otway-Rees*.

können durchaus die Vorteile von asymmetrischer Kryptographie simuliert werden. Ob und wie man diese Bausteine zu einem stimmigen Bild zusammenfügen kann, wird in Kapitel 4.3.1 auf Seite 82 gezeigt.

3.4.3. Problemfelder und Fazit

Alle vorgestellten Modelle besitzen in bestimmten Teilen ihres Konzeptes Vorteile für den Einsatz in VANETs. *PKIs* profitieren stark von hervorstechenden Eigenschaften asymmetrischer Kryptographie. Die sachlichen Anforderungen, die in diesem Kapitel betrachtet wurden, erreichen sie auf elegante Weise, bezahlen dies aber mit größeren Schlüsseln, u. U. längeren Nachrichten und hohen Berechnungszeiten. Weiterhin muss geklärt werden, auf welche Weise und für wen Zertifikate ausgestellt werden, wer die Infrastruktur betreibt und wie der Lebenszyklus von VANET-Knoten organisatorisch begleitet wird.

Die letzten Punkte betreffen natürlich auch die Infrastrukturen, die ausschließlich symmetrische Kryptographie einsetzen. Deren Eigenschaften verhalten sich komplementär zu denen der *PKIs*: Sie profitieren sehr von der Berechnungsgeschwindigkeit ihrer Kryptoalgorithmen, die Fragen nach der eindeutigen Zurechenbarkeit von Nachrichten und das Problem der permanenten Abhängigkeit von einer stationären *TTP* bleiben in diesem Teil der Arbeit noch unbeantwortet.

Das Ziel, kompromittierte Schlüssel schnell und effizient zurückzurufen oder gar böswillige oder eigensinnige Knoten vom VANET auszuschließen, verlangt von allen betrachteten Infrastrukturen gewisse Anpassungen.

webs of trust kommen dem Ad-hoc-Charakter von VANETs sehr nahe, aber bereits zu diesem Zeitpunkt muss angezweifelt werden, ob eine geeignete Infrastruktur tatsächlich ohne zentrale und stationäre Komponenten auskommen kann.

KAPITEL 4

Sicherheitsinfrastrukturen

In diesem zentralen Kapitel werden zunächst unter Berücksichtigung der ermittelten Anforderungen und der geschaffenen Grundlagen die Basiskonzepte einer geeigneten Sicherheitsinfrastruktur für VANETs erarbeitet.

Nachdem bestehende Vorschläge der Forschungsliteratur kritisch untersucht wurden, werden die Basiskonzepte in ausgewählte Verfahren eingebettet. Abschließend werden zwei dieser Verfahren gegenübergestellt, um daraus eine Empfehlung zu abzuleiten.

4.1. Basiskonzepte

Noch unabhängig davon, ob feste Basisstationen (hybrid) vorausgesetzt werden können und ob symmetrischer oder asymmetrischer Kryptographie der Vorzug zu geben ist, diskutiert dieses Kapitel die Kernelemente einer Sicherheitsinfrastruktur. Wenn möglich, werden also für beide „Welten“ Vorschläge erarbeitet. Im Fall einer *PKI* sind dabei bereits in diesem Kapitel konkrete Aussagen möglich, da deren Funktionsweise und Aufbau bekannt sind. Bei Verfahren basierend auf symmetrischer Kryptographie ist dies nicht immer möglich.

Der erste Abschnitt widmet sich den Identitäten und Rollen in einem VANET und wie sie in einer Sicherheitsinfrastruktur repräsentiert werden. Daraufhin stellt sich die Frage, wie solche Identitäten so anonym wie möglich gestaltet werden können, ohne das Schutzziel der Zurechenbarkeit zu beeinträchtigen. Im dritten Abschnitt wird schon etwas praxisorientierter der Lebenslauf einer VANET-Identität – vom Bau eines Fahrzeugs zum Ausschluss desselben – behandelt. Wer die dafür notwendigen Einrichtungen betreibt, ist Thema des nächsten Abschnitts. Für das Basiselement *Kryptographie* wurde in Java eine Performance-Messung von gängigen Verschlüsselungs- und Signaturalgorithmen implementiert, die zusammen mit entsprechender Forschungsliteratur diesbezügliche Vorschläge erlauben. In einem tabellarischen Fazit werden alle Basiskonzepte übersichtlich zusammengefasst.

4.1.1. Identitäten und Rollen

Wie bereits in Kapitel 3.2.1 auf Seite 24 festgestellt, ist die Anonymität der VANET-Teilnehmer nicht mit dem Schutzziel der Nichtabstreitbarkeit (I2) vereinbar. Pseudonyme schränken die geforderten Schutzziele jedoch nicht ein und werden anstelle der wirklichen Identitäten verwendet (vgl. Tabelle 3.1 auf Seite 24). Eine Identität liefert wie in anderen Netzen die Grundlage für jegliche Authentifizierung, sie stellt ein gewisses Wiedererkennungsmerkmal dar, anhand dessen man z. B. korrekt funktionierende bzw. kooperative Teilnehmer zum VANET zulassen und fehlerhafte bzw. böswillige ausschließen kann. Das impliziert natürlich, dass ein Knoten seine Identität nicht beliebig oder wenigstens nur mit extrem hohen Aufwand ändern kann; sonst greifen sämtliche Maßnahmen der Regulierung ins Leere:

> *In VANETs beinhaltet eine Identität ein oder mehrere unabänderliche Merkmale eines Knotens, die ihn eindeutig charakterisieren und unterscheidbar von anderen machen.*

Die Natur einer Identität in automobilen Ad-hoc-Netzen ist zunächst nicht eindeutig festgelegt. Eine VANET-Identität kann nämlich Identitätsmerkmale des Fahrzeugs, des aktuellen Fahrers oder Halters oder von beiden zusammen beinhalten.[1]

Rein fahrzeugbezogene Identität

In VANETs treten neben eventuell personenbezogenen Daten (A3) jedoch in Masse fahrzeugbezogene Daten auf (A1, A2). Zudem ist es durchaus möglich, dass der aktuelle Fahrer nicht für Falschmeldungen verantwortlich ist, sondern sie von einem Defekt des Fahrzeugs oder von Manipulationen (AM2,3,5) herrühren. Hier wäre es also nicht zielführend, einen bestimmten Fahrer zu verfolgen, sondern das Fahrzeug; vor allem, wenn es sich um Firmenfahrzeuge handelt, deren Fahrer oft wechseln.

Denkt man an die Verfolgung gestohlener oder in Straftaten verwickelter Fahrzeuge, so muss man mit Berücksichtigung vorheriger Argumente die Identitätsmerkmale des Fahrzeugs als zwingend notwendigen Bestandteil einer VANET-Identität[2] betrachten. Dies entspricht in digitaler Form der gegenwärtigen Situation: Ein Nummernschild pseudonymisiert den Halter eines Fahrzeugs, der Fahrer kann zudem nicht mit Sicherheit bestimmt werden.

Diese Metapher verwendet auch HUBAUX in [HCL04], S.51 f., und nennt die fahrzeugbezogene Identität „electronic license plate“ – elektronisches Nummernschild.

[1] Definitionen der Begriffe Identität und Identifikator liefert [KSW05], S. 2

[2] Ein Zertifikat würde zum Beispiel in diesem Fall Daten des Nummernschilds, die Fahrgestellnummer, o.ä. beinhalten.

Rein personenbezogene Identität

Man könnte argumentieren, dass der gestiegenen Einflussmöglichkeit[3] eines jeden Fahrzeugführers auch die Verfolgbarkeit und Zurechenbarkeit seiner Aktionen angepasst werden muss. Personenbezogene Identitäten verfolgen also hauptsächlich das Schutzziel der Nichtabstreitbarkeit (I2).

Folgt man diesen Überlegungen und den ermittelten Schutzzielen, so genügt es, den jeweiligen Fahrer als Identität festzulegen, da er der Urheber der Nachrichten ist. Ein Dokument (z. B. ein Zertifikat) oder eine Institution, die in vertrauenswürdiger Weise Identitäten Schlüssel zur Signatur- oder MAC-Bildung zuordnet, würde hier also Personenpseudonyme speichern.

Dieser Ansatz erleichtert die Rekonstruktion von Unfall- und Fahrerflucht-Situationen; bisher kann ein Unfallfahrzeug und damit der Halter sehr wohl, der Fahrer aber nicht mit Sicherheit bestimmt werden, wenn er sich dem Tatort entzogen hat.[4]

Dem steht jedoch die aktuelle Gesetzgebung gegenüber, die grundsätzlich den Fahrzeughalter (§7 StVG) haftbar macht[5]. Die Information des Fahrzeughalters leisten ohne großen Aufwand auch rein fahrzeugbezogene Identitäten, da die Exekutive bereits heute die Halter über Dokumente wie Fahrzeugschein und -brief oder über ihre zentralen Speicher ermittelt. Daher ist es sinnvoll, nur für diejenigen Teilnehmer eines VANETs personenbezogene Dokumente zu verwenden, die über erhöhte Privilegien im Straßenverkehr verfügen. Das Übertragen des Attributzertifikats, das im Fall normaler VANET-Teilnehmer keine nützlichen Informationen trägt, kann somit entfallen. Die unterschiedlichen Rollen, die ein VANET-Teilnehmer einnehmen kann, werden im Folgenden aufgelistet:

- R-PKW: PKWs
- R-LKW: LKWs
- R-SCHIENE: schienengebundene Fahrzeuge (Züge, Straßenbahn, etc.)
- R-POLIZEI: Einsatzfahrzeuge der Polizei
- R-TECH: Einsatzfahrzeuge von Feuerwehr, THW, etc.
- R-WARTUNG: autorisiertes Wartungspersonal in Werkstätten
- R-ZEICHEN: Verkehrszeichen

[3] Durch Übermittlung bestimmter Informationen kann ein Knoten erheblichen Einfluss auf das Verhalten anderer Verkehrsteilnehmer ausüben.

[4] In Unternehmen und Familien ist der Halter nicht unbedingt Fahrer eines Fahrzeuges, vielmehr kommen u. U. eine große Anzahl von Fahrzeugführern in Frage.

[5] Der Fahrzeugführer ist ebenfalls ersatzpflichtig (§18 StVG).

Diese können sehr schnell untereinander wechseln, man denke an zivile Polizeifahrzeuge, die zu einem Einsatz gerufen werden. Sie geben sich im VANET zunächst als normale PKWs (R-PKW) aus, nehmen aber dann die Rolle R-POLIZEI an. Ein LKW-Fahrer wechselt nach getaner Arbeit von R-LKW nach R-PKW.

In einem *PKI*-Szenario ist es allerdings nicht sinnvoll, für jede mögliche Rolle ein Schlüsselzertifikat zu erstellen. Dies leisten zusätzliche Attributzertifikate wesentlich besser. Sie beinhalten keinerlei Schlüssel und zertifizieren nur Eigenschaften oder Privilegien. Bei R-PKW und bei Fahrten, in denen Fahrzeuge von Polizei, Feuerwehr, etc. nicht im Einsatz sind, ist es – wie weiter oben schon angedeutet – nicht nötig, die Attributzertifikate zu übertragen und damit überhaupt erst auszustellen. Diese Zertifikate kommen also nur dann zum Einsatz, wenn die Situation oder die Anwendung die enthaltenen Privilegien explizit fordern. Da sich Teilnehmer der Rollen R-SCHIENE und R-ZEICHEN immer im „Einsatz“ befinden, müssen sie stets beigefügt werden.

Kommt keine *PKI* zum Einsatz, könnte der Besitz eines geheimen Schlüssels oder eines anderen Geheimnisses ein Privileg nachweisen[6].

In diesem Ansatz ist es bisher nicht möglich, bei Bedarf die Fahreridentität festzustellen, da diesbezügliche Daten in den Nachrichten nicht enthalten sind. Solch ein Übertragen von personenbezogenen Daten entspräche nicht dem Schutzziel der Vertrauchlichkeit. Setzt man jedoch einen *EDR* voraus, können hierin die Fahrer zu den Fahrzeiten festgehalten werden, um im Nachhinein Verfolgbarkeit zu realisieren.[7]

Die Vielzahl möglicher Fahrer wirft die Frage auf, wo diese personenbezogenene Identität gespeichert wird. Eine Vorinstallation auf dem Fahrzeug selbst scheidet aus, da man nicht vorhersehen kann, welche Personen ein Automobil benutzen werden. Als weitere Variante eignen sich auch Fahrzeugschlüssel nicht: Zum Einen kann aus Kostengründen nicht für jeden Fahrer ein eigener Schlüssel vorausgesetzt werden, zum Anderen läge die Verwaltung und Speicherung des kryptographischen Materials ohne Ausweichmöglichkeit in den Händen der Automobilhersteller.

Elektronische Führerscheine hingegen bieten sich an: Jeder Fahrer muss einen gültigen besitzen und ihn bei Bedarf nachweisen[8]. Dies bedeutet kaum einen Komfortverlust für die VANET-Benutzer, eröffnet aber zugleich die Möglichkeit, das Fahren ohne Führerschein in gewisser Weise einzudämmen, indem das Fahrzeug ohne gültigen Führerschein nicht gestartet werden kann, die Fahrt sinnvoll einschränkt, o.ä.

Eine solche Funktionalität bringt mit Sicherheit Synergieeffekte in Bezug auf die Neuregelung der Lenk- und Ruhezeiten (VO 3820/85) mit sich, deren Umsetzung für das Jahr 2007 erwartet

[6] In Systemen, die ausschließlich symmetrische Kryptographie einsetzen, ist in diesem Schritt eine vertrauenswürdige Instanz notwendig. Beim Prüfen des Nachweises eines Privilegs ist in diesem Fall nämlich der verwendete Schlüssel erforderlich, der geheimzuhalten ist und nicht übermittelt werden darf. Die *Trusted Third Party* mimt dabei den Mittler zwischen den beiden Parteien. Ansonsten hätte auch der Prüfer des Nachweises nun die Möglichkeit, fortan dieses Privileg zu genießen.

[7] In den Nachrichten sollte ein Zeitstempel eingebunden sein, der eine nachprüfbare Rekonstruktion ermöglicht.

[8] Laut § 2 Abs. 1 StVG ist die Fahrerlaubnis „durch eine amtliche Bescheinigung (Führerschein) nachzuweisen.“ Wenn er während der Fahrt nicht mitgeführt wird, also verloren oder vergessen wurde, begeht eine Ordnungswidrigkeit nach §75 Nr. 4 FeV. (vgl. [DDD04], S. 724 u. 821)

wird. Zum Nachweis dieser Regelungen wird in Zukunft ein digitales Kontrollgerät Pflicht, das die Fahrer und Fahrzeiten speichert – eine offensichtliche Parallele zum *EDR*.[9]

Identität aus personen- und fahrzeugbezogenen Merkmalen

Ein Ansatz, der Nachrichten sowohl dem Fahrzeug als auch dem Fahrer zurechenbar macht, muss personen- und fahrzeugbezogene Identitätsmerkmalen kombinieren. Wegen des Umstands, dass ein Fahrzeug nicht fest einem Fahrer zugeordnet werden kann, ist dies nicht leicht umzusetzen.

Eine wenig praktikable Möglichkeit – z. B. bei der Nutzung einer *PKI* – ist, sowohl ein Zertifikat für das Fahrzeug als auch eines für den jeweiligen Fahrer bei jeder Nachricht zu übertragen. Dies würde die Nachrichtenlänge und die Berechnungsdauer gerade bei *beacons* über die Maßen stark vergrößern, da ja neben den beiden Zertifikaten auch die entsprechenden Signaturen erzeugt und gesendet werden müssen.

Daher ist es ratsam, die beiden Zertifikate bzw. deren enthaltene Informationen zu einem zu verschmelzen. Um aus dem Fahrzeug- und Personenzertifikat ein gültiges zu erstellen, muss dieses mit dem privaten Schlüssel einer bzw. der CA signiert werden. Dies kann grundsätzlich folgendermaßen erreicht werden:

- Beide Zertifikate werden an die CA geschickt. Sie erstellt ein temporäres Zertifikat, das alle benötigten Informationen vereint, und versendet es sicher an den Teilnehmer. Hier wird die CA einer erhöhten Belastung ausgesetzt, die durch intelligente Aufwandsverteilung kompensiert werden muss. Der Umstand, dass zwischen dem Fahrzeug- und dem Personenzertifikat keinerlei Verweis oder Beziehung bestehen darf[10], macht es nötig, dass der Antragsteller mit beiden privaten Schlüsseln Signaturen leistet. Nur so kann die zentrale Instanz überprüfen, ob die beiden übermittelten Zertifikate auch der Konstellation in der Realwelt entsprechen oder ob die Beteiligung eines Fahrzeugs oder einer Person in böswilliger Absicht vorgetäuscht wird.

  Ein Nachteil entsteht beim Aspekt der Nichtabstreitbarkeit von gesendeten Meldungen. Trotz der großen Kombinationsvielfalt von Fahrern und Fahrzeugen und der damit resultierenden Datenmenge müssten diese zumindest eine gewisse Zeit aufbewahrt werden. Wesentlich gravierender wiegt jedoch, dass für dieses eine Schutzziel I2 ein anderes – die Verhinderung der Erstellung von Bewegungsprofilen (V2) – gefährdet wird. Allgemein trägt ein solch gemischtes Zertifikat erheblich mehr Information als ein rein fahrzeugbezogenes, das für den Beobachter zumindest den Fahrzeugführer variabel und zunächst unbestimmbar lässt.

[9] Auf der Webseite der IHK Frankfurt (Oder) wird diese Neuregelung (subjektiv und) zusammenfassend dargestellt, siehe http://www.ihk-ffo.de/content/artikel/10800.html

[10] Dies ist nötig, um Fahrzeug- und Personenzertifikate beliebig kombinieren zu können, d.h. um keine Einschränkung zu erheben, welcher Fahrzeugführer welches Fahrzeug bewegen darf.

- Im Fahrzeug selbst ist in manipulationssicherer Hardware ein privater Schlüssel hinterlegt, mit dem ein temporäres Zertifikat ausgestellt wird. Dies würde zwar ein zentrales *trust center* obsolet machen und damit dem Problem der Netzabdeckung mit festen Basisstationen begegnen, dieser Schlüssel wäre allerdings zentraler Focus von Angriffen und langfristig sehr schwer zu schützen.

  Des Weiteren ist in dieser Lösung der Rückruf von Zertifikaten stark erschwert: Ein böswilliger Teilnehmer ist ja so lange in der Lage, sich gültige Zertifikate auszustellen, bis sein Fahrzeug oder die VANET-Ausstattung stillgelegt wird. In der Zwischenzeit müsste sich die Information, dass diesem Teilnehmer nicht mehr vertraut werden darf, im Rest des Netzes verteilen – ohne zentrale Organisationseinheit.

Im Fall, dass rein symmetrische Kryptographie eingesetzt würde, müsste neben dem geheimen Fahrzeugschlüssel vor Fahrtantritt auch der Schlüssel des aktuellen Fahrers bereitgestellt werden. Dies ließe sich mit individuellen Hardware-Tokens wie dem elektronischen Führerschein vernünftig bewerkstelligen. Im Gegensatz zu Lösungen mit asymmetrischer Kryptographie stellt die Berechnungsdauer von integritätssichernden *MACs* keine Hürde dar (siehe Tabelle 4.3 auf Seite 69). Neben der größeren Nachrichtenlänge – es werden *MACs* sowohl mit Hilfe des Fahrzeug- als auch des Fahrerschlüssels erzeugt – fallen vor allem die erschwerten Bedingungen bei Überprüfung der *MACs* ins Gewicht. Einem Knoten, der den *MAC* einer empfangenen Nachricht prüfen will, darf der zur Prüfung notwendige geheime Schlüssel nicht ausgehändigt werden, oder zumindest nicht in einer Weise, bei der ihm dieser Schlüssel verwendbar vorliegt. Ansonsten wäre er fortan in der Lage, kaum überprüfbar seine Identität zu wechseln.

Im Rahmen von Ad-hoc-Netzwerken stellt dies jedoch ein allgemeines Problem von Sicherheitsinfrastrukturen dar, die auf symmetrischer Kryptographie beruhen. Inwieweit tatsächliche Vorschläge für Sicherheitsinfrastrukturen diese Probleme lösen, wird in Kapitel 4.3.1 auf Seite 82 eingehend betrachtet.

Fazit

Die praktikabelste Lösung stellt sicher die in Abschnitt 4.1.1 auf Seite 47 vorgestellte dar: VANET-Teilnehmer mit erhöhten Privilegien – alle definierten Rollen außer R-PKW – weisen bei deren Inanspruchnahme diese durch ein separates Dokument, eine gesonderte Authentifizierung, o. ä. nach. Ansonsten finden fahrzeugbezogene Identitäten Anwendung, wobei als Identifikatoren eindeutige Merkmale eines Fahrzeugs wie Fahrgestellnummer, Nummernschild, etc. durchaus geeignet sind. Diese müssen jedoch im Sinne der Schutzziele V1,2 noch in eine pseudonyme Form übertragen werden, was in Abschnitt 4.1.2 auf Seite 52 näher vorgestellt wird.

Kryptographische Schlüssel und Dokumente sind diesem Ansatz nach dem Fahrzeug und nicht einer Person zugeordnet. Um dennoch den Fahrer unabstreitbar einer Fahrt mit einem bestimmten Automobil zu einer bestimmten Uhrzeit zuweisen zu können, wird der Gebrauch eines *EDR* vorgeschlagen, der die Benutzung dieses Fahrzeugs manipulationssicher und nur für autorisierte Teilnehmer (R-POLIZEI) einsehbar protokolliert.

Die Verwendung von Zertifikaten, die sowohl personen- als auch fahrzeugbezogene Daten beinhalten, oder ihre Pendants beruhend auf symmetrischer Kryptographie erscheint zwar zunächst elegant und sinnvoll, erleichtert aber in gewisser Weise Angriffe auf die Vertraulichkeit und ist überflüssig, gesetzt den Fall, dass die Personendaten nur dem Zweck der nachträglichen Verfolgbarkeit dienen.

Der Vorschlag, einen elektronischen Führerschein als Zugangskontrollmaßnahme und als Basis für einen elektronischen Fahrtenschreiber (*EDR*) zu verwenden, ist dagegen positiv zu bewerten, jedoch nur als nützliche Zusatzeigenschaft zu betrachten: Hier fehlen rechtliche Grundlagen.

4.1.2. privacy vs. auditability

privacy Als eines der wichtigsten Schutzziele wurde in Kapitel 3.2.1 auf Seite 20 die Vermeidung von Bewegungsprofilen V1 herausgestellt.

Um aus der Sicht eines Angreifers dazu überhaupt in der Lage zu sein, muss man entlang einer gewissen Strecke entweder so viele Basisstationen wie möglich unter seine Kontrolle bringen oder selbst systemkonforme Einrichtungen betreiben. Dies bedingt also einen starken Angreifer AM4,5, der eine entsprechende Verbreitung und Speicher- und Verarbeitungskapazitäten aufweist.

Mit einer nahezu lückenlosen Kette solcher Angreiferstationen gelingt es ihm dann, Aktionen und Verkehrsverhalten von Teilnehmern zu überwachen. Um zu verhindern, dass (für den Angreifer) darüber hinaus die gesammelten Informationen Fahrzeugen zurechenbar sind, wurde im Verlauf dieser Arbeit vorgeschlagen, sich auf Pseudonymität und dem Vermeiden oder Verschleiern von identifizierenden Nachrichtenbestandteilen zu stützen (vgl. Kapitel 3.2.1 auf Seite 20).

Dies ist jedoch aufgrund der besonderen Merkmale von VANETs nicht genug. Mit nur einem unveränderlichen Pseudonym könnte eine Verbindung von Pseudonym und Identität ermittelt werden, z. B. einfach dadurch, dass man ein Zielfahrzeug bis zur Wohnung verfolgt. Als logische Konsequenz wird in [RH05a] u. a. die Verwendung einer Vielzahl von pseudonymen Schlüsseln vorgeschlagen, die es auch starken Angreifern (AM4,5) sehr schwer macht, einerseits gezielt ein Fahrzeug zu verfolgen und andererseits beobachtete Aktionen und Nachrichten einem bestimmten Urheber korrekt zuzurechnen.

Der Wechsel dieser Schlüssel muss jedoch unter bestimmten Voraussetzungen stattfinden, wie Abbildung 4.1 auf der nächsten Seite veranschaulicht: Wenn ein Fahrzeug zwischen zwei angreifenden Basisstationen oder Fahrzeugen seinen Schlüssel und damit sein Pseudonym wechselt, so können die Angreifer das alte und neue einer (noch nicht bekannten) Identität zuordnen[11]. Die Angreifer haben also Informationen gewonnen und vom Schlüsselwechsel profitiert. Um dies zu verhindern, wird eine untere Grenze für die Benutzungsdauer eines Pseudonyms festgelegt, die sich mit d_r (Sendereichweite eines Fahrzeugs), d_v (Fahrtstrecke, in der Geschwindigkeit und

[11] Dies fällt ihnen umso leichter, je weniger Fahrzeuge sich zwischen ihren beiden „Messstationen“ befinden und wie sehr sich das Zielfahrzeug hinsichtlich Geschwindigkeit, Fahrspur, etc. von anderen Verkehrsteilnehmern unterscheidet, vgl. Schutzziel V2 in Kapitel 3.2.1 auf Seite 20.

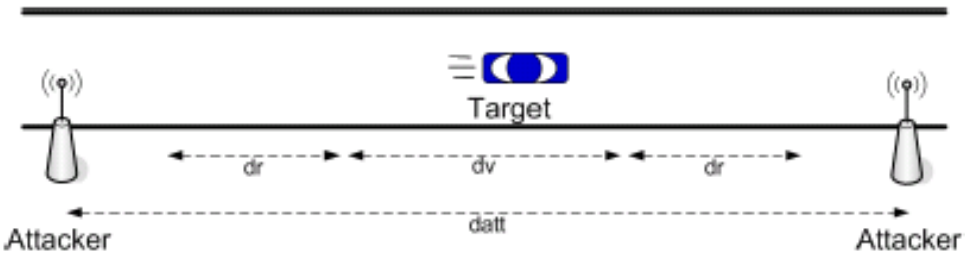

Abbildung 4.1.: Die Einflussgrößen d_r (Sendereichweite eines Fahrzeugs), d_v (Fahrtstrecke, in der Geschwindigkeit und Fahrtrichtung konstant bleiben) und d_{att} (Abstand zwischen zwei Angreiferstationen) bestimmen die Wechselfrequenz von Teilnehmerschlüsseln.

Fahrtrichtung konstant bleiben) und v_t (Geschwindigkeit des Zielfahrzeugs) folgendermaßen errechnet:

$$min(T_{key}) = \frac{d_v + 2d_r}{v_t}$$

Bei einer Fahrzeuggeschwindigkeit von 100 km/h beträgt T_{key} rund eine Minute (vgl. [RH05a], S. 8).

Die obere Zeitgrenze, innerhalb derer ein Schlüsselwechsel erfolgen muss, ist logischerweise durch den Abstand der Messstationen des Angreifers bestimmt, der in der Regel unbekannt sein dürfte. Deswegen wird in der Praxis der errechnete Wert $min(T_{key})$ leicht erhöht[12].

DOETZER wendet in [Doe05], S. 12, berechtigterweise ein, dass dies nur wirksam sein kann, wenn der Pseudonymwechsel gleichzeitig in einer angemessen großen Gruppe erfolgt – es wird eine Anonymitäts- bzw. Pseudonymitätsgruppe geschaffen (vgl. Abschnitt 3.2.1 auf Seite 23). Eine gewisse Nachbarschaftsgröße als Bedingung für einen gemeinsamen Schlüsselwechsel sollte also grundsätzlich vorherrschen.

Als Konsequenz beider Überlegungen lässt sich zum Nachteil der Netz-Performance ein stark erhöhtes Nachrichtenaufkommen prognostizieren. Gerade hier steigt die Chance, dass wichtige *Warnungen* ihre Empfänger nicht erreichen, zusätzlich an, da mit jedem Pseudonymwechsel auch die Hard- und Netzwerkadressen geändert werden müssen. Aus diesen Gründen sind weitere Aussagen zu dieser Problematik erst möglich, wenn alle Seiteneffekte und Einflussgrößen durch Tests und Simulationen genauer erforscht wurden. Grundsätzlich sollte im Zweifel der Verkehrssicherheit Vorrang gewährt werden.

auditability Um im Gegenzug die Rückverfolgbarkeit I1 zu gewährleisten, sind im Fall einer *PKI MANET-IDs*, die in [KSW05], S. 4, und [Kar04], S. 122 ff., präsentiert werden, durchaus geeignet. Eine *MANET-ID* besteht aus dem öffentlichen Schlüssel eines Teilnehmers, der durch ein CA-Zertifikat bestätigt wird. Im Sinne von V1,2 lässt das Zertifikat keine Rückschlüsse auf

[12] Im Beispiel von [RH05a], S. 8, beträgt dieser Aufschlag 10 Prozent.

Personen und Fahrzeuge zu[13]. Durch ihre gesicherte Eindeutigkeit im gesamten Netz sind öffentliche Schlüssel bzw. *MANET-IDs* geradezu prädestiniert als Identifikator und sollten deshalb ins Auge gefasst werden. Als Alternative, die auf Verfahren mit asymmetrischer wie symmetrischer Kryptographie anwendbar ist, könnten Pseudonyme über kryptographische Funktionen aus der Fahrgestellnummer (im Folgenden beispielhaft verwendet), dem Kennzeichen, etc. erzeugt werden.

Ist eine *PKI* die bevorzugte Sicherheitsinfrastruktur, so könnte in jedem Zertifikat eines Teilnehmers ein Pseudonym verankert sein, das die Fahrgestellnummer (*Vehicle Identification Number, VIN*) und den öffentlichen Schlüssel dieses Zertifikats in verschlüsselter Form beinhaltet. Als Schlüssel bietet sich natürlich der öffentliche Schlüssel der CA an, nur sie wäre dann in der Lage, die beiden Inhalte des Chiffrats zu enthüllen.

$$\mathsf{ID} = \mathsf{E}_{PK_{CA}}(VIN_{\mathrm{TN}}, PK_{\mathrm{TNx}}),$$

mit VIN_{TN} als die eindeutige Fahrgestellnummer und PK_{TNx} als der öffentliche Teilnehmerschlüssel dieses vorliegenden Zertifikats.

Folgende Vorteile ergeben sich aus dieser Abbildung:

- Bei der Ausstellung jeden Zertifikats stehen der CA beide Informationen zu Verfügung, sie sind nämlich Erhebungsgrößen beim *enrollment*.
- Der öffentliche Teilnehmerschlüssel ist als variable Einflussgröße notwendig, wenn ein Satz an Schlüsselpaaren vorgesehen sind. Würde nur die Fahrgestellnummer in die Einwegfunktion eingehen, ginge der Nutzen vieler pseudonymer Schlüsselpaare verloren: Alle Zertifikate eines Teilnehmers würden durch den unveränderten Identifikator verkettbar.
- Alle benötigten Elemente zur Zuordnung der Fahrgestellnummer zu einer Identität sind nun innerhalb einer vollständigen Nachricht vorhanden.
- Natürlich kann jeder Angreifer, der der asymmetrischen Verschlüsselung mächtig ist (ab AM1), solche Identifikatoren fälschen; er scheitert aber bei der Einbettung in das Zertifikat.
- Gemäß der Annahmen über die Angreifer in Kapitel 3.3 auf Seite 31 sind sie nicht fähig, die verwendete Verschlüsselung zu brechen. Natürlich muss auch verhindert werden, dass der Schlüssel der CA ihnen nicht in die Hände fällt.

Analog sähe die Verfahrensweise bei symmetrischer Kryptographie aus. Es wäre zwar denkbar, statt des öffentlichen einen der symmetrischen Teilnehmerschlüssel einfließen zu lassen; auch dieser würde eine eindeutige Verkettung liefern. Es sollte jedoch stets vermieden werden, geheime Schlüssel zu übertragen. Stattdessen könnten andere veränderliche Komponenten wie die Zeit zum Zuge kommen (vgl. Kapitel 4.3.1 auf Seite 84 und [JCH04], S. 4).

[13] Die CA hat und muss im Zuge von Netzausschließungen trotzdem die Möglichkeit dazu haben.

Die Exekutive AM5 wäre in diesem Modell alleine nicht fähig, Beziehungen zwischen den Zertfikaten und Fahrzeugen herzustellen.

Möchte man auch die CA als einen jetzt starken Angreifer ausschließen, müsste der private Schlüssel der CA auf mehrere Instanzen verteilt werden, so dass erst bei deren Zusammenarbeit die Fahrgestellnummern auslesbar sind (vgl. Abschnitt 4.1.4 auf Seite 61.

Auch der Fahrer könnte in für ihn kritischen Situationen gewillt sein, seine Identität bzw. seine Pseudonyme derart zu fälschen, dass eine Rückverfolgung nicht mehr möglich ist. Dies gilt es mit technischen Mitteln zu verhindern oder zumindest zu erschweren:

1. Es wird ein *tamper-proof module* als Schlüsselspeicher verwendet.
2. Der Ein- und Ausbau oder Austausch dieser sicheren Module ist bauartbedingt zu erschweren; jedoch dürfte es auch im Erfolgsfall für den Angreifer schwierig sein, anderen Personen Anreize zu bieten, ihre Schlüsselspeicher zu tauschen (Schutzziel V4).

4.1.3. Management von Schlüsseln und Zertifikaten

Mehrere Schlüssel und Pseudonyme pro Teilnehmer implizieren, dass der Aufwand für die Erzeugung, Speicherung und die Sperrung dieser großen Masse die Praktikabilität – zumindest in einem gewissen Maß – negativ beeinflusst. Gegenüber gewöhnlichen mobilen Ad-hoc-Netzen kann die Umsetzung dieser drei kritischen Prozesse nicht nur aufgrund der weniger strikten Hardwarebeschränkungen leichter umgesetzt werden, wie dieses Kapitel zeigen soll.

Erzeugen und Aufbewahren von Schlüsseln

Klassischen Ad-hoc-Netzwerken ist das bootstrapping-Problem inhärent: Für den Route-Discovery-Prozess (siehe Kapitel 2.2.4 auf Seite 9) müssen sich die beteiligten Parteien authentifizieren und die Integrität der Nachrichten gesichert werden; dafür sind jedoch verifizierte Schlüssel notwendig, die wiederum etablierte Routen voraussetzen, um sie überhaupt verteilen zu können. FRANK KARGL beschreibt diese Situation in [Kar04], S. 110, als Henne-Ei-Problem und merkt zudem an:

> *Die Lösung, Schlüssel im Vorfeld offline zu übertragen, kann in Ad hoc Netzen kaum überzeugen.*

Diese Aussage ist in Bezug auf VANETs jedoch nicht haltbar. Im „Lebenszyklus“ existieren zumindest zwei Stationen, die sich als Zeitpunkte für das Speichern der Schlüssel im Fahrzeug[14] anbieten und das bootstrapping-Problem umgehen:

[14] Im Kapitel 4.1.1 auf Seite 46 wurde festgestellt, dass kryptographische Schlüssel der Fahrzeugidentität zugeordnet werden sollten.

- Die Herstellung des Fahrzeugs.
- Die Zulassung/Umschreibung des Fahrzeugs in der Zulassungsstelle.

Erstere Variante hat den Vorteil, dass keine nachträgliche Speicherung – womöglich noch über die Luftschnittstelle – erfolgt, sondern dies bereits vor Ort von autorisierten Personal durchgeführt wird. Da gegen die Angreifer AM3-5 *Tamper Proof Modules* zur Aufbewahrung jeglichen kryptographischen Materials eingesetzt werden, kann diese manipulationssichere Hardware von dafür geeigneten Parteien (z. B. auch von den Automobilherstellern selbst, Details in Kapitel 4.1.4 auf Seite 61) bespielt und unabhängig davon ins Fahrzeug eingebaut werden.

In [RH05a], S. 5, und [PP05], S. 3 f., wird darauf hingewiesen, dass im Zuge der periodischen Inspektionen eines Fahrzeugs auch geprüft wird, ob diese Schlüsselspeicher in der Zwischenzeit manipuliert wurden. Das dafür vorausgesetzte autorisierte Wartungspersonal übernähme im Bedarfsfall auch den Austausch von gesperrten Schlüsseln oder dem gesamten Modul. Die Benutzung vorhandener Infrastrukturen wie Werkstätten und Kundendienstzentren schützt vor hohen Investitionen und lässt für die Teilnehmer die Ansprechpartner bezüglich ihres Fahrzeugs unverändert.

Neben einer adäquaten Schulung verfügt das Wartungspersonal über ein Attributzertifikat, das ihm die Rolle R-WARTUNG für das Fahrzeug überprüfbar attestiert. Sollte der Preis eines *Tamper Proof Modules*, das bereits mit allen Schlüsseln und Zertifikaten ausgestattet ist, in einem günstigen Rahmen liegen, ist es auch durchaus denkbar, dem Personal nur den Aus- und Einbau in das Fahrzeug zu erlauben bzw. zu ermöglichen. Ist dies aus finanziellen Gründen nicht durchzusetzen, sollte es grundsätzlich vermieden werden, geheime Schlüssel über die Luftschnittstelle zum Automobil zu übertragen.

Bilden *PKCS*[15] die Basis für die eingesetzte Sicherheitsinfrastruktur, sollte man in jedem Fall davon absehen, geheime Teilnehmerschlüssel zentral zu speichern. Im Kapitel 4.1.4 auf Seite 61 wird gezeigt, wie dies bereits mit dem Schritt der Schlüsselgenerierung vermieden werden kann; selbst wenn entsprechende gesetzliche Vorschriften bestehen, kann die CA nie mit Sicherheit als Angreifer AM5 ausgeschlossen werden.

Bei Verwendung von symmetrischer Kryptographie kommt man jedoch nicht umhin, geheime Schlüssel auch innerhalb zentraler Instanzen aufzubewahren; für jede Kommunikation zwischen einem Teilnehmer und einer *Trusted Third Party* muss ein gemeinsamer geheimer Schlüssel zur Verfügung stehen. In diese *TTP* muss ein weit höheres Vertrauen aus Teilnehmersicht gegeben sein (vgl. [FHK95], S. 5). Zwar muss auch der private Schlüssel der CA in höchstem Maße geschützt werden[16], im Fall symmetrischer Kryptograhpie kommen hierbei aber noch die Millionen an geheimen Schlüsseln hinzu, was zusätzlich weitaus höhere technische, organisatorische und finanzielle Ansprüche stellt.

[15] Public Key Crypto Systems

[16] Dies lässt sich gut durch eine Entkopplung vom Teilnehmerstamm erreichen, wobei nur hierarchisch untergeordnete Instanzen mit der CA zwecks Signieren von Zertifikaten direkt kommunizieren.

Sperren von Schlüsseln

Aus verschiedenen Gründen können Schlüssel und Zertifikate ihre Gültigkeit verlieren, z. B. weil sie von Vornherein nur eine limitierte Gültigkeitsdauer aufweisen, sie aus Sicherheitsgründen periodisch gewechselt werden oder gar komprimittiert wurden. Im Normalfall ist ein symmetrischer Schlüssel nur zwei Parteien bekannt, entweder teilen sich ihn zwei gleichberechtigte Teilnehmer oder ein Teilnehmer und eine *TTP*. Hier erscheint es zunächst leicht, den jeweiligen Kommunikationspartner über die Sperrung des gemeinsamen Schlüssels zu informieren. Trotzdem ist die Sperrung von Schlüsseln Gegenstand genauerer Untersuchung in konkreten Vorschlägen (siehe Kapitel 4.3.1 auf Seite 82).

In *PKCS*-Systemen jedoch stellt sich die Situation deutlich schwieriger dar, auch wenn eine zentrale Instanz wie REV oder die CA selbst permanent von jedem Teilnehmer zu erreichen ist. In einem hybrid-VANET kann eine solche Kommunikationsverbindung zur *TTP* nicht immer vorausgesetzt werden, in self-organized-VANETs sind gar keine zentralen Instanzen vorhanden oder werden durch die Kooperation einiger exponierter Knoten simuliert. Dies macht es schwierig, *revocations* zügig und netzdeckend durchzusetzen, ohne dabei überhaupt mögliche Angriffe in Erwägung zu ziehen.

Daraus ergeben sich weitere Fragestellungen, die von einer geeigneten *revocation*-Lösung innerhalb von VANETs beantwortet werden müssen:

- Wie beeinflusst die große Zahl der teilnehmenden Knoten den Umfang der Sperrdokumente oder -protokolle?
- In welcher Zeitspanne erreicht die Information einer Schlüsselsperrung sämtliche VANET-Teilnehmer?
- Wie verbindlich lassen sich *revocations* durchsetzen?
- Welche technischen Voraussetzungen müssen dafür vorhanden sein oder geschaffen werden?

Inwiefern aktuelle *PKI*-Sperrverfahren diese Fragen beantworten, zeigt deren Gegenüberstellung in den folgenden Abschnitten.

Certificate Revocation Lists In der ersten Herangehensweise veröffentlicht die REV oder die CA eine Liste (*Certificate Revocation Lists, CRLs*), in der alle gesperrten Zertifikate angeführt sind. Diese nach und nach wachsende Liste kann bei entsprechend großen Netzen enorme Ausmaße annehmen, was einen gewissen Übertragungs- und Speicheraufwand bedeutet. Abhilfe schaffen *Delta CRLs*, die zu einer gegebenen *Basis CRL* nur neu hinzugekommene Zertifikatsperrungen beinhalten. Ein weiterer Ansatz, die Übetragung kompletter *CRLs* zu vermeiden, teilt die Informationen über rückgerufene Zertifikate in *verteilte Sperrlisten* auf. Dies verringert zwar nicht die eigentliche Datenmenge; für einen einzelnen Teilnehmer, der in der Regel nur

einen bestimmten Kreis an Kommunikationspartnern besitzt, genügt es aber, nur die zutreffenden Sperrlisten periodisch abzurufen. Um für ein Zertifikat die zutreffende Sperrliste zu ermitteln, ist in X.509 eine Erweiterung namens *CRLDistributionPoints* vorgesehen, die die Verteilungspunkte benennt, unter denen der Benutzer Statusinformationen zum vorliegenden Zertifikat erhalten kann.

Online Certificate Status Protocol Die zweite Möglichkeit sieht vor, Statusinformationen für jede Zertifikatsüberprüfung online abzurufen, was zwar bedarfsgerecht und sehr aktuell ist, aber für jede Prüfung eine Onlineverbindung zum *trust center* voraussetzt. Ein einfaches Protokoll, *Online Certificate Status Protocol, OSCP*, regelt den Nachrichtenverkehr zwischen Teilnehmer und dem *trust center*.

(vgl. [Sch01], S. 333 ff.)

Sperrbäume Einen Kompromiss zwischen einer Online-Sperrabfrage und *CRLs* realisieren Sperrbäume (*Certificate Revocation Trees, CRTs*). Ziel bei der Verwendung von *CRTs* ist es, dem Client nur die tatsächlich benötigten Sperrdaten zukommen zu lassen und dabei auch den Aufwand für das *trust center* gering zu halten. In diesem Verfahren werden die Informationen, die Inhalt einer oder mehrerer *CRLs* wären, in einer anderen Datenstruktur, einem Baum, repräsentiert. Die CA stellt zunächst anhand ihrer gespeicherten Sperreinträge eine umfassende Liste von statements auf, mit denen sich ermittelt lässt, ob ein Zertifikat gesperrt ist.

> ***Beispiel:***
> *Wenn die Zertifikate Nr. 1438467, 1438468 und 1438469, ausgestellt von der CA Nr. 3, gesperrt sind, lautete das entsprechende statement:*
> *STATEMENT 5: IF CAid = 3 AND (ZNR $>$ 1438466 AND ZNR $<$ 1438470) then GESPERRT*

Mit Hilfe dieser Liste ist es einem Client zwar möglich, herauszufinden, ob ein Zertifikat gesperrt ist, aber das Herunterladen dieser Liste brächte keinerlei Vorteile gegenüber einer *CRL*. Stattdessen generiert die CA für jedes statement einen kryptographischen Hash-Wert und dann in der Folge aus jeweils zwei benachbarten Hash-Werten wiederum einen Hash-Wert. Am Ende ergibt sich ein einziger Hash-Wert, der von der CA zudem signiert wird. Abbildung 4.2 auf der nächsten Seite veranschaulicht diese Vorgehensweise.

Der VANET-Knoten lädt statt der gesamten Liste nun nur das zutreffende statement 5, die Hash-Werte A4, B5, C1 und die Signatur von D1 herunter, die grün markierten Hash-Werte sind mit diesen Informationen berechenbar. Ziel ist es, den letzten Hash-Wert zu berechnen und die Signatur zu verifizieren. Er braucht also nicht umfangreiche CRLs herunterladen, sondern holt sich nur Sperrdaten, die er aktuell benötigt. Die Certification Authority kann den Aufwand, der durch vielfache Generierung von Signaturen entsteht, vermeiden.

Bei der Aktualität bewegen sich *CRTs* auf gleichem Niveau wie die *CRLs*, die Clients benötigen aber in jedem Fall Netzzugriff, wenn sie ein Zertifikat prüfen wollen. (vgl. [Sch01], S. 340 f., [Koc98], S. 4 f.)

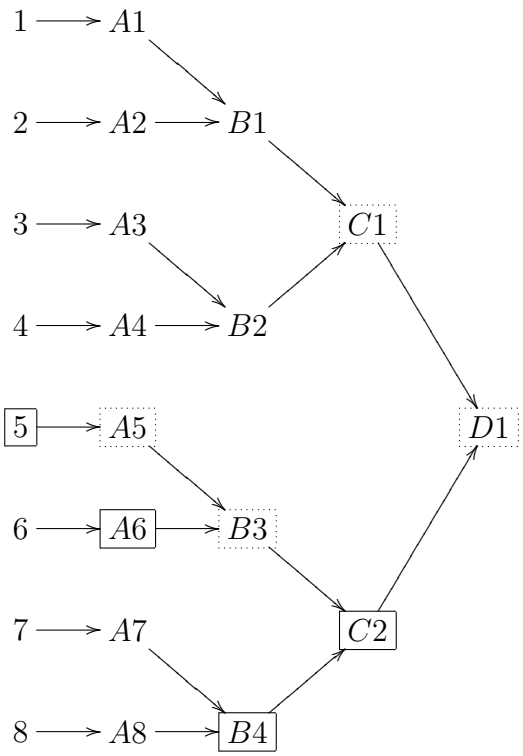

Abbildung 4.2.: In diesem (Teil-)Certificate Revocation Tree müssen nur Hash-Werte in durchgezogenen Kästchen heruntergeladen werden, die in gepunkteten Kästchen kann der Knoten selbst berechnen.

Certificate Revocation System nach Micali und Kargl MICALI stellte bereits 1998 in [Mic96] ein weiteres *Certificate Revocation System* vor, das bewusst die Schwachstellen von *CRLs* vermeiden soll, zugleich aber auch keine ständige Kommunikationsverbindung zur CA fordert. Zunächst legt die CA fest, wie lange ein Zertifikat gültig ist und in welchen Abständen die Gültigkeit geprüft werden kann. MICALI geht in seinen Ausführungen beispielhaft von einem Jahr Gültigkeit und einem Tag als Prüfperiode aus.

Für dieses Verfahren fügt die CA beim Ausstellen eines Teilnehmerzertifikats zwei Felder Y und N hinzu, die beide dieselbe Länge von 100 Bit aufweisen, jeweils einmalig im Netz verwendet und folgendermaßen gebildet werden:

- Die CA wählt einen Startwert $Y0$[17] und bildet davon eine Hash-Chain, indem sie 365 Mal

[17] Dieser Wert kann zufällig gewählt oder durch eine Abbildung eines geheimen Schlüssels und z. B. der Seriennummer des Zertifikats gebildet werden. Letztere Möglichkeit vermeidet die Speicherung einer großen Anzahl von Startwerten.

(die Anzahl der Perioden) Y0 einer Hash-Funktion übergibt:

$$\mathsf{Y} = F^{365}(\mathsf{Y0})$$

F sei eine kryptographische Hash-Funktion, der Wert Y wird im Zertifikat verankert.

- Der zweite Wert N entsteht aus der einmaligen Ausführung der Hash-Funktion von N0[18]:

$$\mathsf{N} = F(\mathsf{Y0})$$

Pro Tag im Gültigkeitszeitraum des Zertifikats (i) und Zertifikat berechnet und speichert die CA ein neues Y auf folgende Weise:[19]

$$\mathsf{Y}_{\text{365-i}} = F^{365-i}(\mathsf{Y0})$$

Ist ein zu prüfendes Zertifikat nicht zurückgerufen worden, übermittelt die CA diesen Wert $\mathsf{Y}_{\text{365-i}}$ – in [Kar04], S. 127, Verifikator V genannt – dem Prüfer des Zertifikats; im anderen Fall wird schlicht N0 zurückgeliefert.

Allein mit dem Verifikator V ist der Prüfer in der Lage, eine Entscheidung über die Gültigkeit des vorliegenden Zertifikats zu treffen; wenn

$$F^{i}(\mathsf{V}) = \mathsf{Y}$$

gilt, sieht er das Zertifikat als gültig an.[20] Es handelt sich bei dem Verfahren von SILVIO MICALI also nicht um den Rückruf eines Zertifikats, es beweist eher die CA die Gültigkeit.

In diesem Stadium liegt ein *Certificate Revocation System* vor, das stark von Onlineverbindungen zur CA abhängt. In [Kar04], S. 128, greift FRANK KARGL die Idee aus [Koc98] auf,

> *dass der Zertifikatsinhaber selbst die Gültigkeit seines Zertifikats beweisen muss.*

Hier wird allerdings auf den Wert N komplett verzichtet, die Funktionsweise von Y wird beibehalten. Ist es einem Knoten möglich, Kontakt zur CA aufzunehmen, holt er sich über die Seriennummer oder ein anderes eindeutiges Merkmal seines Zertifikats den aktuellen Verifikator V_{i} und sendet diesen fortan zusammen mit seiner Nachricht, der Signatur und seinem Zertifikat.

Da er gültige Verifikatoren keineswegs selbst erstellen kann – er kennt den Wert Y0 nicht – sind seine Kommunikationspartner stets in der Lage, die Gültigkeit seines Zertifikats zu prüfen, ohne Kontakt mit der Certification Authority aufnehmen zu müssen.

Falls auf dem Übertragungsweg zwischen Knoten und CA manipuliert wurde, erkennt dies der Knoten durch die oben beschriebene Prüfung sofort und kann eine weitere Anfrage stellen. Alternativ, wenn das Zertifikat abgelaufen ist, schickt die CA eine entsprechende und signierte

[18] Dieser Startwert wird analog zu Y0 gewählt.
[19] Dies kann natürlich auch im Voraus und damit nicht in Echtzeit berechnet werden.
[20] $F^{i}(\mathsf{V}) = F^{i}(\mathsf{Y}^{365-i}) = F^{i}(F^{365-i}(\mathsf{Y0})) = F^{365}(\mathsf{Y0})$

Nachricht an den Knoten. Nimmt der VANET-Teilnehmer in diesem Fall weiterhin am Netz teil, können andere seine Nachricht zurückweisen, da entweder kein oder ein inkorrekter Verifikator[21] enthalten ist.

Für jede Verlängerung eines Zertifikats, d.h. wenn es nicht vorzeitig abgelaufen ist, werden die beschriebenen Schritte wiederholt, allerdings ohne den (hier nicht dargestellten) Prozess des *enrollments*.

Kurzzeit-Zertifikate Den umgekehrten Ansatz verfolgen Kurzzeit-Zertifikate. Ein Benutzer kann erst wieder am Netz partizipieren, wenn ihm ein neues Zertifikat ausgestellt wird. Die Zeit, in der eine *revocation* in Kraft tritt, ist hier also nicht von der Verteil- und Erneuerungszeit der *CRLs*, *CRTs*, etc. (Aufgabe der REV und der CA) abhängig, sondern nur von der Gültigkeitsdauer der Zertifikate und der Bearbeitungszeit der CA. Je kürzer man diese Gültigkeitsdauer gestaltet, desto schneller werden zwar die *revocations* umgesetzt, aber desto größer wird der Aufwand der CA für die Prüfung, ob Knoten noch teilnahmeberechtigt sind, und die Neuerstellung der Zertifikate. Dieser Umstand sollte bei der Realisierung dieses Vorschlags in Bezug auf die Verfügbarkeit und die Performance starke Berücksichtigung finden.

Vorteil dieser Lösung ist die geringe Abhängigkeit von Verbindungen zum *trust center*. Nur zum Ende einer Gültigkeitsperiode ist eine Kontaktaufnahme zur CA notwendig, um ein neues Zertifikat zu erhalten. In 3.2.4 auf Seite 30 wurde die Verfügbarkeit von festen Basisstationen als Herausforderung festgestellt, der Kurzzeit-Zertifikate sehr gut begegnen.

Fazit Ein *revocation system*, das für jede Zertifikatsüberprüfung zwingend eine Verbindung zur CA benötigt, ist für den Einsatz in einem VANET nicht geeignet, da diese Verfügbarkeit in VANETs nicht vorausgesetzt werden kann. Zudem würden die Echtzeitanforderungen, die vor allem von der Anwendungsgruppe A1 (Bereich Telematik) gestellt werden, nicht leicht zu erfüllen sein. Für die Zertifikatsprüfung ständig und stark wechselnder Kommunikationspartner wären reine Online-Verfahren wie *OCSP* und die eleganten *CRTs* wenig praktikabel.

Unter den Offline-Verfahren weisen die *CRLs* die gravierendsten Nachteile auf: Neben ihrer unzureichenden Aktualität und ihrem Aufwand für die CA beeinträchtigt am meisten ihre schiere Größe die Praktikabilität in VANETs: In [Mic96], S. 2, geht der Autor von einer jährlichen Rückrufquote von 10 % aus, so hätte eine umfassende *CRL* für Deutschland ungefähr 5 Millionen Einträge, wenn man beispielhaft davon ausgeht, dass jedes Kraftfahrzeug am VANET teilnimmt. Geht man pro Zertifikat nur von einer 32 Bit Seriennummer aus, so beträge die Größe dieser *CRL* rund 20 MByte (vgl. [Kar04], S. 127).[22]

Natürlich bieten sich in VANETs verteilte *CRLs* an, die jeweils nur eine geographische Region bedienen und so nur einen Bruchteil der vorher ermittelten Größe aufweisen. Als neues Problem treten hier all diejenigen Verkehrsteilnehmer auf, die zwischen solch definierten Regionen

[21] In diesem Fall ist entweder Manipulation vonseiten eines Angreifers oder des Knoten selbst ununterscheidbar aufgetreten.

[22] Weitere Nachteile von *CRLs* zeigt [Gut02], S. 7 ff. anschaulich auf.

wechseln, vor allem Personen, die am Rand von Regionen wohnen, Pendler, Lastkraftwagenfahrer: In diesen Fällen müssten nun doch auch andere Teil-*CRLs* bei Bedarf verfügbar sein. Eine solche Implementierung erforderte zusätzlich die ursprüngliche Herkunft eines Teilnehmers, um die korrespondierende *Certificate Revocation List* zu ermitteln. Ein solcher Mechanismus würde allerdings einer Bewegungsprofilbildung durch AM4 und AM5 zuträglich sein.

Wie in Abschnitt 4.1.3 auf Seite 54 angedeutet, ist es denkbar, pro Teilnehmer nicht ein Schlüsselpaar, sondern mehrere zu vergeben. Der Größe von *CRLs* würde damit vielfach multipliziert. Auch der Aufwand für die Erstellung einer Vielzahl von Kurzzeit-Zertifikaten pro Teilnehmer scheint nicht vertretbar, es stellt außerdem kein *revocation system*, sondern eher eine Ausweichmöglichkeit dar.

In seiner ursprünglichen Form stellt auch das Verifikatoren-System von MICALI nur ein nicht praktikables *online revocation system* dar, mit der Idee von KARGL, den Benutzer die Beweislast der Zertifikatsgültigkeit zu übertragen, steht jedoch ein vielversprechener Vorschlag im Raum. Nachteil dieser Lösung ist die gestiegene Nachrichtenlänge, sie besteht nun aus Nachricht, Signatur, Zertifikat und dem Verifikator.

Deshalb ist zu überlegen, ob der Verifikator tatsächlich jeder einzelnen Nachricht angefügt werden muss (vgl. [Kar04], S. 129 und Kapitel 4.3.2 auf Seite 93). Speichert ein Knoten alle empfangenen Verifikatoren und die Adressaten seiner Nachrichten für eine kleine Zeitspanne, so ist einer Nachricht der Verifikator nur anzufügen, wenn es sich um einen „neuen“ Adressaten handelt. Der Inhalt dieses Caches muss dabei mit Ablauf der Verifikatorgültigkeit (z. b. ein Tag) gelöscht werden.

4.1.4. Betreiber einer Sicherheitsinfrastruktur

Betreiber einer Sicherheitsinfrastruktur werden in dieser Arbeit all diejenigen Stellen, Unternehmen, Organisationen, etc. genannt, die eine oder mehrere Aufgaben in einer Sicherheitsinfrastruktur wahrnehmen. Komponenten, wie sie in 3.4.1 auf Seite 37, 3.4.2 auf Seite 41 und 3.4.2 auf Seite 41 vorgestellt wurden, können demnach auf mehrere Instanzen verteilt werden (z. B. hierarchisch strukturierte CAs) oder in einer Instanz zusammengefasst werden (z. B. können die RA und/oder der Schlüsselgenerierungsserver in die CA integriert werden).

In diesem Kapitel soll erörtert werden, wer als Betreiber einer Sicherheitsinfrastruktur in VANETs geeignet ist.

Erzeugen von Schlüsseln

Die Erzeugung von Schlüsseln bzw. Schlüsselpaaren kann in die Hände

- der Endbenutzer
- dezentraler Registrierungsstellen (z. B. RAs)

- zentraler vertrauenswürdiger Instanzen (z. B. CAs)
- der Hersteller oder Installateure[23] von Schlüsselspeichern
- von kombinierten Varianten

gelegt werden. Im ersten Fall muss sich eine zentrale Instanz (hybrid-Szenario), die für das Schlüsselmanagement zuständig ist, zusätzlich vergewissern, dass der Antragsteller – und ausschließlich er – im Besitz des symmetrischen bzw. des privaten Schlüssels ist. Liegt dem VANET eine *PKI* zugrunde, signiert schließlich die CA das neue Zertifikat, sofern der Antragsteller nicht schon ein gültiges besitzt.

Im Allgemeinen ist jedoch davon abzuraten, die Schlüsselerzeugung allein dem Knoten zu überlassen; denn FRANK KARGL beweist in [Kar04], S. 120, und [KSW05], S. 3:

> *„Ohne Beteiligung einer* Trusted Third Party *können keine verlässlichen Identifikatoren generiert werden.“*

In self-organized-Szenarien, in denen keine feste Infrastruktur angedacht ist, muss die Schlüsselerzeugung von einem, besser mehreren Endbenutzern (EE) geleistet werden. Das Einbinden von mehreren Parteien in den Erzeugungsprozess bereitet zwar größeren Aufwand, verringert jedoch die Chance, dass Schlüsselmissbrauch betrieben wird.

Die Hersteller von Automobilen eignen sich für alle Aufgaben einer Sicherheitsinfrastruktur, die vorab ohne eine Zuordnung eines Fahrzeugs zu einem späteren Käufer bzw. Halter ausgeführt werden können, da während der Produktion der spätere Besitzer nicht bekannt ist und diese Information damit auch für Angriffe nicht zur Verfügung steht. Dazu gehört neben dem Einbau der *Tamper Proof Modules*, die kryptographische Schlüssel sicher beherbergen, auch die Schlüsselgenerierung selbst. Dies setzt die Vertraulichkeit und Integrität dieses Prozesses voraus.[24]

Natürlich bedeutet dies zunächst Investitionen in Hardware, Applikationen und die Absicherung dieser, eine Standardisierung unter den Automobilproduzenten und die periodische Kontrolle dieses Prozesses durch unabhängige Sachverständige. Sollten die dafür notwendigen und hohen Ansprüche vonseiten der Automobilhersteller nicht zu erfüllen sein, ist die Generierung und Speicherung der Schlüssel auf dem *TPM* von einem dritten Unternehmen oder einer geeigneten staatlichen Stelle in Erwägung zu ziehen.

Die beiden letztgenannten Schritte des Erzeugens und des Speicherns dürfen nicht getrennt werden; vom Übertragen von geheimen Schlüsseln von einer Institution zu einer anderen ist dringend abzuraten.

Durch die Entkopplung der beiden Vorgänge der Schlüsselerzeugung und der Zertifikaterstellung durch die CA, wird verhindert, dass eine Instanz alle zentralen Aufgaben in sich vereint und so zu

[23] Der Begriff Installateur meint hier diejenigen Unternehmen, die den Schlüsselspeicher im Automobil verbauen: Die Fahrzeughersteller selbst, ihre Partner-, Tochter- und Mutterunternehmen oder Zulieferer.

[24] Diese Ziele sollte das jeweilige Unternehmen selbst zusichern, z. B. über eine Selbstverpflichtung, eine vertragliche Zusicherung gegenüber dem Endkunden, etc. (vgl. [FHK95], S. 4)

einem sehr starken Angreifer oder aber zum Focus von Angreifern würde. Natürlich lastet jetzt auf den Automobilherstellern hoher Druck, die besonderen Sicherheitsmaßnahmen umzusetzen.

Bei der Schlüsselgenerierung handelt es sich jedoch um ein reines Offline-Verfahren, das durch strikt autorisiertes Personal und Zugriffs- und Zugangsverfahren gut zu schützen ist. Im Laufe der letzten Jahre wurden in der Automobilbranche in interne *PKI*-Systeme investiert[25], was Know-How und eine gewisse Erfahrung in Sicherheitsthemen voraussetzt.

Für jedes dieser Unternehmen besteht der Anreiz, das Vertrauen, das ihnen ihre Kundschaft mit dem Kauf eines Fahrzeugs entgegenbringt, nicht zu enttäuschen und die Vertraulichkeit und Integrität während der Schlüsselerzeugung zu wahren. Das Bekanntwerden eines Missbrauchs würde der Konkurrenz noch mehr Wettbewerbsvorteile einbringen als z. B. die Bekanntgabe einer Rückrufaktion.

Es erscheint sinnvoll, dass der Automobilhersteller auch die aktuellen Zertifikate der zentralen *PKI*-Komponenten im Fahrzeug speichert; dies ist ja unabhängig von den folgenden Schritten des *enrollments* und der Zertifizierung[26].

Diese Überlegungen gelten natürlich auch für Ansätze, die symmetrische Kryptographie einsetzen. Auch hier ist es sinnvoll, die *TPMs* innerhalb der *Trusted Third Party* zu bespielen.

enrollment

Für die Komponente RA sind Automobilproduzenten denkbar ungeeignet, sie müssten flächendeckend regionale Registrierungsstellen aufbauen, ihre Kompetenzen stark erweitern und neuerdings personenbezogene Daten speichern. Zudem bedeutete dies hohe Investitionshürden (Aufbau, Unterhalt), deren Kostendeckung wohl schwer auf die VANET-Teilnehmer umgewälzt werden könnte.

Stattdessen werden als LRAs die Zulassungsstellen des Kraftfahrt-Bundesamtes[27] vorgeschlagen. Sie erfüllen bereits jetzt viele Voraussetzungen:

- Sie sind den Fahrzeughaltern vertraut als diejenige Stelle, die staatliche Belange im Fahrzeugwesen vertritt.
- Sie sind regional sehr gut verfügbar und haben ihren Sitz zentral in Landratsämtern und ähnlichen staatlichen Verwaltungseinrichtungen.
- Sie sind bereits jetzt mit ihrer Zentrale, dem KBA – Kraftfahrt-Bundesamt, vernetzt.

[25] Bei der Audi AG besitzt jeder Mitarbeiter einen Ausweis, der die Zugangskontrolle zu Unternehmensbereichen, die Zugangskontrolle zu Arbeitsplatzrechnern und das Verschlüsseln und Signieren von E-Mails gestattet. Dieses Unternehmen unterhält außerdem einen öffentlichen keyserver: http://keyserver.audi.de

[26] In 4.1.4 auf der nächsten Seite wird dieser Vorschlag weiter begründet.

[27] Beispielhaft für die entsprechenden staatlichen Institutionen anderer Länder.

Nun wird der Prozess der Fahrzeugzulassung um die Überreichung der öffentlichen Schlüssel an die CA und die Erzeugung und Rückübertragung der Teilnehmerzertifikate erweitert. Für den ersten Schritt folgen aus den Ergebnissen des letzten Abschnitts zwei Möglichkeiten:

1. Im Schritt der Schlüsselerzeugung könnte dem *TPM* zusätzlich ein Dokument oder ein Hardware-Token beigelegt werden, das alle auf dem Modul gespeicherten öffentlichen Schlüssel selbstsigniert auflistet. Da zu diesem Zeitpunkt noch keine Verbindung zum Fahrzeug oder gar dem späteren Halter besteht, muss nur sichergestellt werden, dass die Information, welche öffentlichen Schlüssel einem *TPM* angehören, vertraulich bleibt. Dies sollte also exakt während des Schlüsselspeicherungsprozesses vollzogen werden, bei dem schon sehr hohe Sicherheitsmechanismen aktiv sind.

2. Das Fahrzeug bzw. das *TPM* könnte jederzeit seine öffentlichen Schlüssel selbst signieren, so dass beweisbar wäre, dass das Fahrzeug diese Schlüsselpaare besitzt. Bei der Registrierung authentifiziert sich die RA gegenüber dem Fahrzeug und überträgt nach sorgfältiger Prüfung alle ihr ausgehändigten Dokumente an die Zentrale. Die aktuellen Zertifikate der zentralen *PKI*-Komponenten müssen demnach schon vorab, z. B. beim Automobilhersteller, im Fahrzeug gespeichert werden.

Die zweite Möglichkeit ist der ersten in mehreren Hinsichten überlegen:

- Bei 1. kann der Fahrzeughalter nicht als Angreifer AM2 ausgeschlossen werden, er kann z. B. das Dokument vertauschen oder fälschen, um in eine fremde Identität zu schlüpfen.
- Bereits im Werk können die *TPMs* oder das Dokument/Token (un)absichtlich vertauscht werden.
- Zur anschließenden Übertragung der Zertifikate von der CA an den Knoten muss in jedem Fall eine authentische Verbindung etabliert werden.
- Die CA kann sich bei der zweiten Möglichkeit sofort überzeugen, dass im Fahrzeug tatsächlich die angegebenen Schlüsselpaare gespeichert sind.

Im Fall symmetrischer Kryptographie erfolgte bereits im vorherigen Abschnitt der Schlüsselaustausch zwischen dem Fahrzeug und der *TTP*. Letzterer fehlt nur noch die Information des Fahrzeughalters, um dem Schutzziel I1 zu entsprechen: Hier bieten sich Hash-Werte der geheimen Schlüssel an, um diese den Teilnehmern zuzuordnen.

Zentrale Instanzen

CA In [RH05a], S. 6, und [PP05], S. 2, entscheiden sich die Autoren nicht zwischen ihren beiden Vorschlägen für den Betreiber der CA. Mit den Automobilherstellern als CAs wäre es sehr leicht, jedes Fahrzeug mit den aktuellen CA-Zertifikaten auszustatten, wenn eine Standardisierung unter den Herstellern zu erreichen ist.

Im Gegenzug wäre es sehr schwierig für die Unternehmen, in jedem Land die gesetzlichen Vorschriften zu beachten und dieselben Sicherheits- und Qualitätsniveaus zu erreichen. Die Dienste der CA müssten von jedem Standort aus erreichbar sein. Das Betreiben eines *trust centers* fällt außerdem nicht in die Kernkompetenzen dieser Unternehmen.

Den Königsweg bildete natürlich eine verteilte CA bestehend aus einem internationalen Automobilkonsortium und einer staatlichen Automobilbehörde; so könnten die staatlichen Interessen und die der Teilnehmer vertreten werden, ohne dass eine Seite die Oberhand gewönne. Zentrale Aktionen wie das Ausstellen von Zertifikaten oder das Ausschließen eines Teilnehmers bedürfen hier der Zusammenarbeit beider.[28]

Derzeit zeichnen sich diesbezüglich aber keine Bestrebungen ab. Deshalb wird im Folgenden stellvertretend für diese präferierte Lösung von zentralen staatlichen Stellen wie das Kraftfahrt-Bundesamt [29] als Betreiber der CA ausgegangen. Sollte es wider Erwarten doch gelingen, eine zweigeteilte CA aus privaten und staatlichen Institutionen zu schaffen, ist diese Lösung natürlich vorzuziehen.

Staatliche Kraftfahrtsämter sind nach den aktuellen gesetzlichen Vorschriften geschaffen und verfügen über bereits ausgebaute Infrastruktur, eine solch große Teilnehmerzahl zu bewältigen. Das KBA[30] betreibt zudem seit August 2005 ein *trust center* für das Zentrale Kontrollgerätekartenregister (ZKR) im Rahmen der Einführung des Digitalen EG-Kontrollgerätes[31].

Relevante Elemente des ZKR [32]

- die *Fahrerkarte*: Speichert Lenk- und Ruhezeiten und enthält die Identitätsdaten des Fahrers. Diese Daten müssen alle 28 Tage auf Datenspeicher gesichert werden.
- die *Unternehmenskarte*: Weist das Unternehmen aus und ermöglicht die Anzeige, das Herunterladen und den Ausdruck der Daten, die in dem Kontrollgerät gespeichert sind.
- die *Werkstattkarte*: Dient zur Prüfung/Reparatur und Kalibrierung des digitalen EG-Kontrollgerätes, sowie zum Herunterladen der Daten und zur Datensicherung.
- die *Kontrollkarte*: Ermöglicht den unbeschränkten Zugriff auf gespeicherte Daten.

Die *Fahrerkarte* entspricht also in etwa dem *EDR*, wobei auf Letzterem neben den Lenk- und Ruhezeiten auch weitere Daten gespeichert werden (vgl. Kapitel 2.2.4 auf Seite 9). Aufgrund dieser

[28] Laut einem Telefongespräch mit Timo Kosch (BMW) am 2. Februar 2006 wäre diese Lösung auch vonseiten der Industrie durchaus gangbar, wenn auch schwer umzusetzen.
[29] Informationen zum KBA und dessen Zentralregistern sind unter http://www.kba.de zu finden.
[30] In diesem Abschnitt wird stellvertretend für die nationalen Kraftfahrtsämter das KBA diskutiert.
[31] Hinter diesem Begriff verbergen sich digitale Fahrtenschreiber, die ihre analogen und nicht fälschungssicheren Pendants verpflichtend ab 1. Mai 2006 ablösen, vgl. http://www.heise.de/newsticker/meldung/68894.
[32] Diese und weitere Informationen sind unter http://www.kba.de/Stabsstelle/ZentraleRegister/zkr/EG_0.htm zu finden.

Tatsache können Straftäter versucht sein, diese der Exekutive vorzuenthalten oder zu zerstören; deshalb wird an dieser Stelle nicht davon abgewichen, diesen *Event Data Recorder* ähnlich einer Blackbox so sicher wie möglich im Fahrzeug zu verbauen. Die Datenspeicherung erfolgt also nicht auf den individuellen Karten jedes Fahrers, sondern auf dem Speicher des Fahrzeugs und muss – wie in Kapitel 4.1.1 auf Seite 47 gezeigt – noch durch einen elektronischen Führerschein, o. ä. personifiziert werden.

Werkstattkarten in solcher oder ähnlicher Form sind in Verbindung mit einer rechtlichen Absicherung durchaus geeignet, Werkstätten für den Austausch von *TPMs* zu autorisieren, der beim Ende der Schlüsselgültigkeit anfällt.

Die *Unternehmenskarte* und die *Kontrollkarte* sind für ihre Zielgruppen (Unternehmen bzw. Exekutive) ebenfalls als sinnvoll zu erachten, solange die Schutzziele der Vertraulichkeit gesetzeskonform gewahrt bleiben.

Es zeigt sich also wiederum, dass das Kraftfahrt-Bundesamt bereits jetzt Aufgaben umsetzt, die in VANETs in ähnlicher, wenn auch detaillierter Form zu bewältigen sind.

Zusammenarbeit von nationalen CAs Der Vorschlag aus Abschnitt 4.1.4 auf Seite 61, auch die Zertifikate von CAs bereits im Werk in das Fahrzeug einzutragen, wird an dieser Stelle erweitert: Da i. A. schon vor der Produktion jedes Fahrzeugs feststeht, für welchen Kontinent oder gar Staat es bestimmt ist, können die CA-Zertifikate dieser betreffenden Region bereits installiert werden, z. B. für Europa, Süd- und Mittelamerika. Einzig in Staaten wie den USA, in denen das Transport- und Verkehrswesen jeder Staat unabhängig von den anderen reguliert (vgl. [PP05], S. 2), dürfte die Zahl der zu speichernden Zertifikate etwas größer ausfallen.

Sollte es dennoch notwendig werden, die Zertifikate eines anderen Kontinents zu beantragen, so besteht die Möglichkeit, dass die „Heimat“ -CA noch vorab die neuen Zertifikate beschafft, cross-zertifiziert und dem Fahrzeug vertraulich zuschickt[33].

Ist dies nicht möglich oder wechselt das Automobil längerfristig seinen Standort, kommt der Halter nicht umhin, es bei der zuständigen Behörde zu registrieren; bei diesem Vorgang werden ihm dann die Zertifikate von der „neuen“ CA zugewiesen.

Die zuletzt beschriebene Vorgehensweise hat neben ihrer geringen Komplexität vor allem einen Vorteil hinsichtlich des Schutzziels V1 und des Angreifermodells AM5: Sähe man nämlich vor, bei jedem Grenzübergang sich das Zertifikat bei der neuen CA zu holen, gäbe dieser Vorgang genauen Aufschluss über einen Teil seines Bewegungsprofils.

REV Würde man auf konventionelle *revocation systems* setzen, würden deren natürliche Probleme beim Überschreiten einer Grenze noch deutlicher zu Tage treten. In jedem Land müssten auch die Sperrinformationen ausländischer Fahrzeuge verfügbar sein, unter der Prämisse, das Bewegungsverhalten von Fahrzeugen nicht zu registrieren. Dies wäre mit *online revocation systems*

[33] Fernfahrern (LKW, Kurierdienste) ständen so in kurzer Zeit alle benötigten Zertifikate zur Verfügung, ohne die Masse ausufern zu lassen.

durchaus zu bewältigen, wenn die Echtzeitanforderungen und die mangelnde Netzabdeckung mit stationären Knoten nicht beständen. Wiederum empfiehlt sich das Verifikatoren-System nach MICALI und KARGL (Abschnitt 4.1.3 auf Seite 58).

Als Verwalter der Verifikatoren ist eine Instanz REV einzurichten, die nur lose mit der CA gekoppelt ist, aber durchaus von derselben Instituion betrieben werden kann. Ihre Aufgabe ist es, die Verifikatoren vorzuberechnen und sie auf Teilnehmeranfrage hin in vertraulicher und authentifizierter Weise bereitzustellen. Sie ist natürlich auch für der Ausschluss von Knoten zuständig. Diese Auslagerung vermeidet direkte Netzzugriffe der Teilnehmer auf die CA, was die Maxime bei der Nutzung einer *PKI* ist.

TSS Da mit *Galileo* zukünftig integritätsgesicherte Zeitdaten zur Verfügung stehen (vgl. Kapitel 2.2.4 auf Seite 9), sollten diese auch als Grundlage für die Zeitangaben in Nachrichten verwendet werden. Diese Daten sind hochverfügbar, unabhängig von Basisstationen und für alle Teilnehmer identisch.

Zusammenfassung

Tabelle 4.1 fasst die Ergebnisse dieses Abschnitts noch einmal zusammen.

***PKI*-Instanz**	**Betreiber**	**Aufgaben**
GEN	Automobilhersteller	Generieren und Speichern von Teilnehmerschlüsseln Einbau der *TPMs*
RA	Zulassungstellen	Registrierung der Teilnehmer Organisation der Zertifizierung
CA	Kraftfahrt-Amt	Zertifikatmanagement
	Werkstätten	Schlüsselaustausch
TSS	*Galileo*	Zeitdienst

Tabelle 4.1.: Die Betreiber in einer *PKI*-basierten Lösung

4.1.5. Kryptographie

Die Anforderung AN2 legte in Abschnitt 3.2 auf Seite 19 dar, dass für die Aufbereitung und den Versand von Nachrichten nur ein kleines Zeitfenster bleibt; parallel dazu müssen auch empfangene Nachrichten verarbeitet werden. Jegliche kryptographische Maßnahme verlängert dabei die Verarbeitungszeit, im Fall der Integritätssicherung auch die Nachrichtenlänge. In diesem Kapitel soll deshalb kurz auf die Wahl der Kryptoalgorithmen, der Schlüssellängen und der daraus resultierenden Nachrichtenlängen eingegangen werden. Dazu wurde in Java ein Performance-Test

namens CRYPTOSPEED implementiert, dessen source code und Ergebnisse ab Anhang A auf Seite xi zu finden sind. Die Angaben für NTRU wurden wegen mangelnder Verfügbarkeit aus den Angaben von [NTR] ermittelt, dies betrifft die Ergebnisse in den Tabellen 4.2 und 4.3 auf der nächsten Seite.

Ver- und Entschlüsselung Außer für den Anwendungsbereich der *beacons*[34] ist die Nachrichtenverschlüsselung das Mittel, um Kommunikationsinhalte gegenüber Dritten abzuschirmen.

Symmetrische Algorithmen			
Algorithmus	**Schlüssellänge**	**Verschlüsselung**	**Entschlüsselung**
3DES	112 Bit	0,134 ms	0.136 ms
AES	192 Bit	0,092 ms	0,071 ms
Blowfish	448 Bit	0,054 ms	0,052 ms
Asymmetrische Algorithmen			
RSA	1024 Bit	2,81 ms	51,07 ms
NTRU	251 Bit	0,487 ms	0,342 ms

Tabelle 4.2.: Verschlüsselungsalgorithmen und ihre Performance. Die kompletten Ergebnisse dieses Laufs sind in Anhang B auf Seite xxiii zu finden.

In Tabelle 4.2 werden nun die Ergebnisse folgenden Aufrufs von CRYPTOSPEED zusammengefasst:

```
java CryptoSpeed -c all -m 100 -l 10000
```

(alle Verschlüsselungsalgorithmen, Nachrichtenlänge von 100 Byte, 10000 Zufallsnachrichten zur Bildung des Durchschnitts)

Hier spielen die symmetrischen Verfahren *3DES*, *AES* und *Blowfish* ihre Stärke aus. Aber auch das asymmetrische Verfahren *NTRU* , das in CRYPTOSPEED wegen mangelnder Verfügbarkeit nicht integriert werden konnte, überrascht mit sehr guten Werten und sollte nach Prüfung seiner Eigenschaften beim Signieren im nächsten Abschnitt und seiner generellen Sicherheit ins Auge gefasst werden.

Signaturen generieren und verifizieren Da *beacons* grundsätzlich kryptographische Merkmale tragen müssen, die Aufschluss über ihre Integrität und Zurechenbarkeit (I1) geben, bilden sie den Flaschenhals. Da ein Fahrzeug meistens mindestens so viele *beacons* empfängt

[34] In Kapitel 3.2.1 auf Seite 20 wurde dargelegt, dass *beacons* nur optionalerweise zu verschlüsseln sind, die Erfüllung der Echtzeitanforderungen hat hier Vorrang.

wie es versendet, ist besonderes Augenmerk auf die Verifikation der Maßnahmen zu richten. Zusätzlich spielen die erforderliche Schlüssel- und vor allem Nachrichtenlänge eine tragende Rolle.

In Tabelle 4.3 werden die bisher genannten Methoden zur Sicherung von Integrität hinsichtlich Schlüssellänge, Berechnungszeit und der resultierenden Länge gegenübergestellt.

Die Berechnungszeit entspringt wiederum CRYPTOSPEED, als Konstanten wurde eine Nachrichtenlänge von 150 Byte und eine Wiederholungszahl von 10000 angenommen.

```
java CryptoSpeed -s all -m 150 -l 10000
```

Symmetrische Algorithmen				
Algorithmus	**Schlüssellänge**	**Länge**	**Generierung**	**Verifikation**
HMAC SHA-1	1024 Bit	160 Bit	0,024 ms	0,026 ms
HMAC MD-5	1024 Bit	128 Bit	0,053 ms	0,053 ms
Asymmetrische Algorithmen				
md5/rsa	1024 Bit	1024 Bit	48,22 ms	2,88 ms
SHA-1/DSA	1024 Bit	320 Bit	20,11 ms	39,66 ms
ECDSA	192 Bit	320 Bit	59,13 ms	117,82 ms
NTRU	251 Bit	251 Bit	0,850 ms	0,606 ms

Tabelle 4.3.: Signatur- und MAC-Algorithmen, ihre Länge und Performance. Die Angaben über die Signaturlänge entstammen [DKKS05], S. 4. Die kompletten Ergebnisse dieses Laufs sind in Anhang C auf Seite xxv zu finden.

Wie zu erwarten war, schlagen die beiden *MAC*-Algorithmen ihre asymmetrischen Konkurrenten, einzig *NTRU* bewegt sich in einer ähnlichen Größenordnung.

Unter den asymmetrischen Signaturverfahren fällt auf, dass *RSA* eine sehr schnelle Verifizierung von Signaturen erlaubt und gegenüber *DSA* klar vorzuziehen ist. Leider spiegeln aktuelle Java-Implementierungen von *ECDSA* nicht das Potential dieses Verfahrens wider, das in den in diesem Kapitel zitierten Quellen in ähnliche Regionen wie *NTRU* eingeordnet wird. In [RH05a], S. 9 f. werden *NTRU* und *ECDSA* ebenfalls verglichen[35]. Hier wird zumindest die Grundtendenz bestätigt, dass *ECDSA* das Signieren schneller leistet als das Verifizieren und *NTRU* in beiden Disziplinen führend ist.

Als Kandidaten bleiben also nur *ECDSA* und *NTRU* übrig, von denen letzterer weit weniger Kryptoanalysen ausgesetzt war. Seine Vorteile liegen neben der überragenden Geschwindigkeit in der im Vergleich zu *RSA* etwas kleineren Schlüssel- und Signaturlänge. Die größten Vorteile

[35] Die Resultate sind aufgrund eines schwächeren Rechners und einer anderen Implementierung nicht mit denen von CRYPTOSPEED vergleichbar.

von *ECDSA* werden in seiner Kompaktheit gesehen, die die Werte der Konkurrenten um Längen unterbieten.

Wählt man das Verifikatoren-System zum Rückruf von Zertifikaten (vgl. Kapitel 4.1.3 auf Seite 58), verlängert es Nachrichten um eine nicht erhebliche Datenmenge. In [Kar04], S. 203 ff., summiert deshalb der Autor die einzelnen Felder eines Zertifikats gemäß dem Verifikatoren-System auf und stellt fest, dass die Größe maßgeblich von der Signatur- und Schlüssellänge abhängt. Sie beträgt bei einem 1024-Bit-RSA-Schlüssel ingesamt 2256 Bit zuzüglich des aktuellen Verifikators von 160 Bit. Kommt jedoch *ECDSA* zum Einsatz, verringert sich der Wert auf 720 Bit zuzüglich dem Verifikator – eine immense Verbesserung, die sich natürlich auch auf die Übertragungszeit positiv auswirkt.

Mit der Senderate von *beacons* von 100 bis 300 ms steht eine obere Zeitgrenze für die Sammlung und Konstruktion der Nachrichteninhalte eines *beacons* fest. Gegeben die in Abschnitt 2.2.5 auf Seite 11 zitierte Übertragungsrate dürfte diese Zeitgrenze sowohl von *NTRU* als auch von *ECDSA* leicht zu erreichen sein. Auf Basis der vorliegenden Daten ist also *ECDSA* der Vorzug zu geben; dennoch müssen sich die beiden Kandidaten erst in weiteren Tests beweisen, wenn die genauen Hardwarespezifikationen vorliegen.

Auch [LRW03], [BJZ05] und [DKKS05] strengen Performance-Vergleiche an – mit größtenteils konformen Resultaten. Weitergehende Informationen zur Wahl von Schlüssellängen sind unter [LV01] zu finden.

4.1.6. Fazit

Die in diesem Kapitel erarbeiteten Basiskonzepte wurden – wenn möglich – sowohl für asymmetrische wie auch für symmetrische Kryptographie ausgelegt und verlangen daher noch nach einer geschickten Kombination und Konsolidierung. Für die Integration in konkrete Infrastrukturgerüste werden in den folgenden Kapiteln Vorschläge der Literatur erläutert und hinsichtlich ihrer Tauglichkeit geprüft.

4.2. Konzepte ohne feste Basisstationen

Nachdem im vorherigen Kapitel bereits allgemeinere, überkonzeptuelle Elemente einer VANET-Sicherheitsinfrastruktur erarbeitet wurden, werden nun in diesem Kapitel konkrete Ansätze der Forschungsliteratur präsentiert, die allein auf dem self-organized-Szenario beruhen, also keiner festen Basisstation bedürfen.

In Ermangelung von Vorschlägen, die direkt auf einen Einsatz in VANETs abzielen, wird mit Konzepten für mobile Ad-hoc-Netzwerke vorlieb genommen. Trotz dieses Umstandes und trotz der Annahme, dass in zukünftigen automobilen Ad-hoc-Netzwerken fest installierte Knoten verfügbar sein werden, können die folgenden Unterkapitel zu einem stimmigen Gesamtkonzept beitragen. Zusätzlich wird die Anwendbarkeit klassischer mobiler Ad-hoc-Netzwerke auf VANETs geprüft.

Basiskonzepte	
Identität	*MANET-ID* oder ähnliche Abbildung pseudonymer, eindeutiger Merkmale
Rollen	Attributzertifikate
Bewegungsprofile erschweren	Pseudonymwechsel im Schutz einer Anonymitätsgruppe
Zurechenbarkeit	Signatur, *MAC*, Einträge des *EDR*
Schlüsselmanagement	Verifikatoren-System (nur *PKI*)
Betreiber	Kraftfahrtsamt (und ein Automobilhersteller-Konsortium)
Kryptographie	Verfahren über elliptischen Kurven bzw. *AES* und *Hmac SHA-1*

Tabelle 4.4.: Basiskonzepte

4.2.1. Verteilte CA

Die in diesem Abschnitt skizzierten Vorschläge versuchen, die Aufgaben einer oder mehrerer zentraler CAs auf mehrere Knoten zu verteilen und bedienen sich u. a. der Schwellenwertkryptographie[36]. Die Knoten selbst werden in allen hier betrachteten Papieren als ursprünglich gleich ausgestattet und berechtigt angesehen.

Verfahren von ZHOU und HAAS

Als Ausgangspunkt ihrer Überlegungen befanden ZHOU und HAAS in [ZH99], S. 4, dass die Verfügbarkeit von CAs in Ad-hoc-Netzwerken nicht durch bloße Replikation gesteigert werden sollte. Dadurch sinke zudem die Sicherheit dieser zentralen Organe.

Stattdessen schlagen sie ein *key management system* vor, das aus mehreren *servern* besteht, von denen jeder ein Schlüsselpaar besitzt – wie jeder „normale“ Knoten auch – und alle öffentlichen Schlüssel im Netz kennt. Es sind n solcher *server* notwendig, um ein Zertifikat zu signieren. Jeder *server* verfügt also über einen Teil des privaten Schlüssels und erstellt damit eine Teilsignatur, die er wie n-1 weitere *server* an einen *combiner*[37] übermittelt. Dabei wird eine bestimmte Anzahl von inkorrekten Teilsignaturen toleriert.

Auf Seite 6 f. zeigen die Autoren, dass der einem *server* eigene Teil des privaten Schlüssels – *share* genannt – durch einen neuen und gleichwertigen ersetzt werden kann. Dies wird durch die Zusammenarbeit der *server* geleistet, ohne dass dabei der private Gesamtschlüssel einem Knoten bekannt wird oder dass der neu erzeugte *share* vom alten, nicht mehr verwendeten ableitbar ist. Diese letzte Eigenschaft ist von großer Bedeutung, da ein solches *share refreshing* vor allem durch das Bekanntwerden eines oder mehrerer *shares* ausgelöst wird.

[36] [MVO96] behandelt ab S. 525 verschiedene *threshold schemes*, ADI SHAMIR stellt in [Sha79] ein eigenes bekanntes Verfahren vor.

[37] Jeder *server* kann die Rolle eines *combiners* übernehmen.

AC-PKI

Anders als beim Verfahren von ZHOU und HAAS bringen die Autoren von [ZLL+05] einen *PKI*-Ansatz vor, der Anonymität und den Verzicht auf Zertifikate verspricht (AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks).

Analog zu den Erkenntnissen in [KKA03] kommt identitätsbasierte Kryptographie zum Einsatz, bei der ein Identitätsmerkmal wie die IP-Adresse, ein eindeutiger Name oder eine E-Mail-Adresse als öffentlicher Schlüssel dient. Er wird also nicht als Teil eines Schlüsselerzeugungsprozesses konstruiert, sondern ist für das Verfahren an sich frei wählbar. Da ein öffentlicher Schlüssel umgekehrt auch eindeutig den Besitzer identifiziert, werden Zertifikate obsolet.

Um zu verhindern, dass Angreifer, die die Identität eines Knotens und damit seinen öffentlichen Schlüssel kennen, auch den passenden privaten Schlüssel berechnen können, wird dieser Vorgang vom Teilnehmerknoten weg zu einer *TTP* verlagert.[38]

Ausgehend von einem geheimen master key berechnet die *TTP* abhängig von der ID eines beantragenden Knotens dessen privaten Schlüssel. Bei einer Komprimittierung dieses master keys wäre also ein Angreifer in der Lage, private Schlüssel zu den ihm bekannten Identitäten ohne Aufwand zu erstellen. Daher verteilt SHAMIRS secret sharing-Technik ([Sha79]) dieses Geheimnis auf mehrere besondere Knoten im Ad-hoc-Netz. Dementsprechend werden einem Antragsteller auf seinen privaten Schlüssel auch nur Teile seines privaten Schlüssels zugesandt, und zwar von einer bestimmten, notwendigen Anzahl solcher *D-PKGs* (Distributed Public Key Generators). Es wird in [ZLL+05], S. 3, gezeigt, dass der Knoten selbst prüfen kann, ob eine *D-PKG* (un)absichtlich einen Manipulationsversuch unternommen hat. Der Knoten ist dann gezwungen, eine „Ersatz“ *-D-PKG* zu finden.

Die Autoren erkannten, dass die verteilten Generatoren von privaten Schlüsseln Angreifern ein Ziel bieten, solange sie unterscheidbar von „normalen“ Knoten sind. Als Abhilfe schlagen sie ein alternatives oder zusätzliches Routingverfahren vor, das von derselben Forschergruppe entwickelt wurde. *MASK* verwendet wechselnde Pseudonyme, um die Identität der Knoten zu schützen (vgl. [ZLLF]).

MOCA

SEUNG YI und ROBIN KRAVETS greifen in [YK] dieses Konzept vollständig auf, integrieren aber zusätzlich den Rückruf von Zertifikaten: Wie auch bei der Ausstellung von Zertifikaten sind eine gewisse Zahl von *MOCAs, MObile Certification Authorities*, notwendig, von denen jede ein teil-signiertes *revocation*-Zertifikat per broadcast verteilt. Die Teilnehmer können also nur dieses Zertifikat für gültig halten, nachdem sie die kritische Zahl von Teilsignaturen empfangen haben.

Des Weiteren liefern die Autoren Möglichkeiten, dieses *MOCA framework* hinsichtlich der Anzahl der benötigten *MOCAs* und routing requests (S. 5 f.) zu verbessern.

[38] Die Autoren sowohl von [ZLL+05] als auch von [KKA03] berufen sich dabei auf das identitätsbasierte Verschlüsselungsverfahren von DAN BONEH und MATT FRANKLIN, vgl. [BF01].

Self-Managed Heterogeneous Certification in Mobile Ad Hoc Networks

In [Li03] wiederum erweitern WEIHONG WANG, YING ZHU und BAOCHUN LI das Prinzip der Schwellenwertkryptographie um drei Aspekte:

- Koexistenz von verschiedenartigen CAs, Knoten haben also individuelle Listen von CAs, denen sie zu einem Zeitpunkt vertrauen.
- Als *D-PKGs* werden neben denen, die in unmittelbarer Nähe des anfragenden Knotens liegen, vor allem auch jene zu Rate gezogen, die nur per multihop zu erreichen sind. Dies soll das Vertrauen in die verteilte *CA* weiter erhöhen.
- Bei jeder kritischen Ende-zu-Ende-Verbindung wird versucht, ein CA-System zu finden, das beide Kommunikationspartner von CAs zertifiziert; d. h. eine Gruppe von CAs, denen sie beide Vertrauen entgegenbringen.

In ihrer Arbeit fordern die Autoren, dass Nachrichten immer über die Kanten eines Vertrauensgraphen ausgetauscht werden. In einem solchen Graphen entsteht eine Kante zwischen zwei regulären Knoten dadurch, dass sie beide nur einen singlehop entfernt sind und zumindest einer bestimmten CA gemeinsam vertrauen.

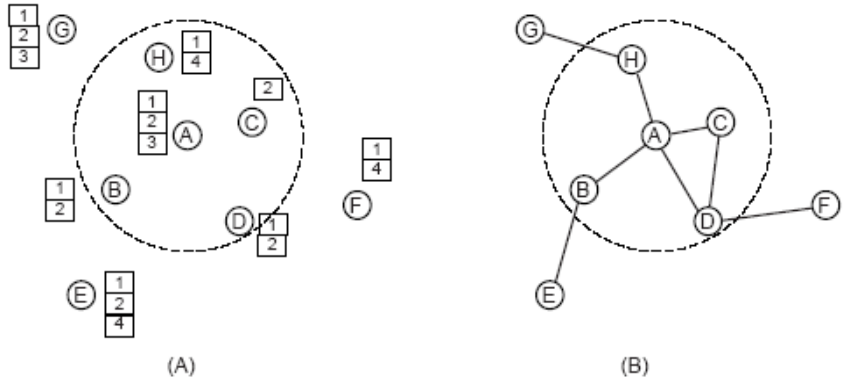

Abbildung 4.3.: (A) zeigt für jeden regulären Knoten (Buchstaben) die Liste ihrer vertrauten CAs (Zahlen), (B) den entstehenden Vertrauensgraphen, sofern es in den CA-Listen der Knoten Gemeinsamkeiten gibt.

Auf der Grundlage dieses Graphen können sich Kommunikationspartner A und B (multihop) gegenseitig als vertrauenswürdig ausweisen, wobei dieses *Distributed Multi-hop Certificate Request (DMCR)*-Protokoll in drei Phasen gegliedert ist:

1. Beide Knoten senden sich jeweils die Liste der CAs zu, denen sie gegenwärtig vertrauen.
2. Verläuft das Überprüfen der ausgetauschten Listen auf Gemeinsamkeiten positiv, so senden sie sich die Zertifikate, die von einer gemeinsamen CA ausgestellt wurden.

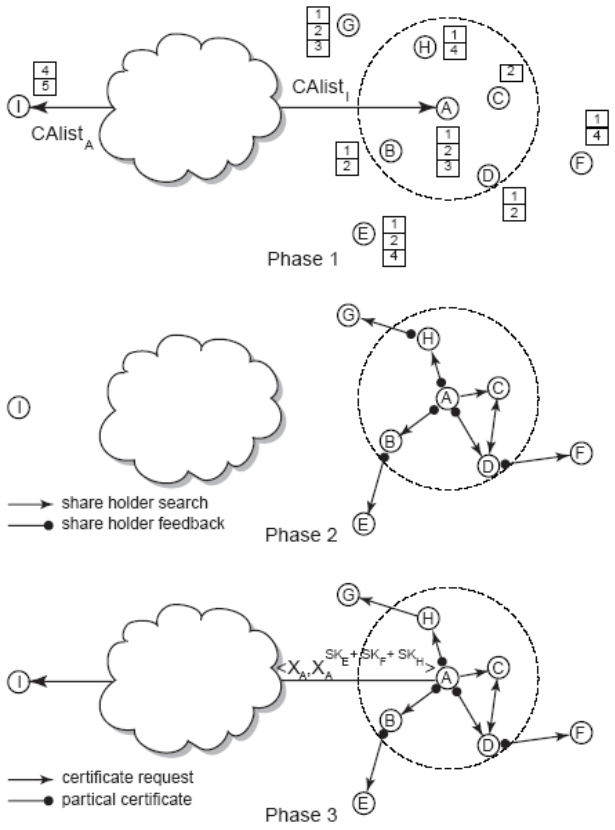

Abbildung 4.4.: Die drei Phasen des *DMCR*

3. Im negativen Fall sucht sich der initiierende Knoten A eine Gruppe von benachbarten Knoten, die mit seinem Kommunikationspartner CAs gemeinsam haben. Erreicht die Mitgliederzahl dieser Gruppe die notwendige Anzahl für die zugrundeliegende Schwellenwertkryptographie, so beantragt er ein Zertifikat von diesen, um damit seine Identität B gegenüber auszuweisen. Diesen Schritt führt auch B aus. Nur wenn beide Zertifikat-requests erfolgreich sind, wird eine Vertrauensbeziehung zwischen A und B erreicht.

Voraussetzung hierfür ist die Unabänderlichkeit und Manipulationsfreiheit der Knotenidentität.

Kritik

Eine zentrale Erkenntnis, die sich durch die Betrachtung der vier verwandten Verfahren ableiten lässt, lautet:

> *Die Unabhängigkeit von festen, zentralen Instanzen steigert die Abhängigkeit eines jeden Knotens von einer Gruppe von anderen und senkt zugleich die Autonomie des einzelnen Knotens und die Zuverlässigkeit des Gesamtsystems.*

Jedes Mal, wenn der private Schlüssel des Systems benötigt wird, muss sich eine Zahl an vertrauenswürdigen Schlüsselteilhabern finden, um diesen zu erstellen. Der Prozess kann dabei auf verschiedene Art und Weise dermaßen beeinträchtigt werden, dass Signaturen und Entschlüsselungen für eine Zeitspanne verhindert werden:

- Es sind nicht genügend *server* in Reichweite.
- Angreifer leiten Pakete von *servern* nicht weiter oder tauschen sie gegen Datenmüll aus.
- Natürliche Paketverluste provozieren einen time out des Prozesses.

Auch wenn alle Voraussetzungen erfüllt sind, sind die vorgestellten Verfahren hinsichtlich AN2 nur als ungenügend zu bewerten; schon grundsätzlich stellt Schwellenwertkryptographie ein sehr aufwändiges Verfahren dar, das enorme Nachrichtenaufkommen disqualifiziert sie schließlich ganz.

Als weiterer gravierender Nachteil ist die fehlende initiale Identifizierung der Knoten zu nennen, hier macht sich das Fehlen einer zentralen Instanz sofort bemerkbar. Aus gleichem Grunde sind auch alle Bemühungen, ein *revocation system* auch ohne zentrale *TTP* zu konstruieren, als nicht effektiv zu betrachten.

Dass eine Auswahl an regulären Knoten die besondere Rolle von *servern* einnimmt, wirft weitere Probleme auf, z. B. die geschickte Auswahl an Knoten für diese Aufgabe. Es müsste ja gesichert sein, dass sie sich in immer verfügbarer Weise regional verteilt halten. Dies ist – im Gegensatz zu Basisstationen – nicht realisierbar. Aus diesen Gründen scheiden alle hier vorgestellten Ansätze für den Einsatz in VANETs aus.

4.2.2. web of trust - ad hoc

In der Gruppe um HUBAUX entstanden mit [HBC01] und [CBH03] zwei sehr ähnliche Ansätze, die das *web of trust* auf Ad-hoc-Netzwerke transferieren.

Obwohl bei PGP bzw. GnuPG bereits ein hoher Grad an Selbstorganisation herrscht, sorgen zentrale Schlüsselserver für die Verteilung von Teilnehmerschlüsseln. Im Vorschlag dieser Forschergruppe wird dieser Dienst ersatzlos in die Hände der Teilnehmer gelegt.

In ihrem Modell werden Zertifikate und ihre Inhaber als Zertifikatsgraph, einem gerichteteten Graphen *G(V,E)*, dargestellt, wobei *V* die Menge der Teilnehmerknoten (vertex) und *E* die Menge der Kanten (edges) bezeichnen. Diese edges stellen nichts anderes dar als die durch ein Zertifikat bestätigte Beziehung zwischen zwei Knoten – vertices.

Beispiel:
Knoten X stellt Knoten Y ein Zertifikat aus: $X \longrightarrow Y$

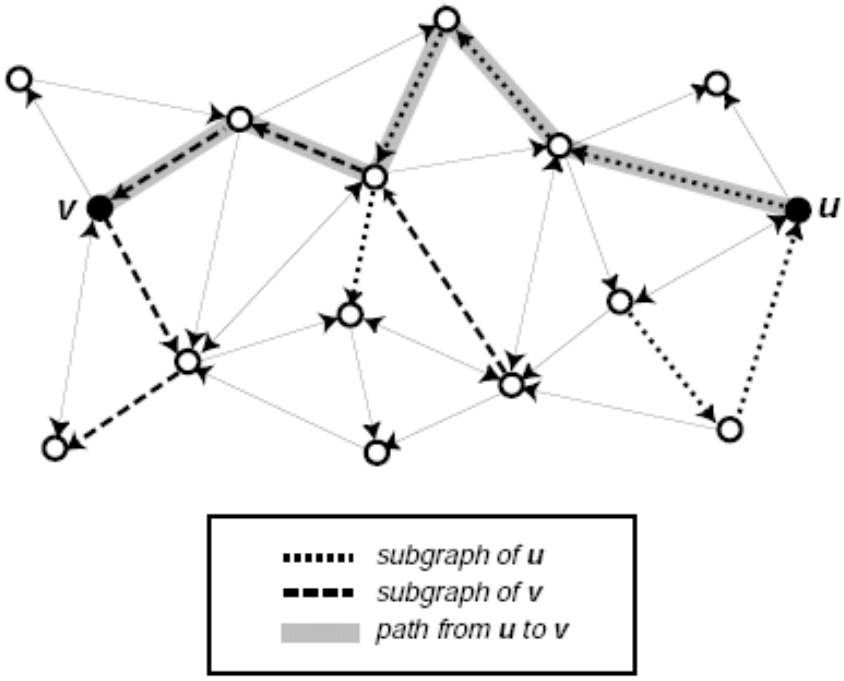

Abbildung 4.5.: Suche einer Vertrauensbeziehung zwischen zwei Knoten u und v

In einem lokalen Schlüsselspeicher jeden Knotens befinden sich alle Zertifikate, die er selbst anderen ausgestellt hat, und zusätzlich ausgewählte Zertifikate, die ihm von anderen Teilnehmern ausgestellt worden sind (vgl. [HBC01], S. 6 und [CBH03], S. 5). Die Menge der letzteren Zertifikate nennt HUBAUX *subgraph*.

Wenn nun zwei Knoten u und v am Anfang ihrer Kommunikationsbeziehung ihre zugesandten Zertifikate prüfen wollen, legen sie ihre *subgraphs* zusammen und versuchen, eine Zertifikatskette von einem Knoten zum anderen zu finden; dies veranschaulicht Abbildung 4.5.

Bei entsprechend großen Netzen und der Vorgabe eines beschränkten Zertifikatsspeichers sind Kommunikationspartner nur zu einer bestimmten Wahrscheinlichkeit in der Lage, sich gegenseitig zu authentifizieren. Der von den Autoren vorgestellte *Shortcut Hunter*-Algorithmus profitiert vom small worlds-Phänomen, das sie bereits in [CBH02] für PGP-Graphen untersuchten.[39] Hierin spielen Abkürzungen, *shortcuts*, die tragende Rolle und werden analog zu [HBC01], S. 7, definiert:

> *Ein* shortcut *ist definiert als die Kante zwischen zwei Knoten, nach deren Entfernen der kürzeste Pfad mindestens die Länge drei hat.*

Gemäß seinem Namen baut nun dieser Algorithmus für einen Knoten iterativ einen optimalen *subgraph* auf, indem er bevorzugt diesen Abkürzungen folgt. Dies entspricht seinen Designprin-

[39] In [Wat99] fand DUNCAN WATTS heraus, dass in Netzen der kürzeste Pfad zwischen zwei beliebigen Knoten erstaunlich kurz ist, obwohl die Netzstruktur vor allem lokale Häufungen – Cluster – und nur wenige „Fernverbindungen“ aufweist. Er führt dabei die Arbeit [Mil67] entscheidend weiter.

zipien, bei vorgegebener Zertifikatsanzahl möglichst große „Netzabdeckung“ zu erreichen. Im nächsten Schritt fordert er genau die Zertifikate an, deren Inhaber die Knoten des errechneten *subgraphs* sind (vgl. auch [Kar04], S. 117 ff.).

Kritik Anders als in [HBC01] werden in [CBH03] nicht nur die Grundprinzipien erläutert, sondern auch Aussagen über die Initialisierung des Netzes und das Management des Zertifikatlebenslaufs gemacht:

1. Das Schlüsselpaar wird vom Knoten selbst erzeugt. Dass dieser Umstand bei unehrlichen Benutzern, die z. B. ihren öffentlichen Schlüssel nicht ihrer eigenen Identität zuordnen lassen wollen, das Vertrauen in das *web of trust* untergräbt, stellen die Autoren in beiden Papieren fest ([HBC01], S. 8 f., [CBH03], S. 6 f.).

   Der Einsatz von Metriken, um diese mit einer gewissen Wahrscheinlichkeit herauszufinden, setzt an der falschen Stelle an: Ein Knoten sollte systembedingt gar keine Möglichkeit haben, sich unter einer falschen Identität auszugeben.

2. Der kontinuierliche Austausch von Zertifikaten zwischen den Teilnehmern des Netzes fällt sehr aufwändig aus (vgl. auch [Kar04], S. 110). Aufgrund der hohen Mobilität von VANET-Teilnehmern und der begrenzten Gültigkeit von Zertifikaten müsste der vorgeschlagene Algorithmus periodisch ausgeführt werden, ohne dabei die Garantie zu haben, dass er aufgrund der kurzen Verbindungszeiten überhaupt erfolgreich terminiert.

   Zusätzlich dürfte den Initialisierungsprozess die Eigenschaft verlangsamen, dass die Authentizität immer nur mit einer bestimmten Wahrscheinlichkeit geprüft werden kann.[40]

   Allgemein scheint die Abhängigkeit vom *Shortcut Hunter*-Algorithmus keineswegs den Echtzeitanforderungen in VANETs zu entsprechen; man stelle sich vor, man müsse bei einer Warnung aus der Gruppe A1 oder A2 zunächst langwierig prüfen lassen, ob die Nachricht authentisch ist, womöglich kommt der Algorithmus wegen fehlender Pfade zu überhaupt keinem Ergebnis.

3. CAPKUN unterscheidet in seinem vorgestellten *certificate revocation system* zwischen *explicit revocation* und *implicit revocation*. Der erste Begriff meint den aktiven Rückruf eines Zertifikats, indem der Aussteller eine Aufforderung zum Rückruf sendet. Dank der vorher beschriebenen Vorgehensweise zur beschränkten Speicherung von Zertifikaten muss diese Aufforderung nur an diejenigen Teilnehmer geschickt werden, die das betreffende Zertifikat tatsächlich in ihrem Speicher gelistet haben. Der implizite Rückruf tritt in Kraft, wenn das Zertifikat seine zeitlich befristete Gültigkeit verliert.

   Hier wirkt sich das Fehlen eines zentral organisierten Zertifikatrückrufs besonders drastisch auf die Umsetzgeschwindigkeit aus; erst wenn bestimmte Knoten Kontakt zu einander haben, setzen sich die *revocation requests* durch. Zudem wird nicht geklärt, ob und

[40] Es ist durchaus vorstellbar, dass die beiden letztgenannten Effekte den Initialisierungsprozess zum erliegen bringen und menschliches Eingreifen, d. h. vermehrtes Ausstellen von Zertifikaten, notwendig wird.

welche Maßnahmen ergriffen werden können, wenn ein Knoten sich weigert, die *revocation* durchzusetzen.

Natürlich bietet die zugrundegelegte asymmetrische Kryptographie genügend Potential die Schutzziele der Integrität zu erfüllen. Wie bereits in den Kapiteln 3.2.1 auf Seite 20 und 4.1.1 auf Seite 46 festgestellt wurde, läuft die Zuordnung eines einzigen Zertifikats zu einem Knoten Gefahr, die Erstellung von Bewegungsprofilen zu begünstigen.

Schlägt man für dieses System pro Nutzer mehrere pseudonyme Zertifikate (und Schlüsselpaare) vor, so steigert sich drastisch das Performance-Problem.[41]

Durch den vollkommenen Verzicht auf zentrale Strukturen und die Gleichberechtigung aller Knoten fallen natürlicherweise auch starke Angreifer (zumindest AM4) weg und die Kosten für den Aufbau und den Unterhalt von zentralen Instanzen. Doch sind bereits Angreifer vom Format AM1 in der Lage, erhebliche Störungen zu verursachen, indem sie unter einer falschen Identität zertifiziert werden.

KARGL bemerkt in seiner Bewertung dieses Verfahrens, dass das manuelle Signieren von Zertifikaten die Nutzer überfordert; zurecht, denn eine digitale Signatur sagt in diesem Entwurf mehr aus: Sie leistet nämlich die Aussage, dass der vorliegende, zertifizierte Knoten selbst in der Lage ist, andere zu prüfen und ihnen gültige Zertifikate auszustellen (vgl. [Kar04], S. 109). Konzepte, die in üblichen *PKIs* und auch in PGP strikt und absichtlich getrennt sind, werden hier also vermischt, was selbst schwachen Angreifern Tür und Tor öffnet.

Diese grundsätzliche Designschwäche, die vorauszusehenden Performance-Schwächen, die geringe Resistenz gegenüber der Bildung von Bewegungsprofilen und das durchsetzungsschwache *revocation system* schließen dieses Verfahren von einer Anwendung in automobilen Ad-hoc-Netzwerken aus.

4.2.3. Verfahren basierend auf symmetrischer Kryptographie

neighborhood key method

Ihren Vorschlag *neighborhood key method* ([LD05]) betten die Autoren JÖRG LIEBEHERR und GUANGYU DONG in ein *Overlay Software System*, dem *HyperCast*, ein; die genauen Spezifikationen und Konzepte hierfür berühren die Sicherheitsbetrachtungen nicht und werden in diesem Rahmen auch nicht behandelt.

Mit diesem Verfahren soll den Schutzzielen Vertraulichkeit und Integrität in Ad-hoc-Netzwerken (self-organized) entsprochen werden; das Modell ist laut ihren Ausführungen auch auf hybrid-Netze ausdehnbar.

[41] Wenn die Pseudonym-Wechsel wie in Abschnitt 4.1.1 auf Seite 46 verlaufen, wird der *Shortcut-Hunter*-Algorithmus bei allen beteiligten Knoten ausgelöst.

Obwohl für den Nachrichtenaustausch symmetrische Verschlüsselung und *MACs* Vertraulichkeit und Integrität leisten sollen, sichert asymmetrische Kryptographie die Verteilung der dazugehörigen symmetrischen Schlüssel ab. Auch die Authentifizierung von Teilnehmern basiert auf dem Austausch von Zertifikaten.

Jeder Knoten besitzt also ein Schlüsselpaar und ein Zertifikat, das von einer *TTP* ausgestellt wurde; zugleich ist er stets Mitglied einer bestimmten *neighborhood*, einer Gruppe von Teilnehmerknoten. Die vorher angesprochenen symmetrischen Schlüssel sind allen Mitgliedern zugänglich, sofern sie sich authentifiziert haben.

Das Verteilen von *KeyUpdate*-Nachrichten, die diesen symmetrischen *neighborhood key* in verschlüsselter und signierter Form enthalten, wird von folgenden Ereignissen ausgelöst:

- Ein neuer authentifizierter Knoten tritt der *neighborhood* bei.
- Ein authentifizierter Knoten sendet eine entsprechende Anfrage.
- Ein authentifizierter Knoten verlässt die *neighborhood*.
- Eine den Nachrichten beigefügte Sequenznummer, der die Gültigkeit der *neighborhood keys* begrenzt, erreicht das definierte Maximum.

So wird also sichergestellt, dass jeder Knoten einer *neighborhood* stets über die aktuellen symmetrischen Schlüssel verfügt, um Nachrichten zu ent- und zu verschlüsseln und *MACs* zu erzeugen und zu prüfen.

multihop-Nachrichten zu übertragen bedeutet in diesem System, Nachrichten zwischen *neighborhoods* zu übersetzen. Es wird nun vorausgesetzt, dass jeder authentifizierte Knoten die *neighborhood keys* seiner Nachbarn kennt. Dies bedeutet jedoch, dass an jedem Knoten die Nachricht mit dem *neighborhood key* des letzten Absenders entschlüsselt[42] und dem des nächsten Empfängers verschlüsselt[43] werden muss. Es kann an dieser Stelle nicht nachvollzogen werden, welches Ziel die im nächsten Schritt eingeführten *message keys* verfolgen. Jede Nachricht soll nämlich mit einem neu erzeugten *message key* vor der Verschlüsselung mit dem *neighborhood key* verschlüsselt werden. Dieser neue symmetrische Schlüssel wird in verschlüsselter Form der Nachricht beigelegt.

Kritik Das *neighborhood key*-Prinzip ist ein gutes Beispiel dafür, dass Sicherheitsmaßnahmen, die für mobile Ad-hoc-Netzwerke konzipiert sind, nicht ohne Weiteres auf automobile Ad-hoc-Netzwerke anwendbar sind. Auch stellvertretend für andere Sicherheitsentwürfe, die Gruppen oder Cluster einführen (vgl. [BHM+02], [HL04]), ist anzunehmen, dass aufgrund der ständig wechselnden Gruppenzugehörigkeiten sehr viel Bandbreite verloren geht. In diesem Hintergrund nehmen natürlich auch die Zertifikatsprüfungen und die *KeyUpdate*-Nachrichten eminent zu. Die

[42] Dies betrifft nicht den ursprünglichen Absender einer Nachricht.
[43] Dies gilt nicht für das Nachrichtenziel.

propagierten Ver- und Entschlüsselungen bei *neighborhood*-Übergängen kosten zudem Performance (AN2)

Weiterhin ist anzumerken, dass Vertraulichkeit nur bezüglich der Gruppe, nicht für zwei dezidierte Kommunikationspartner vorgesehen ist (Schutzziel V2). Auch die Zurechenbarkeit auf einen bestimmten Nachrichtenurheber ist nicht gegeben, selbst auf Gruppenebene ist dies nur so lange möglich, wie der *neighborhood key* gültig ist (Schutzziel I1).

Ein *revocation system* wie das Verifikatoren-basierte aus Kapitel 4.1.3 auf Seite 54 könnte zwar leicht implementiert werden. Es wäre jedoch geschickter und den Anforderungen angemessener, die in jedem Fall notwendige *PKI* auf die Sach- und Schutzziele anzupassen als auf diesen Ansatz zu vertrauen.

Trotzdem dient die Idee der *neighborhood keys* als Grundlage für die Erweiterung des Verfahrens von JONG YOUL CHOI, MARKUS JAKOBSSON und SUSANNE WETZEL in Abschnitt 4.3.1 auf Seite 84.

Pairwise Keys

In [ZXSJ03] wird das *Pairwise Key Establishment Protocol* vorgestellt, das Schwellenwertkryptographie mit einer probabilistischen Schlüsselverteilung kombiniert, die folgendermaßen von den Autoren skizziert wird:

In einer Initialisierungsphase wählt ein Schlüsselspeicher eine bestimmte Anzahl symmetrischer Schlüssel aus einer sehr großen Menge und händigt sie dem neuen Teilnehmer aus. Diese Schlüsselverteilung wird dahingehend optimiert, dass ein beliebiges Teilnehmerpaar zumindest einen dieser Schlüssel mit einer hohen Wahrscheinlichkeit gemeinsam hat. Fortan wird zwischen folgenden beiden Knotenbeziehungen unterschieden:

1. *direct paths*: Zwei Knoten besitzen einen gemeinsamen vorverteilten Schlüssel.
2. *indirect paths*: Zwei Knoten können über eine Kette von *direct paths*, die einen oder mehrere Knoten als sog. *proxies* involviert, eine Kommunikationsbeziehung aufbauen.

Da beim Vorliegen eines *direct path* nicht ausgeschlossen werden kann, dass auch ein dritter Teilnehmer den gemeinsamen Schlüssel besitzt, muss auf andere Weise ein session key erstellt werden.

Hier findet nun die zweite Komponente dieses Vorschlags ihre Anwendung. Anstatt den privaten Schlüssel einer CA auf mehrere Instanzen zu verteilen (vgl. Kapitel 4.2.1 auf Seite 71), wird hier der vom Sender erzeugte session key in Schlüsselteile zerlegt und auf unterschiedlichen *direct paths* und *indirect paths* an den Empfänger übermittelt. Dieser ist nach Erhalt einer ausreichenden Menge an Schlüsselteilen in der Lage, den session key zu rekonstruieren und ihn für die Kommunikation mit dem Sender verwenden.

Kritik Dieses Beispiel zeigt offenkundig, dass allgemein probabilistische Schlüsselverteilverfahren in automobilen Ad-hoc-Netzwerk denkbar ungeeignet sind.[44] Bei der Größe von VANETs müsste jeder Knoten eine sehr hohe Anzahl an vorinstallierten Schlüssel speichern, um die Wahrscheinlichkeit eines gemeinsamen Schlüssels auf einem hohen Niveau zu halten. Dann aber verursacht bei diesen flüchtigen Netzstrukturen allein die Suche nach gemeinsamen Schlüsseln so viel Netzlast und Zeitaufwand, dass zeitkritische Nachrichten nicht übermittelbar sein würden.

Hinzu kommt, dass der empfangende Knoten warten muss, bis eine bestimmte Anzahl von Schlüsselteilen bei ihm eintrifft und er überhaupt den Schlüssel wiederherstellen kann. Genau dieser Prozess kann durch netzbedingte Paketverluste und *proxies*, die die *shares* nicht oder in manipulierter Form weiterleiten, so stark beeinträchtigt werden, dass nicht nur die Echtzeitanforderungen nicht erfüllt werden können, sondern gar kein session key ausgetauscht werden kann.[45]

Außerdem besteht immer die Möglichkeit, dass nicht genügend *(in)direct paths* verfügbar sind, um überhaupt die erforderliche Anzahl an *shares* abzudecken.[46]

4.2.4. Fazit

Von allen in diesem Kapitel vorgestellten Verfahren kann nur die *neighborhood key*-Methode einen Beitrag zu einem stimmigen Konzept (vgl. Kapitel 4.3.1 auf Seite 88) liefern. Alle anderen scheiden schon früh als Variante aus, da

- ihre Funktionsweise meist nur zu einer gewissen Wahrscheinlichkeit verfügbar ist,
- oft nicht klar ist, wie die ersten Schritte beim Aufbau eines VANETs („bootstrapping“) zu leisten sind,
- in keinem Fall die Schutzziele (AN1) die Echtzeitanforderungen (AN2) zur Gänze erfüllbar sind.

4.3. Konzepte mit festen Basisstationen

Dieses Kapitel dominieren ein Vorschlag auf Basis einer traditionellen *PKI* und ein Ansatz, der hauptsächlich auf symmetrische Kryptographie setzt. Während es für die erste Lösung bereits in Kapitel 4.1 auf Seite 45 möglich war, Elemente zu konkretisieren, wird die zweite Möglichkeit ausführlich dargelegt, diskutiert und erweitert. Zusätzlich wird auch ein auf *Kerberos* basierender Vorschlag ausgeführt.

[44] [LHE05] stellt eine weiteres Verfahren dieser Gattung dar, das sich von seiner Grundauslegung her nicht für VANETs eignet und in dieser Arbeit nicht betrachtet wird.

[45] Auch RAYA fordert in [RH05a], S. 5, dass die Authentifizierung in automobilen Ad-hoc-Netzwerken in jedem einzelnen Fall und in einem engen Zeitrahmen durchführbar sein muss.

[46] Dies ist zum Beispiel der Fall, wenn gar nicht genügend Fahrzeuge in Reichweite sind oder diese zu wenig gemeinsame Schlüssel besitzen.

Um den Rahmen der Arbeit nicht zu sprengen, wurde auf die Darstellung der Verfahren aus [WW03a] und [WW03b] verzichtet, die in diesem Stadium nicht für VANETs ausgelegt sind und überdies nur den Teilaspekt der Authentifizierung behandeln.

4.3.1. Verfahren basierend auf symmetrischer Kryptographie

Im Kapitel 3.2.2 auf Seite 28 wurden die Echtzeitanforderungen von VANETs formuliert. Genau diese geforderte Kompaktheit, Berechnungsgeschwindigkeit und die Resistenz gegenüber Kryptoanalysen ist die Domäne symmetrischer Kryptographie. Wie es gelingt, die Flexibilität von *PKIs* nachzuahmen und dabei die genannten Vorteile zu bewahren, zeigt dieses Kapitel. In manchen der vorgestellten Ansätze kommt für Verwaltungsaufgaben asymmetrische Kryptographie zum Zuge, wenn dies mit symmetrischer Kryptogpraphie allein zu umständlich zu lösen wäre; darauf wird in der Beschreibung explizit hingewiesen. Zusätzlich zur Bewertung der Verfahren werden ggf. Vorschläge zur Verbesserung oder Erweiterung angefügt.

KAMAN

Mit *KAMAN* stellen ASAD AMIR PIRZADA und CHRIS MCDONALD eine nach [CKM00] modifizierte Kerberos-Variante für Ad-hoc-Netzwerke vor. Hier ersetzt eine Authentifizierungseinheit bestehend aus mehreren replizierten Servern die Zweistufigkeit des ursprünglichen Kerberos 5-Prozesses, der aus einem *Authentication Server AS* und einem *Ticket Granting Server TGS* besteht.

Folgende drei Annahmen treffen die beiden Autoren, die die Basis für das Kernprotokoll dieses Vorschlags bilden:

1. Jeder Teilnehmer besitzt einen geheimen Schlüssel (alternativ ein Passwort).
2. Jeder Server speichert die
   - *ID* eines Teilnehmers,
   - die Hash-Abbildung des Teilnehmerschlüssels,
   - einen Wert *priority* abhängig vom Vertrauen, das dem Teilnehmer entgegengebracht wird,
   - einer Wert *lifetime*, der je nach Grundsicherheit des Netzes die Gültigkeitsdauer des Teilnehmerschlüssels widerspiegelt.
3. Jeder Server teilt mit den anderen einen gemeinsamen geheimen Schlüssel.

Das eigentliche Protokoll, das über einen Authentifizierungsprozess bei einem Server zwei Kommunikationspartner mit einem gemeinsamen session key versorgt, besteht aus vier Phasen. In der

ersten schickt der initiierende Teilnehmerknoten an einen Server eine Ticketanfrage, die u. a. ein Transaktionskennzeichen und die *IDs* der beiden beteiligten Knoten enthält.

Aufgrund dieser beiden Teilnehmeridentifikatoren kann der Server den *lifetime*-Wert beider Knoten überprüfen. Falls beide Werte noch nicht abgelaufen sind, generiert er einen session key und eine Nachricht an den ersten Knoten, die zwei Bestandteile aufweist:

- Das eigentliche Ticket, das neben Zusatzinformationen den neuen gemeinsamen Schlüssel und die Gültigkeitsdauer dieses Tickets aufnimmt. Dieses Ticket ist für die Weiterleitung an den zweiten Knoten bestimmt und deshalb schon ab dem Server mit dessen geheimen Schlüssel verschlüsselt.
- Der neue gemeinsame Schlüssel, dessen Gültigkeitsdauer und Zusatzinformationen, verschlüsselt mit dem geheimen Schlüssel des ersten Knotens.

Im dritten Schritt überreicht der erste Teilnehmer dem zweiten das Ticket, der im vierten und letzten Schritt einen time stamp – bereits mit dem gemeinsamen Schlüssel chiffriert – zurücksendet.[47]

Dieser Schlüsselaustausch wird schon in Phase 2 abgebrochen, wenn der *KAMAN*-Server aufgrund der ihm vorliegenden *lifetimes* feststellt, dass ein oder beide Schlüssel der Beteiligten abgelaufen oder zurückgezogen wurden. Dies kommt zumindest für den Anwendungsbereich des verschlüsselten Datenaustausch einem *revocation system* gleich.

Kritik Durch die Verwendung einer schlankeren Kerberos-Modifizierung konnte ein Teil des Overheads konventioneller Kerberos-Implementierungen vermieden werden. Der Schutz der Vertraulichkeit (V1,2) kann jedoch nur dann aufrechterhalten werden, wenn tatsächlich jede Nachricht mit einem session key verschlüsselt wird. Durch dies und einen Hash über jeder Nachricht soll auch implizit das Schutzziel der Integrität abgedeckt werden. Tatsächlich dürfte es für Angreifer AM1-3 unmöglich sein, den session key zu ermitteln, eine Nachricht abzufangen, zu verändern und verschlüsselt an den eigentlichen Adressaten zu leiten. Die Zurechenbarkeit ist wie in vielen Modellen, die auf symmetrischer Kryptographie beruhen, nur auf beide Kommunikationspartner und eventuell den beteiligten Server (AM4) einschränkbar.

Gerade *beacons* als die zentralen Elemente eines VANETs müssen in einer geeigneten Sicherheitsinfrastruktur möglichst autonom von jedem Teilnehmer hinsichtlich der Integrität und Zurechenbarkeit abzusichern sein. Genau dies erlaubt das vorliegende Verfahren nicht: Sämtliche multicast- und geocast-Techniken sind dahingehend nicht praktikabel abzusichern, der Schutz unverschlüsselter Nachrichten ist gar nicht vorgesehen.

Weiterhin sind – nunmehr weniger bedeutsam – als Nachteile zu nennen, dass keine unterschiedlichen Teilnehmerrollen angedacht oder implementierbar sind und die beiden Eigenschaften *priority* und *lifetime* nicht fundiert bestimmbar erscheinen.

[47] Die detaillierten Nachrichteninhalte sind in [PM04], S. 3, zu finden.

Verfahren von Choi, Jakobsson und Wetzel

Schon der Titel von „Balancing Auditability and Privacy in Vehicular Networks“ ([CJW05]) bestätigt zwei der ermittelten Schutzziele, V1-4 und I1. Dies soll im Modell dieses Vorschlags durch eine starke Infratstrukturunterstützung gewährleistet werden: Zum Einen sind dies die *front-end authorities*, genannt *Base Stations BS*, die mit den Teilnehmerknoten in direkter Kommunikation stehen. Zum Anderen bleibt die *back-end authority*, *Ombudsman*, im Hintergrund und kommuniziert nur mit den *BS*. Ihre Aufgabe besteht darin, bei Bedarf und rechtlichem Hintergrund die Beziehung zwischen Teilnehmerpseudonymen und ihren wahren Identitäten zu enthüllen.

Setup Die Teilnehmerknoten, *Nodes*, besitzen eine Reihe von Parametern:

- eine eindeutige Identifikationsnummer ID,
- einen seed-Wert zur Generierung von Pseudonymen SD,
- eine Menge an Kurzzeitpseudonymen PS mit korrespondierenden session keys KS, die der Knoten selbständig innerhalb eines kleinen Zeitintervalls t (etwa minütlich) wechselt,
- eine Menge an Langzeitpseudonymen HD, *handle* genannt, die der Knoten selbständig innerhalb eines längeren Zeitraums T (etwa jeden Tag) wechselt,
- einen Speicher DB_N, der gleich einem *EDR* Nachrichtentupel aufnimmt. Ein Datenbankeintrag besteht dabei aus der lokalen Empfängerzeit, dem Pseudonym des Senders, der Nachricht und der lokalen Senderzeit.

Der *Ombudsman* besitzt seinerseits

- einen Speicher DB_{OM}, der Tupel von Knoteninformationen (HD, ID, T, SD) aufnimmt,
- eine öffentlich verfügbare Einwegfunktion F_{OM}, mit der aus IDs HDs generiert werden.

Folgende Parameter weist jede *BS* auf:

- einen öffenlichen Schlüssel PK und einen dazugehörigen privaten Schlüssel SK
- eine öffentlich verfügbare Einwegfunktion F_{BS}, mit der aus HDs PSs generiert werden
- eine Datenbank DB_{BS}, die während des im Anschluss beschriebenen Initialisierungsvorgangs zwischen N und BS die Daten PS, KS, t, HD des Knoten speichert.
- einen weiteren Speicher, der genau DB_N entspricht.

Modell In der Registrierungsphase eines jeden Knotens N wählt *Ombudsman* zufällig die Werte ID und SD, die er auf einem sicheren Kanal auch dem Knoten zukommen lässt. Beide, *Ombudsman* und der Knoten, sind von nun an übereinstimmend und nachvollziehbar in der Lage, HDs zu generieren:

$$\mathsf{HD} = \mathsf{F}_{\mathrm{OM}}(\mathsf{ID},\mathsf{SD},\mathsf{T})$$

T ist dabei eine systemweit festgelegte Zeitspanne. Eine gewisse Menge dieser *handles* legt *Ombudsman* in seiner $\mathsf{DB}_{\mathrm{OM}}$ ab.

Nach der erfolgreichen Registrierungsphase ist ein Knoten bereit, am VANET teilzunehmen. Dazu muss er sich gegenüber einer *BS* in Reichweite authentifizieren und übermittelt hierfür das für T korrekte *handle*, verschlüsselt mit PK.

Die *Base Station* entschlüsselt das *handle* mit SK und reicht es bei besonders kritischen Anwendungsfällen an *Ombudsman* weiter, um dessen Gültigkeit überprüfen zu lassen.

Analog zur Registrierungsphase fließen nun in die Einwegfunktion F_{BS} das übermittelt *handle* und t ein:

$$\mathsf{O} = \mathsf{F}_{\mathrm{BS}}(\mathsf{HD},\mathsf{t})$$

Die entstehende Bitkette O, die davon unabhängig auch der Knoten berechnet, enthält sowohl den session key KS, als auch das Kurzzeitpseudonym PS. Abhängig vom Kurzzeitintervall t liefert diese Funktion also eine Menge von PS und KS, die in den Speichern DB_{N} und $\mathsf{DB}_{\mathrm{BS}}$ vorgehalten werden.

Zusammenfassend kann also festgestellt werden, dass

- *Ombudsman* allein die Zuordnung Knotenidentität – Langzeitpseudonyme (*handles*) kennt,
- die *Base Stations* für jeden Knoten das *handle* und alle Kurzzeitpseudonyme PS der Periode T kennen und abfragbar gespeichert haben.
- die Knoten unabhängig von den Basisstationen ihre Kurzzeitpseudonyme und die dazugehörigen session keys berechnen können.

Ablauf der Kommunikation Bei jeglicher Kommunikation in diesem Modell verwendet ein Knoten niemals seine Langzeit- sondern stets seine Kurzzeitpseudonyme PS, über denen zusammen mit dem korrespondierenden session key KS, der lokalen Zeit LT und der Nachricht M ein *MAC* gebildet wird. Ein typisches Nachrichtenpaket sieht also folgendermaßen aus:

$$\mathsf{P} = \mathsf{PS}|\mathsf{LT}|\mathsf{M}|\mathit{MAC}$$

Die Autoren JONG YOUI CHOI, MARKUS JAKOBSSON und SUSANNE WETZEL unterscheiden dabei zwischen zwei Kommunikationsarten: v2v und v2i. In der ersten speichert der empfangene Knoten die Nachricht ab und leitet sie später bei Bedarf an eine Basisstation weiter. Eine Prüfung des *MAC* ist hier nicht vorgesehen, da dem diesem Teilnehmer sonst der aktuelle session key ausgehändigt werden müsste.

Eine Verifikation ist also den *BS* vorbehalten (v2i), da sie sind ja in der Lage sind, PS und KS eines Teilnehmers in ihrer Datenbank DB_{BS} nachzuschlagen. Für die umgekehrte Senderichtung, von der Basisstation zum Knoten, werden dieselben Datenbankeinträge benutzt, um einem Teilnehmer ein Nachrichtenpaket konform zu diesem Schema zukommen zu lassen.[48]

Kritik Die selbstgesteckten Ziele, einen Spagat zwischen Vertraulichkeit und Verfolgbarkeit bzw. Zurechenbarkeit, erreicht die vorgeschlagene Lösung zunächst gut und in einer leichtgewichtigen Form. Die umfangreiche Datenspeicherung bei den Basisstationen und bei *Ombudsman* erlaubt einerseits eine genaue Verfolgung aller Knoten in pseudonymer Form. Andererseits können erst im Zusammenspiel mit *Ombudsman* die wahren Teilnehmeridentitäten enthüllt werden.

Diese Trennung in Verbindung mit den Kurz- und Langzeitpseudonymen wirkt sich vorteilhaft auf das Schutzziel der Vertraulichkeit aus:

Trotz der klar vorhersehbaren Zeitintervalle für den Wechsel der Kurzzeitpseudonyme gewinnen selbst starke Angreifer AM4,5 wenig verwertbare Informationen, da sie damit noch nicht auf die Identität schließen können. Selbst wenn eine Basisstation in Angreiferhände fällt, verfügt er pro Teilnehmer nur über ein *handle*; er kann daher keine weiteren vorhersagen (Einwegfunktion F_{BS}) und Teilnehmeraktionen nur im Zeitraum T zuordnen.

Wenn nun ein Teilnehmer diese eine angreifende Basisstation in Zeitabständen größer als T passiert, so ist diese Verkettung für jene BS nicht nachvollziehbar. Im vorliegenden Modell ist problematischerweise von einem Handover die Rede, in dessen nicht näher erläuterten Verlauf Basisstationen Statusinformationen und das *handle* eines Knotens austauschen, wenn er den Reichweitenbereich des einen verlässt und den eines anderen betritt ([CJW05], Seite 6). Wenn es nun Angreifern gelänge, mehrere benachbarte Basisstationen zur gleichen Zeit unter ihre Kontrolle zubringen, könnten weitreichendere Profile erstellt werden. Wie schon erwähnt, beziehen sich diese weiterhin auf die Langzeitprofile.

Angreifer AM3 oder niedriger erlangen nicht einmal Kenntnis über das derzeitige *handle*.[49] *Ombudsman* als eigentliche *TTP* bleibt in den meisten Fällen offline, kommuniziert ausschließlich mit den *BS*[50] und bietet wenig Angriffsfläche.

Überhaupt überzeugen die schlanken Protokolle und die für alle Beteiligten autonome Generierung der Pseudonyme (AN2). Dennoch gibt es einige Punkte, eine Verbesserung oder Erweiterung an diesem an sich durchdachten und realisierbaren Modell rechtfertigen:

1. Im Modus v2v ist nicht vorgesehen, dass reguläre Knoten die angehängten *MACs* verifizieren. Durch die komplett fehlende Authentifizierung zwischen den *Nodes* gelingt es in

[48] Nachrichten von einem Knoten zu einer Basisstation sind also auf dieser Betrachtungsebene nicht von von denen zu unterscheiden, die eine Basisstation zu diesem Knoten sendet.

[49] eine entsprechend überprüfte Einwegfunktion vorausgesetzt

[50] In [CJW05] nicht erwähnt, aber anzunehmen ist, dass auf Basis des öffentlichen und privaten Schlüssels sichere Kanäle verwendet werden.

diesem setup sogar Angreifern der Stufe AM1, Nachrichten zu versenden, deren Inhaltskomponenten (siehe Abschnitt 4.3.1 auf Seite 85) sie beliebig wählen können, ohne dass ihre Kommunikationspartner die Korrektheit überprüfen können. In Hinblick auf das Anwendungsgebiet A1 stellt dieser Umstand den weitaus kritischsten Einschnitt dieses Vorschlags dar.

2. Aufgrund dieser Beobachtung wird auch klar, dass Knoten ohne Sanktionen am Netz teilnehmen können, ohne sich bei den *BS* zu registrieren. In diesem Fall ist es völlig gleichgültig, ob der nun unregistrierte Teilnehmer PS und KS korrekt bildet. Er bewegt sich nämlich nahezu anonym im Netz; denn keine Instanz kann irgendeine Aussage oder Annahme über die vorliegenden Kurzzeitpseudonyme und session keys machen. Dies gilt vor allem im Bereich v2v, aber auch in v2i, wenn die Identität über *Ombudsman* nicht überprüft wird.

3. Der letztgenannte Umstand könnte auch schwache Angreifer ab AM1 dazu verleiten, sich bei einer Basisstation mit einem selbstgewählten *handle* zu initialisieren. Er kann so in allen Anwendungsbereichen, in denen keine Identitätsprüfung stattfindet, ein falsches Pseudonym vortäuschen. Seine Aktionen bleiben dabei in v2v- und v2i-Kommunikation nicht zurechenbar, ohne dass die *BS* Verdacht schöpfen.[51]

4. Dadurch ist ein *revocation system* nur inkonsequent durchzusetzen. Die Sperrung eines Teilnehmers müsste konform zu den Ergebnissen von Kapitel 4.1 auf Seite 45 bei *Ombudsman* ansetzen. Bei der Onlineprüfung durch eine Basisstation würde diese *TTP* eben die Sperrinformation übermitteln. Doch auch wenn diese Identitätsprüfung konsequent durchgeführt würde, hätten die *Base Stations* in diesem Modell keine Handhabe gegenüber gesperrter Knoten.

5. In der vorliegenden Form ist das System nicht in der Lage, *beacons* hinsichtlich Integrität und Zurechenbarkeit (I1) abzusichern.

6. In [CJW05] ist keine Nachrichtenverschlüsselung zwischen Teilnehmerknoten implementiert, sie verfügen dafür noch nicht einmal über eigenes kryptographisches Material.[52]

7. Auch das Konzept der Teilnehmerrollen kann noch nicht umgesetzt werden.

8. Die Autoren treffen weiterhin keine Aussagen zum genauen Hergang der Aufteilung von O in den session key und das Kurzzeitpseudonym.

9. Hinsichtlich AN1 fällt die absolute Abhängigkeit von den Basisstationen negativ auf: Bei korrekter und vollständiger Implementierung des Modells müsste das Straßennetz

[51] Die Basisstationen überprüfen ja nur die korrekte Bildung von PS und KS auf Grundlage des übermittelten *handles*.

[52] Dies mag beim Versenden von *beacons* nicht ins Gewicht fallen; im Schutzziel V1 wurde festgestellt, dass die Verschlüsselung von *beacons* keinen vorrangigen Charakter hat, solange die daten geringstmögliche Rückschlüsse auf explizite Teilnehmer erlauben (vgl. Kapitel 3.2.1 auf Seite 20)

vollständig abgedeckt werden – eine unrealistische Annahme, besonders im Anfangsstadium des VANETs. Zudem werden an deren Hardwareausstattung und Verarbeitungskapazität hohe Ansprüche (Aufwand, Absicherung, Kosten) gestellt.

2MAC - Fazit und zusammenfassendes Konzept

Als Fazit über die Verfahren basierend auf symmetrischer Kryptographie lässt sich ziehen, dass im Gegensatz zu *KAMAN* nur das zweite vorgestellte Verfahren genügend Potential besitzt, als Sicherheitsinfrastruktur in VANETs eingesetzt zu werden.

Wie der Kritik zum Verfahren von CHOI, JAKOBSSON und WETZEL (Kapitel 4.3.1 auf Seite 84) zu entnehmen ist, fehlen dennoch einige Detaillösungen, die unter Einfluss der Basiskonzepte aus Kapitel 4.1 auf Seite 45 im Folgenden als Erweiterung präsentiert werden.

Als wichtigste und zentrale Erweiterung wird ein *vanet key* vK vorgeschlagen, der den Kritikpunkten 1 bis 5 entgegentritt. Mit besagtem *vanet key* erzeugt jeder Knoten, BS wie N, zusätzlich zu seinem persönlichen *MAC*, der aus dem aktuellen KS gebildet wird, einen zweiten *MAC* über jeder Nachricht. Der *vanet key* ändert sich periodisch, innerhalb eines Zeitintervalls, in dem Knotenausschließungen durchsetzbar sein sollen. Daher wird wie auch in der gesamten Arbeit ein Zeitraum von einem Tag (T) angenommen.

Wie in Abschnitt 4.3.1 auf Seite 85 muss sich in diesem erweiterten Verfahren jeder Knoten bei einer Basisstation authentifizieren. Dies geschieht nach demselben Prinzip bis auf den Unterschied, dass nun die Basisstation die Gültigkeit eines Knotens immer dann bei *Ombudsman* prüfen lässt, wenn er den aktuellen *vanet key* nicht vorweisen kann (Kritikpunkt 3). Nur wenn der Knoten nicht ausgeschlossen wurde, übermittelt die *BS* den *vanet key*:

$$\mathsf{PS}_t | \mathsf{E}_{\mathsf{KS}_t}(\mathsf{vK}_\mathsf{T})$$

PS_t und KS_t bezeichnen das in t gültige Pseudonym respektive den geheimen Schlüssel; beide können durch das *handle* sowohl von *BS* als auch von *N* autonom berechnet werden. Dem mit KS_t verschlüsselten *vanet key* ist ein gewisser Zeitstempel inhärent, da ein von der *BS* falsch gewählter KS_t vom Teilnehmerknoten ignoriert würde.

Fortan fügt jeder authentifizierte Knoten an alle Nachrichten

1. $\mathsf{MAC}_{\mathsf{KS}_t}(\mathsf{M})$
2. $\mathsf{MAC}_{\mathsf{vK}_\mathsf{T}}(\mathsf{M})$

an. Damit wird der Grad der Autonomie der Teilnehmerknoten gegenüber den Basisstationen drastisch erhöht: Jeder Knoten ist nun in der Lage zu prüfen, ob eine Nachricht von einem authentifizierten Kommunikationspartner stammt (Kritikpunkt 1). Da der Initialisierungsvorgang nur einmal in T erfolgen muss, kann auch eine zeitweilige oder regionale Unverfügbarkeit von Basisstationen toleriert werden (Kritikpunkt 9).

Den Fall, dass ein korrekt authentifizierter Knoten seine *MACs* korrekt bildet, seine Nachrichteninhalte jedoch böswillig sind, meldet ein betroffener Knoten an REV, die mit Hilfe von *BS* die Abmahnung bzw. den Auschluss des Angreiferknotens in die Wege leitet. Dieser nachträgliche Ausschluss stellt keinen Nachteil gegenüber einer *PKI*-Lösung dar, da das Verfahren analog gestaltet wird (Kritikpunkt 2 und 4). Ein weiteres „Schlupfloch“ bestände darin, dass ein ebenfalls korrekt authentifizierter Angreifer (wie oben ab AM2) $\mathsf{MAC}_{\mathrm{vK_T}}(\mathsf{M})$ richtig bildet, aber den persönlichen *MAC* fälscht, um die Zurechenbarkeit zu umgehen. Dieser Angriff stellt eine klaren Nachteil dieses Verfahrens gegenüber einer *PKI* dar; denn hier kann erst auf Ebene der Basisstationen der Betrug erkannt werden. In einer *PKI*-Lösung ist dies schon auf Teilnehmerebene möglich. Aus diesem Grund wird ein *tamper-proof module* vorgeschlagen, dass genau diesen Angriff nicht zulässt.

Kritikpunkt 5 beanstandete die fehlende Sicherung von *beacons*. Zwar verdoppeln sich mit dem zweiten *MAC* die Nachrichtenlänge und die Berechnungszeit[53], machen aber immer noch einen kleinen Bruchteil einer *PKI*-Lösung aus (vgl. Kapitel 4.1.5 auf Seite 67).

Zusätzlich zum $\mathsf{MAC}_{\mathrm{vK_T}}(\mathsf{M})$ wäre es durchaus denkbar, einen weiteren, kürzeren Schlüssel aus dem *vanet key* zu extrahieren, um durch *AES* die *beacons* zusätzlich zu verschlüsseln. Ein systemweites Schema zu dieser Extraktion vorausgesetzt, können sämtliche Netzteilnehmer diesen autonom erzeugen und damit auch die *beacons* und ähnliche Nachrichten aus A1,2 gegenüber Außenstehenden (AM1) hinsichtlich Inhaltsvertraulichkeit absichern.[54]

Sollte es darüberhinaus notwendig sein, einen sicheren Kanal zwischen zwei Kommunikationspartnern aufzubauen, fungiert eine *BS* als *KDC* gemäß Abbildung 3.2 auf Seite 42 oder gemäß dem Verfahren *KAMAN* (Abschnitt 4.3.1 auf Seite 82).

Einbettung der Basiskonzepte Mit der Einführung des *vanet keys* wird bereits dem Baustein privacy vs. auditability entsprochen.

Deutlich schwerer fällt die Bearbeitung des Kritikpunkts 7, einer Domäne von *PKIs*. Die besonderen Rollen privilegierter Teilnehmer wie R-POLIZEI, etc. treten fast ausschließlich im Anwendungsgebiet A2, den Einsatzsignalen, zu Tage. Da diese Nachrichten im geocast-Modus zu verteilen sind, erscheint symmetrische Kryptographie hier weder zweckmäßig noch effizient. Vertraute man erneut *tamper-proof modules* könnte man für jede einzelne Rolle einen geheimen Schlüssel in jedem Fahrzeug vorinstallieren, mit denen solch erzeugte MACs geprüft werden können.

Beharrt man jedoch nicht auf symmetrischer Kryptographie, bieten sich als Alternative anonyme Zertifikate an, in denen *Ombudsman* die Rolle in Erweiterungsfeldern zusichert. Gemäß des Basiskonzepts Identitäten und Rollen (Kapitel 4.1.1 auf Seite 46) werden nur dann Nachrichten derart signiert und mit diesem Rollenzertifikat versehen, wenn es die Lage erfordert. Die technische Kapazität eines regulären Knotens muss dagegen nicht erweitert werden, da

[53] Weitere 160 Bit Nachrichtenlänge, weitere 0.024 ms Berechnungszeit im Fall von *SHA-1*, vgl. Tabelle 4.3 auf Seite 69

[54] Auch für diese weitere Maßnahme bleibt genügend Zeit, wie Tabelle 4.2 auf Seite 68 veranschaulicht.

1. er für die Initialisierungsphase sein *handle* mit dem öffentlichen Schlüssel für eine *BS* verschlüsseln muss und damit auch Signaturen prüfen kann,
2. Einsatzsignale die Ausnahme, nicht die Regel darstellen.
3. der Zeitaufwand immer noch weit unter dem einer *PKI*-Lösung liegt.

Auch die Einbettung der Konzepte des Schlüsselmanagements und der Betreiber der Sicherheitsinfrastruktur gestaltet sich geringfügig schwieriger als bei einer *PKI*. Die beiden geheimzuhaltenden Merkmale eines Teilnehmers IDs und SD müssen nämlich sowohl im Fahrzeug als auch bei *Ombudsman* gespeichert werden, ohne dass Dritte davon erfahren. Entsprechend den Vorschlägen aus Kapitel 4.1.4 auf Seite 63 speichert *Ombudsman* vorab systemweit eindeutige ID und dazu jeweils noch einen Wert SD in *tamper-proof modules* und lässt diese vom Fahrzeughersteller in die Automobile verbauen.[55] Parallel dazu speichert *Ombudsman* diese Werte in seiner Datenbank DB_{OM}.

Mit der Eindeutigkeit von ID ist die Wahrscheinlichkeit sehr hoch, daraus auch eindeutige Hash-Werte zu bilden. Diese werden nämlich anstatt der IDs verwendet, um den Registrierungsprozess durchzuführen:

Ein neuer Fahrzeugbesitzer erscheint mit diesem Hash-Wert und den sonstigen notwendigen Dokumenten bei der RA – den Zulassungsstellen. Die RA übermittelt nur den Hash-Wert und Informationen zu Fahrzeug und -halter an die *TTP*, die nun in der Lage ist, ID und Fahrzeughalter in ihre Datenbank einzutragen. Kein Angreifer ist in der Lage, von diesem Hash-Wert auf ID zu schließen, von der die Lang- und Kurzzeitpseudonyme generiert werden.

Die Funktion einer Instanz REV und der Mechanismus des *revocation systems* wurde zu Beginn dieses Abschnitts bereits erläutert; genauso wie die RA ist die REV dafür zuständig, Aufgaben wie Angriffe von der zentralen Instanz des *Ombudsman* abzuschirmen. Als optionale Komponente zur Fahrtprotokollierung kann die Lösung aus Kapitel 4.1.1 auf Seite 46 unabhängig von bisher genannten Maßnahmen realisiert werden: Hier wurde vorgeschlagen, mit Hilfe eines elektronischen Führerscheins Fahraktivitäten im *EDR* festzuhalten.

Mit *AES* und einer *MAC*-Funktion wie *HMAC SHA-1* wurden die sichersten, schnellsten und kompaktesten Kryptoalgorithmen gewählt, die im Baustein Kryptographie vorgestellt wurden.

Im Folgenden werden nun zusammenfassend zentrale Punkte im lifecycle eines VANET-Teilnehmers gelistet und durch Abbildung 4.6 auf der nächsten Seite erläutert:

1. *Ombudsman* speichert ID und SD in einem *tamper-proof module*.
2. Ein Fahrzeughersteller verbaut dieses Modul.
3. Der neue Fahrzeughalter registriert sich mit dem Hash-Wert aus ID und SD über die RA bei *Ombudsman*.

[55] Dieser Schritt gestaltet sich wesentlich unkomplizierter, wenn eine *TTP* aus Kraftfahrtsamt und einem Herstellerkonsortium gebildet werden kann.

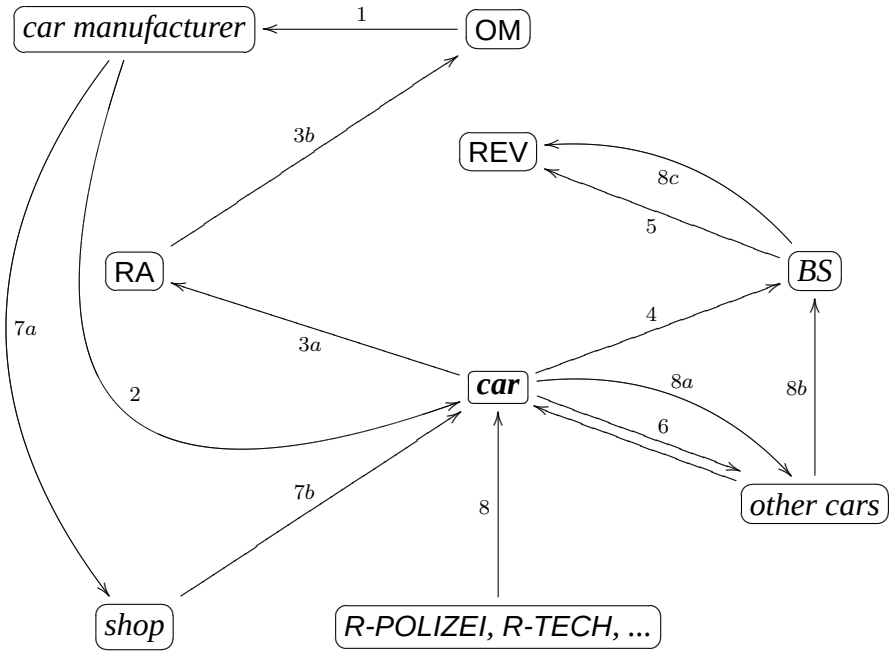

Abbildung 4.6.: Das 2MAC-Verfahren

4. Das Fahrzeug generiert das für T gültige *handle* und authentifiziert sich bei einer *BS*. Diese liefert den aktuellen vK_T nur aus, wenn

5. das *handle* von *REV* auf Gültigkeit geprüft wurde.

6. Fortan übermittelt das Fahrzeug zusätzlich zu seinen Nachrichten $MAC_{KS_t}(M)$ und $MAC_{vK_T}(M)$ – unter Verwendung seines Kurzzeitpseudonyms PS.

7. Im Zuge der zweijährigen Fahrzeuginspektion bestellt eine autorisierte Werkstatt vom Automobilhersteller ein neues *TPM* und verbaut es in das Fahrzeug. Die Authentifizierung basiert in beiden Fällen auf dem Attributzertifikat der Werkstatt.

8. Attributzertifikate weisen dem Fahrzeug besondere Privilegien anderer Teilnehmerrollen wie R-POLIZEI oder R-TECH aus.

9. Trotz korrekter *MACs* enthalten die Nachrichten des Fahrzeugs nun irreführende Nachrichten. Dieser Tatbestand ist über die Datenbanken DB_N, DB_{BS} und DB_{OM} nachvollziehbar und diesem Fahrzeug zurechenbar. REV legt einen Sperrvermerk für dieses Fahrzeug an.

10. Am nächsten Tag will sich das Fahrzeug gemäß den Punkten wieder in das VANET einbuchen, aufgrund seines Sperrvermerks in DB_{OM} wird ihm vK_T nicht ausgehändigt; seine Nachrichten finden bei anderen Teilnehmern keine Beachtung, da er $MAC_{vK_T}(M)$ nicht bilden kann.

Fazit Mit dem Basiskonzept von CHOI, JAKOBSSON und WETZEL und mit den vorgeschlagenen Erweiterungen wird den Anforderungen AN1,2 in befriedigender Art und Weise entsprochen. Einzig die Problematik der Teilnehmerrollen und in gewisser Weise auch das *enrollment* konnte nicht eleganter gelöst werden. Im Gegensatz dazu zeichnet sich das vorgeschlagene Verfahren hinsichtlich Performance und Kompaktheit der Nachrichten und des *revocation systems* aus. Eine abschließende Gegenüberstellung mit einer *PKI*-Lösung erfolgt in Kapitel 4.4 auf Seite 99.

4.3.2. Verfahren basierend auf asymmetrischer Kryptographie

LKN-ASF

CHRISTIAN SCHWINGENSCHLÖGL und STEPHAN EICHLER legen ihrem „LKN Ad Hoc Security Framework“ ([SE04]) eine traditionelle *PKI* zugrunde, wie sie in Kapitel 3.4 auf Seite 36 erläutert wurde. Sie legen in ihrer Arbeit zwar nicht die genauen Umstände der Implementierung dar, einer Adaption der Basiskonzepte (Kapitel 4.1 auf Seite 45) steht jedoch nichts im Wege.

Ohne alle übereinstimmenden Elemente[56] aufzuzählen, wird stattdessen auf abweichende oder neue Ideen hingewiesen.

Die erste Abweichung besteht im *certificate revocation system*, dem hier wenig Aufmerksamkeit gewidmet wurde. Die Autoren erkennen zwar die Notwendigkeit, böswillige Teilnehmer vom Netz auszuschließen, die von ihnen eingeführten *revocation messages* entsprechen im Grunde *Certificate Revocation Lists*. Diese Nachrichten werden zusätzlich mit Sequenznummern ausgestattet, um sicherzustellen, dass ein Knoten alle *revocation messages* einer Periode empfangen hat. Dieser Ansatz schließt natürlich alle Nachteile von *CRLs* ein, die bereits in Abschnitt 4.1.3 auf Seite 54 ermittelt wurden. Neben der geringen Aktualität und der wachsenden Masse an Daten erkennen auch SCHWINGENSCHLÖGL und EICHLER in [SE04] auf Seite 6:

> *Depending on the sending frequency, the number of messages traveling the network can be very high.*

Dieses Schema zum Rückruf von Zertifikaten sollte deswegen durch Verifikatoren-System aus Kapitel 4.1.3 auf Seite 58 ersetzt werden.

Die Idee des *certificate caching* ist hingegen durchaus als praktikabel und vorteilhaft einzuschätzen. Obwohl eine genauere Implementierungsbeschreibung nicht geliefert wird, lässt sich der Grundgedanke dieses Prinzips wie folgt darstellen. In einem Cache werden die letztgeprüften Zertifikate der Kommunikationspartner zwischengespeichert und beschleunigen laut den Simulationstests in [SE04] die Abarbeitung vor allem dann, wenn quasi gleichzeitig Nachrichten von verschiedenen Empfängern bearbeitet werden müssen – dies entspricht genau der Situation in VANETs.

[56] Z. B. wird das Prinzip, Attributzertifikate für besondere Privilegien auszustellen, wie in Abschnitt 4.1.1 auf Seite 46 aufgegriffen. Diese weisen besonderen Rollen (z. B. R-POLIZEI) besondere Attribute zu, die über das Teilnehmerzertifikat verifizierbar sind.

Kritik *LKN-ASF* leistet zwar kaum Neues gegenüber traditionellen *PKIs*, der Beitrag des *certificate caching* könnte jedoch die Praktikabilität und Performance einer *Public Key Infrastructure* (AN1) erheblich verbessern. Dazu müssten aber weitere Tests in der Praxis stattfinden, die noch konkreter als in [SE04] auf automobile Ad-hoc-Netze abzielen.

SAM

In seiner Dissertation [Kar04] stellt FRANK KARGL *SAM* - eine *Sicherheitsarchitektur für Mobile Ad hoc Netzwerke* vor, die Sicherheit durch drei Komponenten (vgl. Abbildung 4.7) erreichen will und deren Funktionen ineinandergreifen und sich gegenseitig stützen:

- *MANET-IDs*
- *SDSR – Secure DSR*
- *MobIDS*

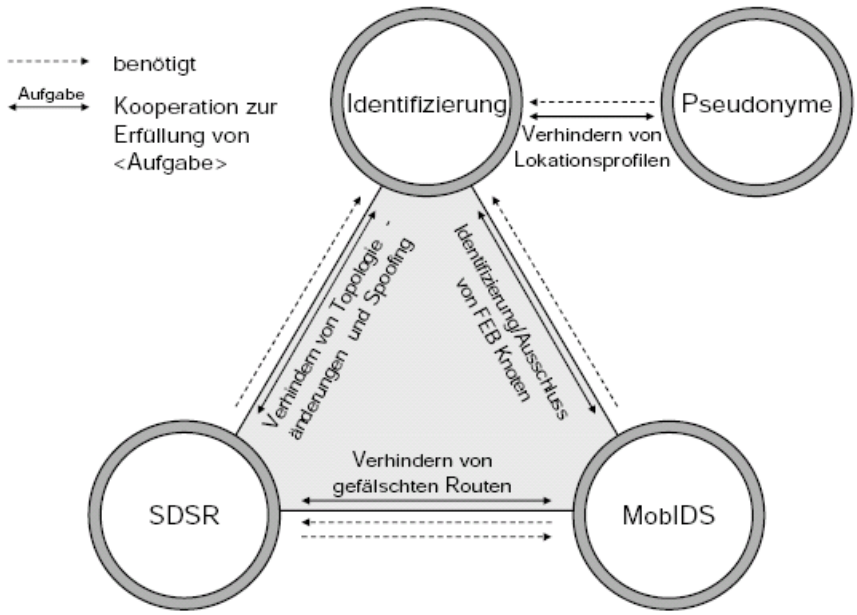

Abbildung 4.7.: Die Komponenten von *SAM*

MANET-IDs In Abschnitt 4.1.2 auf Seite 52 bewährten sich die *MANET-IDs* als eindeutige und pseudonyme Identifikatoren. Hinter dem Ansatz von KARGL verbirgt sich aber darüberhinaus eine *PKI*-Lösung, die ein *revocation system*, eine Umsetzung von *MANET-IDs* in Adressen und eine Authentifizierungsmethode bereitsstellt.

Das System zum Rückruf von Zertifikaten heißt *MANET-CRS* und wurde in Kapitel 4.1.3 auf Seite 58 als das zu favorisierende *revocation system* vorgestellt. Jeder Teilnehmer ruft in periodischen Zeitabständen einen Verifikator-Wert von der CA ab, mit der er selbst andere Knoten von der Gültigkeit seines Zertifikats überzeugen kann.

Weiterhin wird vorgeschlagen, die Netzwerk- und Hardwareadressen direkt aus der *MANET-ID* abzuleiten. Diese Vorgehensweise setze kollisionsresistente Hash-Funktionen voraus.

Bei der Authentifizierung in *SAM* wird geprüft, ob die abgeleitete Hardwareadresse, das Zertikat (der Verifikator), die Signatur des Teilnehmers und der *CA* gültig sind und dieser Teilnehmer nicht durch das *MobIDS* gesperrt wurde.

SDSR – Secure DSR Dieser Authentifizierungprozess wird schon in den route request des *SDSR* vorgelagert, so dass nur Routen zu authentifizierten Knoten etabliert werden. Während dieses Vorgangs wird ein geheimer und symmetrischer Schlüssel generiert, der fortan für unicast-Kommunikation zur Verfügung steht. In [Kar04], S. 156 ff., erläutert der Autor detailliert sein entwickeltes Routing-Verfahren, das eine modifizierte Version von *DSR* darstellt.[57]

MobIDS Die dritte Säule von *SAM* besteht in einem *Intrusion Detection System*, deren Design in Abbildung 4.8 dargelegt wird. Auf der untersten Ebene agieren einige ausgewählte Kno-

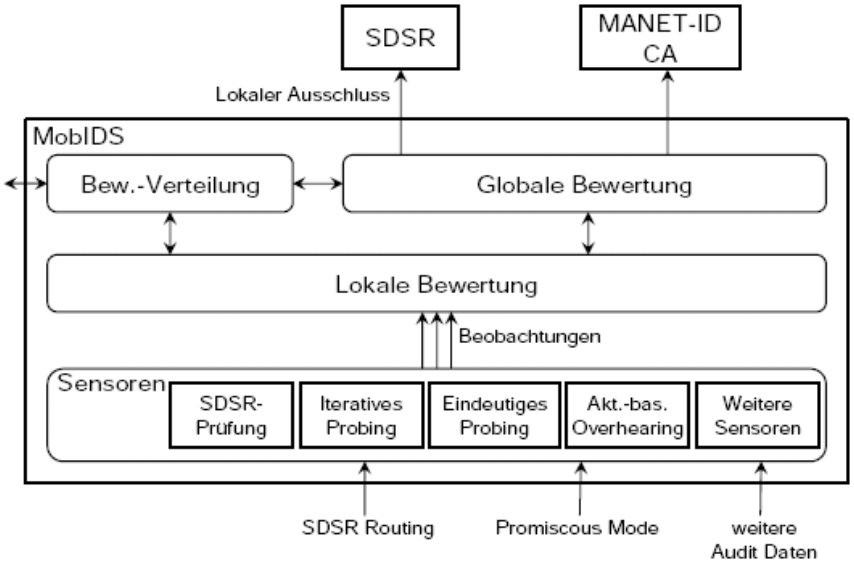

Abbildung 4.8.: Die *Intrusion Detection*-Komponente von *SAM*: *mobIDS*

ten als sog. *Sensoren*, die durch Aggregieren von empfangenen Nachrichten und beobachtetem Teilnehmerverhalten eine *lokale Bewertung* errechnen. Diese gehen wiederum in eine *globale Bewertung* ein, nachdem sie zu anderen Knoten im Netz verteilt wurde. Unterschreitet nun die

[57] Routing-Verfahren sind nicht Teil dieser Arbeit; *SDSR* wird deshalb nicht weiter verfolgt.

globale Bewertung eines Knotens einen gewissen Wert, so erfolgt der Ausschluss dieses Knotens aus dem lokalen Netz.

Wird schädliches Verhalten dieses Teilnehmers wiederholt beobachtet, so kann er auch global ausgeschlossen werden, indem die CA über die Vorfälle in Kenntnis gesetzt wird und dessen Zertifikat zurückzieht.

Kritik Bei korrekter Implementierung und Schlüsselwahl sind wie bei allen *PKI*-Varianten die Schutzziele der Integrität durch Signaturen sehr gut zu erreichen. Besonders schwer wiegt dieser Vorteil beim Versenden von *beacons*, da die Adressaten für die Maßnahme der Integritäts- und Zurechenbarkeitssicherung nicht bekannt sein müssen.

Nicht erwähnt blieb bisher in der Vorstellung dieses Verfahrens, dass KARGL zur Vermeidung von Bewegungsprofilen zusätzliche, hinsichtlich ihrer Zahl limitierte Pseudonyme einführen will (vgl. [Kar04], S. 135 ff.). Dies erscheint in Anbetracht der geringen Hardwarebeschränkungen nicht sinnvoll, sondern ist durch einen kompletten Satz an *MANET-IDs*, Schlüsselpaaren und Zertifikaten zu ersetzen, um die Schutzziele der Vertraulichkeit erfüllen zu können. Für Szenarien, in denen zwei mobile Knoten größere, zusammenhängende Nachrichten über einen längeren Zeitraum austauschen, ist das Aushandeln eines symmetrischen Schlüssels zwischen Knoten und Basisstationen durchaus sinnvoll und denkbar.[58]

Das Ableiten der Hardware- und Netzadressen aus den öffentlichen Schlüsseln ist eine Idee, die sich kompakt umsetzen lässt und das Schutzziel V1 elegant fördert. Durch die Reichweitenbeschränkung und den periodischen Wechsel der Schlüssel ist die Wahrscheinlichkeit, Adressenduplikate in Kommunikationsreichweite anzutreffen, als so klein einzustufen, dass sie keiner weiteren Erörterung bedarf.

Einen großen Beitrag zum Erreichen von AN2 (Performance) leistet das Cachen von Zertifikaten, das auch auf die Verifikatoren ausgeweitet werden könnte. Obwohl in [Kar04] weitere Angaben dazu fehlen, ist anzunehmen, dass diese Maßnahme in der Verarbeitung von *beacons* eine größere Zeitersparnis mit sich bringt. Zusätzlich ist festzustellen, dass die maximale Verweildauer im Cache von der Gültigkeitsdauer der gespeicherten Zertifikate und Verifikatoren begrenzt ist[59]. Die Wahl dieser Grenze garantiert, dass die Reaktions- und Durchsetzungszeit von Knotenausschließungen – hier durch die Gültigkeitsspeiode der Verifikatoren bestimmt – berücksichtigt wird. Die minimale Caching-Zeit stellt den typischen Trade-Off zwischen Speicherplatz und der Wahrscheinlichkeit dar, Daten aus dem Cache verwenden zu können. In jedem Fall ist er nach dem *FIFO*-Prinzip (*First In – First Out* zu handhaben. Im Gegensatz dazu ist das System *mobIDS* als *Intrusion Detection System* aus mehreren Gründen nicht in gleicher Weise für VANETs umzusetzen:

[58] z. B. für den Abruf der aktuellen Verifikatoren-Liste oder dem Übermitteln von aggregierten Daten zur Verkehrslage einer Region.

[59] Diese Zeit stellt ein Zertifikatsfeld dar, bei Verifikaten ist sie systemweit festgelegt. Aufgrund der Mobilität und der Vielzahl der Knotenkontakte dürfte diese maximale Verweildauer in den seltensten Fällen erreicht werden.

- *Sensoren* werden nach [Kar04] aus gewöhnlichen Knoten rekrutiert. Dies wirft die Frage auf, nach welchen Kriterien diese auszuwählen sind und wie überhaupt die zusätzliche Rechen- und Speicherlast zu bewältigen ist.
- Nicht nur aus Performance-, sondern auch aus Sicherheitsgründen ist davon Abstand zu nehmen, andere Knoten durch einen „promiscuous mode“[60] zu belauschen, selbst wenn dafür Sicherheitsvorkehrungen getroffen werden.
- In [Kar04], S. 195 erkennt der Autor selbst, dass die beschränkte Reichweite und die Mobilität der Teilnehmer schon in MANETs widersprüchliche Bewertungen hervorrufen können. Dieser Einwand schließt das *IDS* für VANETs erst recht aus – zumindest in der gezeigten Form.[61]

Ansätze von HUBAUX et al.

Mann kann korrekterweise nicht davon sprechen, dass die zahlreichen Veröffentlichungen von HUBAUX und seinem Team ([RH05a], [HCL04], [RH05b] u. a.) eine konkrete Sicherheitsinfrastruktur vorschlagen; dazu werden zu wenige Entscheidungen getroffen. Dennoch leisteten die einzelnen Vorschläge in dieser Arbeit wichtige Beiträge zu einem stimmigen Gesamtvorschlag einer *PKI*-basierten Lösung. In einer kurzen Zusammenfassung sollen hier noch einmal die Säulen seiner Forschung auf diesem Gebiet dargestellt werden.

- Digitale Signaturen bilden das Herzstück der Sicherheitsmaßnahmen und werden entsprechend den Schutzzielen I1,2 über jede zu versendende Nachricht gebildet. Digitale Zertifikate, ausgestellt von einer zentralen *CA*, bestätigen die Zusammengehörigkeit von Teilnehmern und ihren öffentlichen Schlüsseln.
- Jeder Teilnehmer verfügt über einen Satz von öffentlichen Schlüsseln, die keine Hinweise auf ihn erlauben.
- Eine nicht näher spezifizierte *Electronic License Plate* dient zur Identifizierung von Teilnehmern.
- Diese wird gemeinsam mit dem privaten Schlüssel und kritischen *Warnungen* in einem *tamper-proof device*, dem *EDR*, gespeichert, was die Rückverfolg- und Zurechenbarkeit von Aktionen erlaubt.

[60] promiscuous mode bezeichnet eigentlich einen Modus eines Netzwerkadapters, alle, nicht nur die an ihn gerichteten Pakete mitzulesen.

[61] Nichtsdestotrotz können *IDS* wertvolle Beiträge zur Sicherheit in VANETs beitragen, sind aber nicht explizit Thema dieser Diplomarbeit.

VPKI - Fazit und zusammenfassendes Konzept

In diesem zusammenfassenden Kapitel werden alle Basiskonzepte und die Vorschläge von KARGL und HUBAUX ET AL. vereint.

Gemäß den Basiskonzepten Identitäten und Rollen und privacy vs. auditability besitzt jedes Fahrzeug eine Menge an Schlüsselpaaren und entsprechenden Zertifikaten, die zunächst eine Gültigkeitsdauer gemäß dem Inspektionsintervall des Fahrzeugs aufweisen.

Ein solches Zertifikat besitzt keinerlei Merkmale, die auf das Fahrzeug oder den Halter hinweisen. Zusätzlich zum Ablaufdatum des Zertifikats wird ein Wert Y hinzugefügt, der ein Element des Verifikatoren-Systems, des bevorzugten *revocation systems*, darstellt.

Als pseudonymer Identifikator dient die MANET-ID, die zusammen mit dem privaten Schlüssel und dem Zertifikat abhängig von der Fahrzeuggeschwindigkeit, der Sendereichweite, dem Vorhandensein einer ausreichend großen Gruppe von Fahrzeugen, u. a. gewechselt wird. Dies soll der Erstellung von Bewegungsprofilen entgegenwirken.

Zur Sicherung der Integrität und Zurechenbarkeit wird für jede Nachricht die Signatur mit dem aktuellen privaten Schlüssel erzeugt und übermittelt. Durch das mitgelieferte Zertifikat können Adressaten prüfen, ob die Nachricht verändert wurde und ob sie tatsächlich von diesem Sender stammt. Ein zusätzlicher Verifikator-Wert macht es jedem Teilnehmer möglich, auch ohne Kontakt zur CA bzw. zur REV die Gültigkeit des Zertifikats anzuerkennen (vgl. Schritt 5 von Abbildung 4.9 auf der nächsten Seite).

Zur Reduzierung der übermittelten Daten könnte weiterhin ein Caching von Zertifikaten und Verifikatoren eingeführt werden, so dass nicht jedem *beacon* das Zertifikat und der Verifikator beigefügt werden muss. Die genaue Frequenz, mit der vollständige *beacons* inklusive Zertifikat und Verifikator gebildet werden müssen, muss von der Sendereichweite, der Verkehrslage, der Kontaktzeit mit anderen Fahrzeugen und weiteren Einflussgrößen abhängen und bedarf weiterer Analyse.

Einmal pro Periode T muss jedes Fahrzeug die Liste an Verifikatoren von der REV abholen (vgl. Schritt 4 von Abbildung 4.9 auf der nächsten Seite). Die Menge an Daten kann durch folgenden Vorschlag reduziert werden:

Nach und nach wird sich für jedes Fahrzeug eine durchschnittliche Fahrzeit innerhalb von T einpendeln. Es ist also möglich, auf Basis dieser Dauer und einer angemessen Pufferzeit[62] die benötigte Menge an Schlüsseln zu schätzen und nur für diese Menge die Verifikatoren abzurufen. Durch Restriktionen in der Schlüsselwahl soll auch einer mittelfristigen Wiedererkennung durch Angreifer entgegengetreten werden. Benutzte Schlüssel wandern in einen „Quarantäne"-Speicher, der verhindert, dass diese innerhalb einer Zeitspanne von z. B. einer oder zwei Wochen wieder eingesetzt werden.

> ***Beispiel:***
> *Bei einer täglichen Fahrzeit von zwei Stunden – 120 Schlüsselpaare – und einer Schlüsselmenge von 10000 könnte diese „Quarantäne"-Zeit auf rund 41 Tage*

[62] Beispielsweise zwei Stunden.

ausgedehnt werden, was diese regional beschränkten Fahrzeugbewegungen (z. B. Fahrt zur Arbeitstätte) trotz ihrer wiederkehrenden Natur weniger leicht beobachtbar macht.
Bei einer täglichen Fahrzeit von acht Stunden verkürzte sich diese Zeit natürlich auf rund 10 Tage. Da jedoch anzunehmen ist, dass dieser Fahrer sich in einem weit größeren Gebiet als AM3,4 bewegt oder in dieselbe Region eine längere Zeit nicht zurückkehrt, wiegt diese Einschränkung nicht schwer.

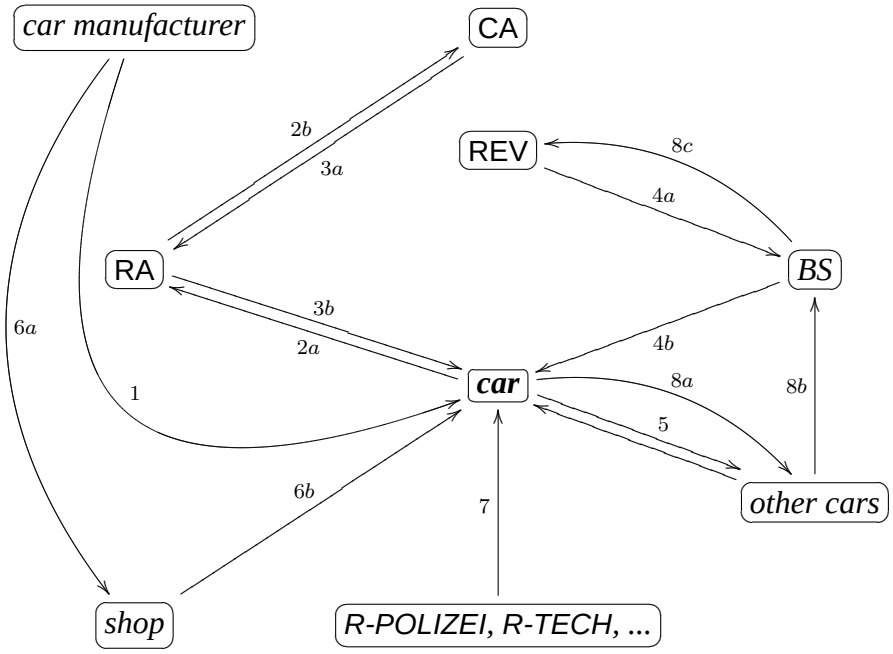

Abbildung 4.9.: Das VPKI-Verfahren

Für den Abruf der Verifikatoren und die Übertragung längerer, aggregierter Meldungen ist es durchaus angebracht, sich mit seinem aktuellen Zertifikat und einer Signatur gegenüber einer Basisstation[63] zu authentifizieren und anschließend einen geheimen und symmetrischen session key auszuhandeln, der für diese Übertragung genutzt wird. Dies vermeidet vor allem die Verkettung von Verifikatoren und damit Zertifikaten und Pseudonymen.

Für einzelne, vertraulich zu behandelnde Nachrichten sind die öffentlichen Schlüssel eines Knotens zu nutzen; der Aufwand, einen session key zu etablieren, wäre angesichts der kurzen Kontaktzeiten nicht rentabel.

Das Schlüssel- und Zertifikatmanagement und die Betreiber wird direkt aus den Basiskonzepten übernommen und hier nur mehr verkürzt dargestellt:

[63] Für die Kommunikation mit einem anderen Fahrzeug dürfte es nicht vorhersehbar sein, ob die Kontaktzeit ausreicht.

Nachdem der Autobauer den Satz an Schlüsselpaaren im *tamper-proof module* des Fahrzeugs zusammen mit dem Zertifikat der nationalen CA[64] gespeichert hat (vgl. Schritt 1 von Abbildung 4.9 auf der vorherigen Seite), geht das Fahrzeug in den Verkauf. Zur Registrierung dieser Schlüssel in der CA erscheint der Halter bei einer RA und übergibt die erforderlichen Dokumente, während die RA im Hintergrund vertraulich die durch das Fahrzeug selbst signierten öffentlichen Schlüssel abruft (Schritt 2) und für den Zweck der Zertifikatserzeugung an die CA übermittelt. Im abschließenden Schritt 3 werden die von der CA signierten Zertifikate zurück auf das Fahrzeug übertragen (wiederum über die RA).

Zum Erneuern von Zertifikaten im Rahmen der Inspektionen eignen sich speziell geschulte und autorisierte Werkstätten, die ihr Privileg durch ein Attributzertifikat gegenüber dem Automobilhersteller und dem Fahrzeug ausweisen (Schritt 6), um ein neues *TPM* zu bestellen und zu verbauen. Es erscheint angebracht, auch den neuerlich *enrollment*-Prozess (Schritt 2 und 3) an dieser Stelle stattfinden zu lassen. Auf demselben Wege, nämlich durch entsprechende Attributzertifikate, nehmen auch andere Teilnehmer wie Polizei, Feuerwehr, etc. ihre Rolle wahr (Schritt 7).

Dem Prinzip des Verifikatoren-Systems folgend, wird ein Teilnehmer vom VANET augeschlossen, indem er statt Verifikatoren Nullwerte von der REV empfängt und so sein Zertifikat von anderen Teilnehmern nicht als gültig angesehen wird.[65] Ein auf das Fahrzeug beschränkter „Fahrtenschreiber“ kann wie auch im Verfahren 2MAC unabhängig von den anderen Maßnahmen nur durch die Verwendung des *EDR* und des elektronischen Führerscheins (vgl. auch Kapitel 4.1.1 auf Seite 46) umgesetzt werden.

Als kryptographische Basis werden wegen ihrer Kompaktheit und Performance Algorithmen auf Basis elliptischer Kurven vorgeschlagen, auch *NTRU* ist eine mögliche Variante.

4.4. VPKI vs. 2MAC

Eine Entscheidung zwischen den beiden Vorschlägen VPKI und 2MAC fällt nicht leicht. Durch die funktionelle Mächtigkeit der digitalen Signatur wirkt VPKI sehr elegant und unkompliziert. 2MAC hingegen erzeugt wesentlich kompaktere Nachrichten, ohne auf Caching zurückgreifen zu müssen. Auch die Berechnungszeiten von Kryptoalgorithmen betragen nur Bruchteile des Konkurrenten. Dies eröffnet zwei Möglichkeiten:

1. Es kann günstigere Hardware im Fahrzeug verwendet werden (AN3).
2. Es stehen wesentlich mehr Ressourcen für andere, zukünftige Anwendungen bereit.

[64] oder idealerweise der Verbindung eines Automobilkonsortiums mit dem nationalen Kraftfahrt-Amt

[65] Wie auch in 2MAC dient die Meldung widersprüchlicher oder unsinniger Nachrichten an die REV als Anhaltspunkt für den Ausschluss eines böswilligen Knotens (Schritt 8 in Abbildung 4.9 auf der vorherigen Seite). Ein *Intrusion Detection System* wie *mobIDS* kann aufgrund verschiedener Mängel noch nicht empfohlen werden.

Ein weiterer Vorteil liegt in der verteilten Speicherung der operativen Pseudonyme PS (in den *BS*) und der wahren Teilnehmeridentität (bei *Ombudsman*): Eine Verkettung beider Informationen gelingt nur bei der Zusammenarbeit dieser *front-end* und *back-end authorities*.

Größter Nachteil von 2MAC ist die umständliche Unterstützung von unterschiedlichen Teilnehmerrollen, die den Aufbau einer kleinen *PKI* erforderlich macht. Dies kann nach den Erkenntnissen dieser Arbeit durch symmetrische Kryptographie nicht einfacher, sicherer und performanter gelöst werden. Im Gegenzug muss sich das damit leicht favorisierte VPKI erst in Praxistests beweisen.

KAPITEL 5

Fazit

5.1. Ergebnisse der Arbeit

Als Grundlage diverser Sicherheitsbetrachtungen war es in dieser Arbeit notwendig, sich die speziellen Eigenschaften und Charakteristika von VANETs vor Augen zu führen. Als limitierende Punkte wurden die hohe Zahl und Mobilität der Teilnehmer und die Flüchtigkeit solcher Ad-hoc-Netzwerke herausgestellt. Als weitere Herausforderung ergaben sich Echtzeitanforderungen, die im Kapitel Anwendungen auch beispielhaft dargelegt wurden. Im Zuge dieser Überlegungen fand auch eine Klassifikation der Anwendungen statt, in die die hybride Struktur von VANETs und die Existenz verschiedener Teilnehmerrollen einflossen. Als zentrale Anforderungen definierten die Schutzziele die Funktionalität einer geeigneten Sicherheitsinfrastruktur. Es gilt dabei besonders, die Erstellung von Bewegungsprofilen, den Verlust von Inhaltsvertraulichkeit und das Vorgeben falscher Identitäten zu verhindern. Zudem soll es möglich sein, die Manipulation von Nachrichten zu erkennen und bei gegebenem Vertraulichkeitsniveau die Zurechenbarkeit für spezielle Fälle zu sichern. Angreifermodelle wurden definiert, um die Wirksamkeit von Konzepten feststellen zu können.

Auf einer tieferen Detailebene wurden sechs Basiskonzepte erarbeitet, die sich erprobter Sicherheitsmechanismen und eigener Beiträge bedienen. Im ersten wird die Identität eines VANET-Teilnehmers dem Fahrzeug zugeordnet; nur wenn ein Teilnehmer eine besondere Rolle wie Polizei, Feuerwehr oder Wartungspersonal aufweist, wird ihm dies durch ein Attributzertifikat personenbezogen beglaubigt.

Um diese Identitäten vor Verfolgung und Profilbildung durch Angreifer zu schützen, wird vorgeschlagen, einen Satz an Pseudonymen mit zugehörigen Schlüsseln pro Teilnehmer zu verwenden, die abhängig von Zeit, Ort, Geschwindigkeit und Sendereichweite gewechselt werden. Trotzdem sollen Angreifer, Straftäter, Unfallflüchtige, etc. in gewisser Weise verfolgbar sein, wenn der gesetzliche Rahmen vorliegt und die zentrale vertrauenswürdige Instanz aus diesem Grund kooperiert. Nur diese *TTP* darf in der Lage sein, die Pseudonyme in reale Identitäten aufzulösen. Als drittes Element wurde der lifecycle von Schlüsseln und Zertifikaten – besonders der Rückruf von Schlüsseln und Zertifikaten und der Ausschluss böswilliger Knoten – an die charakteristischen Eigenschaften von VANETs und den Schutzzielen angepasst.

Eng verflochten mit dieser Thematik ist die Frage nach den Betreibern einer Sicherheitsinfrastruktur. Als Ideallösung für die zentrale Instanz wurde eine Verteilung auf ein Automobilhersteller-Konsortium und nationalen Kraftfahrtsämtern vorgestellt, realistischerweise wurde für die weitere Arbeit die *TTP* nur durch das Kraftfahrtsamt verkörpert. Die beiden Teilinstanzen RA und REV schirmen zentrale Aufgaben und Angriffe ab.

Als letzter Teilaspekt verglich eine eigene Implementierung gängige Verschlüsselungs- und Signaturalgorithmen, die zusammen mit externen Quellen qualitative Einschätzungen und Empfehlungen zuließ.

Noch konkreter wurden bestehende Ansätze für Sicherheitsinfrastrukturen in den beiden folgenden Kapiteln erörtert und hinsichtlich ihrer Tauglichkeit analysiert. Trotz des Versuchs, möglichst viele und verschiedenartige Ansätze zu erforschen, erwiesen sich alle Vorschläge, die explizit keine Basisstationen voraussetzen, als unbrauchbar. Indirekt wurden dabei die Unterschiede zwischen MANETs und VANETs herausgearbeitet.

Im Gegensatz dazu kristallisierte sich eine brauchbare *PKI*-Lösung heraus, die elegant die Anforderungen erfüllt. Ein zweiter Vorschlag wurde um das Konzept des *vanet keys* erweitert und stellt eine Variante zur *PKI* dar, die weniger elegant, aber wesentlicher kompakter und performanter die Anforderungen erfüllt. In beide Ansätze flossen die Basiskonzepte erschöpfend ein.

5.2. Offene Punkte und Ausblick

Im Umfangs dieser Arbeit konnten einige interessante Forschungsgebiete nur gestreift werden, allen voran die Einbettung der erarbeiteten Sicherheitsinfrastrukturen in ein sicheres Routing-Verfahren. *Intrusion Detection Systems* können sicherlich zur Sicherheit in VANETs beitragen – der betrachtete Ansatz dazu erschien jedoch in vielerlei Hinsicht problematisch. Des Weiteren wäre es notwendig, die genauen Hardware-Spezifikationen künftiger VANET-Knoten in Erfahrung zu bringen, um die Performance-Analyse von Kryptoalgorithmen praxisnäher gestalten zu können. Überhaupt muss sich die Praktikabilität der beiden Vorschläge erst in realen Prototypen und verschiedenen Szenarios beweisen.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Die Literaturangaben sind alphabetisch nach den Namen der Autoren sortiert. Bei mehreren Autoren wird nach dem ersten Autor sortiert.

[ABD+05] AIJAZ, Amer ; BOCHOW, Bernd ; DÖTZER, Florian ; FESTAG, Andreas ; GERLACH, Matthias ; KROH, Rainer ; LEINMÜLLER, Tim: Attacks on Inter-Vehicle Communication Systems - An Analysis. (2005). http://www.network-on-wheels.de/downloads/NOW_TechReport_Attacks_on_Inter_Vehicle_Communications.pdf

[AHK] AAD, Imad ; HUBAUX, Jean-Pierre ; KNIGHTLY, Edward W.: *Denial of Service Resilience in Ad Hoc Networks*. citeseer.ist.psu.edu/aad04denial.html

[BF01] BONEH, Dan ; FRANKLIN, Matt: Identity-Based Encryption from the Weil Pairing. 2139 (2001), S. 213 ff.

[BHM+02] BECHLER, Marc ; HAUCK, Achim ; MÜLLER, Daniel ; PÄHLKE, Frank ; WOLF, Lars C.: Ein Sicherheitskonzept für clusterbasierte Ad-hoc-Netzwerke. In: *WMAN*, 2002, S. 135–152

[BJZ05] BLASS, Erik-Oliver ; JUNKER, Holger ; ZITTERBART, Martina: Effiziente Implementierung von Public-Key Algorithmen für Sensornetze. In: *Lecture Notes in Informatics: Informatik 2005, Vol. 2* GI

[BSBHJ06] BEN SALEM, Naouel ; BUTTYAN, Levente ; HUBAUX, Jean-Pierre ; JAKOBSSON, M.: Node Cooperation in Hybrid Ad hoc Networks. 5 (2006), Nr. 4

[CAM05] CAMP Vehicle Safety Communications Consortium: *Vehicle Safety Communications Project Task 3 Final Report Identify Intelligent Vehicle Safety Applications Enabled by DSRC*. http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/1665CAMP3web/images/CAMP3scr.pdf. Version: März 2005

[CBH02] CAPKUN, S. ; BUTTYAN, L. ; HUBAUX, J. P.: Small Worlds in Security Systems: an Analysis of the PGP Certificate Graph. In: *Proceedings of The ACM New Security Paradigms Workshop 2002*. Norfolk, Virginia Beach, USA, September 2002

[CBH03] CAPKUN, S. ; BUTTYAN, L. ; HUBAUX, J. P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. (2003), Januar

[CJW05] CHOI, Jong Y. ; JAKOBSSON, Markus ; WETZEL, Susanne: Balancing auditability and privacy in vehicular networks. In: *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. New York, NY, USA : ACM Press, 2005. – ISBN 1–59593–241–0, S. 79–87

[CKM00] CARMAN, D. W. ; KRUUS, P. S. ; MATT, B. J.: Constraints and Approaches for Distributed Sensor Network Security. (2000), September

[DDD04] DR. JAGOW, Joachim ; DR. BURMANN, Michael ; DR. HESS, Rainer: *Straßenverkehrsrecht*. 18. Verlag C.H.Beck, 2004

[DFM05] DOETZER, Florian ; FISCHER, Lars ; MAGIERA, Przemyslaw: VARS: A Vehicle Ad-Hoc Network Reputation System. In: *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*. Taormina, Italien, Juni 2005

[DKKS05] DOETZER, Florian ; KOHLMAYER, Florian ; KOSCH, Timo ; STRASSBERGER, Markus: Secure Communication for Intersection Assistance. In: *Proceedings of the 2nd International Workshop on Intelligent Transportation*. Hamburg, Germany, März 2005

[DKS05] DOETZER, Florian ; KOSCH, Timo ; STRASSBERGER, Markus: Classification for traffic related inter-vehicle messaging. In: *Proceedings of the 5th IEEE International Conference on ITS Telecommunications*. Brest, France, Juni 2005

[Doe05] DOETZER, Florian: Privacy Issues in Vehicular Ad Hoc Networks. In: *Workshop on Privacy Enhancing Technologies*. Cavtat, Croatia, Mai 2005

[Eck03] ECKERT, C.: *IT-Sicherheit*. 2. Oldenbourg Verlag München Wien, 2003. – ISBN 3–486–27205–5

[Eur03] Europäische Kommission: *The Galilei Project – GALILEO Design Consolidation*. http://europa.eu.int/comm/dgs/energy_transport/galileo/doc/galilei_brochure.pdf. Version: 2003

[Fed99] FEDERRATH, Hannes: *Sicherheit mobiler Kommunikation - Schutz in GSM-Netzen, Mobilitätsmanagement und mehrseitige Sicherheit*. Vieweg-Verlag, 1999. – ISBN 3–528–05695–9

[Fed05a] FEDERRATH, Hannes: *Datenschutzfreundliche Techniken im Internet*. http://www-sec.uni-regensburg.de/security/Folien/11DSfrdlT.pdf. Version: Oktober 2005

[Fed05b] FEDERRATH, Hannes: *Einführung Sicherheit*. http://www-sec.uni-regensburg.de/security/Folien/01EinfSi.pdf. Version: Oktober 2005

[Fed05c] FEDERRATH, Hannes: *Management von Informationssicherheit*. http://www-sec.uni-regensburg.de/security/Folien/03SecMgmt.pdf. Version: Oktober 2005

[FHK95] FOX, Dirk ; HORSTER, Patrick ; KRAAIBEEK, Peter: Grundüberlegungen zu Trust Centern. (1995), S. 1–10

[Fra04] FRANZ, Walter: *Car-to-Car Communication - Anwendungen und aktuelle Forschungsprogramme in Europa, USA und Japan*. http://www.network-on-wheels.de/downloads/car-to-car_uebersicht.pdf. Version: 2004

[Gor] GORZNA, Tino: *DSRC - Dedicated Short Range Communication*. http://www.ihp.ffo.de/systems/lv/ws0405/TG_Text.pdf

[Gut02] GUTMANN, Peter: *Everything you never wanted to know about PKI but were forced to find out*. http://www.cs.auckland.ac.nz/~pgut001/pubs/pkitutorial.pdf. Version: 2002

[HBC01] HUBAUX, J. P. ; BUTTYAN, L. ; CAPKUN, S.: The Quest for Security in Mobile Ad Hoc Networks. In: *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*. Long Beach, CA, Oktober 2001

[HCL04] HUBAUX, Jean-Pierre ; CAPKUN, S. ; LUO, Jun: The Security and Privacy of Smart Vehicles. 2 (2004), Nr. 3, S. 49–55

[HL04] HALVARDSSON, Mattias ; LINDBERG, Patrik: Reliable group communication in a military Mobile Ad hoc Network. (2004). http://www.vxu.se/msi/forskn/exarb/2004/04006.pdf

[Int02] INTERNET ENGINEERING TASK FORCE: *Proceedings of the 53. Internet Engineering Task Force*. http://www3.ietf.org/proceedings/02mar. Version: 2002

[JCH04] JAKOBSSON, Markus ; CAPKUN, Srdjan ; HUBAUX, Jean-Pierre: Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks. Version: 2004. http://infoscience.epfl.ch/getfile.py?mode=best&recid=508. – Forschungsbericht. – Online–Ressource

[JJ04] JOHNSTON, Roger ; JON, Warner: Think GPS Offers High Security? Think Again! In: *Business Contingency Planning Conferencel*

[Kar04] KARGL, Frank: Sicherheit in Mobilen Ad-hoc Netzwerken. (2004), September. http://medien.informatik.uni-ulm.de/~frank/research/dissertation.pdf

[KKA03] KHALILI, A. ; KATZ, J. ; ARBAUGH, W.: Toward Secure Key Distribution in Truly Ad-Hoc Networks. (2003)

[Koc98] KOCHER, Paul C.: On Certificate Revocation and Validation. In: *Financial Cryptography*, 1998, S. 172–177

[Kos05] KOSCH, Timo: Technial Concept And Prerequisites of Car-To-Car Communication. In: *5th European Congress and Exhibition on ITS*

[KSW05] KARGL, F. ; SCHLOTT, S. ; WEBER, M.: Identitäten in Mobilen Ad hoc Netzwerken. (2005), September. http://medien.informatik.uni-ulm.de/~frank/research/wman2005.pdf

[Kur03] KURIHARA, T. M.: *ITS Radio Service (DSRC) Security*. August 2003

[LD05] LIEBEHERR, Jörg ; DONG, Guangyu: An Overlay Approach to Data Security in Ad-Hoc Networks. (2005). http://www.comm.toronto.edu/hypercast/papers/hcast-secadhoc.pdf

[LHE05] LUO, Jun ; HUBAUX, Jean-Pierre ; EUGSTER, Patrick T.: DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks. 2 (2005), Nr. 4, S. 311–323

[Li03] LI, Weihong Wang; Ying Zhu; B.: Self-Managed Heterogeneous Certification in Mobile Ad Hoc Networks. 3 (2003), Oktober, S. 2137–2141

[LRW03] LAMPARTER, B. ; RIEDEL, I. ; WESTHOFF, D.: Anmerkungen zur Nutzung digitaler Signaturen in Ad Hoc Netzwerken. (2003), September. http://www.iponair.de/publications/Lamparter_PIK03.pdf

[LV01] LENSTRA, Arjen K. ; VERHEUL, Eric R.: Selecting Cryptographic Key Sizes. 14 (2001), Nr. 4, 255–293. citeseer.ist.psu.edu/lenstra99selecting.html

[Mic96] MICALI, S.: Efficient Certificate Revocation. (1996), Nr. MIT/LCS/TM-542b. citeseer.ist.psu.edu/micali96efficient.html

[Mil67] MILGRAM, Stanlay: The Small World Problem. 67 (1967)

[Mle05] MLETZKO, Christian: *Sicherheit in automobilen Ad-hoc-Netzen*, Universität Regensburg, Diplomarbeit, März 2005

[MVO96] MENEZES, Alfred J. ; VANSTONE, Scott A. ; OORSCHOT, Paul C. V.: *Handbook of Applied Cryptography*. 1. Boca Raton, FL, USA : CRC Press, Inc., 1996. – 816 S. – ISBN 0849385237

[MWH01] MAUVE, Martin ; WIDMER, Jörg ; HARTENSTEIN, Hannes: A Survey on Position- Based Routing in Mobile Ad Hoc Networks. (2001), Nr. 6, 30-39. http://citeseer.ist.psu.edu/496653

[NDJB02] NASH, Andrew ; DUANE, William ; JOSEPH, Celia ; BRINK, Derek: *PKI e-security implementieren*. 1. mitp-Verlag, Bonn, 2002. – ISBN 3–8266–0781–3

[NTR] http://www.ntru.com/products/Neo_Java1.pdf

[Pfi00] PFITZMANN, Andreas: *Sicherheit in Rechnernetzen: Mehrseitige Sicherheit in verteilten und durch verteilte Systeme*. http://dud.inf.tu-dresden.de/~pfitza/DSuKrypt.pdf. Version: 2000

[PH04] PFITZMANN, Andreas ; HANSEN, Marit: *Anonymity, Unobservability, Pseudonymity, and Identity Management - A Proposal for Terminology*. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.21.pdf. Version: September 2004

[PM04] PIRZADA, Asad A. ; MCDONALD, Chris: Kerberos assisted Authentication in Mobile Adhoc Networks. In: *CRPIT '04: Proceedings of the 27th conference on Australasian computer science*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2004, S. 41–46

[PP05] PARNO, Bryan ; PERRIG, Adrian: Challenges in Securing Vehicular Networks. In: *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, 2005

[Ref05] REFAEI, Mohamed T.: *Securing Emerging Wireless Networks*. http://www.irean.vt.edu/research_workshop_feb2005/refaei_mohamedtamer.pdf. Version: 2005

[RH05a] RAYA, M. ; HUBAUX, J. P.: The Security of Vehicular Ad Hoc Networks. In: *Proceedings of SASN'05*, 2005

[RH05b] RAYA, Maxim ; HUBAUX, Jean-Pierre: The Security of Vehicular Networks. In: *EPFL Technical Report IC/2005/009*, 2005

[Rob02] ROBERT BOSCH GMBH: *Autoelektrik Autoelektronik*. 4. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braundschweig/Wiesbaden, 2002. – ISBN 3–528–13872–6

[Sch00] SCHWABE, Jürgen: *Entscheidungen des Bundesverfassungsgerichts*. 7. Hamburg, 2000. – 42–54 S. – ISBN 3–980 1080–4–X

[Sch01] SCHMEH, Klaus: *Kryptographie und Public-key-Infrastrukturen im Internet*. 2. dpunkt-Verlag, Bonn, 2001. – ISBN 3–93258–90–8

[SE04] SCHWINGENSCHLÖGL, Christian ; EICHLER, Stephan: Certificate-based Key Management for Secure Communications in Ad Hoc Networks. (2004), Februar

[Sha79] SHAMIR, Adi: How to share a secret. 22 (1979), Nr. 11, S. 612–613. – ISSN 0001–0782

[SHL+05] SAMPIGETHAYA, Krishna ; HUANG, Leping ; LI, Mingyan ; POOVENDRAN, Radha ; MATSUURA, Kanta ; SEZAKI, Kaoru: CARAVAN: Providing Location Privacy for VANET. (2005), November

[Sto04] STOLTZE, Tjark S.: *DSRC - Dedicated Short Range Communication*. http://www.tu-harburg.de/~sess0173/mat/dsrc_handout.pdf. Version: Mai 2004

[TC03] TIAN, Jing ; COLETTI, Luca: Routing approach in CarTALK 2000 project. In: *Proceedings of the IST Mobile & Wireless Communications Summit*

[TMN+03] TIAN, J. ; MAIHOEFER, C. ; NELISSE, M. ; PROVERA, M. ; DAGLI, L. ; TEPFENHART, M. ; BRENZEL, C.: *Routing Protocol Implementation*. http://www.cartalk2000.net/bausteine.net/file/showfile.aspx?downdaid=6642&sp=E&domid=687&fd=0. Version: 2003

[Wat99] WATTS, Duncan J.: *Small Worlds – The Dynamics of Networks between Order and Randomneess*. Princeton, New Jersey : Princeton University Press, 1999

[Why05] WHYTE, William: *Safe at Any Speed: Dedicated Short Range Communications (DSRC) and On-road Safety and Security*. Februar 2005

[WS02] WOOD, Anthony D. ; STANKOVIC, John A.: Denial of Service in Sensor Networks. 35 (2002), Nr. 10, 54–62. http://dx.doi.org/http://dx.doi.org/10.1109/MC.2002.1039518. – DOI http://dx.doi.org/10.1109/MC.2002.1039518. – ISSN 0018–9162

[WW03a] WEIMERSKIRCH, André ; WESTHOFF, Dirk: Zero Common-Knowledge Authentication for Pervasive Networks. In: *Selected Areas in Cryptography*, 2003, S. 73–87

[WW03b] WEIMERSKIRCH, André ; WESTHOFF, Dirk: Identity Certified Authentication for Ad-Hoc Networks. In: *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03), George W. Johnson Center at George Mason University, Fairfax, VA, USA, October 31, 2003* ACM

[YEY+04] YIN, Jijun ; ELBATT, Tamer ; YEUNG, Gavin ; RYU, Bo ; HABERMAS, Stephen ; KRISHNAN, Hariharan ; TALTY, Timothy: Performance evaluation of safety applications over DSRC vehicular ad hoc networks. In: *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM Press. – ISBN 1–58113–922–5, 1–9

[YK] YI, S. ; KRAVETS, R.: *MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks*. citeseer.ist.psu.edu/676460.html

[ZH99] ZHOU, Lidong ; HAAS, Zygmunt J.: Securing Ad Hoc Networks. 13 (1999), Nr. 6, S. 24–30

[ZLL+05] ZHANG, Yanchao ; LIU, Wei ; LOU, Wenjing ; FANG, Yuguang ; KWON, Younggoo: AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks. (2005), Mai

[ZLLF] ZHANG, Yanchao ; LIU, Wei ; LOU, Wenjing ; FANG, Yuguang: MASK: An anonymous routing protocol for mobile ad hoc networks. http://ece.wpi.edu/~wjlou/publication/TW-mask.pdf

[ZXSJ03] ZHU, Sencun ; XU, Shouhuai ; SETIA, Sanjeev ; JAJODIA, Sushil: Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach. (2003), November, S. 326–335

ANHANG A

Performance-Messung von kryptographischen Algorithmen in JAVA

Die Performance-Messung wird ausschließlich über Kommandozeilenparameter aufgerufen, die Listing A.1 aufzeigt. Pro Aufruf ist entweder die Messung von Signatur- und *MAC*-Algorithmen oder von Verschlüsselungsalgorithmen möglich. Dies erfolgt mit dem Parameter -s bzw. -c und dem gewünschten Algorithmus. Wählt man statt einem Algorithmus „all“, so werden alle verfügbaren Algorithmen dieser Kategorie nacheinander gemessen.

Um aussagekräftigere Berechnungszeiten zu erhalten, wird jeder Algorithmus mehrfach ausgeführt; ohne den dafür erforderlichen Parameter -l wird eine Wiederholungszahl von 100 angenommen. Für jede Wiederholung wird eine neue Zufallsnachricht erstellt (vgl. Listing A.3 auf Seite xxii), deren Länge in bytes durch den Parameter -m festgelegt werden kann (ansonsten 100 bytes). Die Ausgabe erfolgt grundsätzlich auf STDOUT und eine Datei, die mit dem Parameter -o angepasst werden kann.

Listing A.1: Usage

```
usage: java CryptoSpeed -s|-c <algorithm> [-m <msg length>] [-l <loops>]
            [-o <outfile>] [-v]
 -c    <blow|rsa|3des|aes|all>       cipher with specified algorithm
 -l    <loops>                       count of loops; optional
 -m    <message length>              message length in bytes; optional
 -o    <file>                        specify output file; optional
 -s    <ecc|rsa|md5|dsa|sha1|all>    sign with specified algorithm
 -v                                  (very) verbose output; optional
```

Listing A.2: Performance-Messung von Kryptoalgorithmen

```
1  /*
2   * CryptoSpeed.java
3   *
4   * Created on 2006-02-03
5   *
6   * Author: Manuel Reil
7   */
8
9  package CryptoSpeed;
10
11 import java.io.*;
12 import java.io.FileWriter;
13 import java.util.Hashtable;
```

```
14 import java.util.Enumeration;
15
16 import java.security.cert.Certificate;
17 import java.security.*;
18 import java.security.spec.ECGenParameterSpec;
19 import java.security.SecureRandom;
20 import javax.crypto.*;
21 import javax.crypto.spec.*;
22 import java.text.DecimalFormat;
23
24 import org.apache.commons.cli.*;
25
26 public class CryptoSpeed {
27
28     private static Hashtable algoList = new Hashtable();
29     private static String tmpAlgo = "";
30     private static String algoType = "";
31
32     private static String KEYSTORE ="";
33     private static String ALIAS ="";
34     private static char[] PASS = null;
35
36     private static boolean verbose = false;
37     private static double begin = 0;
38     private static double end = 0;
39
40     private static byte[] sign = null;
41     private static byte[] cipherText = null;
42
43     private static RandMsg rand = null;
44
45     private static KeyStore ks;
46     private static Mac mac = null;
47     private static PublicKey pub = null;
48     private static PrivateKey priv = null;
49     private static Key key = null;
50     private static Signature signature;
51
52     private static String keyLength = "";
53     private static boolean cipherMode = false;
54
55     private static String outStr  = "";
56     private static StringBuffer out = new StringBuffer();
57     private static FileWriter fw = null;
58
59     /*-------------------------
60     *  DEFAULTS
61     */
62
63     private static final String KEYSTORE_DSA = ".keyDSA";
64     private static final String KEYSTORE_RSA = ".keyRSA";
65
66     private static final String ALIAS_DSA     = "test";
67     private static final String ALIAS_RSA     = "rsa";
68
69     private static final char[] PASS_DSA    = {'t','e','s','t', '1', '2'};
70     private static final char[] PASS_RSA    = {'r','s','a','r', 's', 'a'};
71
72     private static DecimalFormat df = new DecimalFormat("#.######");
73     private static int loops = 100;
74     private static String outFile = "";
75     private static int msgLength = 100;
76
77     //-------------------------
78
```

```
79      static byte[] msg;
80
81      public static void main(String[] args)
82      {
83      // handle the args
84          if(!handleARGS(args))
85          {
86              System.exit(0);
87          }
88
89          msg = new byte[msgLength];
90
91          // random msgs
92          rand = RandMsg.getInstance(msgLength);
93
94          // all algorithms specified in args are prepared (key generation or loading) and measured
95          for(Enumeration e = algoList.keys(); e.hasMoreElements();)
96          {
97              tmpAlgo = (String) e.nextElement();
98              algoType = ((String) algoList.get(tmpAlgo));
99              if(algoType.equals("a"))
100             {
101                 prepareAsymmetric(tmpAlgo);
102             }
103             else if(algoType.equals("s"))
104             {
105                 prepareSymmetric(tmpAlgo);
106             }
107             else
108             {
109                 prepareMAC(tmpAlgo);
110             }
111
112             // start single run
113             measure(tmpAlgo, algoType);
114         }
115
116         // if specified, output file is written
117         if(!outFile.equals(""))
118         {
119             try
120             {
121                 fw = new FileWriter(outFile);
122                 fw.write(out.toString());
123                 fw.flush();
124                 fw.close();
125             }
126             catch(IOException ioe)
127             {
128                 ioe.printStackTrace();
129             }
130         }
131     }
132
133     // measure single algorithm
134     private static void measure(String algo, String algoType)
135     {
136       // to save the time for each run
137       double signOrCipherTimes = 0;
138       double veriOrDecipherTimes = 0;
139       double maxSignOrCipher = 0;
140       double maxVeriOrDecipher = 0;
141       double minSignOrCipher = 10000000;
142
143       double minVeriOrDecipher = 10000000;
```

```
144         double current = 0;
145 
146         for(int i = 0; i < loops; i++)
147         {
148 
149                 if(i%10 == 0)
150                 {
151                         System.out.print(".");
152                 }
153                 msg = rand.nextMsg();
154 
155                 // encrypt
156                 if(cipherMode)
157                 {
158                         if(algo.equals("RSA"))
159                         {
160                             current = doCipher(msg, pub, algo, Cipher.ENCRYPT_MODE);
161                         }
162                         else
163                         {
164                             current = doCipher(msg, key, algo, Cipher.ENCRYPT_MODE);
165                         }
166                 }
167                 else
168                 {
169 
170                 // sign
171                     current = sign(msg, key, algo, algoType);
172                 }
173 
174                 signOrCipherTimes += current;
175                 if(current > maxSignOrCipher)
176                 {
177                     maxSignOrCipher = current;
178                 }
179                 if(current < minSignOrCipher)
180                 {
181                     minSignOrCipher = current;
182                 }
183 
184                 // decrypt
185                 if(cipherMode)
186                 {
187                     current = doCipher(cipherText, key, algo, Cipher.DECRYPT_MODE);
188                 }
189                 else
190                 {
191 //                   verify
192                     if(algoType.equals("m"))
193                     {
194                         current = sign(msg, key, algo, algoType);
195                     }
196                     else
197                     {
198                         current = verify(msg,sign,pub);
199                     }
200                 }
201 
202                 // update min and max times
203                 veriOrDecipherTimes += current;
204                 if(current > maxVeriOrDecipher)
205                 {
206                     maxVeriOrDecipher = current;
207                 }
208                 if(current < minVeriOrDecipher)
```

```
209             {
210                 minVeriOrDecipher= current;
211             }
212         }
213
214         // average
215         signOrCipherTimes/=loops;
216         veriOrDecipherTimes/=loops;
217
218         // print output to STDOUT
219
220         String action = "signing";
221         String deAction = "verifying";
222
223         if(cipherMode)
224         {
225             action = "ciphering";
226             deAction = "deciphering";
227         }
228
229         outStr = "\n-------------------------------------------\n"+
230                  algo+" | msg length: "+msgLength+" | loops: " + loops +"\n"+
231                  "-------------------------------------------\n"+
232                  "average "+ action + " time: "
233                  + df.format(signOrCipherTimes/1000/1000) + " ms\n"+
234                  "maximum "+ action + " time: "
235                  + df.format(maxSignOrCipher/1000/1000) + " ms\n"+
236                  "minimum "+ action + " time: "
237                  + df.format(minSignOrCipher/1000/1000) + " ms"+"\n\n"+
238                  "average "+ deAction + " time: "
239                  + df.format(veriOrDecipherTimes/1000/1000) + " ms\n"+
240                  "maximum "+ deAction + " time: "
241                  + df.format(maxVeriOrDecipher/1000/1000) + " ms\n"+
242                  "minimum "+ deAction + " time: "
243                  + df.format(minVeriOrDecipher/1000/1000) + " ms\n"+
244                  "-------------------------------------------\n";
245          System.out.println(outStr);
246
247         // save this output for the output file
248         if(!outFile.equals(""))
249         {
250             out.append(outStr);
251         }
252     }
253
254     // build signatures und macs
255     private static double sign(byte[] msg, Key signKey, String algo, String algoType)
256     {
257         double current = 0;
258         try
259         {
260             if(algoType.equals("m"))
261             {
262                 mac = Mac.getInstance(algo);
263
264                 mac.init(signKey);
265                 begin = System.nanoTime();
266
267                 sign = mac.doFinal(msg);
268
269                 end = System.nanoTime();
270                 current = end - begin;
271             }
272             else
273             {
```

```
274                 signature.initSign((PrivateKey)signKey);
275
276                 begin = System.nanoTime();
277                 signature.update(msg);
278
279                 sign = signature.sign();
280
281                 end = System.nanoTime();
282                 current = end - begin;
283             }
284
285             String sigString = new String(sign);
286
287             if(verbose)
288             {
289                 System.out.println("------------------------------------");
290                 System.out.println("message: " + new String(msg));
291                 System.out.println("\n\nsignature: "
292                                    + sigString + " | length: " + sign.length);
293                 System.out.println("signing time: " + current/1000/1000 + " ms");
294                 System.out.println("-------");
295             }
296
297             return current;
298         }
299         catch (Exception e)
300         {
301             e.printStackTrace();
302             return 0;
303         }
304     }
305
306     // verify signatures (only)
307     private static double verify(byte[] msg, byte[] sign, Key veriKey)
308     {
309         boolean isOk = false;
310
311         try
312         {
313             signature.initVerify((PublicKey)veriKey);
314
315             begin = System.nanoTime();
316
317             signature.update(msg);
318
319             isOk = signature.verify(sign);
320
321             end = System.nanoTime();
322             double current = end - begin;
323
324             if(verbose)
325             {
326                 System.out.println("verifying time: "
327                                    + current/1000/1000 + " ms" + " | signature is "
328                                    + (isOk ? "valid" : "invalid"));
329                 System.out.println("------------------------------------\n");
330             }
331             return current;
332         }
333         catch (Exception e)
334         {
335             e.printStackTrace();
336             return 0;
337         }
338     }
```

```
339
340     // encrypt or decrypt according to mode
341     private static double doCipher(byte[] msg, Key key, String algo, int mode)
342     {
343       double current = 0;
344       try
345       {
346             Cipher cipher = Cipher.getInstance(algo);
347             cipher.init(mode, key);
348
349             begin = System.nanoTime();
350
351             // encrypt the cleartext
352             cipherText = cipher.doFinal(msg);
353
354             end = System.nanoTime();
355             current = end - begin;
356
357             if(verbose)
358             {
359                 System.out.println("-----------------------------------");
360                 System.out.println("message: " + new String(msg));
361                 System.out.println("\n\nciphertext: "
362                                    + new String(cipherText) + " | length: " + sign.length);
363                 System.out.println("ciphering time: " + current/1000/1000 + " ms");
364                 System.out.println("-------");
365             }
366
367             return current;
368         }
369         catch (Exception e)
370         {
371             e.printStackTrace();
372             return 0;
373         }
374     }
375
376     // manage the args and supported algorithms
377     private static boolean handleARGS(String[] args)
378     {
379         Hashtable cipherAlgos = new Hashtable();
380         Hashtable signAlgos = new Hashtable();
381         String[] str;
382
383         int i = 0;
384         String givenAlgo = "";
385         String algoType = "";
386
387         String signHelp = "<";
388         String cipherHelp = "<";
389
390 /*-------------------------------------------------------------------------
391  *       for adding further algorithms add a complete entry in the corresponding
392  *       Hashtable
393  *       a -> asymmetric cipher or signature algorithms
394  *       s -> symmetric cipher algorithms
395  *       m -> mac algorithms
396  *-------------------------------------------------------------------------
397  */
398         // add sign algorithms to repository
399         signAlgos.put("rsa",new String[]{"MD5withRSA","a"});
400         signAlgos.put("dsa",new String[]{"SHA1withDSA","a"});
401         signAlgos.put("ecc",new String[]{"ECDSA","a"});
402         signAlgos.put("md5",new String[]{"HmacMD5","m"});
403         signAlgos.put("sha1",new String[]{"HmacSHA1","m"});
```

```
404
405         // add cipher algorithms to repository
406         cipherAlgos.put("aes",new String[]{"AES","s"});
407         cipherAlgos.put("3des",new String[]{"DESede","s"});
408         cipherAlgos.put("blow",new String[]{"Blowfish","s"});
409         cipherAlgos.put("rsa",new String[]{"RSA","a"});
410
411 //------------------------------------------------------------------------
412
413         // concatenate help strings
414         for(Enumeration e = signAlgos.keys(); e.hasMoreElements();)
415         {
416             signHelp += (new String[]{(String) e.nextElement()})[0]+"|";
417         }
418         signHelp += "all>";
419
420         for(Enumeration e = cipherAlgos.keys(); e.hasMoreElements();)
421         {
422             cipherHelp += (new String[]{(String) e.nextElement()})[0]+"|";
423         }
424         cipherHelp += "all>";
425
426         // create Options object
427         Options options = new Options();
428
429         // add options
430         options.addOption("o", true, "<file>                          "+
431             "specify output file; optional");
432         options.addOption("s", true, signHelp+"     "+
433             "sign with specified algorithm");
434         options.addOption("c", true, cipherHelp+"       "+
435             "cipher with specified algorithm");
436         options.addOption("l", true, "<loops>                         "+
437             "count of loops; optional");
438         options.addOption("m", true, "<message length>                "+
439             "message length in bytes; optional");
440         options.addOption("v", false, "                                "+
441             "(very) verbose output; optional");
442
443         CommandLineParser parser = new PosixParser();
444         CommandLine cmd = null;
445         try
446         {
447             cmd = parser.parse(options, args);
448         }
449         catch(ParseException e)
450         {
451             e.printStackTrace();
452         }
453
454         HelpFormatter formatter = new HelpFormatter();
455
456         if(args.length < 2)
457         {
458             formatter.printHelp( "java CryptoSpeed -s|-c <algorithm>"+
459                     "[-m <msg length>][-l <loops>][-o <outfile>][-v]", options);
460             return false;
461         }
462 //        set mode and algorithms
463         if(cmd.hasOption("s") && !cmd.hasOption("c"))
464         {
465 //            sign mode
466            givenAlgo = cmd.getOptionValue("s");
467            if (!givenAlgo.equals("all"))
468            {
```

```
469                 if(signAlgos.containsKey(givenAlgo))
470                 {
471                    algoList.put(((String[]) signAlgos.get(givenAlgo))[0],
472                          ((String[]) signAlgos.get(givenAlgo))[1]);
473                 }
474                 else
475                 {
476                    formatter.printHelp( "CryptoSpeed", options);
477                    return false;
478                 }
479              }
480              else
481              {
482 //                 add all signature algos
483                 for(Enumeration e = signAlgos.elements(); e.hasMoreElements();)
484                 {
485                     str = (String[]) e.nextElement();
486                     algoList.put(str[0], str[1]);
487                 }
488              }
489           }
490           else
491           {
492 //             cipher mode
493              if (cmd.hasOption("c"))
494              {
495                  cipherMode = true;
496                  givenAlgo = cmd.getOptionValue("c");
497                  if (!givenAlgo.equals("all"))
498                  {
499                     if(cipherAlgos.containsKey(givenAlgo))
500                     {
501                           algoList.put(((String[]) cipherAlgos.get(givenAlgo))[0],
502                                 ((String[]) cipherAlgos.get(givenAlgo))[1]);
503                     }
504                     else
505                     {
506                           formatter.printHelp( "CryptoSpeed", options);
507                           return false;
508                     }
509                  }
510                   else
511                  {
512                      for(Enumeration e = cipherAlgos.elements(); e.hasMoreElements();)
513                      {
514 //                          add all cipher algos
515                           str = (String[]) e.nextElement();
516                           algoList.put(str[0], str[1]);
517                      }
518                  }
519              }
520              else
521              {
522                   formatter.printHelp( "CryptoSpeed", options);
523                   return false;
524              }
525           }
526 //     set loops
527           if(cmd.hasOption("l"))
528           {
529               String l = cmd.getOptionValue("l");
530               try
531               {
532                     loops = new Integer(l);
533               }
```

```
534             catch(Exception e)
535             {
536                   formatter.printHelp( "CryptoSpeed", options);
537                   return false;
538             }
539         }
540
541 //        set outfile
542         if(cmd.hasOption("o"))
543         {
544             outFile = cmd.getOptionValue("o");
545         }
546 //        set verbosity
547         if(cmd.hasOption("v"))
548         {
549             verbose = true;
550         }
551 //        set message length
552         if(cmd.hasOption("m"))
553         {
554             try
555             {
556                 msgLength = new Integer(cmd.getOptionValue("m"));
557             }
558             catch(Exception e)
559             {
560                 e.printStackTrace();
561             }
562         }
563         return true;
564     }
565
566     private static void prepareAsymmetric(String algo)
567     {
568      KeyPair keyPair = null;
569
570       if(algo.endsWith("SA") && !algo.endsWith("ECDSA"))
571       {
572             if(algo.endsWith("RSA"))
573             {
574                KEYSTORE = KEYSTORE_RSA;
575                ALIAS = ALIAS_RSA;
576                PASS = PASS_RSA;
577             }
578             else
579             {
580                KEYSTORE = KEYSTORE_DSA;
581                ALIAS = ALIAS_DSA;
582                PASS = PASS_DSA;
583             }
584             try
585             {
586                 ks = KeyStore.getInstance("JKS");
587                 FileInputStream ksin = new FileInputStream(KEYSTORE);
588
589                 ks.load(ksin, PASS);
590                 ksin.close();
591
592 //                Privaten üSchlssel lesen
593                 key = (PrivateKey) ks.getKey(ALIAS, PASS);
594                 Certificate cert = ks.getCertificate(ALIAS);
595                 pub  = cert.getPublicKey();
596
597                 if(!cipherMode)
598                 {
```

```
599 //                  Signatur-Objekt erstellen
600                     signature = Signature.getInstance(algo);
601                 }
602             }
603             catch(Exception e)
604             {
605                 e.printStackTrace();
606             }
607         }
608         else if(algo.endsWith("ECDSA"))
609         {
610             try
611             {
612                 KeyPairGenerator keyGen =
613                       KeyPairGenerator.getInstance(algo, "BC");
614                 ECGenParameterSpec ecSpec =
615                       new ECGenParameterSpec("prime192v1");
616
617                 keyGen.initialize(ecSpec, new SecureRandom());
618
619                 keyPair = keyGen.generateKeyPair();
620                 signature = Signature.getInstance(algo);
621             }
622             catch(Exception e)
623             {
624                   e.printStackTrace();
625             }
626             key = keyPair.getPrivate();
627             pub = keyPair.getPublic();
628         }
629     }
630
631     private static void prepareSymmetric(String algo)
632     {
633       try
634       {
635             KeyGenerator keygen = KeyGenerator.getInstance(algo, "SunJCE");
636             key = keygen.generateKey();
637       }
638       catch(Exception e)
639       {
640             e.printStackTrace();
641       }
642     }
643
644     private static void prepareMAC(String algo)
645     {
646       byte[] keyData = { (byte)0x01, (byte)0x02,
647                          (byte)0x03, (byte)0x04,
648                          (byte)0x05, (byte)0x06,
649                          (byte)0x07, (byte)0x08 };
650       try
651       {
652             KeyGenerator kgen = KeyGenerator.getInstance(algo);
653             kgen.init(1024);
654             key = kgen.generateKey();
655       }
656       catch(Exception e)
657       {
658             e.printStackTrace();
659       }
660     }
661 }
```

Listing A.3: Erstellung von Zufallsnachrichten

```
1  /*
2   * RandMsg.java
3   *
4   * Created on 9. November 2005, 15:34
5   *
6   * @author Manuel Reil
7   */
8
9  package CryptoSpeed;
10
11 import java.security.*;
12 import java.util.Timer.*;
13
14 //    singleton
15 public class RandMsg {
16
17     private static RandMsg randMsg;
18
19     private byte[] msg;
20     private SecureRandom rand;
21
22     /** Creates a new instance of RandMsg */
23     private RandMsg(int length)
24     {
25         msg = new byte[length];
26         try{
27             rand = SecureRandom.getInstance("SHA1PRNG");
28             rand.setSeed(1234567890);
29         }catch(NoSuchAlgorithmException e){
30             e.printStackTrace();
31         }
32     }
33
34     public static RandMsg getInstance (int length)
35     {
36         return new RandMsg(length);
37     }
38
39     public byte[] nextMsg()
40     {
41         rand.nextBytes(msg);
42         return msg;
43     }
44
45     public byte[] nextMsg(int length)
46     {
47         msg = new byte[length];
48         rand.nextBytes(msg);
49         return msg;
50     }
51 }
```

ANHANG B

Performance von Verschlüsselungsalgorithmen

Intel Celeron 1100 MHz, 384 MByte RAM, 10000 Wiederholungen pro Algorithmus, Zufallsnachrichten (100 Byte)

```
DESede | msg length: 100 | loops: 10000
average ciphering time: 0,134322 ms
maximum ciphering time: 60,239104 ms
minimum ciphering time: 0,049024 ms

average deciphering time: 0,136166 ms
maximum deciphering time: 60,369024 ms
minimum deciphering time: 0,048896 ms
```

```
RSA | msg length: 100 | loops: 10000
average ciphering time: 2,814783 ms
maximum ciphering time: 137,714048 ms
minimum ciphering time: 1,243008 ms

average deciphering time: 51,070995 ms
maximum deciphering time: 1151,995008 ms
minimum deciphering time: 10 ms
```

```
AES | msg length: 100 | loops: 10000
average ciphering time: 0,091523 ms
maximum ciphering time: 48,091904 ms
minimum ciphering time: 0,02496 ms

average deciphering time: 0,070579 ms
maximum deciphering time: 32,336 ms
minimum deciphering time: 0,021888 ms
```

```
Blowfish | msg length: 100 | loops: 10000
average ciphering time: 0,054021 ms
maximum ciphering time: 40,077952 ms
minimum ciphering time: 0,015872 ms

average deciphering time: 0,051831 ms
maximum deciphering time: 41,071104 ms
minimum deciphering time: 0,016 ms
```

Performance von Signatur- und MAC-Algorithmen

Intel Celeron 1100 MHz, 384 MByte RAM, 10000 Wiederholungen pro Algorithmus, , Zufallsnachrichten (150 Byte)

```
MD5withRSA | msg length: 150 | loops: 10000
average signing time: 48,216395 ms
maximum signing time: 717,135104 ms
minimum signing time: 10 ms

average verifying time: 2,879545 ms
maximum verifying time: 192,606976 ms
minimum verifying time: 1,318016 ms
```

```
SHA1withDSA | msg length: 150 | loops: 10000
average signing time: 20,11394 ms
maximum signing time: 405,441024 ms
minimum signing time: 10 ms

average verifying time: 39,660623 ms
maximum verifying time: 685,629952 ms
minimum verifying time: 10 ms
```

```
HmacMD5 | msg length: 150 | loops: 10000
average signing time: 0,023748 ms
maximum signing time: 2,963968 ms
minimum signing time: 0,014976 ms
average verifying time: 0,026043 ms
maximum verifying time: 16,926976 ms
minimum verifying time: 0,014976 ms
```

```
HmacSHA1 | msg length: 150 | loops: 10000
average signing time: 0,053155 ms
maximum signing time: 9,163008 ms
minimum signing time: 0,038912 ms

average verifying time: 0,053137 ms
maximum verifying time: 5,661056 ms
minimum verifying time: 0,038912 ms
```

```
ECDSA | msg length: 150 | loops: 10000
average signing time: 59,127876 ms
maximum signing time: 284,13504 ms
minimum signing time: 10 ms

average verifying time: 117,815192 ms
maximum verifying time: 1248,829952 ms
minimum verifying time: 10 ms
```

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig angefertigt sowie keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Diese Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Regensburg, den 13. März 2006

MANUEL REIL