Institut für Informatik der Technischen Universität München

Security Concepts for Robust and Highly Mobile Ad-hoc Networks

Florian Dötzer

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. J. Schlichter

Prüfer der Dissertation:

1. Univ.-Prof. Dr. U. Baumgarten
2. Univ.-Prof. Dr. Claudia Eckert,
   Technische Universität Darmstadt

Die Dissertation wurde am 19. September 2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 23.04.2008 angenommen.

ABSTRACT

Security Concepts for Robust and Highly Mobile Ad-hoc Networks

Florian Dötzer

The emergence of widespread wireless communication and ad-hoc networking technologies enables new approaches in traffic related and mobile communications. Mobile devices, including cars as a special variation thereof, are becoming pervasively connected through broad availability of low-cost commercial wireless communication hardware. The advantages of wireless ad-hoc networks - decentralization and self-organization - allow the design of cost effective communication systems for cars and keep driver involvement to a minimum. Cars offer a couple of advantages for the operation of a decentralized network, such as unlimited power supply, periodic maintenance and availability of on-board navigation systems and tamper resistant devices. But they also introduce new challenges, especially through their high degree of mobility. The automotive environment demands high standards on reliability and quality in general. Security is especially critical for a communication system that is designed to support safety-critcal decisions. However, conventional security concepts cannot solve all issues in such a inter-vehicle communication (IVC) system adequately: security infrastructure is not available continuously, privacy is a strong user requirement and automated credential revocation remains an unsolved question in this scenario.

Therefore, this thesis analyses existing security concepts with respect to their fit in an IVC system, develops new approaches in response to the shortcomings encountered by a specific threat analysis for this scenario and introduces a component-based architecture.

After confining the scope of the IVC system to three main application types, vehicle to vehicle single-hop, vehicle to infrastructure single-hop and multi-hop communication, a threat analysis has been conducted based on a custom system model, consisting of predefined constraints, traffic models, hardware components and non-functional requirements. This resulted in specific security requirements for information authenticity, dependability and privacy leading to the design of a component-based security architecture for IVC systems.

In order to provide authentic information in the traffic-related communication domain, an analysis and classification of the traffic-event parameter space has been performed. A key feature of this work is the new paradigm of information-based trust. Application-level information, large-scale effects, availability of local sensors and longterm knowledge about the traffic environment are used to rate the quality of confidence in received messages. This allows the implementation of a trust scheme that does not rely on trust authorities, while it may still take opportunity of such infrastructures if available. Trust authorities are costly and difficult to operate in a global environment compared to a decentralized solution.

The proposed paradigm also overcomes the fundamental problems of a centralized trust system, such as the loose correlation between node credentials and message content, the problem of robust and scalable credential revocation and the yes/no trust statement of credentials.

In order to meet stringent privacy requirements, a trusted third party solution has been adapted to fit the chosen scenarios by modifiying cryptographic protocols and has been implemented prototypically. The information-based approach is utilized to renounce otherwise mandatory node authentication for traffic-related messaging. However, the privacy architecture also supports services that require node authentication, typically required in vehicle to infrastructure and end to end user communication settings. A key exchange protocol relying on this architecture has been implemented on a smart card to prove the feasibility in a vehicle to infrastructure scenario, while a secure routing protocol has been designed and implemented for connection oriented multi-hop communication.

Scalability and availability of vehicle to vehicle communication depends on efficient usage of the available wireless bandwidth. Therefore, message priorization schemes and data aggregation patterns have been developed. In order to prevent the broadcast storm problem, forwarding algorithms have been evaluated for usage in traffic-related messaging. These mechanisms help to mitigate the effects of operability denial attacks.

The final outcome of this thesis is that a security solution must be highly customized to be effective, due to the abundance of requirements given by the diversity of application scenarios and expandability of the security architecture. The proposed security architecture, which has been tailored to the needs of inter-vehicle communications, offers significant advantages with respect to possibility of maintenance in a global context, privacy and scalability by exploiting the properties of a car-based network. The results of simulations and prototypical implementations suggest that a information-based view supported by scenario-building will strongly enhance the local reasoning about specific traffic-related events. Communication can be conducted while preserving privacy towards third parties and routing mechanisms can be hardened against the most imminent threats with support from a security backend. If security issues are elaborately evaluated and the implementation is carefully crafted for those applications that are suited for this kind of communication, inter-vehicle communication provides many benefits and can contribute to traffic safety, traffic efficiency and driver comfort significantly. The information-based paradigm provides an enormous potential for other scenarios as well, where mobility and decentralization are decisive factors.

Acknowledgements

First and foremost I want to thank Uwe Baumgarten for his support and integration in the field of computer science, his guidance throughout my PhD studies and a great working atmosphere at his research group. His comments have always been helpful and encouraging. He is one of those professors that are there when you need them and listen to you when you have something to say.

I am most grateful to Timo Kosch not just for being such a supportive thesis advisor at BMW but also for becoming a real friend. His understanding and support in all kinds of situations is clearly something special.

I would like to especially thank Karl-Ernst Steinberg, the head of the Information and Communication group and Raymond Freymann, the head of BMW Group Research and Technology, for giving me the opportunity to pursue my PhD at BMW Group.

I am deeply grateful to BMW Group Research and Technology for providing me with an excellent work environment and simply for consisting of so many great people. It was a real pleasure to work with my colleagues in the Information and Communication group, mainly Hans-Jörg Vögel, Mikhail Smirnov, Michael Dinkel, Markus Strassberger, Franz Bernt and many other people whose names I cannot list here alltogether. Also, I would like to thank all the students whose work I supervised and who have a great part in realizing many of the concepts and finding new solutions, literally all of them being fascinated with the potential of the technologies we have been working on. Wernhard Markwitz has been reviewing much of my work during different phases and gave invaluable comments on scientific methodology and presentation of the contents. These comments and the constant push to move on in the finishing phase have added a significant incentive to proceed.

Last but not least, I would like to sincerely thank my parents for their love and unshakable belief in me, for supporting all my big steps in life and for respecting all the decisions I made whether they agreed with them or not. I would definitely had enjoyed my life a whole lot less without them.

Contents

ABSTRACT iii

Acknowledgements v

List of Figures xi

List of Tables xv

List of Abbreviations xvii

Chapter 1. Introduction 1

1.1. Motivation 1

1.2. Related Work 2

1.3. Problem Outline 4

1.4. Structure and Contributions 5

Chapter 2. Inter-Vehicle Communication 9

2.1. Ad-hoc Network Basics 9

2.2. Automotive Environment 14

2.3. Relevant VANET Properties 17

2.4. VANET Applications 19

2.5. Requirements 22

Chapter 3. System Model and Threat Analysis 25

3.1. System Model 25

3.2. Information Security 29

3.3. Threat Analysis 35

3.4. Security Objectives 43

Chapter 4. Architecture and Building Blocks 47

4.1. Definitions and Terms 47

4.2. SARI - Secure Architecture for Robust Inter-vehicle Communication 47

4.3. SARI and Privacy 53

4.4. SARI Building Blocks 63

4.5. Conclusion 76

Chapter 5. Vehicle to Vehicle Scenario 77

5.1. Terminology, Requirements and Concepts 77

5.2. Classification of Traffic-Related Applications 82

5.3. Distributed Data Aggregation of Area Events 87

5.4. Local Reasoning 91
5.5. Reputation System 93
5.6. Voting Schemes 103
5.7. Confidence Level and Decisions 107
5.8. Conclusion 108

Chapter 6. Vehicle to Infrastructure Scenario 111
6.1. Use case and Applications 111
6.2. Communication Patterns, Requirements and Security 112
6.3. V2I Protocol 113
6.4. Conclusion and Security Evaluation 118

Chapter 7. Connection-Oriented Communication Scenario 119
7.1. Secure Routing in Ad-hoc Networks 119
7.2. Performance of Secure Routing Protocols 127
7.3. AODV-SEC 130
7.4. Conclusion 134

Chapter 8. Prototyping and Simulation 135
8.1. Development and Verification using Prototyping and Simulation 135
8.2. Prototyping of Privacy Concepts 135
8.3. Traffic Light Lab Setup 136
8.4. Simulation Environment 140
8.5. Information Distribution Related Implementations 144
8.6. VARS Simulation 146
8.7. Voting Schemes 147
8.8. AODV-SEC 153
8.9. Summary 156

Chapter 9. Conclusion 157
9.1. Conclusion 157
9.2. Directions for Future Work 161
9.3. Outlook 161

References 163

Glossary 175

Glossary 175

Appendix A. Pseudonymous Credentials: Key Encryption Protocol 183
A.1. Encryption of Credentials 183
A.2. Decryption of Credentials 184

Appendix B. Statecharts 185
B.1. Statechart JavaCard Implementation 185
B.2. (UML) Action Diagrams Distributed Data Aggregation 186

Appendix C. Simulation Charts 189

List of Figures

1.1 Overview Topics 5

2.1 Different Types of Ad-hoc Networks 10

2.2 Hidden and Exposed Nodes 11

2.3 Type-1: V2V Traffic-related Applications 19

2.4 Human Machine Interfaces 21

2.5 Type-2: V2I Applications 21

2.6 Type-3: COC Applications 22

3.1 Sensors and Human Input 27

3.2 Determining the Traffic Scenario 28

3.3 Security Engineering Abstraction Layers 31

3.4 Graphical and Textual Representation of Attack Trees 33

3.5 An example Attack Tree: Steal a Car 33

3.6 Attackable Components 37

3.7 Overview Attack Trees 38

4.1 General Architecture 49

4.2 TRM Lifecycle 52

4.3 Initialization Phase & Architecture 56

4.4 Operational Phase 57

4.5 Revocation Phase 57

4.6 Classification of Existing Trust Establishment Schemes 64

4.7 SARI Trust Model 66

4.8 Satellite Positioning Principle 68

4.9 Attacks on Satellite Positioning 69

4.10Broadcast Storm 71

4.11Re-transmission Optimization 72

4.12Flowchart Message Received 73

4.13Virtual Backbone 75

5.1 Event Situation Timeline 77

5.2 Phases Traffic-Related Messaging 79

5.3 Geographic Areas 81

5.4 Dynamics 83

5.5 Driver Related Determinism 83

5.6 Configuration Dependency 84

5.7 Aggregation Locality 85

5.8 Data Collection 85

5.9 Impact 86

5.10Detecting a Hazard 89

5.11Recognition Rectangle and Curvy Road 89

5.12Generating an Recognition Area 90

5.13Adding a Margin Point 90

5.14Shrinking the Recognition Area - Inbound 91

5.15Shrinking the Recognition Area - Leaving 91

5.16Combining Recognition Areas 91

5.17Event Generation 92

5.18Sample Reputation System 95

5.19VARS Architecture 96

5.20Piggybacking of Reputation Information 98

6.1 Vehicle to Infrastructure Communication Protocol 116

7.1 50 nodes, 200s, 60% CBR, BMW GenMobTrace 129

7.2 100 nodes, 200s, 60% CBR, CMU Setdest 131

7.3 Hash chain example (AODV-SEC) 133

8.1 Setup of a Large Intersection 138

8.2 Simulator Coupling 143

8.3 A Screenshot of a VANET Simulation Run Playback 144

8.4 Simplified Algorithm for Efficient Message Distribution Algorithm 145

8.5 VARS Simulation 147

8.6 Comparison of the Four Voting Methods 150

8.7 Progression diagram visualizing the development of important key figures of the simulation run: Majority of Freshest 22 with Threshold 2, Flip Attack, 20% Attackers 151

8.8 Packet Delivery for Scenario 1 154

8.9 Route Discovery Delays for Scenario 2 (no pause time) 155

8.10Data-Throughput Results for Scenario 2 with increasing load 155

B.1 Statechart for JavaCard Implementation 185

B.2 Distributed Data Aggregation: addMessage(M) 186

B.3 Distributed Data Aggregation: addDetection(M) 187

C.1 Overview Application Simulation Implementation 189

C.2 Overview Secure Routing Protocols 190

C.3 50 nodes, 200s, 30% CBR, BMW GenMobTrace 191

C.4 100 nodes, 200s, 30% CBR, BMW GenMobTrace 192

C.5 50 nodes, 100s, 30% CBR, CMU Setdest 193

C.6 Normalized Routing Load for Scenario 2 194

C.7 Normalized MAC Load for Scenario 2 194

List of Tables

2.1 IVC Applications 20

2.2 Applications Type Comparison 22

3.1 Sensors and Human Input 27

3.2 Overview Traffic Scenarios - Factors derived from Local Parameters 29

3.3 Attacker Model 36

3.4 Subtree: Alter User Information (V2V) 38

3.5 Subtree: Attack Privacy (V2V) 38

3.6 Subtree: Deny Services (V2V) 39

3.7 Subtree: Attack Privacy (V2I) 39

3.8 Subtree: Deny Services (V2I) 40

3.9 Subtree: Inject Messages (COC) 40

3.10Subtree: Attack Privacy (COC) 40

3.11Rating of attacks according to the DRED model 42

3.12Security Objectives per Application-Type 45

4.1 Variables and Parameters for Algorithms 60

5.1 Event Classes 87

5.2 Event Class Presets 87

5.3 Example: Event Presets for Small Hazard Class 88

5.4 Variables and Parameters 100

5.5 Comparison of "Confidence Evaluation" Schemes 109

6.1 Variables and Parameters for Protocol 115

7.1 Simulation Parameters 128

7.2 Comparison of Routing Protocols 134

8.1 Memory Consumption for Key Length 1024 bit 136

8.2 V2I Protocol Performance 139

8.3 Simulation Parameters 149

8.4 Time delays due to cryptographic operations 153

8.5 Route discovery times of different multihop connections 153

A.1 Variables and Parameters for Key En-/Decryption Algorithm 183

List of Abbreviations

ACC	Adaptive Cruise Control
C2C-CC	Car to Car Communications Consortium
CA	Certificate Authority
CBR	Constant Bit Rate
COC	Connection Oriented Communication (type-3)
CRL	Certificate Revocation List
DARPA	Defense Advanced Research Projects Agency
DREAD	Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability
DRED	Damage Potential, Reproducibility, Exploitability, Discoverability
DSRC	Dedicated Short Range Communication
ECU	Electronic Control Unit
EMP	Electro-Magnetic Pulse
FCD	Floating Car Data
ETSI	European Telecommunications Standards Institute
GLOMOSIM	Global Mobile Information Systems Simulation
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HUD	Head Up Display
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
IT	Information Technology
IVC	Inter-Vehicle Communication
MAC	Media Access Control or Message Authentication Code
MANET	Mobile Ad-hoc Network

NIC	Network Interface Card
NS-2	Network Simulator 2
OBU	On-board Unit
OSI	Open Systems Interconnection
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
PKI	Public-Key Infrastructure
POI	Point of Interest
RA	Registration Authority
RERR	Route ERRor
RREP	Route REsPonse
RREQ	Route REQuest
RSU	Road-side Unit
SAE	Society of Automotive Engineers
SARI	Secure Architecture for Robust Inter-vehicle communication
SUMO	Simulator of Urban Mobility
TTL	Time To Live
TTP	Trusted Third Party
UML	Unified Modeling Language
V2I	Vehicle to Infrastructure (type-2)
V2V	Vehicle to Vehicle (type-1)
VANET	Vehicle Ad-hoc Network
VARS	Vehicle Ad-hoc network Reputation System
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

CHAPTER 1

Introduction

The first section of this chapter will explain current developments in communication and information technology, motivate why inter-vehicle communication is a promising new technology and especially point to security issues in such systems. The second section outlines the current state-of-the-art in ad-hoc networking, inter-vehicle communication and ad-hoc network security. The third section provides a top-level overview on the objectives of this thesis. The last section lists the scientific fields that are used in this work, explains the structure of this thesis, and student activities as part of this work.

1.1. Motivation

New wireless technologies and the increasing number of mobile devices change the way people are using information technologies. ”Ubiquitous Computing” and ”Pervasive Computing” are terms that reflect the omnipresence of information and communication technologies in our lives. This shift from immobile computing to mobile computing also requires to rethink the paradigms of communication. Static, wired communication infrastructure is extended to accommodate mobile usage, such as cellular networks, local hotspots provide locally stored information and also act as access point to other networks. Mobile devices themselves may be connected together using point-to-point technologies, either using cables, infrared or radio communications.

All mobile devices equipped with wireless communication may be operated as nodes forming an ad-hoc network. This kind of network offers a simple and efficient way for local connectivity and exchange of location-related information, because it does not require connections to infrastructure.

One of the main advantages of an ad-hoc network is complete decentralization during normal operation. This means that the introduction of such a technology in principle only requires upgrading the software of devices equipped with a wireless communication technology. If one thinks about the great numbers of such devices already brought into the market, there is a huge potential for ad-hoc networking.

Before the benefits of ad-hoc networks can be exploited, there are a number of issues that have to be solved first. Trust establishment is difficult to achieve in a decentralized context, scalability is a major concern in a self-organized scheme that cannot rely on managed infrastructure. In such a network, where devices communicate freely and forward others’ data, designers also have to think about privacy-related topics and cooperation among nodes. Security in general is difficult to achieve, since all reactions have to work in a distributed way or at least reflect the special architecture of ad-hoc networks.

A special class of mobile devices are cars. Future cars, that are equipped with powerful on-board computers and wireless communication technologies will provide all essential features of other mobile devices as presented by [**KMSB01**]. Additionally, they have some properties, that make them well suited for the operation of decentralized, self-organizing ad-hoc networks. For instance, cars have a constant power supply while driving, which drastically reduces the problem of limited battery power. There will be an in-depth discussion about VANET properties in section 2.3. Unfortunately, a network consisting of cars also introduces some new challenges, such as dealing with a very large number of nodes, high average mobility or very limited user interactions. This thesis will focus on the specifics of an automotive environment.

Many applications have been proposed for IVC - Inter-Vehicle Communication. In this work they are roughly categorized into vehicle to vehicle traffic-related, vehicle to infrastructure and connection oriented communication, see section 2.4.2. The benefit for car owners is that local information can be exchanged directly among participating nodes without causing extra costs. Communication can increase traffic safety by mitigating dangerous situations, it can increase traffic efficiency through optimized traffic routing, it can support driver assistance systems, it can improve comfort functions and provide additional, situation-related information for the driver.

The security and dependability of traffic-related messaging is obviously critical for safety in two ways, on one hand it may provide additional information and thus helps to lower the number of accidents or improve traffic efficiency. On the other hand serious problems will arise if the system can be easily manipulated. Another important thing to consider is that the user's privacy must not be diminished while using such a system. This can have an impact on user acceptance of such a system and therefore also influence market introduction, which is in turn critical for the overall system's performance.

1.2. Related Work

This section presents related work to create an overview on the scientific topics relevant for this work in creating a Secure Architecture for Robust Inter-vehicle communication - SARI. A more detailed discussion on related work will be given in the context of specific technologies in respective chapters and sections.

1.2.1. Ad-hoc Networks

Ad-hoc networks have first been discussed with military applications in mind, [**HDL+02**] gives a good overview on the whole topic. A lot of work has been dedicated to radio technologies, medium access schemes or routing that reflect the specific properties of ad-hoc networks.

Humayun Bakht [**Bak05**] divides the development of Mobile Ad-hoc NETworks in three generations. The first generation are Packet Radio Networks, starting at 1972, which employed first medium access approaches and simple distance-vector routing for usage in combat environments. The second generation emerged in 1980s, driven by DARPA's Survivable Radio Network (SURAN) project, providing packet-switched network for a mobile battlefield environment without infrastructure. In the 1990s, commercial ad-hoc networks appeared and IEEE 802.11 subcommittee had adopted the term ad-hoc networks. Notebook computers and later mobile devices such as PDAs and cellphones offered the technical basis for ad-hoc networking and research conferences discussed specific problems with such equipment such as limited battery life.

Current developments include advanced coding and radio technologies, efficient medium access schemes and routing algorithms suitable for various scenarios where ad-hoc networks are used.

1.2.2. Inter-vehicle Communication

Inter-vehicle communication can be seen as a special discipline of mobile ad-hoc networks. However, the first ideas of communicating cars were merely dedicated to extend the range of local sensors, a good overview is presented in [**Per03**]. Later, efficient information dissemination schemes have been proposed, that consider high-density and high mobility effects of vehicle-based networks [**KSA02**]. In addition to direct communication among cars, extending on-board sensors, inter-vehicle communication has been seen in a wider context in a number of papers, such as [**KMSB01**].

Research projects, involving aspects of inter-vehicle communication, have been sponsored by the EU, such as Cartalk - [**Pro01a**] and Preventive Safety - [**Pro03a**]. National research agencies have sponsored additional research projects, such as Vehicle Safety Communications - [**Pro05b**], Vehicle to Infrastructure Initiative - [**oT**] and the PATH project - [**Pro05a**] in the USA and Network on Wheels - [**oWP04**], Fleetnet - [**Pro04**] and Invent - [**Pro01b**] in Germany.

Some of those projects have operational prototypes: FleetNet [**FEL01**], Carisma [**Kos04b**] and one as part of the VSC-project [**Pro05b**].

The car-to-car communications consortium [**tCCC04**] is an organization founded by european car makers to form an international standard for inter-vehicle communication.

1.2.3. Security in distributed Systems

Military scenarios have stimulated the first major applications of ad-hoc networks. Obviously, military networks have a demand for security and many different approaches have been introduced to achieve secure communication. However, the military setting is different from a civilian one:

Single Root of Trust: A military unit follows the orders of a single superior commander, meaning there are no conflicting parties being sources of trust and there are processes that define how equal-level authorities resolve decisions. This does not have to be true in a commercial / civilian environment, where customers may buy products from competing companies or where national boundaries hamper the cooperation between organizations.

Common User Motivation: In the military, the users of communication devices and other equipment have a common goal, which has a higher priority than communication itself. Thus, user behavior within a communication system can be enforced by measures that are outside the technical domain. In civilian environments this is usually not the case. Users exploit the benefits of their equipment independently of other users and there is little opportunity to hold users accountable for the inappropriate usage of their devices.

Low Average Mobility: The average mobility of military units moving offroad and in hazardous environments is lower than average speeds of nodes in a comparable civilian setting with intact road infrastructure. Also, due to their common objectives, mobility patterns of military units correlate to a higher degree and relative speeds are lower on average - think of an armored unit moving towards a target area compared to cars moving in opposite directions that meet on a highway.

Hierarchical Structure and Infrastructures: A military organization must have a chain of command and procedures for handling sensitive information. The existence of an infrastructure that accommodates the hierarchical structure is useful for setup and maintenance operations of an ad-hoc network, especially with respect to a security infrastructure. In civilian systems however, the operation of such an extended infrastructure will most likely be too costly and due to market competition it will be most likely heterogeneous.

This explains why some of the concepts developed in a military context do not work in the context of this work.

Researchers realized that security is difficult to achieve in decentralized systems. A number of approaches have been proposed, [**BFL96**] and [**SH02**] summarize some of the most promising ones.

1.3. Problem Outline

Preceding sections help to understand that highly-mobile ad-hoc networks are a new field of technology, where the resulting combination of scientific fields, such as information and communication security, wireless communications, ad-hoc networking, traffic management systems, demands new approaches, because conventional ones are usually not directly applicable. Even fundamental questions, such as how to protect vehicle systems against manipulation, how to ensure privacy in mobile environments or how to efficiently handle traffic-related information are still unanswered and the resulting scientific field itself remains immature. This requires in-depth involvement in a broad range of topics such as automotive (vehicle architecture, traffic management, industrial processes, etc.), information security (security engineering, tamper resistant devices, etc.), key management (key establishment protocols, organizational structures, algorithms, etc.), cryptographic methods (modular arithmetics, cryptographic functions, algorithms, etc.), privacy (threats, criteria, architectures, etc.), communication protocols, simulation and practical realization issues in automotive environments.

A consequence of the broad perspective is that an effective and secure architecture cannot be completed without a more detailed specification of relevant components that themselves depend on the system's operational environment. This requires a thorough evaluation of environmental/external conditions and their effects on system design.[1] System architecture and prototypical implementations will have to respond to the automotive usage profile considering inherent customer needs, while fundamental concepts (e.g. distributed architecture, adaptability, modularity) presented in this thesis should be universally valid. This requires an abstraction and structuring of the whole field of inter-vehicle communication, including the main use-cases. Starting from a generic system model, threats have to be analyzed and security objectives have to be defined to be able to synergize new concepts and existing technologies to an overall system architecture.

In short, the problem can be outlined as follows: To develop a inter-vehicle communication system and application framework that exploits advantages of the automotive basis while responding to special conditions given in this context. The proposed solution should address the variety of use cases or scenarios and be robust and secure in a sense that it will sustain dedicated attacks and extreme conditions occurring during normal usage. Overall complexity and operational overhead should be as low as possible and reflect characteristics of a global usage.

This work will evaluate the requirements for an inter-vehicle communication network and assess security issues including relevant dependability topics in order to achieve the ultimate goal of designing a scalable ad-hoc communication system suitable for use in highly mobile scenarios. The analysis, design and implementation will be complemented by theoretical and simulative evaluation of the component's performance to verify feasibility. A detailed description of the topics covered in this thesis is given in the next section, paragraph 1.4.1. This thesis however does not provide full validation of all concepts and

[1] Later in this thesis conditions that restrict system design will be distinguished between constraints, determined by external factors and assumptions that have been made according to logical estimations, best practice approaches and simply common sense.

their interactions as required for a mature prototype. Also, an complete overall simulation including all components and proposals has not been implemented due to the complexity of an overall communication, traffic flow and application simulation with extensive test-cases on top of that. Concepts have been tested to verify their basic principle, not to ensure full validation under all circumstances which would be necessary for serial production.

1.4. Structure and Contributions

1.4.1. Covered Topics and Structure

Referring to the problem outline in the previous chapter, this paragraph will indicate the topics that are related to the work included in this thesis. The first part of this work discusses factors to be considered in the given operational environment and identifies criteria and requirements affecting system design. Building on the definition of a basic system model is designed. A threat analysis building on that model leads to the definition of security objectives. This concludes the first part of this work, containing chapters 1 to 3. The second part starts with a chapter about the synthesis of a system architecture, called SARI - Secure Architecture for Robust Inter-vehicle communication and introduces basic concepts for it. The following chapters discuss other parts of the technical solution, an overview is given in Figure 1.1.

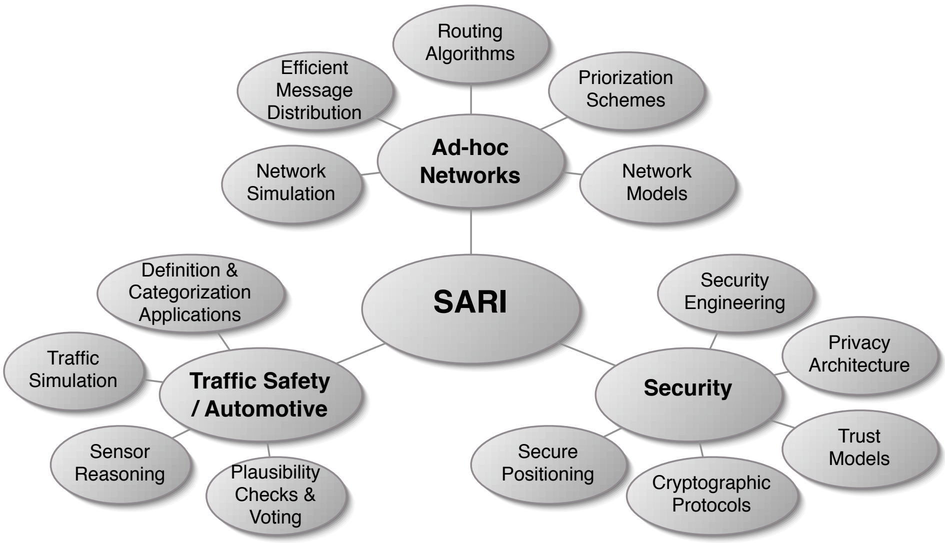


Figure 1.1. Overview Topics

The technical solutions relate to three major areas of research: ad-hoc networks, information security and traffic safety / automotive applications. In the field of ad-hoc networks: efficient message distribution, routing algorithms, priorization schemes, and network models. In the security area: security analysis, privacy architecture, trust models, cryptographic protocols (privacy, V2I), and secure positioning. And in the traffic safety / automotive application field: definition and categorization of applications, simulation, data structures for distributed data aggregation, sensor reasoning and information-centric plausibility checks. The second part includes chapters 4 to 7.

The last part, consisting of chapter 8 and 9, presents implementations and simulations that relate to the technical solutions in the second part. While the second part provides a theoretic view on the solution and a theoretical analysis - where possible, the third part focuses on implementation specific findings and practical evaluation of these concepts. The final chapter concludes this thesis, sums up the major contributions and points to future work.

1.4.2. Methodology

The rationale behind the structure of this thesis is straightforward:

(1) Introduction to the general topic and definition of the problem.
(2) Introduction of fundamental concepts, definition of terms, description of the operating environment's specifics and identification of constraints, assumptions and overall requirements.
(3) Structuring of use-cases, development of a system model, security analysis and definition of security objectives.
(4) Synthesis of a system architecture and supportive concepts.
(5) Elaboration of different approaches according to three application scenario types.
(6) Evaluation of approaches, concepts and techniques. Conclusion.

1.4.3. Associated Work

Student work that has been supervised as part of this thesis includes:

- Ein verteiltes nicht-interaktives Authentisierungskonzept für mobile Ad-Hoc Netze
  Richard Wimmer - Diploma Thesis
- Seminar on Ad-hoc Networks including the work of
  - Adam Burg: Ad-hoc network specific attacks
  - Michael Dyrna: Peer2peer Network Service Discovery for Ad-hoc Networks
  - Ying Li: Micropayment-schemes for ad-hoc networks
  - Andreas Meier: IDS in Ad-Hoc Networks
  - Johann Niklas: Evaluation of distributed trust concepts
  - Nicolas Padoy: Secure routing in ad-hoc networks
  - Fabian Schilcher: Key management and distribution for threshold cryptography-schemes
- Analysis of Ad-hoc Network Routing Security using Network Simulation
  Nicolas Padoy - Technical Report
- Reputationssysteme in großen, hoch-mobilen Ad-hoc-Netzen
  Przemyslaw Magiera - Diploma Thesis
- Secure Routing in an IEEE 802.11 based Vehicular Ad-hoc Network
  Javier Fabra - Diploma Thesis
- Entwicklung eines Motivationssystems für Ad-Hoc Netzwerke
  Florian Schreiner - Diploma Thesis
- Mobile Ad-Hoc Netzwerk Kommunikation mit fester Infrastruktur
  Florian Kohlmayer - Diploma Thesis
- Implementation of Algorithms for Information Aggregation in VANET Environments
  Christian Härdt - Technical Report
- Analysis and Implementation of Forwarding Algorithms in Vehicle Ad-hoc Networks
  Borislav Vangelov - Technical Report

- Integration of Traffic and Application Simulation into a NS-2 Environment
  Rumen Tashev - Technical Report
- Analysis and Improvement of Inter-Vehicle Communication Security by Simulation of Attacks
  Benedikt Ostermaier - Diploma Thesis

This work has been largely funded by BMW Research and Technology. Furthermore it was carried out under supervision of Professor Baumgarten, TU Munich as member of his mobile distributed systems research group. In addition, some parts of this work have been done in close collaboration with the institute for data processing, the institute for communication networks of TU Munich and the research group IT-security of TU Darmstadt.

CHAPTER 2

Inter-Vehicle Communication

The goal of this chapter is to give an introduction and point out relations to associated technical disciplines. Building on that, the specifics of the operational environment are described and evaluated. After identifying the scope of applications and general constraints, assumptions and overall system requirements are presented.

The first section of this chapter describes ad-hoc networks and explain different technologies used for ad-hoc networks, while in the second section specifics of inter-vehicle communication in relation to other forms of ad-hoc networks are discussed. In the third section those properties of the automotive environment that affect system design are analyzed one by one. The fourth section is about applications types that will be in the focus of this work and why they are of special importance. Finally in the fifth section overall requirements for the system design are derived.

2.1. Ad-hoc Network Basics

Royer [**RT99**] defines Ad-hoc Networks as follows:

Definition 1 (ad-hoc network)**.** *An ad-hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes.*

The properties of ad-hoc networks according to this definition clearly separate them from conventional wireless networks, such as cellular networks or hotspot-oriented architectures:

Decentralized: In this work a decentralized system is a distributed system which functions without an organized center or authority. All nodes have the same capabilities, while they may perform different roles at a specific point of time, such as cluster-head and cluster-node. [1]

Self-Organizing: In the case of ad-hoc networks self-organization means that all functions that are necessary to establish and maintain data communication are realized without requiring human intervention during normal operation.

Wireless: Nodes of an ad-hoc network communicate via wireless communication technologies.

Multi-Hop: An ad-hoc network allows forwarding or routing of network packets.

Due to these properties, ad-hoc networks are ideally suited for applications where the amount of local information exchange exceeds long-range communication and infrastructure is not or only partially available, such as in disaster relief, law-enforcement, military, etc. Also in other scenarios, those properties provide advantages over conventional systems, even if these are not considered fully-fledged

[1] Gerard Tel lists three distinct properties of decentralized systems in [**Tel94**]: 1. Lack of knowledge of a global state, 2. Lack of a global time frame and 3. Non-determinism. In this work # 2. does not apply because of the availability of a global time reference, see 1

ad-hoc networks for example the ad-hoc mode of IEEE's 802.11 type networks or Bluetooth personal area networks.

2.1.1. Relation Between Different Self-organizing Network Technologies

Ad-hoc networks and peer-to-peer networks share the concept of self-organization. However, peer-to-peer networks are an application running over a given network (a so-called overlay network), whereas the notion of ad-hoc networks usually refers to the combination of a dedicated network and applications running over it. In contrast to peer-to-peer networks, ad-hoc networks have to deal with issues on medium access and networking layers, as well as mobility and decentralization.

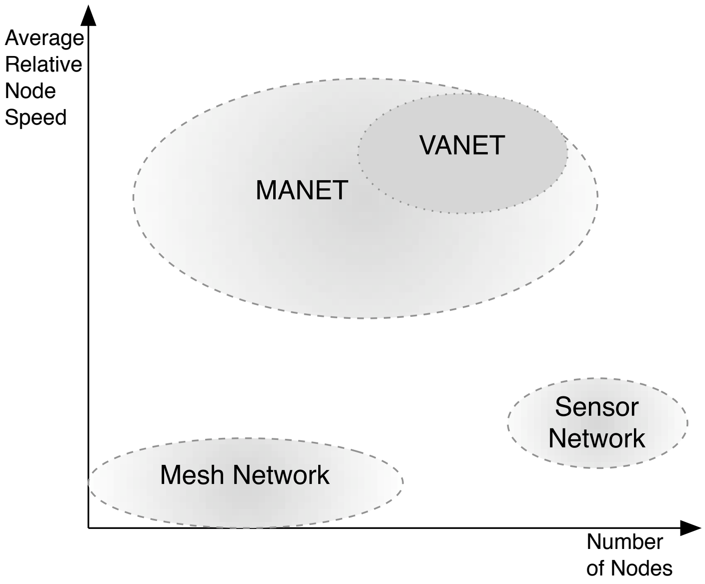


Figure 2.1. Different Types of Ad-hoc Networks

There exist various specializations of the generic ad-hoc network definition. An overview is given in Figure 2.1. One of them are so-called mesh networks. This term is generally used for networks with fixed nodes, which will not be considered in this work.[2] Another specialization are MANETs, where node mobility is an essential property of network nodes. Some general effects of mobility will be discussed in section 2.1.5. Yet a more specialized version of MANETs are VANETs, the main focus of this work. A detailed discussion about the implications of a vehicle environment is given in section 2.3. Sensor Networks are intelligent sensors that are connected among each other and to a gateway node using ad-hoc network technologies. Refer to [**KKP99**] for an example of sensor networks.

The following subsections give a short introduction to the technology of mobile ad-hoc networks (MANET) and vehicle ad-hoc networks (VANET), their theoretical scalability and factors related to ad-hoc network performance evaluation.

2.1.2. Radio Technologies

Although the concepts presented in this work will work with different radio technologies and physical layer technologies are not within the focus of this work, it is important to understand some basic principles of

[2] To avoid confusion: fixed nodes are used in this thesis, but not fixed nodes that are directly connected to each other with an ad-hoc protocol.

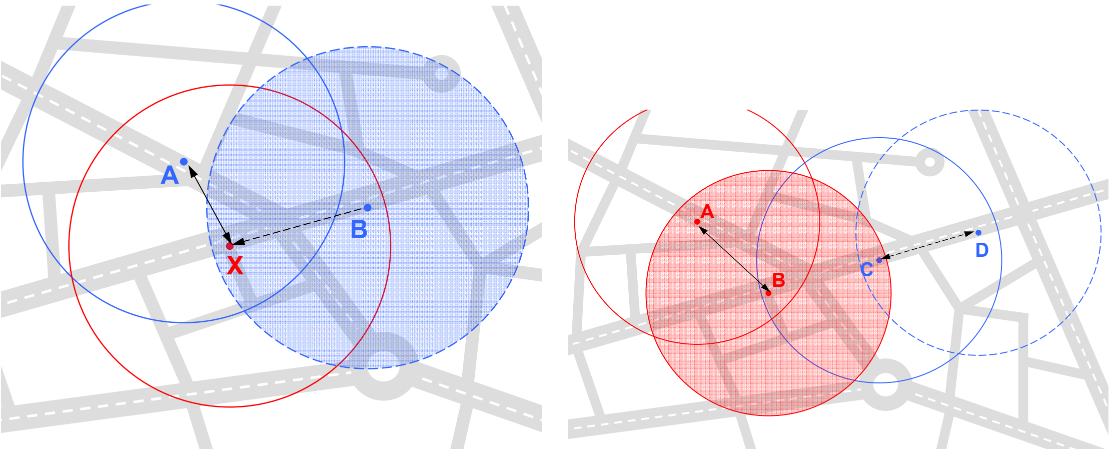
the methods used in these standards. This is important for realizing their implications on system design with respect to security and dependability. For a deeper understanding of radio technologies in general, refer to [**Rap99**] and [**Pro00**]. The physical layer technologies as used in IEEE's 802.11 standards will be used as a reference in this thesis.

Almost all modern wireless communication systems use some kind of spread spectrum technology, such as frequency hopping, pseudo random noise spreading, etc. Spread spectrum generally makes use of a sequential noise-like signal structure to spread the normally narrow-band information signal over a relatively wide band of frequencies. The receiver correlates the received signals to retrieve the original information signal. This technique decreases the effects of narrow-band interferers, such as other nodes, other electrical devices (e.g. microwave ovens) or weather phenomenons. If the spreading range is large enough, this can potentially reduce the effect of radio jamming, a property that is exploited by military communication systems.

Transmission quality is usually not the same in both directions, due to different frequencies, interferences, and path losses. This is especially true for mobile environments and if directional antennae are used. As we will see later, this will also have implications on systems design since it complicates handshake protocols and overhearing the correct forwarding of messages.

2.1.3. Medium Access

Medium access is critical for scalability in large ad-hoc networks, since a wireless channel is always a shared resource. Note that the way of modeling medium access effects will have a direct impact on simulation results in densely populated network environments. Distributed wireless communication systems face additional challenges, because medium access management has to work in a decentralized way. The following paragraphs give a short overview on relevant topics.



(a) Hidden Node Problem

(b) Exposed Node Problem

Figure 2.2. Hidden and Exposed Nodes

2.1.3.1. Hidden Node / Exposed Node. In wireless communication systems the hidden node problem (see Figure 2.2(a)) occurs when a node A is visible from a wireless node X, but not from other nodes communicating with this node, such as node B. Thus, node B tries to establish communications with X without knowing that this interferes with A. As a result, fragmented packets have to be dropped or transmission quality degrades until the situation is resolved.

The exposed node problem (see Figure 2.2(b)) occurs when two nodes A and B communicate with each other, while node C overhears B's transmissions. Node C therefore rejects communication request from node D to prevent interference although this would not occur since B and D are not within range of each other. This results in limited communication bandwidth.

In a vehicle-based network, these situations may occur quite often due to the nodes' high average mobility and they do have an effect on the system's performance in high node density situations. Section 4.4.4 will present a methodology for efficient message distribution in densely populated situations, that also considers the hidden and exposed node problem.

2.1.3.2. Power Control vs. Adaptive Coding. In high node-density situations it is questionable whether to adjust the transmission power of nodes or not. On one hand, adaptation of transmission power reduces the range of nodes and therefore increases the reuse of frequency bands in a given geographic area. In other words, if there are many nodes in a given area, they can reduce their transmission power to such a degree that they only reach their immediate neighbors. This allows nodes that are two hops away to use the same frequency band without interference. Since frame retransmissions due to collisions on communication channels depend to a large degree on the number of nodes within range, this can increase bandwidth efficiency significantly. On the other hand however, neighborhood information is essential to adjust transmission power accordingly. But neighborhood information means overhead and can be in conflict with privacy requirements. Also, power control may become too complex in a highly dynamic, decentralized system.

The alternative is to adapt coding schemes to a given Bit Error Rate (BER). The effects of higher bandwidth efficiency may be reached by employing larger symbol alphabets for modulation if higher signal to noise ratios, resulting in a lower bit error rate, are available as shown by Beyer in [**Bey02**].

In this work it is assumed that there is no power control mechanism and the modulation depends on the wireless channel's bit error rate, according to 802.11 standards.

2.1.4. Wireless Distribution

There are two major principles how information can be selectively distributed in an ad-hoc network:

Definition 2 (Routing). *Routing is a means of selecting paths in a ad-hoc network along which information should be sent. This is performed by a network layer, which means that routing is independent from application specific information.*

Definition 3 (Forwarding). *Forwarding is the distribution of information using application specific information. Thus, a forwarding metric has to be defined for every application using forwarding.*

Additionally, packets can also be broadcasted (or "X-casted", see next paragraph), where nodes drop packets they are not interested in. In some cases this is the easiest way, but as the node density increases the bandwidth efficiency drops dramatically. One of the effects that affects connectivity negatively is called broadcast storm, discussed in section 4.4.4 (see also [**Kos05**]).

2.1.4.1. X-cast. In communication networks, a distinction has to be made between following addressing methodologies:

Definition 4 (Unicast). *Unicast is the act of transmitting data to a single node with a distinct destination address.*

Definition 5 (Broadcast)**.** *Broadcast is the distribution of data to a number of recipients. Since the wireless channel is a shared medium, broadcast may be achieved on MAC layer where all nodes listening on a specific channel will receive the data. However, a logical broadcast may also be achieved on higher layers in the OSI-model, where nodes distribute information, while using unicast mechanisms at the MAC layer.*

Definition 6 (Multicast)**.** *Multicast is the distribution of data to a number of recipients that are members of a multicast group. A multicast group usually refers to address-ranges or groups of node IDs.*

Definition 7 (Geocast)**.** *Geocast is a form of multicast, where the multicast group is not specified through address-ranges or node IDs but through geographical positions of nodes at a given point in time.*

2.1.5. Mobile Ad-hoc NETworks - MANETs

MANETs according to Royer [**RT99**] have properties that are beneficial for use in a car-based network (which makes them VANETs), when compared to conventional technologies:

- **Available** Multiple distribution paths among nodes increase the probability of successful information dissemination.
- **Decentralized** Ability to work in environments or situations where infrastructure is not available. Additionally, users maintain control over their equipment, which is a significant property with respect to privacy.
- **Self-organizing** No network maintenance necessary.
- **Mobile Environment** Dynamic topology changes are not only supported, the network can also take advantage of mobility.

Other properties are rather problematic, when compared to infrastructure-based mobile communication systems:

- **Limited Resources** Nodes of a MANET have limited resources, such as memory, computational power, energy. While in some cases the distributed concept does compensate the effects on global scalability, in other cases it does not.
- **Limited Bandwidth** MANETs are usually based on publicly available frequencies, meaning that communication bandwidth is limited. Efficient bandwidth usage is a challenging task in mobile distributed systems.
- **Hostile Environment** The whole network is decentralized which means that there is no "back end" that is professionally maintained and secured. Literally all equipment is "in the field" and controlled by normal users.

Section 2.2 will point out additional points for car-based MANETs, called VANETs - Vehicle Ad-hoc NETworks.

2.1.6. Scalability of Ad-hoc Networks

The theoretical background to scalability of ad-hoc networks has been studied, among others, in [**LBC$^+$01**] and [**GK00**]. Both conclude that scalability in ad-hoc networks strongly depends on local communication load and how far information is forwarded in terms of number of hops. If the average number of hops exceeds a certain limit, the system performance is severely affected. The MANET routing community has established a general rule of thumb that this limit is roughly in the order of seven hops for existing implementations. The conclusion drawn from this constraint is that routed applications (in an ad-hoc network) should focus on local distribution of messages. Furthermore it can be deduced that wireless ad-hoc networks will not replace conventional forms of networking on a global scale. It is crucial to recognize this, since it affects the whole paradigm of how different forms of networks co-exist and cooperate.

2.1.7. Connectivity Parameters

Connectivity parameters are important metrics for determining the communication situation in a mobile network. First, there are global connectivity parameters that can only be calculated by centralized observation. Some of this information has been used in simulations to evaluate the effectiveness of different system configurations. Second, there are local connectivity parameters that can be calculated by nodes themselves to determine the communication environment, explained in section 3.1.5.

Global Connectivity Parameters:

- **Clustering Coefficient** Watts and Strogatz [**WS98**] introduce the clustering coefficient graph measure to determine whether or not a graph is a small-world network. The clustering coefficient Ci for a vertex vi is the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them.
- **Number of Neighbor Distribution** Nodes in a MANET are usually keeping track of all the neighbors in their communications range. The distribution of the number of neighbor in a MANET is a good indicator for connectivity homogeneity.

Local Connectivity Parameters:

- **Average Hop Count** The number of hops that an average route has. In a forwarding network an analogy would be the average number of forwarding hops that a message is delivered, before an application triggers some action [3].
- **Neighborhood Information** The overall number of neighbors and their average persistence encountered by a node can be useful information regarding traffic patterns and network density.
- **Link Duration** The duration wireless links between two nodes (MAC-layer) persist can be measured. The output of link durations observed by a node ranges from the average link duration to a classification according to predetermined patterns of link duration distributions.

2.2. Automotive Environment

Apart from purely technological questions, there are some additional factors that influence the design of a VANET:

- The car as a product differs from other products in a couple of ways.
- A car-based messaging system can only work efficiently, if a multitude of car makers is supporting this technology. This requires good collaboration and agreements about central components of such a system.

[3] In a traffic-related messaging scenario the action taken by an application is displaying information to the driver of a car

- Companies that sell their products on the world's markets must reflect the diversity of cultures and regulations in their target markets.

In the following sections these criteria will be elaborated.

2.2.1. The Car as a Product

Cars are sometimes called "the most expensive consumer product", meaning that they can be seen as products that are necessary for daily life to a great extent and they are not necessarily luxury goods. But if compared to other consumer goods, especially information and communication products, there are significant differences. This greatly affects how technologies are used.

2.2.1.1. Extended Lifecycle. Cars are used for a much longer period than information and communication products. The expected lifetime is around 7-8 years and they can be in use for a much longer time. Due to their complexity and different requirements regarding safety and reliability, the development usually takes longer as well. A typical period from the beginning of a new product design to start of production is around 60 months (5 years). This has several implications for product developers:

- Maintenance has to be guaranteed for the whole lifetime and in all markets.
- Spare parts have to be available for a very long time, up to 30 years.
- It takes a long time before new technologies can be integrated into a new product.

2.2.1.2. High Expectations on Reliability. Cars are relatively expensive and their operation is safety-critical. This is why expectations on quality and availability are a lot higher than on other products. Electronic devices and systems have to pass a variety of criteria before they are integrated into a new car model, such as electromagnetic compatibility, power consumption, influence on other systems, etc.

2.2.2. Inter-OEM Systems

A communication system such as the one which is envisioned by certain organizations in Europe and the USA has to be standardized in order to ensure interoperability between devices of different manufacturers. There will most likely be multiple standards and standards bodies be involved, such as IEEE or ETSI for communication, SAE for automotive application message formats, etc. This requires a considerable amount of coordination among involved entities. Another aspect is that central components must not be within organizational control of a single (car-) manufacturer. That means that multiple companies must agree on the modes of operation, standards have to be defined and a business model has to be created for the operation of central components. The issues with centralized infrastructures will be discussed in-depth in chapter 2.5.

2.2.3. Global Business

A company that sells products worldwide has to consider many different social values and legislations. When designing a system that can be operated in all countries without drastic modifications, it has to be thought about customer needs as well as legal requirements.

2.2.3.1. Legal Aspects. Many countries have import regulations on military material that also include cryptographic devices and/or algorithms, an important regulation in this respect is the Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies, [**Sta96**].

If the messaging system uses any cryptographic primitives, it has to be checked whether they can be used worldwide without major import restrictions. Other regulations affect the obligation to store connection data, such as the German "Ordination concerning the Technical and Organizational Implementation of Measures for the Interception of Telecommunications - TKÜV" [**fW05**] or the limitation to store personal data, such as the directive 95/46/EC of the European Union on the protection of personal data [**Uni81**].

Liability is another legal aspect of major importance. Every part of a car has to fulfill certain requirements with respect to traffic safety. How the influence of a communication system on traffic safety is seen by the law differs from country to country.

2.2.3.2. Global Products. If products are marketed worldwide it is desirable that units sold in different countries have the same physical configuration. Local adjustments must be cost effective. Regarding IVC, there has to be an effective process to update systems with locally different parameterizations.

2.2.3.3. Central Infrastructure. Central Infrastructures are bad in a global environment. There are other significant disadvantages, that will have an effect on system design as shown in later chapters:

Costly Operation: Operating a central entity requires resources that someone has to pay for. If the system's security for instance depends on the security of a central authority this can reach huge expenses, as for instance a trust center.

Lack of Control: Car manufacturers naturally would not operate such a facility, but outsourcing this task means lack of direct control, while still being held responsible of what happens under the name of a brand.

Single Point of Failure: Central entities are always susceptible to system failures or even dedicated attacks. Denial of service is very hard to prevent for centralized architectures.

Interworking: It is unlikely that a single, global entity will be able to handle all tasks. Interworking between different centralized entities is inevitable for a global system. But how different operators, subject to different legal systems and working for different contractors can efficiently work together remains an open question.

Scalability: The larger the system grows, the more difficult it will be to scale centralized tasks. For example the maintenance of revocation lists in large systems is a difficult task.

Import Regulations & Legal System: Centralized services that have been specified for one region may not be operated legally in other countries due to import regulations and the legal system.

Connectivity: Connectivity to central services cannot be guaranteed in ad-hoc networks.

2.2.3.4. Cultural Differences. The public opinion on privacy and security of communication systems depends on cultural background, legal systems and political situation. Since cars are products that are bought by citizens, the culture and social atmosphere of a specific region as well as the reputation of the manufacturer has a large impact on their purchase decisions. System design therefore has to accommodate the different requirements derived from market needs. While privacy may be a strong issue in some countries, other societies may find it more useful if cars can be tracked for various reasons.

2.2.3.5. Business Model. A company will only decide to introduce a new technology if it expects a benefit that exceeds the efforts. In other words, there has to be a business model. The implication on system design is that apart from traffic related messaging, there will most likely be other applications that support the deployment of the system. Some examples are:

- production: wireless access to upload firmware and other basic software without physical connections
- maintenance: remote access to car-computers to read error memories and/or upload updates

In chapter 3 attacks on the system users' privacy will be discussed. But how successful attacks affect the life of persons depends on their own perception, the culture they live in and the threats they face. On one hand, civil rights are handled quite differently and privacy is something that people cannot rely on in many countries, because government requires systems to provide access to central databases. On the other hand, especially in western countries, people have a strong desire to maintain their privacy[4]. Regarding this work, it has to be determined how both goals can be met.

A problem arising with a central database is that once it has been created, there will always be voices in favor of exploiting it for reasons not in the best interest of users.

2.2.4. Deployment

If some kind of technology is installed in a car, there has to be an appropriate business model as pointed out in the previous chapter. Most likely, there will be more than one party that profiting from a system according to the architecture presented in chapter 4. Specific market and deployment aspects have been studied in [**MMP+05**].

Proposals for either reducing end-customer prices through cross-subvention or increasing customer interest:

- **Production** The software for a car's ECUs [5] has to be flashed during production. Wireless transmission of this data would enhance the production process considerably.
- **Maintenance** Wireless access to cars that enter a dealer's premises would accelerate diagnosis and offer a the dealers a much faster response to customer needs.
- **Central Traffic Management Systems** Existing traffic management systems that gather data from special stations or mobile units could greatly capitalize on traffic-related messaging based on inter-vehicle communications.
- **Public Hotspots** Public hotspots could offer information services, such as POIs [6], schedules, local content, podcasts, etc. that has been provided by public entities or local merchants.
- **Gas station** Gas stations could offer special treats to increase customer loyalty similar to existing payback concepts (Payback, Speed pass, etc.).
- **Home** Cars can be connected to local wireless networks to download private music files, addresses, navigation routes and other personal or car-related information.

2.3. Relevant VANET Properties

When comparing VANETs with MANETs, there are some characteristics that influence a system's design. Some properties of vehicles serving as network nodes are beneficial, others are not:

Potential advantages:

[4] The incidence documented on [**fPP03**] shows that even in countries with strong privacy laws such as Germany it is a difficult task to maintain privacy.

[5] In automotive electronics, an electronic control unit (ECU) is an embedded system that controls one or more of the electrical subsystems in a vehicle

[6] Point Of Interest: landmarks, food, gas stations, shops, cultural locations, etc.

- **Time and Positioning** All vehicles that are part of the system are assumed to provide state-of-the-art navigation systems, which caters global time and position.
- **No Energy Constraints** Cars have large on-board power generators. Everything in the order of small communication and computer systems will not affect the on-board power system significantly.
- **Tamper Resistant Device** Every future service that requires some security function will need secure storage of cryptographic keys and a secure computation unit to execute cryptographic algorithms. Additionally, future cars will require ECU's to register at a central security component, which has to withstand tampering for obvious reasons. Hence, cars will have a built-in tamper resistant device that may be used by various on-board systems or applications.
- **Computing Platform** Future cars will carry computers and memory that exceeds that of currently available mobile devices.
- **Physical Access** Cars are usually owned by one person and the number of people who have physical access is limited.
- **Periodic Maintenance** Modern cars require periodic maintenance for proper operation.
- **Standards** Car manufacturers install the system and therefore control the standards being used for IVC. These standards are inter-organizational and on a global level.
- **Long-term / Known Users** The main user of a car or at least the owner are usually known to the car's manufacturer or authorities. Additionally, in most cases cars are owned and used by the same person or group of persons for a long period.

Potential disadvantages:

- **Privacy** Privacy is an important requirement in order get users' acceptance. There are a couple of attacks on privacy that one can think of, such as the generation of movement patterns for specific nodes, establishing communication profiles or even automated detection of infringements. These issues will be discussed in section 3.4.
- **Decentralization** The strong demand for decentralization in a VANET influences the system in many ways. It affects trust establishment in general, since it is difficult to trust unknown node. Without a central infrastructure, the distribution and revocation of cryptographic key material is a serious problem. Another issue is that some nodes may decide not to participate in the network if it is not useful for them, thus crippling the network and degrading overall performance.
- **High Mobility** The high average speed that cars may have and especially the passing of cars traveling in the opposite direction on highways places seriously affects routing protocols and the network's connectivity in general. Additionally, the vehicles are not uniformly distributed, resulting in areas with redundant connections, while other areas may be fragmented into small isolated network islands.
- **No User Interaction** Persons driving a car have to concentrate on this task. A system that supports human drivers must not require user interaction apart from displaying context-related information and simple inputs. It can therefore not be assumed that any person supports the operation of the system by giving serious input.
- **Large Number of Nodes** A VANET consists of a large number of nodes. [**BE04**] assumes that VANETs become the largest and most important ad-hoc networks deployed in the near

future. Potentially, every future car on the world could be equipped with a communication device. Although in near-term this number will be much smaller, the system's design still has to reflect the issues related to a very large number of nodes.

2.4. VANET Applications

In various projects dedicated to inter-vehicle communication (e.g. [**tCCC04**], [**oWP04**], [**Pro05b**], [**Pro03b**], [**oT**]) a number of applications for inter-vehicle communication have been proposed. This section will introduce the most relevant applications and categorize them into the three categories safety, comfort / efficiency and infotainment for a better understanding. Most importantly, the applications will be grouped into three application types. This categorization is fundamental for all following chapters, because it divides applications according to their usage profile which greatly affects system design and security concepts.

2.4.1. Table of Applications

Table 2.1 shows some applications that have been discussed in various expert groups. This gives an idea about the diversity of the challenges that have to be tackled when designing an IVC system.

2.4.2. Application Types

For the following chapters, the applications will be categorized according to usage profiles and communication patterns. This has great influence on the security architecture. They are categorized into three types:

Type 1: Vehicle to Vehicle Traffic-related (V2V):

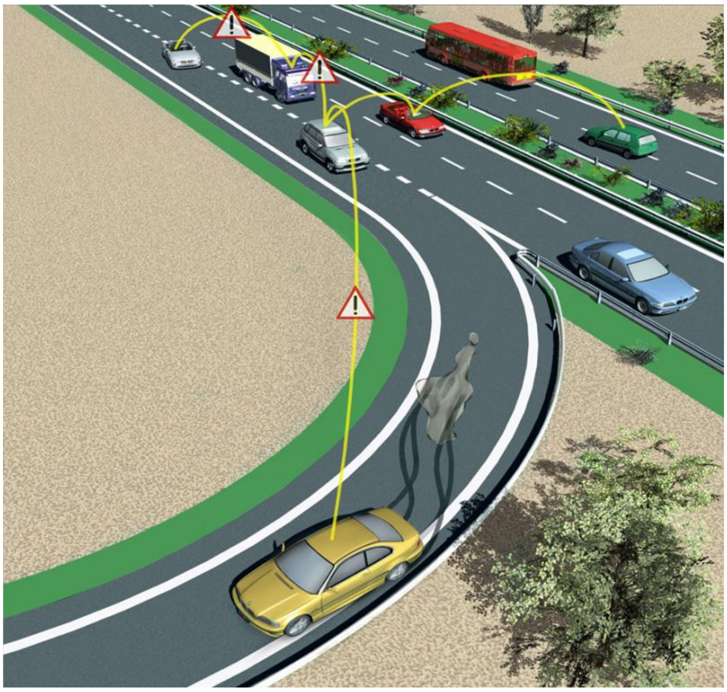


Figure 2.3. Type-1: V2V Traffic-related Applications

The basic idea of vehicle to vehicle traffic-related messaging (V2V) is shown in Figure 2.3. Cars equipped with a V2V application try to automatically detect hazards and other traffic-related events while driving, using their on-board sensors. When a hazard is detected, a new

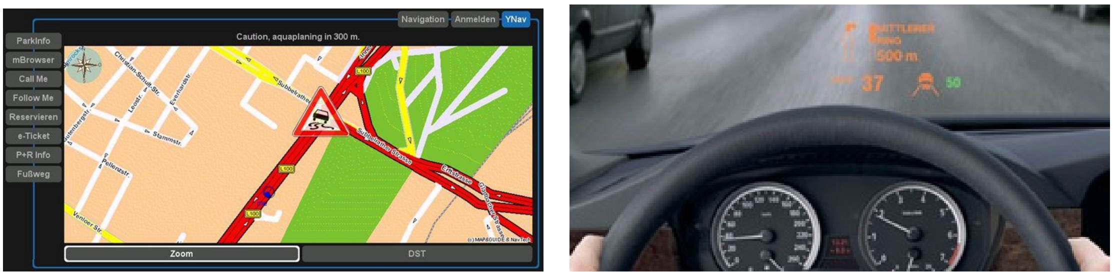
Safety Applications			Type
Local Danger Warning	Low Visibility Warning	Fog, smoke, rain, snow, etc.	1
	Approaching Emergency Vehicle Warning		1
	Curve Speed Warning		2
	Low Bridge Warning		2
	Low Parking Structure Warning		2
	Wrong Way Driver Warning		2
	Work Zone Warning		1
	Obstacle Warning	A generic type for all kinds of obstacles	1
	Emergency Brake Warning	To avoid rear-end collision	1
	Pre-crash Sensing		1
	Post Crash Warning	To notify others	1
	Blind Spot Warning		1
Intersection Assistance	Traffic Signal Violation Warning		2
	Stop Sign Violation Warning		2
	Left Turn Assistant	Checking for oncoming traffic	2
	Stop Sign Movement Assistance		2
	Intersection Collision Warning		1
	Blind Merge Warning	Checking for others at merging lanes	1
	Pedestrian Crossing Information		2
	Emergency Vehicle Signal Preemption	Em. vehicles controlling traffic lights	2
	Braking Distance Estimation	Useful to check one's approaching speed	2
Maintenance	Safety Service Point		2
	Remote Diagnostics	Sending maintenance data to the dealer	2
	Safety Recall Notice		2
	Just-In-Time Repair Notification		2

Comfort and Efficiency Applications			Type
Traffic Performance Assistance	Automated Profiling	Recording profiles of driving behavior	2
	Lane Assistants	Choice of lane, keeping lane and lane change	1
	Platooning	Grouping of different vehicles into a single coordinated platoon	3
	Dynamic Right of Way	Clustering of cars and adaptive right of way to improve traffic flow	1
	Distributed Floating Car Data	Traffic flow and density information	1,2
	Intelligent Traffic Flow Control	Cooperative flow optimization	1
	Intelligent On-Ramp Metering	Wireless tolling, etc.	2
Comfort	Cooperative Cruise Control	Cooperating vehicles add comfort to cruise control	3
	Drive-through payment	Product ordering and payment via IVC	2
	In-Vehicle Signage	Traffic signs are displayed inside a car	2
	Cooperative Glare Reduction	Pixel light, headlamp aiming, . . .	1
	Free-Flow Tolling	Pass toll stations without stopping	2
	Adaptive Chassis Control	Adjust chassis settings according to road condition known from other cars	1,3
	Cooperative GPS Correction		1
	Adaptive Drivetrain Management	Adjust drive train settings according to road condition known from other cars	1,3

Infotainment Applications			Type
Communicating Convoys	Dynamic Grouping	Cars / passengers with specific properties will be added to a group on route	3
	Information Services	Notification on predefined events within travel group (convoy)	3
	Teleconferencing	Audio, video, file sharing, messaging, . . .	3
Local Information Points	Point of Interest Notification	Notification on predefined POIs	2
	Map Downloads and Updates	Automated digital map data updating	2
	File Subscription	Automated podcast, weather, . . . download	3
Interaction between mobile devices	Instant Messaging	IVC transports messaging traffic	3
	Multihop cell-phone forwarding	IVC forwards cell-phone connections, also critical SOS calls out of tunnels	3

Table 2.1. IVC Applications

warning message is generated and sent subsequently. Cars[7] receiving such a message evaluate its content and decide on the message's forwarding. Additionally, the receiving system has to make a presentation decision whether the information is appropriate and relevant enough to alert the driver[8]. The information can be presented either on a navigation display, shown in Figure 2.4(a), on a Heads-Up-Display (HUD), shown in Figure 2.4(b), or by specific (warning) sounds.


Caution, aquaplaning in 300 m.

(a) Navigation Display

(b) HUD

Figure 2.4. Human Machine Interfaces

After a driver has been warned, he can change his route or driving style accordingly, thus avoiding the hazard or mitigating its effects. This increases road safety and also leads to an improvement of traffic flow, since congestions may be circumvented. It has to be noted here that the network may also include fixed units. It can be distinguished between hotspots and RSUs. Hotspots provide gateway functionalities and may optionally support IVC specific protocols and technologies. RSUs have similar capabilities as OBUs with respect to communication and information processing. But they may have different sensors or none at all and they are not moving.

Type 2: Vehicle to Infrastructure (V2I):

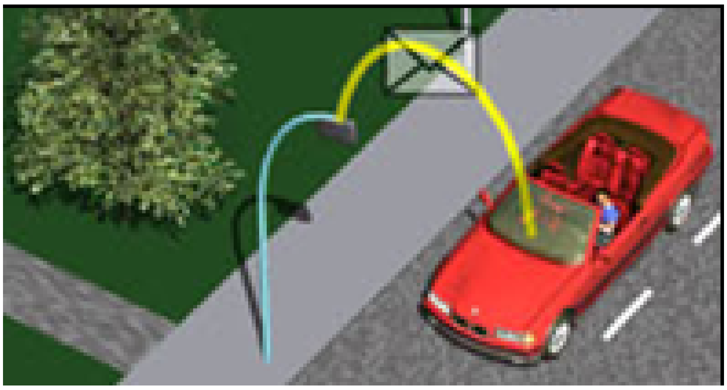


Figure 2.5. Type-2: V2I Applications

Vehicle to infrastructure (V2I) communication is all communication between cars and fixed infrastructure that is connected to other networks [9].

Type 3: Connection Oriented Communication (COC):

[7] The terms "car" and "vehicle" will be used similarly in this work.

[8] Since messages may be delivered over large distances, where information is not (yet) relevant to the driver, the information may be stored and then displayed at some later time when it becomes relevant to the driver

[9] If the fixed infrastructure is not connected to a network other than the IVC network, the infrastructure is called Road-Side Unit (RSU)

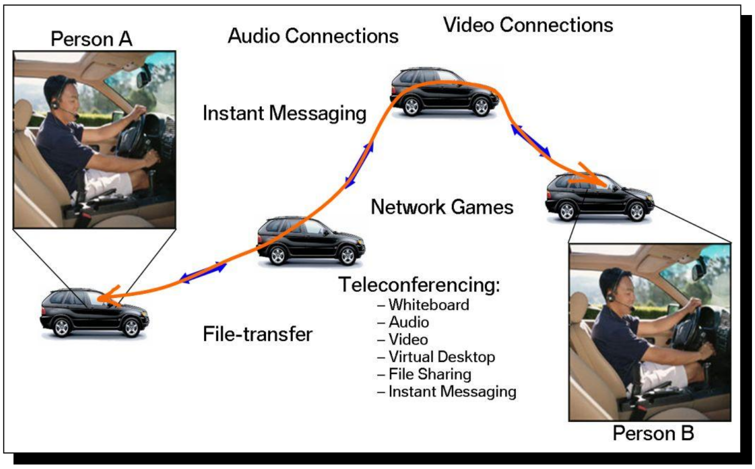


Figure 2.6. Type-3: COC Applications

Connection oriented communication (COC) refers to all kinds of communication, where a car is either source and/or destination of the communication, the communication is multi-hop, a route is established and at least one of the routing nodes is a car. A schematic view is shown in Figure 2.6. The establishment and maintenance of routes must function while cars are moving. This type of application can be seen as an extension to single-hop communications by adding routing mechanisms, which are the main focus when looking at this type of application.

For a better understanding Table 2.2 gives an idea of the application types' typical use:

	V2V	V2I	COC
Communication Type	Forwarded Multi-Hop	Single-Hop	Routed Multi-Hop
Beneficiary	Receiver	Sender & Receiver	Sender
Required Penetration Rate	High	Low	High

Table 2.2. Applications Type Comparison

Obviously, there are many differences between applications of one type. A detailed classification of Type-1, which is the most diverse and most important type will be presented in section 5.2.

2.5. Requirements

Following the problem outline in section 1.3 and the arguments in this chapter, overall requirements for a IVC system are defined in this section. Regarding the focus of this work on security and dependability the requirements will be revisited in chapter 3 and elaborated on security objectives.

Requirement 2.1 (Available & Robust)**.** *Availability and robustness are important properties, especially so if the services are related to traffic and travel-safety. Availability means the system is usable in as many situations as possible and robustness means that the system stays operational in presence of detrimental events.*

Requirement 2.2 (Secure)**.** *The preceding requirement does only consider "natural" events. Requiring security means that the system must be protected against the persons that deliberately attack parts of the system.*

Requirement 2.3 (Decentralized)**.** *In section 2.2.3.3 the downsides of centralization have been pointed out, leading to the requirement of decentralization. However, if complete decentralization cannot be achieved, it is desirable to limit centralization to the situations, where it can be handled easier such as during production.*

Requirement 2.4 (Short Connection Time)**.** *The high degree of mobility and relative speeds lead to the requirement of a system being able to cope with short connection times, sometimes only being able to transfer a single network packet.*

Requirement 2.5 (No User Interaction)**.** *The system must reflect the peculiarities of a traffic environment by requiring a minimal degree of user interaction*[10] *during normal usage. This requirement includes self-organization of the network, situation adaptive filtering of contents and automated trust establishment among nodes. No user interaction is strictly required for type-1 applications. Type-2 and type-3 applications may require some user input for the initiation of new connections or service usage, but this interaction has to be limited to the degree that is necessary and appropriate in the user's current situation.*

Requirement 2.6 (Cost Effective)**.** *The automotive industry has very tight cost calculations and customers probably would not want to pay more money than necessary leading to a preference for cost effective solutions.*

Requirement 2.7 (Global Feasibility)**.** *The market structure and complexity of nation specific homologation demands car makers to design products that can be deployed globally without major modifications. An IVC system must therefore reflect cryptographic export regulations, employ global address space, etc.*

Requirement 2.8 (Application Priorities)**.** *Traffic-related applications and messages are in general more important than multimedia messages. And some traffic events are more urgent than others. Prioritizing the distribution of messages is required to ensure maximum availability of critical application data.*

[10] User interaction means that the user is required to respond to the system by making inputs

CHAPTER 3

System Model and Threat Analysis

In this chapter, a lightweight system model is introduced, forming the boundary conditions for security analysis and based on given facts, reasonable assumptions and best-practice approaches. After introducing security engineering basics, a threat analysis is presented in the third section. The effective result of this chapter is the identification of the relevant subset of security services called security objectives in this thesis.

3.1. System Model

This section introduces a basic system model that is required for the following threat analysis in section 3.3. The first part lists given constraints and assumptions that have been made, describes the communication system being used, gives an overview on available sensors and data sources, discusses the trust model in an automotive environment and introduces traffic scenarios.

3.1.1. Constraints and Assumptions

Before thinking about a system model, constraints that are given beforehand will be listed. In addition to these settings, there are additional assumptions that have been made according to logical estimations, best practice approaches and simply common sense.

Constraints affecting system design are:

Constraint 1 (Processing Power)**.** *The processing power will continue to grow over the next years with the same rate as Gordon Moore has stated in* [**Moo65**]*. Cars will therefore have considerable computational power, but costs remain a critical factor. The equivalent processing power is assumed to reach that of a contemporary mobile device in the range of a 500Mhz StrongARM Processor.*

Constraint 2 (Memory)**.** *The arguments that support the further growth of processing speed are even stronger for the growth of memory. Over the past years, the price for memory and the size of storing devices have decreased dramatically. Persistent memory is assumed to be in the range of multiple GBytes.*

Constraint 3 (Available Bandwidth)**.** *Communication bandwidth for wireless communications is physically limited because the wireless channel is a shared medium and the spectrum is divided into different frequency bands that are regulated by governmental bodies. Additionally, due to the broadcast nature of the wireless transmission, collisions may occur which impair the communication throughput. Source coding technologies that will improve data rates substantially are not to be expected, bandwidth will remain scarce in future. As a rough estimation, dedicated communication bandwidth available for inter-vehicle communications will be two channels with 10 MHz bandwidth each in Europe, according to* [**Mok05**]*.*

Constraint 4 (Communication Range)**.** *Common radio technologies which are the basis for inter-vehicle communications have limited transmission power, also due to bandwidth efficiency reasons. Communication ranges up to 1000m have been measured using dedicated antennae, but experiences of various research groups suggest a more conservative approach and an estimated maximum communication range of 500m is assumed.*

Constraint 5 (Costs)**.** *Costs are a dominant factor in the automotive industry. Estimated on-board unit costs are estimated to be less than 100 Euro, excluding antenna and wiring.*

In summary, the limiting factors for this work are communication bandwidth and cost.

In order to use available technologies and to exploit specific properties of the automotive environment **assumptions** have been made to facilitate possible approaches toward a secure architecture:

Assumption 1 (Secure Positioning and Global Time)**.** *All vehicles are equipped with satellite positioning systems*[1]*. This offers information about position and global time, which are protected by a combination of the measures discussed in section 4.4.3 to accommodate the required level of security.*

Assumption 2 (Tamper Resistant Module)**.** *All vehicles will have an on-board tamper resistant module that is protected against removal and replacement or will delete all kept secrets when removed*[2]*.*

Assumption 3 (Trust Authorities)**.** *For every manufactured vehicle, there exists a trust authority that is able to attest its specific features or operational behavior. The trust authority may either be the manufacturer or the vehicle's operator.*

Assumption 4 (Physical Transport)**.** *Since cars are moving, they are able to transport information physically. This property may be exploited to improve communication performance and extend the reach. In the literature this is sometimes called store-and-forward (e.g.* [**KSA02**]*).*

Assumption 5 (Limited Message Set)**.** *In type-1 applications, the set of local danger warning messages assigned to specific situations is limited. This allows a comparison of message-contents from different sources. Thus the content originating from different nodes can be correlated to calculate a level of confidence for this information. This is fundamental to the concepts presented in chapter 5.*

Assumption 6 (Power Consumption)**.** *Power consumption of a inter-vehicle communication system is low compared to overall electrical power consumption in a car. The effect on the power system can therefore be neglected.*

3.1.2. Communication System

The communication system will be based on IEEE 802.11a or IEEE 802.11p standards, with modifications to medium access schemes being discussed in section 4.4.4. [**Fes05**] gives a short overview on IEEE 802.11p.

The antennae being used in prototypes usually have one main lobe to the front and one to the back of the car, which increases the range in positive and negative moving direction. Current research

[1] However, not all vehicles are assumed to have digital maps or route allocation functionality

[2] Note that future cars will most likely provide such a module for other reasons, as noted in section 2.3.

papers, such as [**SS03, CUBV05**] have considered beam-forming or at least separate usage of front/back antennae, but this increases complexity for power control and channel access schemes significantly and will therefore not be considered in this thesis. For theoretical analysis, the antennae are considered to be omnidirectional.

Networking will strongly depend on application needs. Routing will be required for Type-3 applications, while Type-1 require forwarding that uses application specific information. Type-2 applications will not require specific networking functionality, since hotspots that act as gateways will use standard protocols. However, if multi-hop routing is used to reach nodes that do not have direct connection to the hotspot, ad-hoc routing schemes will be used and addresses have to be mapped between the internet address-space and the IVC address space.

3.1.3. Data Sources

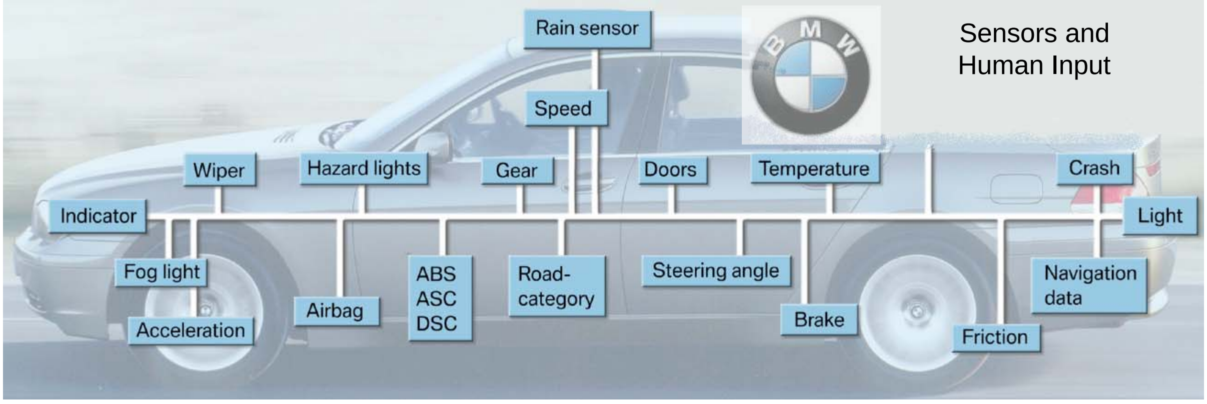


Figure 3.1. Sensors and Human Input

There are a number of sources where an on-board system may get original information[3] from: internal sources, such as environment sensors, movement sensors or human input, or external information, such as positioning data or measurements from fixed road-side sensors. Figure 3.1 and Table 3.1 provide an overview.

Sensors	**Human Input**
RADAR	Brake-, Clutch-, Gas-Pedal
Rain / Visibility Sensor	Windscreen Wipers
Thermometer	Indicators
Gas Sensor	Gear
Forward-looking Infrared (FIR)	Doors and Windows
Crash Sensor	Sunroof
Steering Angle & Wheel Sensor	Radio Volume
Acceleration Sensor	
Brightness Sensor	
Ultrasound Sensor	
Engine Control Sensors	

Table 3.1. Sensors and Human Input

[3] Original information denotes that kind of information that is not received from other vehicles in the network.

3.1.4. Trust Establishment

The given automotive environment restricts the availability of security infrastructures that are generally used for trust establishment due to the reasons given in paragraph 2.2.3.3. Access to such infrastructures may only be assumed during production of a car, when it enters an official garage for maintenance and when it is registered at local authorities. Assuming other forms of connections to infrastructures, such as sporadic access to gateways or hotspots, connections via WAN networks or through portable memory devices is too restrictive for a general approach. This thesis will therefore focus on approaches that avoid access to security infrastructures as much as possible.

Section 4.4.1 will elaborate on the topic trust establishment.

3.1.5. Traffic and Communication Environment Categories

As pointed out before, the settings an IVC system has to operate in are very diverse. The traffic situation can be analyzed according to the speed distribution of nodes, the vehicle density, the relative speed distribution, social factors such as drivers' behavior, etc. These parameters are interacting with communication related parameters, such as the number of neighbors, the communication range, the type and quality of the wireless channel, etc. The combination of these parameters' actual values will be denoted as traffic environment and communication environment respectively.

These parameters may be evaluated at a local level with the perspective of a single node or on a macro-level where the settings of a part of the system are analyzed[4]. Ideally, the local analysis will lead to the same judgments as a macro-level analysis.

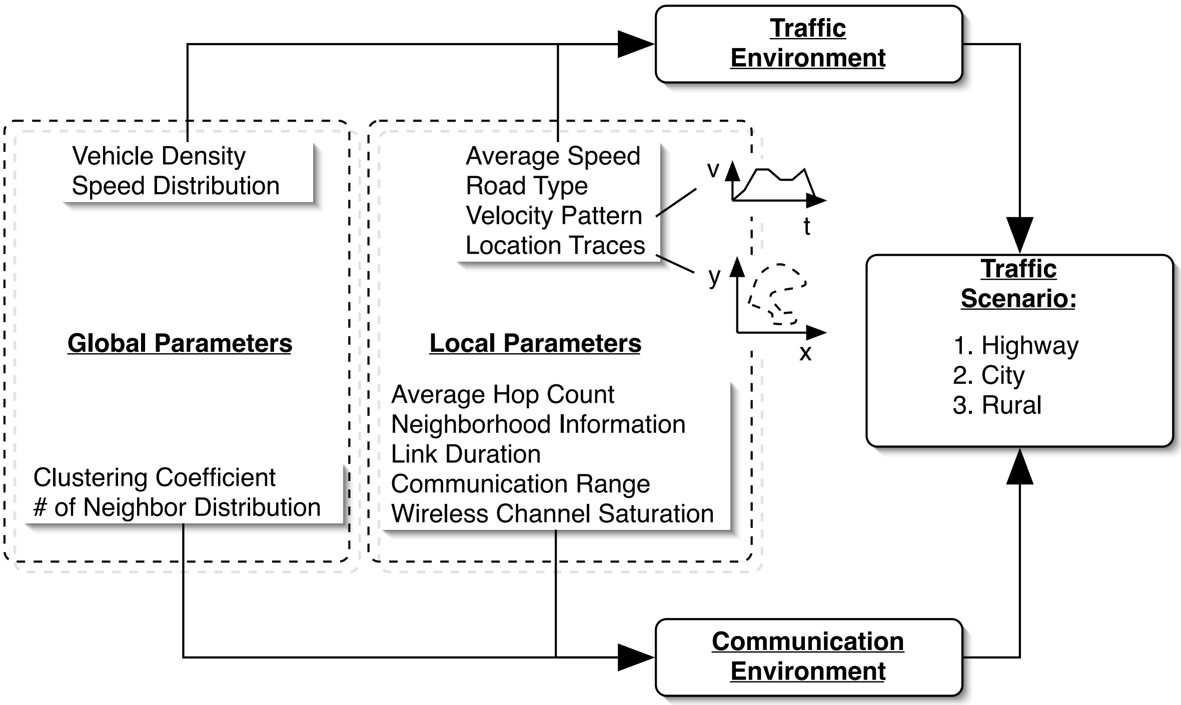


Figure 3.2. Determining the Traffic Scenario

In order to be able to study the effects of different communication system designs and the performance of different security architectures three traffic scenarios are defined:

[4] A macro-level view on global parameters can only be reached in theory or simulation where complete data is available

Definition 8 (Highway). *The highway scenario stands for a set of cars moving along a road or in opposite direction, but with only few intersecting traffic. The relative speed divergence is usually very high since cars moving in the same direction approximately have the same speed, while the head-on traffic has a relative speed of about double the average speed. The average speed is higher than in all other scenarios. The communication channel is in most cases unobstructed and the wireless range is high.*

Definition 9 (City). *In the city scenario vehicles are moving in various directions, average speed is low, since maximum allowed speed is low, traffic congestions occur sometimes and cars have to make frequent stops at intersections, traffic lights, etc. The number of neighbors is usually very high due to high two dimensional traffic density*[5]*. The wireless channel however is often constructed by buildings, which limits the wireless range and decreases the channel quality. Traffic jams are a variation of this scenario, where speed is very low (zero).*

Definition 10 (Rural). *The rural scenario is something like a mixture between the highway and city scenarios when it comes to average speed, relative speed divergence and movement patterns. The main challenge in this scenario is the low density of cars, making physical packet delivery an essential functionality of the system (see assumption 4).*

Parameter	**Highway**	**City**	**Rural**	**Derived from Parameter(s)**
Average Speed	high	low	medium	Average Speed
Relative Speed Divergence	high	low	medium	Link Duration
Movement Pattern	linear	2-dimensional	2-dimensional	Location Traces
# of Neighbors	medium	high	low	Wireless Channel Saturation, Neighborhood Information
Communication Range	high	low	high	Communication Range
Velocity Pattern	smooth, no stops	choppy, frequent stops	smooth, rare stops	Velocity Pattern

Table 3.2. Overview Traffic Scenarios - Factors derived from Local Parameters

This categorization is shown in Figure 3.2. The local and global parameters are used to determine the traffic scenario. An overview of a combination of locally measurable data such as speed, connection times and the number of neighbors is shown in Table 3.2. If digital maps containing long-term, preknown registered information such as road-types and urban boundaries are available[6], this can be used to improve this process. The concepts presented in this work can be applied universally. Relying on locally measurable data suffices for the analysis in this chapter. Section 5.4.3 will point out a more specific model using RSUs to improve the quality of confidence decisions.

3.2. Information Security

This section gives a basic understanding of information security and how it differs from other technical disciplines. The design principles with respect to information security of this thesis will be introduced and it will be explained why they are important. Relevant technologies, algorithms and processes will be presented that are required to understand the security issues identified in this chapter, the weighting of threats and design approaches in following chapters.

[5] As opposed to the highway scenario where traffic density might also be high, but only in a one dimensional space along the highway.

[6] Note that due to cost reasons vehicles may only be equipped with satellite navigation instead of a full navigation system with digital maps, see Constraint 6

3.2.1. Information Security Basics and Design Principles

3.2.1.1. Why the Security Business is Different. Dealing with security is different from other engineering tasks in some ways. The following paragraphs will highlight some of the important points and thus provide a common understanding for the design choices of this thesis:

First, attackers cannot be assumed to be in conformance with overall statistics, because they will always try to exploit an a priori setup in such a way that their attack becomes most effective. One example is if there exists a system malfunction that only occurs through a combination of unlikely incidents. During normal system operation this may never lead to a serious problem due to the low probability of occurrence. However, an attacker will focus on fabricating exactly those conditions that lead to such an unstable state where he can then exploit abnormal system behavior. This makes it much harder to design secure systems, because an engineer has to evaluate the threats and their implications instead of relying on fault tolerance statistics[7].

Second, it is hard to measure a system's security by explicit system parameters such as the length of cryptographic keys or the number of rounds an encryption algorithm performs. Matt Blaze, a well-known security specialist, once said "one of the most dangerous aspects of cryptology, is that you can almost measure it." [**Sch96**]. Security is about the weakest spot in a system, not about the strongest one. Consequently, in security engineering is more important to anticipate critical weaknesses than reinforcing strong cryptographic algorithms alone.

Third, another difference when compared to fault tolerance approaches is that well-performed attacks are hard to detect. They are in most cases invisible to their victims when they happen and some are never revealed. This makes assessment of the damage very difficult and the success of security measures is hard to quantify. But even if an attack is detected, it is very hard to quantify all its implications. Apart from directly affected employees and customers, the image of a company may be affected, potential customers may not choose the product because of negative publicity, competitors may have gained information that is harmful even if it is not used directly against the victim, etc.

Fourth, in many cases security is not a feature that can be easily added. This means that increasing the level of security is always a trade off. In some cases security affects system performance, in other cases it affects usability. Therefore, a system designer must carefully balance the degree of security against performance and usability. For some security goals, such as anonymity, it is even more difficult, they cannot be built-in afterwards at all. This means that security objectives have to be identified at early stage. On the other hand, security solutions often cannot be fully implemented until system design has been specified, because of system specific vulnerabilities. This makes security engineering a cross-sectional topic that accompanies the whole system development process, most of the time without adding functionality, but putting new constraints on other developers.

Fifth, security has many social and economic aspects, such as pointed out in [**And01a**]. It is important to understand the environment a system operates in and to incorporate this into the threat analysis. One example is that the organization that is responsible for maintaining a system's security is often not identical to the one who is suffering from a successful attack.

3.2.1.2. Design Principles. Following paragraphs will introduce the design principles followed in this work in response to lessons learned in security engineering and the arguments provided above:

[7] It has to be noted however, that in the design of fault tolerance systems, there exist also design pattern for byzantine fault tolerance, meaning the system must not fail even if the worst possible combination of factors occurs.

Requirement 3.1 (Open Security Design)**.** *Auguste Kerckhoffs stated in 1883 that the security of a cryptosystem must not rely on how encryption/decryption is done, but on a secret that is shared among originator and recipient of a message in* [**Ker83**]*. This rule is now known as Kerckhoffs' law.*
Following this rule, the design in this thesis assumes that all information about the system is known to attackers except secret keying material that can be revoked and/or updated.

Requirement 3.2 (Security as Process)**.** *An issue that is easy to understand logically, but has been disregarded many times, is that nobody can foresee all attacks on complex systems in advance. Secure systems have to be flexible enough to respond to new attacks. Security expert Bruce Schneier* [**Sch00**] *put it like that: "Security is a process, not a product."*
Therefore, when developing the security architecture, the lifecycle of its components and the ability to modify the system in response to an attack are important.

Requirement 3.3 (Facilitate Practical Implementation)**.** *Many security problems are caused by bad implementation: secrets are not deleted after using them, random number generators produce predictable output, buffer overflows happen, etc. In fact, security can be completely compromised by deficient implementation.*
Therefore, critical parts will be pointed out and complex solutions will be avoided wherever possible.

Requirement 3.4 (Attack Detection complements Attack Prevention)**.** *The main goal of the security architecture is to prevent attacks. In those cases where attacks cannot be prevented it is desirable that attacks are detected.*
This work is directed to achieve at least attack detection whenever possible.

3.2.2. Security Engineering

Security engineering is the effort to achieve and maintain optimal security and survivability of a system throughout its life cycle. There are different levels of abstraction, Figure 3.3 gives an overview (based on a picture in [**And01b**]).

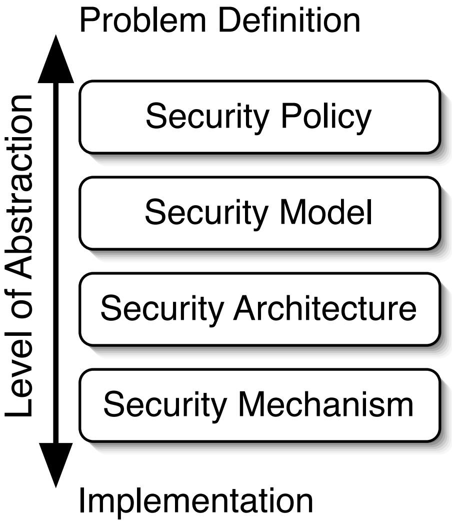


Figure 3.3. Security Engineering Abstraction Layers

3.2.2.1. Phases of Security Engineering. The process of security engineering has been described in [**Eck03**], [**And01b**] and [**Bis02**]. They all agree that before designing a security architecture, threats have to be assessed and the risks have to be analyzed. [**Gör05**] provides a methodology for mobile distributed systems. The analytical part of security engineering consists of following steps:

(1) Describe the general system model and point out specific properties that affect security. See 3.1.
(2) Describe the general threats that affect the system and generate attack trees that reflect the various attacks accordingly [8]. See 3.3.4.
(3) After the generation of specific attack trees and the identification of vulnerabilities, apply a cost function in order to develop a proper risk analysis. See 3.3.6.

3.2.2.2. Security Services. Security services are defined by [**Shi00**] as processing or communication services that give a specific kind of protection to system resources. ISO IS 7498-2 [**fS89**] defines following network security services:

Authentication: Ensures that a principal (user, process, host) is really what it claims to be.

Access Control: Ensures that only authorized principals can gain access to protected resources.

Data Confidentiality: Ensures that only authorized principals can understand the protected data.

Data Integrity: Ensures that no modification of data has been performed by unauthorized principals.

Non-Repudiation: Ensures that a principal cannot deny performing some action on the data (e.g. authoring, sending, receiving).

Other services or properties related to security have been mentioned in the literature, such as **Authorization**, ensuring that only authorized principals get access to relevant resources (see also **Access Control**). **Freshness**, ensuring that security-related data is not a copy from an earlier instance, thus preventing replay attacks. **Reliability**, the ability of a system to perform a required function under stated conditions for a specified period of time. **Availability**, the property of being accessible and usable upon demand by an authorized entity. This provides only a general view on security services and they have to be specified as part of the security engineering process. In section 3.4 those specific security services that have been identified in the threat analysis will be presented as security objectives.

3.2.2.3. Attack Trees. Attack trees as in [**Sch99**] provide a structured and standardized means to classify and refine attacks on a system and they have been used before successfully (cf. [**MEL01**]). The root of each tree represents a general attack on the system such as, e.g., Denial of Service. Attack trees specify attacks in terms of attack goals and their subordinate goals. The overall attack goal is then further refined in the tree structure using AND and OR logical connections.

Figure 3.4 depicts the graphical and textual representations of AND and OR connections. Usually, the textual representation is preferred over the graphical one, since the graphical representation becomes hard to read and quite space consuming for more complex attack scenarios.

As an example, consider the simple scenario that an attacker wants to steal a car. An attack tree could look like the one depicted in Figure 3.5. The attacker can shortcircuit the car to make it move or obtain a copy of the key. The aforementioned two attack goals can again be subdivided into subordinate

[8] Attack trees have been mentioned in the literature as threat trees as well.

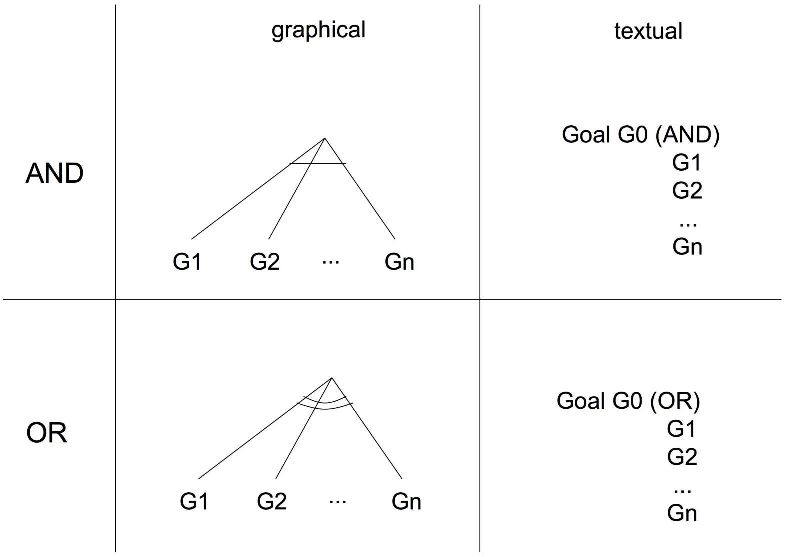


Figure 3.4. Graphical and Textual Representation of Attack Trees

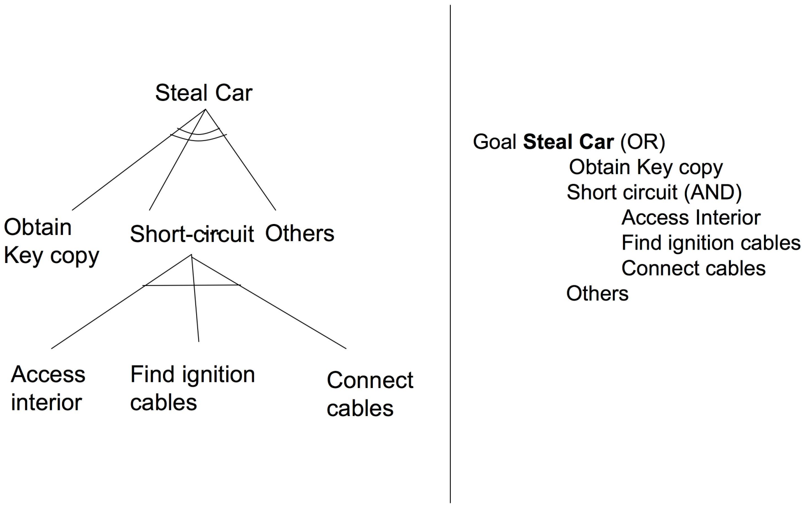


Figure 3.5. An example Attack Tree: Steal a Car

goals. For example, to short-circuit the car, the thief may need to break a car-window to access the car interior, and find the right ignition cables to shortcircuit them.

3.2.3. Security Mechanisms

This thesis will build on cryptographic functions, algorithms, protocols and concepts that are briefly introduced here to better understand following sections and chapters. A more detailed view on required mechanisms will be given in the according sections where required.

3.2.3.1. Basic Cryptographic Functions. This paragraph presents only a brief explanation of basic cryptographic functions and what they are used for. The mathematical background is substantial and can be found for instance in [**Sch96**] or [**MOV96**].

One-Way Hash Functions: One-way hash functions are a fundamental building block for many cryptographic protocols. A hash function takes variable-length input (called *pre-image*) and converts it to a fixed-length output - the *hash value.* A good (cryptographically that is) one-way hash function has two properties:

(1) It works in one direction: It is easy to compute a hash value from a pre-image, but it is hard to generate a pre-image that hashes to a particular value

(2) It is collision-free: It is hard to generate two pre-images with the same hash value

Message Authentication Codes (MAC): A message authentication code is the application of a one-way hash function to a given pre-image and a secret key. In order to be able to verify the hash value, the key has to be known.

Block Ciphers: A block cipher is an encryption function for fixed-size blocks. It requires a secret key for encryption and decryption, thus such an operation is *symmetric cryptography.*

Stream Ciphers: Stream ciphers in contrast to block ciphers are encryption functions for continuous streams of data. Many symmetric encryption algorithms have a mode[9] which produces a key-stream that can be XORed with an input data stream.

3.2.3.2. Public Key Cryptography. In 1976[10], Whitfield Diffie and Martin Hellman changed the whole paradigm of cryptography [**DH76**] when they introduced public-key cryptography. The main problem with conventional symmetric key ciphers (as mentioned above) is the exchange of the secret key which must obviously remain secret during the exchange. Public key mechanisms use a pair of keys - a public key and a private key. It is computationally hard to deduce the private key from the public key. Anyone with the public key can encrypt a message but not decrypt it. Only the entity holding the corresponding private key is able to do so. It should be noted here that public key encryption (also called asymmetric encryption) is computationally much more expensive than symmetric encryption. Therefore, large amounts of data are usually encrypted using symmetric encryption, where the (symmetric) key is encrypted using public key schemes for key exchange, thus significantly reducing the effort. Interestingly, if the process is reversed and data is "encrypted" using the secret key (normally held by a single entity), everyone having the corresponding public key can verify who originally "encrypted" the message. This is the concept of *digital signatures*[11].

3.2.4. Privacy Concepts

Privacy is difficult to achieve in a world, where information systems dominate all aspects of daily life. Many people do not see that combining all information that is available about them electronically would result in a complete summary of their life. The difference between the "real world" and the "virtual world" is that information can be processed automatically and in high volumes in the digital world, something which is not possible to the same degree in the physical world. In addition to that information

[9] Examples are *Output feedback* mode and *Counter* mode

[10] NSA (National Security Agency - USA's intelligence agency focused on cryptography) claimed to have had knowledge of public key cryptography concepts since 1966

[11] For completeness it has to be noted that the message is first hashed using a one-way hash function before signing it to ensure integrity afterwards.

generally persists for a much longer time in the digital environment: articles in newsgroups can still be looked up in a common search engine after many years, while a letter published in a newspaper will not be stored for a long time.

When designing a system it is important to realize that critical information will be hard, if not impossible, to remove it after the system has been built. Although there are some concepts that try to "add" privacy to an existing system, such as encryption for confidentiality or mixes for communication privacy. Identity management systems can be used to anonymize certain aspects of identification. But these approaches only solve specific issues and all have major drawbacks. In the end the degree of privacy is something that has to be defined before system design and then appropriately implemented in order to achieve required levels of privacy.

Additionally, attacks on privacy are usually hard to detect since they are mostly passive attacks. This makes privacy assessment a task that is close to impossible to solve.

3.3. Threat Analysis

For any serious work about security concepts for a system design or architecture it is essential to identify the threats to that system. [**Sch96**], [**Eck03**] and others proposed various methodologies for threat analysis. Since this is not the main focus of this dissertation and since many system specifics are only known on a conceptual level, a relatively simple approach has been taken. After defining the threat model, consisting of attacker model and system model, those system components are identified, where attacks are considered to be of interest. After defining tangible security goals, attack trees are used for threat analysis. Then, attacks that are out of scope for this work are identified and a risk assessment using the DRED (Damage, Reproducibility, Exploitability, Discoverability) model is given.

3.3.1. Attacker Model

Attackers can generally be categorized according to:

Insider Information: Do they have access to confidential insider information, such as knowledge about technologies, business models, people involved, processes, etc.?

Mobility: Are they working from a specific location or are they able to move to specific locations, where attacks are easier to perform? Can they even chase persons or devices to maintain access?

Motivation: What is the motivation for an attack? Is it related to money, revenge, glory, personal satisfaction, etc.?

Skills: What are the skills in terms of education, technical proficiency, physical abilities, etc.?

Resources: What kind of resources are accessible to the attackers? This could be money, special equipment, time, manpower or any other kind of resource that helps in mounting an attack.

Collaboration: Is the attacker an individual, a group of individuals or a coordinated organization consisting of multiple groups in various locations? To which degree do they collaborate, coordinate attack times, exchange information or fully share all resources?

Attack Target: Which part of the system is targeted? Software, hardware, people? How far do they want to go, breaking regulations, breaking the law, committing crimes (bribing, violence, homicides) or even preparing an armed conflict?

The threat model in this thesis uses a generic attacker model with five groups of attackers:

(1) **Hackers** Attackers with a programmable radio transmitter/receiver.

(2) **Owners** Attackers with access to an un-modified OBU (On-Board Unit) who can therefore control the inputs, sensors, etc.

(3) **Crackers** Attackers who have access to a modified OBU unit and who have obtained the keying material.

(4) **Insiders** "Inside" attackers who have access to records and equipment operated by the vehicle manufacturer or the OBU unit manufacturer.

(5) **Organisations** Industry- or state-funded organizations that have access to internal and classified information and possess almost unlimited resources.

Groups of Attackers					
	Hackers	**Owners**	**Crackers**	**Insiders**	**Organizations**
Access to car	no	yes	yes	yes	yes
Access to OBU	no	no	yes	yes	yes
Insider Infos	no	no	no	yes	yes
Motivation	all	all	all	all	only rational
Skills	medium	medium	high	high	high
Resources	low	low	medium	high	high
Collaboration	high	medium	medium	high	high
Attack Target	software	software	SW & HW	SW & HW & people	SW & HW & people

Table 3.3. Attacker Model

In the following chapters only the groups **Hackers**, **Owners** and **Crackers** will be considered. Defending against **Insiders** requires organizational measures that are out of scope. **Organizations** are very hard to defend against, they must in any case be treated as if they were able to break the system's security, especially if they have the capability to cryptanalyze algorithms and protocols.

3.3.2. Attackable Components

Anticipating the system architecture in chapter 4 there are roughly following approaches toward attacking an IVC system, as shown in figure 3.6:

- **OBU Input**
- **OBU/RSU**
- **Lower OSI Layers**
- **Message**
- **Position and Time Information**
- **Protocols**
- **Security Infrastructure**

Note that this is a component oriented viewpoint based on an architectural perspective. It is used to develop a basic understanding about where in a system attacks are to be expected. However, it does not provide a view on what the effects of attacks are and how they are conducted. While the first viewpoint is mostly related to security mechanisms, the latter relates more to security architectures and security models (see also Figure 3.3 on page 31). A detailed threat analysis and reasoning of this work's focus on security will be given in the following sections.

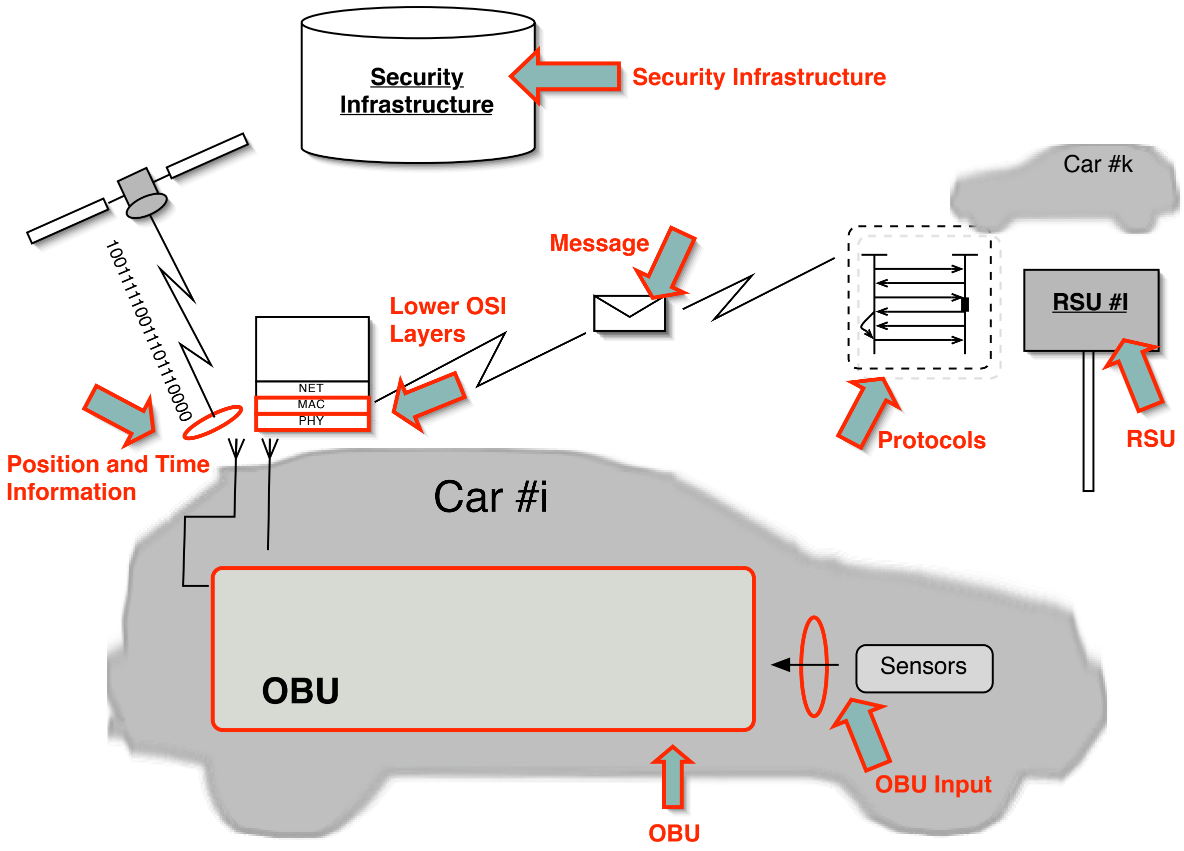


Figure 3.6. Attackable Components

3.3.3. Security Goals

This section describes general security goals that follow the problem outline (1.3), requirements (2.5) and design criteria (3.2.1.2). These goals are a starting point for security analysis and will be reformulated in section 3.4 as security objectives following the analysis.

- every authorized receiver must be able to check whether a message is valid and unaltered.
- there has to be a way to offer receiving nodes a possibility to estimate the trustworthiness of a sending node and possibly the sending nodes classification about the confidence into the message's content.
- the system has to offer a way of dealing with security breaches, new attacks and other forms of unpredictable unconveniences
- it must be (computationally) impossible for anyone to gain knowledge about a node's identity, position or sent messages except for a priori knowledge or using the appropriate system functions.
- nodes must not be able to abuse system resources for a long period of time

3.3.4. Attack Trees

Relevant attacks will be structured in in this paragraph using attack trees. There is one attack tree for every major application type as defined in paragraph 2.4.2. Figure 3.7 shows how the subtrees are structured.

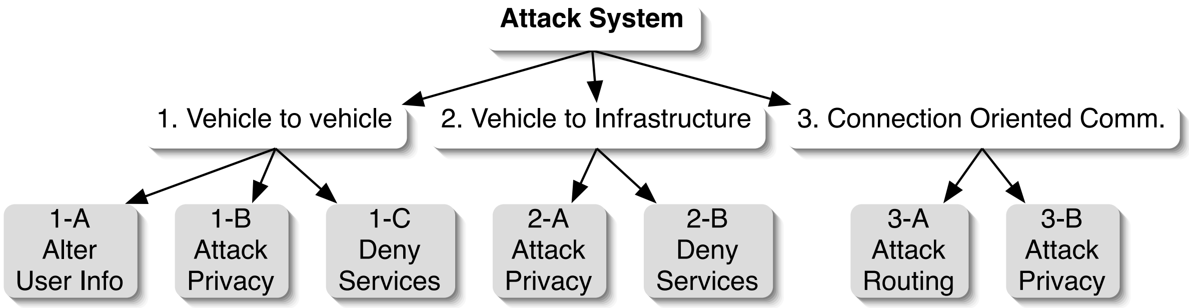


Figure 3.7. Overview Attack Trees

3.3.4.1. Type 1: Vehicle to Vehicle Traffic-related. This paragraph introduces attack trees referring to type-1 applications. Note that subtree 1-C: Deny Services does only include external attacks, since attacking the own node without affecting other parts of the system is not considered to be a relevant threat.

Subtree 1-A: Alter User Information

```
OR  1.  Manipulate generation of messages
        OR  1.  Attack cryptographic system
            2.  Alter sensor input
    2.  Alter message dissemination
        OR  1.  Selectively forward messages
            2.  Alter time-to-live, target region, lifetime, etc.
    3.  Manipulate reasoning process on OBU
        OR  1.  Directly manipulate reasoning of vehicles by exploiting
                vulnerabilities in reasoning algorithms
            2.  Indirectly manipulate reasoning
    4.  Manipulate decision process
```

Table 3.4. Subtree: Alter User Information (V2V)

Subtree 1-B: Attack Privacy

```
OR  1.  Track a specific node
        OR  1.  Track locally (e.g. driving nearby)
                AND 1.  Intercept messages from target node
                    2.  (Re-)recognize node
                    3.  Generate traces by correlating messages
            2.  Track remotely (e.g. through compromised hotspots)
                AND 1.  Collect messages via distributed receivers
                    2.  Correlate and filter out target node's messages
                    3.  Generate traces
    2.  Link person and node-identifier
    3.  Reveal communication relations
    4.  Extract private information
```

Table 3.5. Subtree: Attack Privacy (V2V)

```
Subtree 1-C: Deny Services

OR  1.    Remote Incapacitation of components
     OR  1.    Generate EMP
         2.    Stimulate System Malfunction
     2.    Suppress Communication
     OR  1.    802.11 jamming
         2.    GPS jamming
         3.    Inhibiting physical environment
         4.    802.11 weaknesses and flaws abuse
     3.    Message Distribution Misbehavior
     OR  1.    Overload nodes
         2.    Disturb forwarding / routing
         3.    Don't participate in message forwarding / routing
     4.    Application Layer Misbehavior
     OR  1.    Generate many false messages
         2.    Generate corrupt messages
```

Table 3.6. Subtree: Deny Services (V2V)

3.3.4.2. Type 2: Vehicle to Infrastructure (Hotspot). In type-2 application scenarios the situation is somewhat different, since hotspots may have a different trust level. Since they are connected to other networks, they can also provide up-to-date credentials. However, hotspots may be operated by different, more or less noble, organizations. They may either provide access to attackers due to unfixed security-holes or even be used to stage attacks directly.

```
Subtree 2-A: Attack Privacy

OR  1.    Location Profiling
     OR  1.    Passive attack
         AND 1.    Correlate encrypted messages
             2.    Communicate with other eavesdroppers
             3.    Break encryption
         2.    Active attack
         AND 1.    Identity theft
             2.    Man-in-the-middle
         3.    Operate own hotspot
     2.    Information Profiling
         AND 1.    Get information
             2.    Link person and information
             OR  1.    Observe behavior
                 2.    Get access to TTP that links person and identifier
             3.    Create Profile
```

Table 3.7. Subtree: Attack Privacy (V2I)

3.3.4.3. Type 3: Connection Oriented Communication. Type-3 application types are relying on routing mechanisms which also represent their main attack targets. There are two specific attack subtrees, one for all attacks related to the routing functionality itself and another one for all privacy related attacks targeted at the routing system.

Subtree 2-B: Deny Services

OR
1. Remote Incapacitation of Devices (see Subtree 1-C: 1.)
2. Suppress Communication (see Subtree 1-C: 2.)
3. Physical Attacks on Hotspot
   OR
   1. Disrupt power supply
   2. Disrupt antenna cable
   3. Shield antennae
   4. Physically destroy components
4. Overload Devices
   OR
   1. Send an abundance of packets
   2. Send packets that require large memory
   3. Send packets that require long processing time

Table 3.8. Subtree: Deny Services (V2I)

Subtree 3-A: Attack Routing

OR
1. Manipulate Neighborhood Information
   OR
   1. Inject manipulated beacon message
   2. Inject manipulated data message
2. Attack Routing Process
   OR
   1. Degrade packet forwarding
   2. Abuse routing to monitor network traffic
3. Take advantage of network

Table 3.9. Subtree: Inject Messages (COC)

Subtree 3-B: Attack Privacy

OR
1. Eavesdrop on messages
   OR
   1. Eavesdrop on forwarded data message
   2. Inject beacon message with falsified node ID,
      position information and/or timestamp
   3. Inject data message with falsified piggybacked information
2. Reveal node positions

Table 3.10. Subtree: Attack Privacy (COC)

3.3.5. Out-of-Scope Attacks

While all attacks are valid approaches from a security perspective, following attacks are out of this thesis' scope:

(1) **Attacks on Infrastructure** Security issues of infrastructure such as satellite navigation or security infrastructure itself is not part of this work and therefore assumed to be secure. However, attacks related to important functionalities (such as satellite navigation) and inherent to the architecture will be discussed.

(2) **Software-Based Compromise of Units** Software related security topics are not part of this work and software is assumed to have the properties and functionalities as described and nothing more or less.

(3) **Attacks on Tamper Resistant Modules (TRMs)** Technologies to improve tamper resistance of these modules are key-knowledge of TRM manufacturers. They are assumed to provide the stated qualities 100

(4) **Misconfiguration (Accidental or Intentional)** Misconfiguration relates to operational issues, which are mainly covered my organizational measures and are not regarded in this work.

(5) **Physical Attacks and Radio Jamming** Radio Jamming and other forms of physical attacks cannot be covered by technological measures provided by this work and are therefore not further discussed.

(6) **Social Attacks** Social attacks cannot be covered by technological measures provided by this work and are therefore not further discussed.

3.3.6. Risk Estimation and Damage Potential

In this thesis, the DRED (Damage potential, Reproducibility, Exploitability, Discoverability) model, that has been proposed by Ostermaier [**Ost05**] following the DREAD model proposed in [**MMV$^+$03**], will be used to assess the risks of IVC system vulnerabilities. Each vulnerability is examined according to the four dimensions of the DRED model and rated with low (1), medium (2) or high (3) risk by assigning the number in brackets. The DRED dimensions are:

- **Damage potential** How much potential does an attacker gain if a vulnerability is exploited?
- **Reproducibility** How easy is it to reproduce the attack?
- **Exploitability** How easy is it to launch an attack?
- **Discoverability** How easy is it to find the vulnerability?

The overall risk is then determined by adding the values of each dimension, and then grouping it accordingly into low, medium and high risk.

The risk estimation process is summed up in table 3.11. The criteria used in this threat analysis are based on following guideline[12]:

- **Damage potential**
  (1) Limited number of nodes affected, mostly within communication range OR occurring only during very short period of time
  (2) Nodes in a certain area affected OR occurring during a period of time long enough to affect stabilizing system variables
  (3) Whole system affected OR persistent until nodes are actively updated
- **Reproducibility**
  (1) Requires H/W modification OR special equipment
  (2) S/W only but requires reconfiguration of code
  (3) Cut and paste style, works on all nodes
- **Exploitability**
  (1) Requires H/W access AND collusion and/or maintenance of multiple equipment
  (2) Requires access to node H/W OR some form of limited collusion
  (3) Remote exploitability
- **Discoverability**

[12] It should be noted here, that the threat analysis already assumed some characteristics of a system architecture and that this process has been carried out iteratively throughout the system architecture's design phase.

(1) Noticeable by every user

(2) Not detectable / Only detectable by specialists, on most system nodes

(3) Not detectable / Only detectable by specialists, only in specific affected areas, only during time of occurrence

Attack	**Description**	**D**	**R**	**E**	**D**	**Sum**	**Rating**
1-A.1.1	Attack cryptographic system	3	3	3	2	11	excluded (2)
1-A.1.2	Alter sensor input	2	1	2	3	8	SO-1
1-A.2.1	Selectively forward messages	1	3	3	3	10	SO-3
1-A.2.2	Alter TTL, target region, lifetime, etc.	1	3	3	2	9	SO-3
1-A.3.1	Directly manipulate reasoning of vehicles	1	2	2	3	8	SO-1
1-A.3.2	Indirectly manipulate reasoning	2	3	1	2	8	SO-1
1-A.4	Manipulate decision process	2	2	2	3	9	SO-1
1-B.1.1	Track locally	1	1	2	3	7	SO-5
1-B.1.2	Track remotely	3	2	3	2	10	SO-5
1-B.2	Link person and node-identifier	2	2	1	3	8	SO-5
1-B.3	Reveal communication relations	2	2	1	3	8	SO-5
1-B.4	Extract private information	3	3	2	3	11	SO-5
1-C.1.1	Generate EMP	2	1	1	1	5	excluded (5)
1-C.1.2	Stimulate System Malfunction	2	2	3	1	8	SO-3
1-C.2.1	802.11 jamming	1	3	2	1	7	excluded (5)
1-C.2.2	GPS jamming	3	2	2	1	8	excluded (5)
1-C.2.3	Inhibiting physical environment	1	1	1	2	5	SO-3
1-C.2.4	802.11 weaknesses and flaws abuse	2	3	3	2	10	excluded (2)
1-C.3.1	Overload nodes	2	1	2	2	7	SO-3
1-C.3.2	Disturb forwarding / routing	2	2	3	2	9	SO-4
1-C.3.3	No message forwarding / routing participation	1	3	3	3	10	SO-3
1-C.4.1	Generate many false messages	1	2	3	1	7	SO-1, SO-3
1-C.4.2	Generate corrupt messages	2	2	3	1	8	SO-3
2-A.1.1	Passive attack	1	3	1	3	8	SO-6, SO-7
2-A.1.2	Active attack	2	2	2	2	8	SO-3, SO-6
2-A.1.3	Operate own hotspot	2	1	1	2	6	SO-2
2-A.2	Information Profiling	3	2	3	3	11	SO-5
2-B.1	Suppress Communication	2	2	2	1	7	SO-3
2-B.2	Remote Incapacitation of Devices	3	2	1	1	7	excluded (2,3,4)
2-B.3.1	Disrupt power supply	1	1	2	2	6	SO-3
2-B.3.2	Disrupt antenna cable	1	1	2	2	6	SO-3
2-B.3.3	Shield antennae	1	1	2	2	6	SO-3
2-B.3.4	Physically destroy components	1	1	2	2	6	excluded (5)
2-B.4.1	Send an abundance of packets	3	2	3	1	9	SO-3
2-B.4.2	Send packets requiring large memory	3	3	3	1	10	SO-3
2-B.4.3	Send packets requiring long processing time	3	3	3	1	10	SO-3
3-A.1.1	Inject manipulated beacon message	3	3	3	2	11	SO-4
3-A.1.2	Inject manipulated data message	2	3	3	2	10	SO-8
3-A.2.1	Degrade packet forwarding	2	2	2	3	9	SO-4
3-A.2.2	Abuse routing to monitor network traffic	1	2	2	2	7	SO-8
3-A.3	Take advantage of network	1	2	2	2	7	SO-4
3-B.1.1	Eavesdrop on forwarded data message	2	2	2	3	9	SO-5
3-B.1.2	Inject beacon message	2	3	3	2	10	SO-5
3-B.1.3	Inject data message	2	3	3	2	10	SO-5
3-B.2	Reveal node positions	3	2	1	3	9	SO-5

Table 3.11. Rating of attacks according to the DRED model

3.4. Security Objectives

This section lists those security services that have been identified in the threat analysis. They have been determined by weighing the attack trees' nodes with risk estimations according to the DRED model. It is a list of all security objectives that the system has to provide. These security services may be implemented differently for each of the three application types, see Table 3.12, and will be discussed in chapters 5, 6 and 7.

3.4.1. Authenticity

Authenticity is defined in RFC 2828 as *"The property of being genuine and able to be verified and be trusted."* The question is **what** is genuine and be able to be verified. In ad-hoc network security this is mostly related to node authenticity, which is also an important factor in this thesis. However, node authenticity is not sufficient in the type-1 scenario, the reasons will be discussed in chapter 5. Information authenticity is required, because ultimately a receiver is not interested in who sent the message (and if the sender can be trusted), but if the information is correct. Note, that this does not meet the stringent definition given in RFC 2828 in a sense that information authenticity cannot be guaranteed 100% in the given context.

3.4.1.1. Information Authenticity.

Requirement 3.5 (Information Authenticity)**.** *Traffic-related vehicle to vehicle communication has to provide information authenticity.*

3.4.1.2. Node Authenticity.

Requirement 3.6 (Node Authenticity)**.** *Vehicle to infrastructure communication and connection oriented communication have to provide node authenticity.*

3.4.2. Robustness

Defining robustness as dependability with respect to external factors includes availability and reliability as primary attributes.

3.4.2.1. Availability. RFC 2828 defines availability as *"The property of being accessible and usable upon demand by an authorized entity."* This general definition about availability includes the case of being accessible even if the number of nodes increases dramatically, a property known as scalability. This case is of great importance in this work, because a drastic increase of nodes reduces message distribution efficiency. Another factor is the robustness of applications[13].

Requirement 3.7 (Availability)**.** *The overall system architecture and especially traffic-related applications must be designed in such a way, that it remains operational under extreme usage conditions.*

3.4.2.2. Reliability. RFC 2828 defines reliability as *The ability of a system to perform a required function under stated conditions for a specified period of time.* In the context of this thesis, this relates to the robustness of routed multi-hop connections as used in connection oriented communication (type-3 scenario). The vehicle to vehicle scenario (type-1) does not have stateful connections and the vehicle to

[13] In contrast to the robustness of routing, which is discussed under reliability.

infrastructure scenario (type-2) is based on single hop connections, where issues such as route stability are not important.

Requirement 3.8 (Reliability). *Connection oriented communication must be designed in such a way that connections are robust against external disturbances and malfunctioning nodes.*

3.4.3. Privacy

3.4.3.1. Pseudonymity.

Requirement 3.9 (Pseudonymity).
It must be possible to use pseudonyms as identifiers instead of real-world identities.
It must be possible to change these pseudonyms, where the number of pseudonym changes depends on the application and its privacy threat model.
Pseudonyms used during communication can be mapped to real-world identities in special situations.
A set of properties and/or privileges can be cryptographically bound to one or more pseudonyms.

3.4.3.2. Link Anonymity. In V2I scenarios, where user credentials are used for authentication, identities have to be protected from eavesdroppers. Otherwise, the static nature of hotspots would make wireless surveillance a concern, especially if attackers are using a network of surveillance stations.

Requirement 3.10 (Link Anonymity). *Vehicle to infrastructure communication requires that identities of two communicating nodes may not be extracted by third party nodes overhearing the wireless channel even if digital credentials are used for authentication.*

3.4.3.3. Link Confidentiality. In V2I scenarios, messages contents may have to be protected from external listeners.

Requirement 3.11 (Link Confidentiality). *Vehicle to infrastructure communication requires that messages sent between a car and an infrastructure node are protected from eavesdroppers.*

3.4.3.4. Route Confidentiality. Many applications that use connection oriented communication require that the content is protected against eavesdropping.

Requirement 3.12 (Route Confidentiality). *Connection oriented communication must provide a way to secure an end-to-end connection.*

3.4.4. Freshness

Time plays an important role in all security approaches. Especially in distributed systems however, avoiding race conditions and replay attacks is challenging because the time-frame of communicating nodes have to be synchronized. The solutions offered will be different for type-1 / type-3 scenarios and type-2 scenarios. In the first case the global time of satellite navigation systems is used as a reference and combined with a real-time clock within the OBUs' tamper-resistant boundary. In the second case, the handshaking protocol ensures freshness of the session, because availability of satellite navigation equipment cannot be assumed for fixed nodes.

Requirement 3.13 (Freshness). *Freshness of messages has to be ensured.*

		V2V	V2I	COC
SO-1	Information Authenticity	yes	no	no
SO-2	Node Authenticity	no	yes	yes
SO-3	Availability	yes	yes	yes
SO-4	Reliability	no	no	yes
SO-5	Pseudonymity	yes	yes	no
SO-6	Link Anonymity	no	yes	no
SO-7	Link Confidentiality	no	yes	no
SO-8	Route Confidentiality	no	no	yes
SO-9	Freshness	yes	yes	yes

Table 3.12. Security Objectives per Application-Type

CHAPTER 4

Architecture and Building Blocks

The architecture is based on the constraints and assumptions of chapter 3 as well as the goals that have been discussed in chapter 2.

The first section defines essential terms and definitions. A high-level view on the overall architecture is presented in the second section. This represents the proposed approach to meet the basic requirements given in section 2.5 and security requirements given in section 3.4. The third section presents an approach to address privacy related requirements. A privacy architecture is layed out, corresponding to the system lifecycle model and new methods for effective pseudonym generation, distribution and usage within the given scenario are developed. Finally, section four introduces and describes building blocks and concepts of a Secure Architecture for Robust Inter-vehicle communication - SARI. They provide basic functions and fundamental features that are universally required for all application types. Some parts of these building blocks however may require extensions and/or adaptations to specific application types. This will be discussed in successive chapters 5, 6, 7. The degree of detail of the concepts presented in this chapter varies greatly due to the different complexities of the problems and difficulty of adapting established solutions.

4.1. Definitions and Terms

Definition 11 (Scenario)**.** *Scenarios are built by classification of multiple sets of environmental settings according to their values. The sets of environmental settings given in this work are either related to the traffic environment (average speed, top speed, standard deviation of speed, number of turns per time, clustering coefficient, etc.) or the communication environment (communication neighbor distribution, average link time, standard deviation of link time, channel quality statistics, distribution of bandwidth usage by given applications, etc.)*

Definition 12 (Component)**.** *In this thesis, a component is defined as a hardware entity, software entity or a communication channel within the system context. Depending on the level of detail required, a component may be subdivided into smaller (sub-)components. In general components are related to functionalities and therefore represent logical parts of a larger system, in some special cases however a logical view on a component will be identical with its physical dimension.*

Definition 13 (Building Block)**.** *SARI building blocks (or just building blocks) provide basic functions and fundamental features that are universally required to operate the system, independent of application types.*

4.2. SARI - Secure Architecture for Robust Inter-vehicle Communication

4.2.1. Architectural Constraints

With respect to a tangible architecture and requirements given in section 2.5 decisions had to be made regarding the vehicles' configuration and used funcionality:

Constraint 6 (No digital maps)**.** *The general requirement of cost effectiveness and the constraint of max. 100 EURO equipment lead to renouncing costly digital maps and using algorithms that use native positioning information instead of relying on digitized map information. This decision also avoids the problem of updating and synchronizing digital map data in a secure way.*

Constraint 7 (Use of smart cards)**.** *Using smart card technology to realize various security features offers the advantages of relying on an established technology that provides a high level of tamper resistance. Due to their widespread usage in large numbers, smart cards are also quite cost effective when compared to other tamper resistant devices.*

Constraint 8 (Restrictive Use of Beacons)**.** *Beacons are short messages*[1] *that are sent periodically to exchange neighborhood information. They are used for routing decisions and other functions in a Vehicle Ad-hoc NETwork. However, they can dramatically increase network traffic in crowded environments due to channel access problems. Additionally, periodic transmission of messages makes it hard to achieve the desired degree of privacy. Therefore, beacons will only be used if exchanging neighborhood information is necessary (depending on active applications) and if this information cannot be appended to other messages during communication. In order to provide adequate privacy, backoff times are used, where beacons are not allowed.*

Constraint 9 (Low complexity software)**.** *This requirement is derived from threat analysis and cost of maintenance for software components. Algorithms and programs have to be as simple as possible in order to be able to verify code before it is installed on smart cards. In addition to that, smart cards' computational power and memory is quite limited, which requires efficient algorithms and exploitation of dedicated hardware such as cryptographic coprocessors.*

Constraint 10 (Lifecycle Management)**.** *When looking at the architecture in Figure 4.1, it can be seen that security depends to a large degree on tamper-resistance property of the secure environment. Thus, tampering is a relevant threat and the security architecture must be able to cope with newly invented attacks which are impossible to foresee universally during system development. This requires that the architecture is designed in such a way that updates are possible and each TRM provides distinct secure states for different phases, such as an operational phase and a maintenance phase (where updates are possible).*

4.2.2. SARI Components

This section will introduce the SARI architecture by explaining its functional components. The perspective using components is a implementation related structure, that is important for understanding the operation of SARI and will be referred to in later chapters. Later, in section 4.4, building blocks will be described, technical concepts or proposals that address functional requirements and are fundamental for the system's operability.

Figure 4.1 provides an overview on the architecture. The overall architecture consists of:

- Nodes, either mobile (Cars) or stationary (RSUs)
- Centralized components: security infrastructure, satellite navigation system, etc.

A single node consists of following components:

[1] sometimes called hello messages

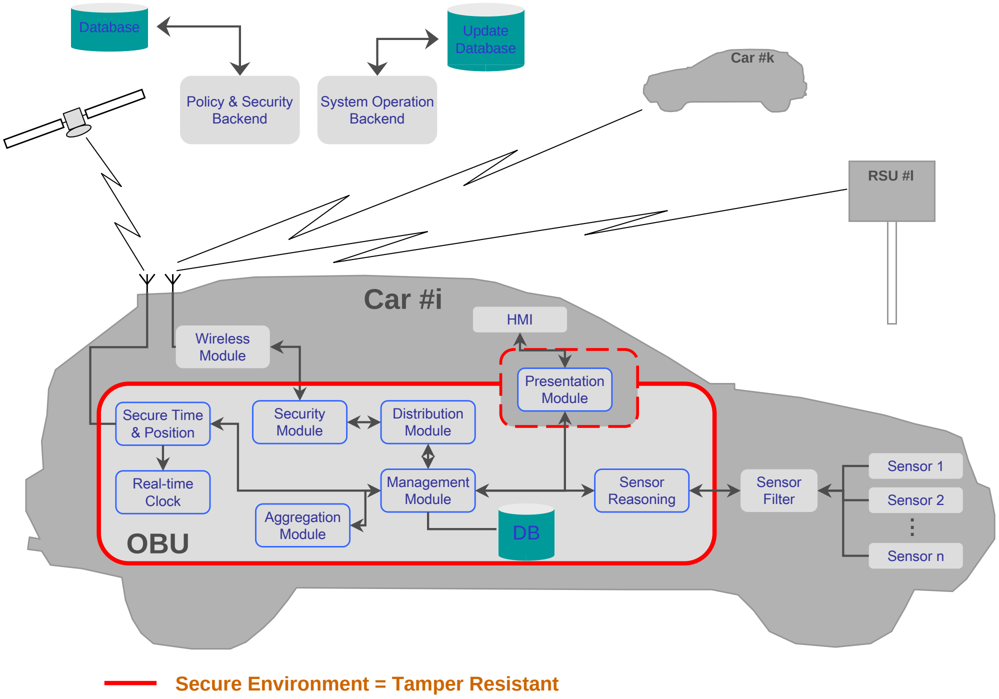


Figure 4.1. General Architecture

WM - Wireless Module: The wireless module represents a communication device that transmits and receives messages covering ISO-OSI layers 1 and 2. It interfaces directly with the security module and other nodes. Logically it also receives control commands and transmission parameters from the central management module. Attacks on wireless modules are out-of-scope, since they only represent basic communication technology that is not focus of this work.

SM - Security Module: The security module performs all kinds of cryptographic operations, such as integrity protection, authentication, encryption/decryption, etc.

For type-1 applications this means mainly adding and verifying integrity protection of sent / received frames. For type-2 applications this means mainly verifying sender authenticity, establishing temporary keys and encryption/decryption according to the protocol presented in section 6.3. For type-3 applications this means mainly assigning integrity protection and sender authentication in support of routing protocol functions as explained in section 4.4.4. In addition to these basic mechanisms, specific applications may use the security module for their own use.

DM - Distribution Module: The distribution module includes all functions that are required for forwarding decisions. This also requires modifications on distribution parameters, size of the distribution area, time-to-live and destination addresses.

PM - Presentation Module: The presentation module includes all functions that are required for presentation decisions. Figure 4.1 shows this component as potentially being located within the secure environment because its operation has an impact on local security. But since the decision module does not directly affect other nodes[2] it is not as critical for overall system security as other components are and may therefore be placed outside the secure environment.

HMI - Human Machine Interface: The Human Machine Interface (HMI) is the component that performs all tasks required for adequate interaction with human users. These tasks include: choice of optimum presentation medium (HUD, display, audible, haptic, etc.), situation-adaptive display of information[3], user inputs, etc. However, human input that is relevant for the reasoning process (refer to section 3.1.3) will be dealt with as a sensor input. Accordingly this data is extracted by the sensor filtering module. In the sensor reasoning module's trust metric however, human input has a different rating than built-in sensors because users may input specific data on purpose and therefore affect security.

SE - Sensors: Most of the sensors mentioned in section 3.1.3 deliver a constant stream of data that is available on the car's internal communication buses, while other inputs may only be single commands (e.g. turning wipers on/off). The datarate may vary as well, comparing RADAR with temperature sensors for instance. From a security perspective, most sensors are not trivial to remove, but may be manipulated physically. Additionally, internal vehicle buses communicate in plaintext and can be accessed with equipment that is publicly available.

SF - Sensor Filter: This module has the task of gathering and filtering sensor data according to the needs of the IVC-system. This includes listening for required data packets on internal buses[4], as well as requesting data from ECUs. Filtering is necessary in order to limit the amount of data that has to be processed by consecutive modules. Security is affected only by lack of relevant information. However, the negative effects will be mitigated to a large degree by the sensor reasoning component. In practical implementations it does not make sense to include a sensor filtering / data gathering unit inside a tamper seal due to cost, performance and spatial considerations.

SR - Sensor Reasoning: After the sensor data has been filtered, the sensor reasoning module performs a number of actions to verify the sensor reading's consistency and checks whether the data matches the local world-model. Additionally, readings from different sensors are correlated with each other according to stored metrics. If unusual patterns are found, this will result in a event detection message to the **management module**. Note that the reliability of the sensor reasoning is also relevant, since a high-confidence reasoning may not be possible if there is only input from a single sensor. This module is an essential building block for the security of type-1 applications, which strongly rely on correct sensor data. If manipulated sensor readings can be detected at this level, wrong sensor data will not lead to wrong presentation decisions or forwarding decisions neither locally nor on remote nodes.

MM - Management Module: A central component of the system, which implements the application and system logic and coordinates the access to the central database.

DB - Database: All relevant data are stored in the central database, including some information from communication protocol stacks. This allows cross-layer interaction, required for the implementation of qualitative service features.

[2] Although the driver's reaction may influence the traffic flow.

[3] This is especially important in critical traffic situation: Is it better to inform the driver about a risky situation or is it better not to divert his attention away from the road?

[4] Modern Cars use field bus systems, such as CAN, Flexray, LIN, etc., to exchange data between different ECUs.

AM - Aggregation Module: The aggregation module merges information received in type-1 messages that belong to a single event. For area events, such as fog, this means that messages from different cars will be merged to a single dataset concerning a specific foggy area if they are correlated. The operation of this module is critical for security, since messages that do not provide additional information on events will be omitted. If for instance an event message concerning a known foggy area is received that lies inside the area, it will be deleted if it is older than the other messages. A positive side-effect of data aggregation is, that it increases the system's performance by removing irrelevant information. The aggregation of area events will be elaborated in section 5.3.

TP - Secure Time & Position: This module provides verifiable time+position pairs. There exist different approaches and this topic will be thoroughly discussed in section 4.4.3.

RTC - Real-time Clock: A local real-time clock keeps track of the global time in absence of satellite positioning signals or other global time information. The RTC has to be secured against manipulation (hence it is situated within the tamper seal) since exact global timing is essential for system security.

The security of SARI requires that some components cannot be easily altered or tampered with. These components have to be placed within a secure environment, as shown in Figure 4.1. However, this measure is expensive and therefore all components where this is not strictly required will be operated in a normal environment.

In addition to node components, there are centralized system infrastructure components that may only be accessible in specific situations according to the decentralization requirement 2.3:

Policy & Security Backend: This includes all infrastructure that is related to security tasks, such as trust centers, emergency response teams, privacy enhancing infrastructure, etc.

System Operation Backend: This term gathers all systems that are required for the operation and maintenance of the system itself. This includes managing software upgrades, fault handling, etc.

4.2.3. Lifecycle

Lifecycle management is an important part of the security concept introduced in this thesis. This concept will be used in conjunction with a tamper resistant module following the architecture used with smart cards. Other (node-)components' lifecycle are derived thereof. Using a tamper resistant module allows the definition of a number of lifecycle-states (see also section 4.3.2 and section 4.4.2). For each of these states, distinct access conditions and permitted methods are defined. The transitions between the states require pre-defined actions, for SARI's state-chart, see Figure 4.2.

Figure 4.2 shows the lifecycle-states that are defined for each single node:

4.2.3.1. OP-Ready. The "OP-Ready" state indicates that the tamper resistant module (TRM) manufacturer has finalized the production in a secure way and delivered the TRM to the vehicle manufacturer.

During production of a car when the tamper resistant module is installed, it has to be in the state "OP-Ready". Following tasks will be performed:

(1) Install vehicle manufacturer specific software including root-certificates, etc.

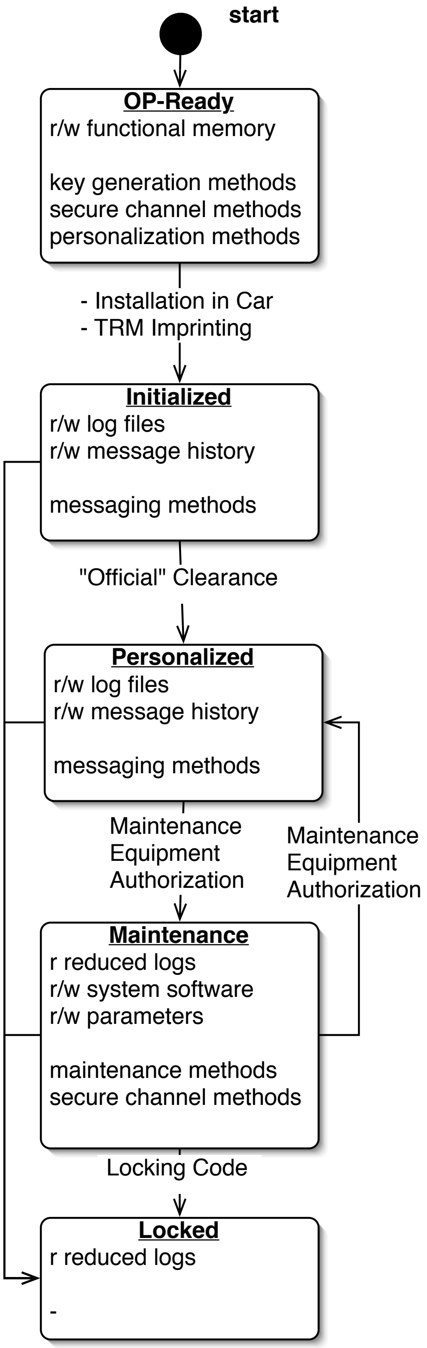


Figure 4.2. TRM Lifecycle

(2) Generate unique keys for this unit[5]

(3) Preparation for the specific car, including TRM imprinting, see section 4.4.2.2.

Regarding the access conditions and permitted methods, it is certainly necessary to have write access for verified software. On the methods side, there must be methods for key generation, personalization and to establish services / applications. But in this state it must not be possible to use methods related to messaging, except special testing routines.

[5] For security reasons this should ideally be performed on the tamper resistant module itself, but for performance reasons it may be necessary to do this on dedicated machines and then transfer the material securely onto the TRM.

4.2.3.2. Initialized. The state "Initialized" is only entered after the module has been installed, other system components have been imprinted and therefore been logically bound together. The transition from op-ready to initialized state requires a qualified approval from the vehicle manufacturer that the installed system works according to system specification and standards. The car is ready to be personalized for a specific user or group of users.

The car will be equipped with services ordered by a user, personal data will be installed and acquisition / installation of credentials for service usage, see section 4.3.2, will be performed.

Access conditions and permitted methods will be the same as during normal operation. However, note that services may not be usable without personalization. Maintenance methods are no more available and critical system data is only accessible read-only. This is thought to improve security because it requires a visible change of state and proper authentication to gain maintenance access. The initialized state is in fact identical to the personalized state, except that the act of formally personalizing the car has not been completed, which may have legal and organizational implications for applications.

4.2.3.3. Personalized. The transition from "Initialized" state to "Personalized" state is enacted by a vehicle manufacturer, authorized dealer or similar entity.

The system provides all functionality required for normal operation. Maintenance methods remain unavailable.

4.2.3.4. Maintenance. The idea of a "Maintenance" state is to be able to update / replace (security) critical system components, both hardware (see also section 4.4.2.2 for details on tamper resistant module imprinting) and software ((root)-certificates, methods, parameters, etc.). This state also provides a way to temporarily shutdown malfunctioning nodes[6]

To guarantee privacy and prevent misuse of personalized data, all privacy-relevant logs are erased and data with individual content (e.g. containing position information) is shrinked to a predefined level and / or scrambled when entering this state. However, some data-sets (usually the very last ones) are kept for analysis.

Obviously, the transition to the maintenance state requires proper authentication. As well as the transition from maintenance to personalized needs authentication to enable messaging functionality.

4.2.3.5. Locked. The "Locked" state is a final state, where the node is inoperable. In this state, all access to internal data such as keys is locked and no methods may be used any more. Only reduced versions of log files are available.

It has to be determined if there is a transition required from "Locked" to any of the other states. In this thesis it will be assumed that locked is a final state and there is no transition to any other state. For practical reasons, this means that once an OBU has been locked (either by an internal routine or by an authorized external command) it must be replaced by another OBU to reestablish functionality.

4.3. SARI and Privacy

Cars are personal devices, they are usually kept for a long time and in the future they will probably store lots of personal information as well. In many societies, cars are status symbols and a lot of personal behavior can be derived from the car a person is driving. Last but not least, most future automobiles will be equipped with navigation systems and therefore technically be able to gather complete movement patterns. If cars become connected, by inter-vehicle communication, conventional internet access,

[6] "Node" in this case refers to the OBU only. The shutdown of a car is certainly not intended here.

maintenance systems, tolling systems, software and media download, off-board navigation, etc., all this data will then potentially be available to unauthorized entities. A very dangerous and often ignored fact about privacy is that innocent looking data from various sources can be accumulated over a long period and evaluated automatically. Even small correlations of the data may reveal useful information. For instance, the knowledge about specific sensor characteristics may give some hints about the make and the model of car. This in turn may be related to other information to identify a specific car and track its location or actions. And once privacy is lost, it is very hard to re-establish that state of personal rights. Privacy sometimes contradicts with security requirements. While system operators want to find or identify attackers to take proper countermeasures, the ability to do so may be used for less noble reasons. Newsome, Shi, Song and Perrig presented a paper about sybil attacks in sensor networks [**NSSP04**]. One of their proposed countermeasures is registering nodes in the network. This concept is somewhat similar to the idea of electronic license plates. While their approach is absolutely reasonable for sensor networks, registration could turn out to be a major privacy concern in VANETs. Nevertheless, there has to be a process to identify and single out malfunctioning units or unacceptable behavior. Fulfilling the objectives concerning privacy that have been stated in secion 3.4.3 is a challenging task. On one hand, users of such networks have to be prevented from misuse of their private data by authorities, from location profiling and from other attacks on their privacy. On the other hand, system operators and car manufacturers have to be able to identify malfunctioning units to maintain system availability and security. These requirements demand an architecture that can manage privacy instead of either providing full protection or no privacy at all. Section 4.3.2 will outline a solution that address both, security and privacy concerns.

4.3.1. Privacy Terminology

Many of the terms used in privacy related topics have different meanings depending on who looks at it, therefore these terms are defined here as follows in this thesis:

Definition 14 (Identification)**.** *Identification is the process of unambigously recognizing an entity within a given set of entities.*

Definition 15 (Anonymity)**.** [**PK01**] *defines an anonymity as "the state of not being identifiable within a set of subjects, the anonymity set."*

Definition 16 (Pseudonymity)**.** *In this work, the defintion provided in Common Criteria ISO IS 15408* [**ISO99**]*, will be used for pseudonymity: "Pseudonymity ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use." Note that applying the more strict terminology provided in* [**PK01**] *would define the kind of pseudonymity used here as "role pseudonymity".*

Definition 17 (Re-recognition)**.** *Here, re-recognition will be defined as follows: "Re-recognition is the ability to recognize a subject within a group that has been recognized at an earlier time." Note that if a system design allows re-recognition, it still provides pseudonymity, but not anonymity.*

When discussing privacy enhancing technologies (PET), following parameters are of interest:

Perseverance of Identifiers: This is related to how often identifiers used by an entity are changed. Related to this question is whether identifiers are only used once and will be discarded after usage or will be re-used after some time. This is relevant for choosing the identifier switching strategy and depends on the attacks that are to be prevented.

Entropy of Content: The question whether the content bears any structure is of great importance for systems to improve privacy. Note that Shannon stated that the degree of information carried by a message is much higher if the probability of this content being sent is lower.

$$I(m) = -logp(m)$$

with p(m) being the probability that message m is chosen from all possible choices in the message space M. The relevance of this is how much critical information is contained in a message. For instance if all messages only carry strings chosen from a very limited set of strings (such as traffic-related events) and the probability of each string being sent is almost equal, this results in a lower privacy threat potential.

Size of Anonymity Set: The effectiveness of pseudonymization and other techniques of hiding a specific entity within similar looking entities depends on the size of the anonymity set (see definition 15). This will become relevant when changing pseudonyms, since this is only effective in a group of pseudonym changing nodes that is large enough so correlation becomes too expensive.

Side-Channels: Almost every system has side-channels where information can be leaked unintentionally. This can pose a serious threat to privacy. Therefore the number and quality of potential side-channels has to be evaluated according to their effects on privacy.

For explanations of other privacy related terminology refer to [**PK01**] and [**SD02**].

4.3.2. Privacy Architecture

The architecture introduced in this section fulfills the requirement 3.9 by relying on a trusted third party. Although this requires a centralized infrastructure, it does not violate the softened requirement of decentralization that is used in this thesis, since a connection is only necessary during the setup phase, when the node / OBU is in initialization state.

The main concept is to rely on a trusted third party approach supported by smart cards. One major requirement is the use of non-interactive protocols in the operational phase, since most messages will be sent in broadcast style.

The fundamental idea of this architecture is that there will be an authority A, that is trusted by all parties participating in the network: customers, manufacturers, system operators, service providers, etc. Gordon Peredo has presented such an architecture in [**Per03**]. The authority must be independent from other parties' interests and obtain special legislative protection. Authority A stores real-world identities and maps one or more pseudonyms to each identity. The mapping is kept secret and will only be revealed in exactly defined situations.

For the privacy architecture, three states in the lifecycle of a single vehicle are of special interest. The initialization state where the systems of a vehicle are set up. The personalized state as the major mode of operation, where vehicles can send messages signed according to a chosen pseudonym. And the maintenance state, where the normal operation of a node is shut down and may lead to a complete lock-down - the locked state - and disclosure of a vehicle's real ID according to pre-defined situations. In order to better explain the different phases provided in the privacy architecture, a distinction will be made between these phases and a single node's states. However, it must be clear that the phases are directly linked (and derived from) a node's states.

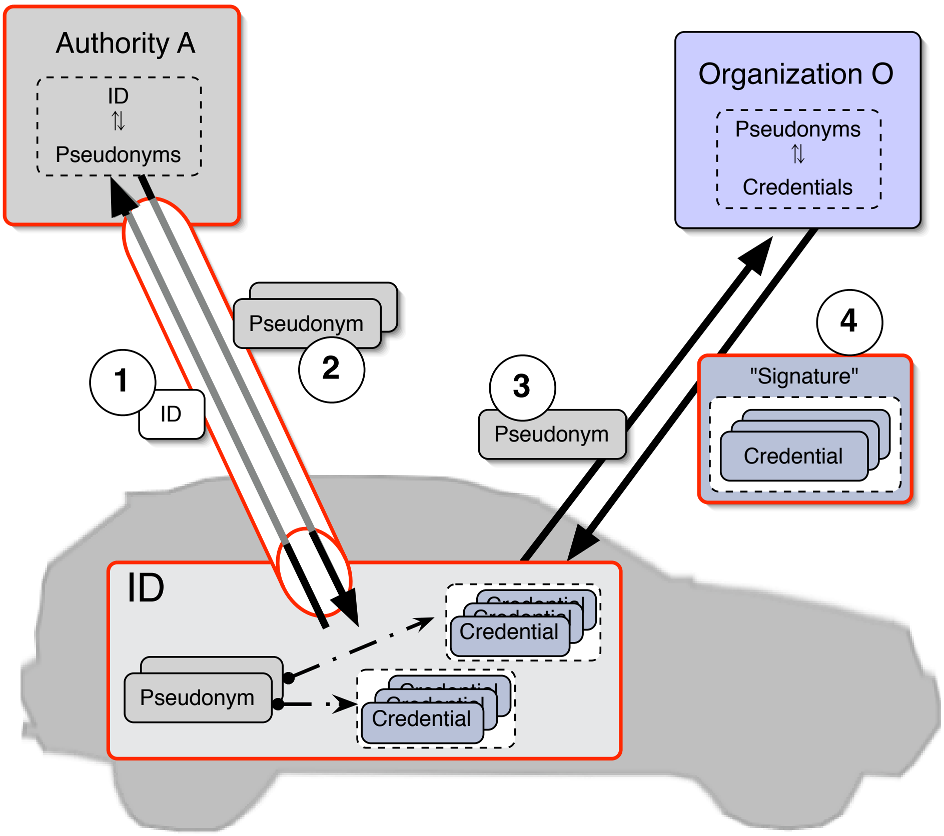


Figure 4.3. Initialization Phase & Architecture

4.3.2.1. Initialization Phase. Figure 4.3 gives an overview on the entities in this architecture. Each car is equipped with a smart card during its production that is fixed physically and cannot be removed without destruction. The smart card (and therefore also the car) is associated with a unique, immutable and non-migratable electronic ID. A secure communication link is established between the smart card and authority A. The smart card transmits the ID (1) and the authority cryptographically derives a set of pseudonyms from that ID after checking it. The pseudonyms are then transmitted back to the smart card (2). Now the car is ready to subscribe for various services. For every pseudonym (3) the car can get multiple service subscriptions from organization O, typically a car manufacturer. For every pseudonym organization O generates a set credentials, one for each service and sends it back (4). The car is now ready for use. This assumes that during production of a car a secure connection between this device and authority A is available. This can be realized using Smart Card Management Systems such as Visa's Global Platform [**Pla01**], which enables "secure channels" from smart cards to backend hardware security modules (HSMs) [**Doe02**].

4.3.2.2. Operational Phase. During normal operation a car may choose any of its pseudonyms and the related credentials to testify its rights or sign messages, see Figure 4.4. A communication partner can therefore always check the credentials in order to verify the car's compliance with common standards, its right to use a service or other properties that have been approved by organization O.

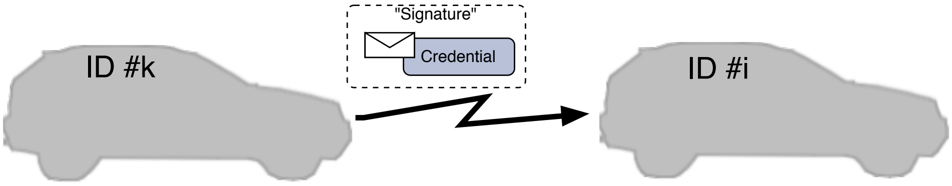


Figure 4.4. Operational Phase

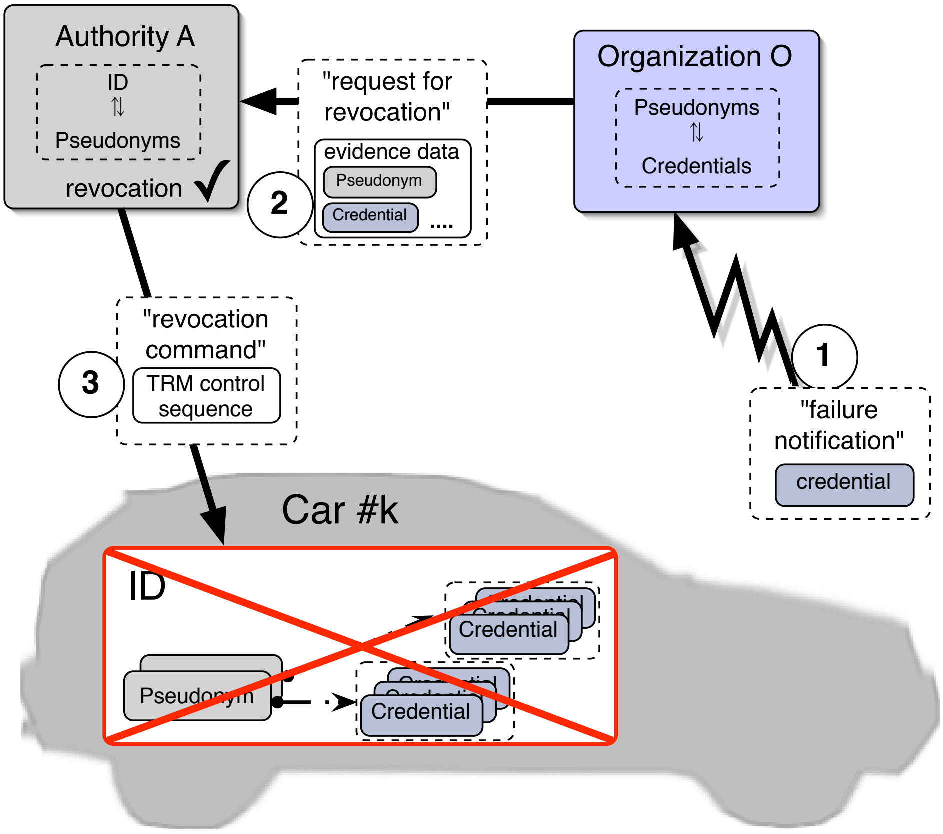


Figure 4.5. Revocation Phase

4.3.2.3. Credential Revocation Phase. Figure 4.5 shows what happens if there is evidence of malfunctioning or malicious nodes. Some entitiy within the network will initiate a revocation request and send their evidence to organization O (1). Organization O must gather all evidence about a malfunctioning unit until the information suffices to appeal to authority A for identification resolution and service shutdown (2). If the evidence is sufficient to allow for ID disclosure according to authority A's guidelines, it will compute a reverse mapping from a pseudonym to the real identity. Thus, organization O can find malfunctioning or malicious nodes and therefore maintain the long-term availability of the system. Smart card operating systems allow the shutdown of the whole OBU. This is one possibility to take action and could be initiated by authority A directly (3). However, it cannot be guaranteed that the malfunctioning / malicious unit is shut down on time.

Note that during the normal operation, the ID, all pseudonyms and credentials are stored in a tamper resistant smart card and therefore significant protection against misuse is provided.

4.3.3. Method for Anonymous-Credential Generation and Distribution

The realization of the privacy architecture mentioned in the previous section requires a way to assign some kind of credentials to nodes which they can use for authentication regarding a related service. The solution has to provide authentication in a sense that a node, the prover P, can prove to have been assigned a credential to another node, the verifier V. Additionally, the integrity of such a message has to be ensured. The specific setup for type-1 applications demands that it works without interaction, according to requirement 2.5. And it has to reflect the privacy architecture with respect to the processes in initialization, operational and revocation phases.

4.3.3.1. Zero Knowledge Proofs. When looking at the requirements given above, it becomes clear that digital certificates such as the ones being used with PKIs are always bound to the public key included in the certificate and thus cannot be used for anonymized or pseudonymized communication. Reflecting the given problem and looking at advanced cryptographic tools, zero knowlege proofs provide a way to meet aforementioned requirements. A zero knowledge proof is a cryptographic protocol where a prover "P", called Peggy, can prove to the verifier "V" or Victor that she has a piece of information without giving this information away. Knowledge of this piece of information can prove an identity, membership of a group, privileges given by someone else to Victor.

Zero knowledge proofs in their basic form are interactive protocols. Victor asks Peggy a series of questions. If Peggy knows the secret, she can answer all the questions correctly. If she does not, she has some chance (50 percent in most implementations) of answering correctly. After a number of rounds, say more than ten, Victor will be convinced that Peggy knows the secret if all answers have been answered correctly. Still, Victor did not gain any information other than becoming convinced about Peggy's knowledge of the secret. For example, there exist implementations using graph isomorphism [**GMW86**] or hamiltonian cycles [**Blu86**]. Using graph isomorphism as an example, a simplified protocol will look like this: A graph is a network of lines (edges) connecting different points. If two graphs are identical except for the names of the points and edges, they are called *isomorphic*. For large graphs, finding out whether two graphs are isomorphic is a hard problem (in a computational complexity sense). Assume that Peggy knows the isomorphism between two graphs G_1 and G_2. The following protocol will convince Victor of Peggy's knowledge[7]:

(1) Peggy randomly permutes G_1 to produce another graph **H** that is isomorphic to G_1[8].

(2) Peggy sends **H** to Victor.

(3) Victor asks Peggy either to:
   - prove that **H** and G_1 are isomorphic
   - prove that **H** and G_2 are isomorphic

(4) Peggy complies and either:
   - proves that **H** and G_1 are isomorphic, without proving that **H** and G_2 are isomorphic
   - proves that **H** and G_2 are isomorphic, without proving that **H** and G_1 are isomorphic

(5) Peggy and Victor repeat steps (1) through (4) n times.

[7] Example taken from [**Sch96**], p.104

[8] Because Peggy knows the isomorphism between **H** and G_1, she also knows the isomorphism between **H** and G_2. For anyone else, finding an isomorphism between G_1 and **H** or between G_2 and **H** is as hard as finding an isomorphism between G_1 and G_2.

Non-Interactive Zero Knowledge Proofs

However, since dissemination of type-1 messages uses broadcast-style methods and a "proof" for included information's authenticity can only be given by a source node, not by relay nodes, interactive protocols cannot be used in the overall architecture[9]. Fortunately, non-interactive zero knowledge proofs are a variation of zero knowledge proofs that can be used in the desired way. A one-way hash function takes the place of Victor[10]:

(1) Peggy uses her information and n random numbers to transform the hard problem into n different isomorphic problems. She then uses her information and the random numbers to solve the n new hard problems.
(2) Peggy commits to the solution of the n new hard problems.
(3) Peggy uses all of these commitments together as a single input to a one-way hash function. She then saves the first n bits of the output of this one-way hash function.
(4) Peggy takes the n bits generated in step (3). For each i-th new hard problem in turn, she takes the i-th bit of those n bits and either
    - if it is a 0, she proves that the old and new problems are isomorphic
    - if it is a 1, she opens the solution she committed to in step (2) and proves that it is a solution to the new problem.
(5) Peggy publishes all the commitments from step (2) as well as the solutions in step (4).
(6) Victor or whoever else is interested, verifies that steps (1) through (5) were executed properly.

Amos Fiat and Adi Shamir presented an authentication and digital signature scheme [**FS87**] that was later modified to become one of the best known zero-knowledge proofs of identity [**FFS87**]. It is based on the factoring problem of large numbers, more specifically quadratic residues and employs a form of asymmetric keys.

4.3.3.2. A Modified Fiat Shamir Method. Coming back to the basic requirements given above, in section 4.3.3's first paragraph, it is also desirable to avoid certificates, due to their added message overhead. Discrete logs and quadratic residues can both be used as a basis for zero-knowledge proofs. With discrete logs, everyone can generate valid key-pairs[11]. The quadratic residue problem however, can be used in such a way, that key-pairs are derived from secret parameters, that are only known to an authority (thus being able to generate valid key-pairs) but are still verifyable by anyone having a public key. This is exactly the feature why the quadratic residue problem has been chosen for this work and the key generation process is the key extension to Feige-Fiat-Shamir scheme to meet the specific requirements of the presented overall architecture.

"Special" parameters, variables, operators and sets used in following paragraphs are shown in Table 4.1.

4.3.3.3. Key Generation. In the same way as in RSA, large prime numbers and a modular arithmetic are important. Here, every public and every private key has a number of k different key parameters which are used to calculate a signature. Those key parameters of a public and private key-pair correspond to each other according to equation 4.1. In order to reduce communication- and computational overhead,

[9] This will also be reflected in requirement 5.5, when discussing type-1 scenario in chapter 5.
[10] Example taken from [**Sch96**], pp.106-107
[11] See example in [**MOV96**]

s_i	Secret key parameters
$\bar{s}$	Secret key
v_i	Public key parameters
$\bar{v}$	Public key
W_i	Parameter vector
r_j	Random number
$\bar{x}$	"Witness" vector
$\bar{h}$	Hash vector
$keyID$	Key identifier
N	Module
p	Prime factor (of N)
q	Prime factor (of N)
$H(x)$	Hash value of x
$\|H(x)\|$	Length of hash value of x (in bit)
$\mathbb{QR}$	Set of quadratic residues
$\mathbb{Z}_N^*$	Cyclic group, modulus N

Table 4.1. Variables and Parameters for Algorithms

all simulated interactions will be parallelized using the hash of a specific key parameter subset. The key parameters v_i and s_i are therefore fundamental parts of the concept introduced here.

$$s_i \equiv \frac{1}{\sqrt{v_i}} \pmod{N} \qquad \forall i = 0, 1, \ldots, k-1 \tag{4.1}$$

In order to be able to distinguish between valid key-pairs that have been generated by organization O or authority A and invalid key-pairs generated elsewhere, the key generation process has to include parameters that are only known to authorized entities. Therefore key identifiers, denoted with *keyID*, will be used that lead to the generation of public keys, *pubKey*, using a hash function *H(x)* with a hash length of $|H(x)|$ bit. The goal of the key generation is an unambiguous assignment of a key identifier *keyID* to public key parameters v_i using a parameter vector $\bar{W}$ and a one-way function *H(x)*, see equations 4.2 and 4.3.

$$v_i = H(keyID, W_i) \qquad where \quad W_i \in N \wedge W_i << N \quad for \quad i = 0, 1, \ldots, k-1 \tag{4.2}$$

$$v_i \in QR \pmod{N} \qquad for \quad i = 0, 1, \ldots, k-1 \tag{4.3}$$

The parameters in $\bar{W}$ have to be chosen in such a way that *H(x)* calculates public key parameters v_i that are quadratic residues modulo N (QR (mod N)). If k different values for W_i are found, the public key can be generated according to equation 4.4.

$$pubKey = (keyID, W_0, W_1, \ldots, W_{k-1}) = (keyID, \bar{W}) \tag{4.4}$$

Note that the public key does not require a certificate, since every receiver can calculate the public key parameter vector:

$$\bar{v} = [v_0, v_1, \ldots, v_{k-1}] \tag{4.5}$$

from pubKey using equation 4.2 and they will only be correct if previously generated by an authorized entity. The reason for this is that only an authorized entity knowing the factors of module N could have chosen parameters W_i such a way that equation 4.3 is true. The length of the public key parameters is always $|H(x)|$ bit. The bit-length of parameters W_i has been chosen to be 16 bit, an evaluation leading to this can be found in [**Wim04**]. For every user to have its own private key parameters, an authorized entity calculates the square roots modulo N for every v_i. Due to the special properties of modular arithmetics[12] every quadratic residue $QR \pmod{N}$ has exactly four solutions of quadratic square root modulo N. In the case of a pseudonym the smallest value becomes the private key parameter s_i. In the case of a credential a slightly different approach is taken because the key-pair has to be encrypted for transmission. This will be described in appendix A. The private key consisting of k entries therefore is:

$$\bar{s} = [s_0, s_1, \ldots, s_{k-1}] \tag{4.6}$$

This finishes the generation of a key-pair and every user receives a pubKey and a privKey for his pseudonym (during personalization over a secure channel) and for every of his credentials received from this time on. The parameters v_i are calculated by every user on its own (see equation 4.2). Only after verifying:

$$s_i^2 \cdot v_1 \overset{!}{=} 1 \pmod{N} \qquad \forall i = 0, 1, \ldots, k-1 \tag{4.7}$$

the data will be used. This corresponds to the verification of a key pair using root certificates in a PKI. State of current research is that calculating the square root modulo N without knowing the prime factors p and q of N has the same complexity as factoring modulus N itself (e.g. [**Tie00**]).

4.3.3.4. Signature Generation. The signature generation of the modified scheme here is in principal the same as in the original Feige-Fiat-Shamir scheme [**FFS87**] based on equation 4.1. Modifications only affect parallelization[13] and using an identity to determine public key-parameters. For signature generation, the source node S's TRM first picks t different random numbers r_j:

$$r_j \in Z_N^* \qquad where \quad j = 0, 1, \ldots, t-1 \tag{4.8}$$

and calculates a vector of "witnesses" with t entries:

$$\bar{x} = [x_0, x_1, \ldots, x_{t-1}] \qquad where \quad x_j = r_j^2 \pmod{N} \quad for \quad j = 0, 1, \ldots, t-1 \tag{4.9}$$

In a next step, the TRM calculates a hash-value using hash function $H(x)$, which is used system-wide:

[12] There are special numbers, called Blum Integers, which have the property that the quadratic residue of a Blum Integer again is a quadratic residue.

[13] As explained before, zero knowledge proofs are interactive protocols. In order to get a non-interactive version, the single steps are parallelized and the verifier is simulated by using a one-way function.

$$\bar{h} = H(message, x_0, x_1, \ldots, x_{k-1}) \tag{4.10}$$

Then, the entries of a bit-matrix are filled using these values line by line, starting with the least significant bit h_0 of the hash value:

$$\bar{\bar{b}} = \left\{ \begin{array}{cccc} b_{00} & b_{01} & \ldots & b_{0(k-1)} \\ b_{10} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b_{(t-1)0} & \ldots & \ldots & b_{(t-1)(k-1)} \end{array} \right\} = \left\{ \begin{array}{cccc} h_0 & h_1 & \ldots & h_{k-1} \\ h_k & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{(t-1)*k} & \ldots & \ldots & b_{t*(k-1)} \end{array} \right\} \tag{4.11}$$

This matrix corresponds to the challenge of the interactive version. The final step is the calculation of the vector

$$\bar{y} = [y_0, y_1, \ldots, y_{t-1}] \qquad where \quad y_j = r_j \cdot \Pi_{m=0}^{k-1} s_m^{b_{jm}} \quad for \quad j = 0, 1, \ldots, t-1 \tag{4.12}$$

In other words, for every line in matrix $\bar{\bar{b}}$ only those s_i will be multiplied with the random number r_i, where the corresponding number in the matrix is not zero. The source node can now send a message with signature:

$$signed - message = \{message, \bar{\bar{b}}, \bar{y}, pubKey(I, \bar{p})\} \tag{4.13}$$

Every receiving node can verify the authenticity of the message, since it is impossible for others to generate such a signature without knowing the private key of the source node S. Additionally, a malicious source node S^* cannot generate a valid signature, since it cannot predetermine which hash-values will be calculated by the hash function $H(x)$ for given values x_i.

4.3.3.5. Signature Verification. In order to verify a signature, a receiver R calculates the vector $\bar{v}$ from the sender S's public key:

$$\begin{aligned} &\bar{v} = [v_0, v_1, \ldots, v_{k-1}] \qquad where \\ v_i = H(keyID, W_i) \quad and \quad W_i \in N \wedge W_i << N \quad &for \quad i = 0, 1, \ldots, k-1 \end{aligned} \tag{4.14}$$

Note that, as it has been discussed above, public keys are not signed, since only an authorized entity, knowing the primes p and q factoring the module N, can generate valid key parameters. The module N is stored on the TRM during "OP-Ready" phase using a secure channel, so that no other parties can inject forged key-pairs. Using the matrix from equation 4.11, the receiver can calculate vector $\bar{z}$.

$$\bar{z} = [z_0, z_1, \ldots, z_{t-1}] \qquad where \quad z_j = y_j^2 \cdot \Pi_{m=0}^{k-1} v_m^{b_{jm}} \quad \forall j = 0, 1, \ldots, t-1 \tag{4.15}$$

Since following is true

$$\begin{aligned} z_j = y_j^2 * \Pi_{m=0}^{k-1} v_m^{b_{jm}} &= (r_j * \Pi_{m=0}^{k-1} s_m^{b_{jm}})^2 * \Pi_{m=0}^{k-1} v_m^{b_{jm}} (see\ eqn.\ 4.12) \\ &= r_j^2 * \Pi_{m=0}^{k-1} (v_m * s_m^2)^{b_{jm}} = r_j^2 * 1 (see\ eqn.\ 4.1) \\ &= r_j^2 \equiv x_j \pmod{N} \quad \forall j = 0, 1, \ldots, t-1 (see\ eqn.\ 4.9) \end{aligned}$$

the vectors $\bar{z}$ and $\bar{x}$ must be identical if the message has not been altered. Since the vector $\bar{x}$ itself is not known, a matrix $\bar{\bar{c}}$ (analog to matrix $\bar{\bar{b}}$ from equation 4.11) is generated by calculating the hash values $h' = H(m, \bar{z})$ using the hash function $H(x)$. If

$$\bar{\bar{c}} \overset{!}{=} \bar{\bar{b}} \tag{4.16}$$

is true, the message has not been altered and was signed by an authorized source node.

In order to be able to sign messages of varying length, a hash-value will be calculated using a cryptographic hash function and then the hash-value will be signed instead of the message. Note that this is true for almost all other signature algorithms as well.

It should also be pointed out, that the pseudonym credentials have to be transferred to the TRM during OP-Ready phase using a secure channel, because the private key parameters must not be seen by other parties. Once the pseudonym credentials from authority A are installed on the TRM, they can be used to establish encryption / decryption to transmit additional subscription credentials over open channels.

4.3.3.6. Key Encryption. The encryption and decryption algorithms are elaborated in appendix A.

4.3.4. Conclusion

In this section a privacy architecture has been introduced and a non-interactive zero knowledge scheme has been modified to meet the requirements of this privacy architecture as part of the overall SARI architecture presented in this work. A prototypical implementation has been realized on smart cards with a JavaCard operating system, which will be presented in chapter 8 together with results of tests and measurements. Regarding the usage of pseudonyms it has to be noted that the intended mode of operation is to switch pseudonyms at least each time a car is started. However, if location profiling is to be avoided, they have to be changed while driving. This is not problematic, as long as the respective car does not have to send out messages frequently and there is a sufficiently large number of other senders. Otherwise, during situations with low traffic density, it would be easy to correlate the old and the new pseudonym, thus rendering this activity useless. Summed up, it must be ensured that a pseudonym change is effective and old & new pseudonym cannot be correlated easily.

4.4. SARI Building Blocks

In this section, main building blocks and concepts of SARI that are fundamental for the general architecture will be introduced and explained.

4.4.1. Trust Establishment

Trust, according to RFC 2828 [**Shi00**] is defined as follows:

Definition 18 (Trust)**.** *"Generally, an entity can be said to 'trust' a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects, that is it does what it claims to do and does not perform unwanted functions."*

Trust is an essential foundation for all security measures, in personal life as well as for computer systems. In information security there are typically three phases that are related to trust:

(1) **Trust Establishment** The process of "assigning" trust to an entity, such as a car.

(2) **Trust Verification** The process of verifying own or third party trust statements[14] in response to an operation.

(3) **Trust Revocation** The process of nullifying or decreasing trust classifications of an entity.

Although this section is named trust establishment, because trust establishment is the most interesting part in this context, technically it would be more appropriate to refer to trust management, because this includes the fact that trust may also be revoked if a component or node does not behave as expected during the lifecycle, which has been discussed in section 4.2.3.

Definition 19 (Trust Management). *"Trust management integrates different trust establishment schemes across multiple administrative boundaries into a single management framework."*

This definition includes the fact that usually a single trust establishment scheme is not sufficient in complex systems and ultimately the management of multiple schemes is required. This is also the case in this work, where different approaches have to be taken for different usage profiles.

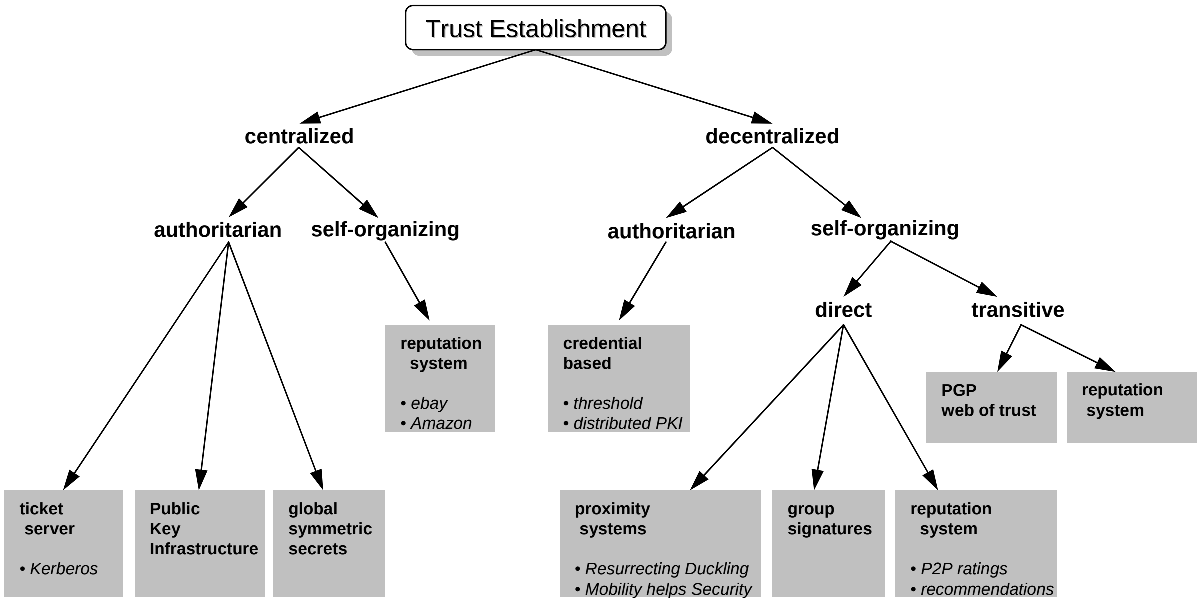


Figure 4.6. Classification of Existing Trust Establishment Schemes

Figure 4.6 presents an overview on different trust establishment schemes. They can be differentiated according to the question whether the trust is established and trust information is stored in a decentralized way or trust records are aggregated at specific entities.

Another distinction can be made with respect to the existence of an authority that acts as a trusted third party. The type of trust relations is yet another point to look at.

Most trust models only consider direct trust statements, but some systems, such as PGP (Pretty Good Privacy) or some kinds of reputation systems, use transitive relations. In a self-organizing model this exploits the linkage between different enities assuming that the majority is trustworthy and no entitiy can easily pretend to be many other entities[15].

Additionally, it is crucial to understand how the trust records are updated and looked up. They can either be verified by contacting a central authority upon request, that is there is a close temporal relation

[14] that have been established before

[15] This attack is called Sybil Attack and will be discussed later.

between a request and the communication with a trusted third party. If some form of credentials are used, they may either be valid until they are explicitly revoked or they may have a validity period. As a response to inevitable attacks on credential, there may exist different update/revocation mechanisms. Either the updates are triggered on-demand, or they are executed proactively, e.g. on a periodic basis.

In the following paragraphs, an overview on existing approaches for distributed systems is given and SARI's overall trust model is introduced. The different application types, as defined in section 2.4.2, have very different requirements regarding the trust model and will therefore be discussed independently in chapters 5, 6, 7.

4.4.1.1. State-of-the-Art. Some work has been done on trust establishment in ad-hoc networks. Matt Blaze et al. discuss the specifics of trust establishment in distributed system in [**BFIK99**] and refer to earlier work by two of the authors on decentralized trust management in [**BFL96**]. Asokan and Ginzboorg [**AG00**] mention the problem of key-agreement in ad-hoc networks with *"little or no support infrastructure"* and examine various alternatives. Laurent Eschenauer, Virgil Gligor and John Barras [**EGB02**] provide a work that investigates the problems of trust establishment in mobile ad-hoc networks. A follow-up work from Rakesh Bobba et al. [**BEGA03**] introduces an approach to bootstrap security associations for MANETs as a foundation for routing security. Srdjan Capkun, Levente Buttyan and Jean-Pierre Hubaux propose a self-organized public-key management system for MANETs in [**CBH02**]. Another work of the same authors [**CHB03**] discusses how the mobility property of MANETs can be exploited to support security. Another prominent paper from Frank Stajano and Ross Anderson [**SA99**] introduces a new scheme to establish trust in distributed systems relying on a secure channel for initialization. Jean-Pierre Hubaux provided a good overview on open questions in ad-hoc network security during his keynote speech at the first European Workshop on Security in Ad-Hoc and Sensor Networks (SASN'04) [**Hub04**]. Brian Parno and Adrian Perrig gave another overview, but focused on VANETs in [**PP05**].

4.4.1.2. SARI Trust Model. The conclusion from the state-of-the-art analysis in the previous paragraph and the multi-faceted vehicle environment is that a hybrid approach will be necessary, providing centralized and decentralized trust establishment methods as required. Figure 4.7 shows a trust graph representing SARI's trust model. The edges represent directed trust relations.

The Graph shows that the driver / owner of a car cannot be trusted by any other entity, since he may or may not behave cooperative. The driver itself however trusts the car manufacturer for the proper functioning of the system built into his car. But he may not trust the car manufacturer to respect his privacy in all aspects. This is the point when a privacy granting trusted third party comes in. A great deal of SARI's security and privacy depend on a tamper resistant module. Therefore the tamper resistant module manufacturer is trusted to provide technology to prevent tampering. The vehicle manufacturer has to trust service providers with trustworthy service provision. Service providers in turn rely on their trust centers in issuing, maintaining and revoking credentials properly.

Note that this covers a basic level of trust, addressing requirement 3.6 *node authenticity*, but it does not meet requirement 3.5 *information authenticity*. This will be elaborated in chapter 5.

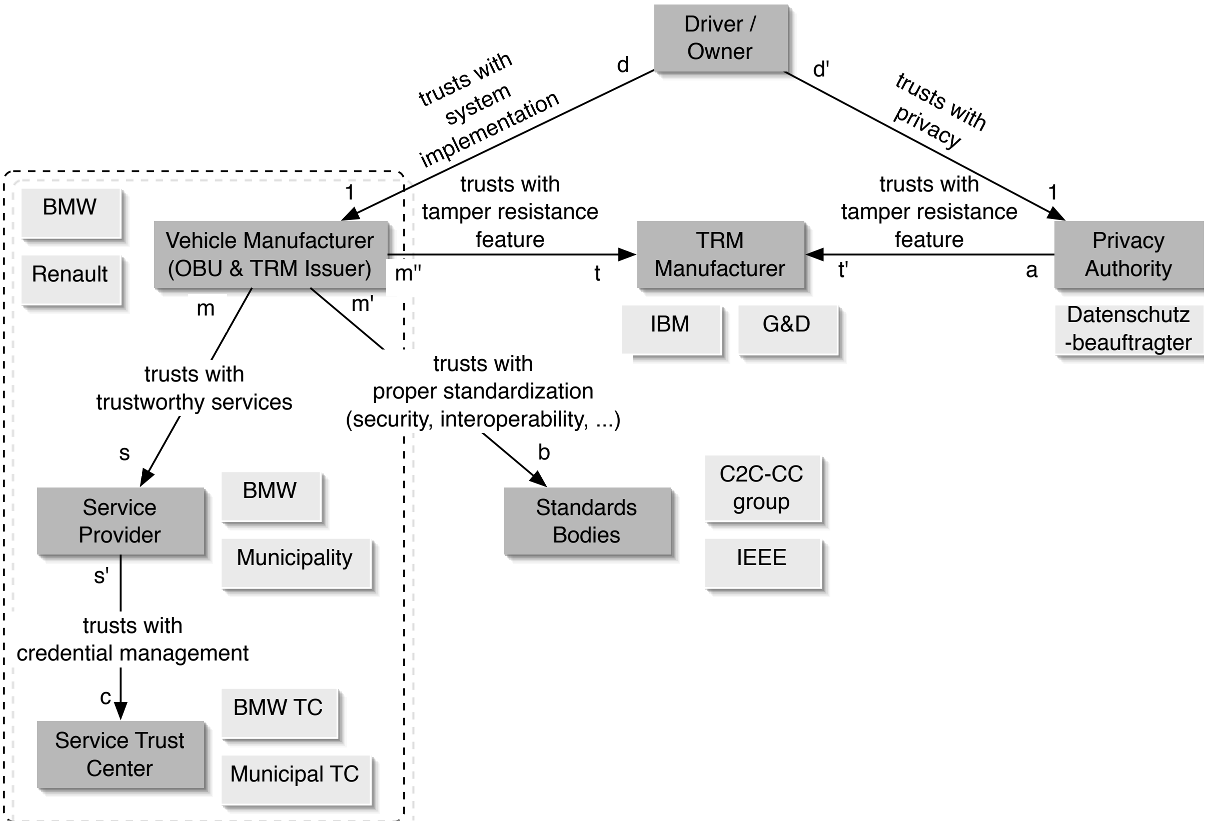

Driver / Owner
d
d'
trusts with system implementation
trusts with privacy
1
1
BMW
Renault
Vehicle Manufacturer (OBU & TRM Issuer)
trusts with tamper resistance feature
m''
t
TRM Manufacturer
trusts with tamper resistance feature
t'
a
Privacy Authority
Datenschutz -beauftragter
IBM
G&D
m
m'
trusts with trustworthy services
trusts with proper standardization (security, interoperability, ...)
s
b
Service Provider
BMW
Municipality
Standards Bodies
C2C-CC group
IEEE
s'
trusts with credential management
c
Service Trust Center
BMW TC
Municipal TC

Figure 4.7. SARI Trust Model

4.4.2. Tamper Resistant Module

In second assumption in section 3.1.1 is the presence of a tamper resistant module. Such devices exist in a variety of form factors ranging from smart cards to tokens. Their key feature is their resistance against physical attacks, hence the name. The countermeasures against different kinds of attacks have been improved over the years and reached a considerable level, see [**RBP$^+$04, KK99**] for an in-depth discussion of this topic and related attacks.

A tamper resistant module provides a secure computing platform that is used to store data, execute algorithms and perform cryptographic operations. The usage of tamper resistant modules represents a fundamental part of SARI's security concept because it assumes that some of the algorithms cannot be manipulated or biased during execution. More specifically, SARI requires following features in addition to tamper resistance:

- **Sandboxing** Applications[16] cannot interfere with each other.
- **Lifecycle Management** The tamper resistant device is a state machine, each state only allowing distinct actions or execution of applications.
- **Secure Installation** Applications can only be installed if certified.

4.4.2.1. Multiapplication Smart Cards. One form factor of tamper resistant modules are smart cards. The most powerful versions are able to run multiple applications and provide a security coprocessor for efficient execution of cryptographic algorithms. The operating system is usually verified code that offers a high degree of security with respect to vulnerabilites in the development process. Access to

[16] Note that in this case application refers to programs running on a smart card, not actually to a fully-fledged application.

cryptographic operations is provided by API's. If additional cryptographic operations require direct access to the cryptographic coprocessor, it is possible to write a new interface. These features and the flexibility of smart cards made them the choice #1 for this thesis.

All smart cards in the field will have to be managed with respect to operating system versions, applications running on them, access privileges for various entities, etc. For these tasks exist smart card management systems with Global Platform [**Pla01**] being the most prominent one. Global Platform fulfills all requirements of the given vehicle scenario, therefore it will be regarded as reference with respect to smart card management in this thesis.

4.4.2.2. Tamper Resistant Module Imprinting. The use of tamper resistant modules as a way to improve security in IVC systems bears the threat that the module can be removed and used elsewhere to mount an attack. On the other hand, the removal and/or replacement of system compononents such as sensors may also have significant effects on the system behavior, thus affecting the security. In the context of this thesis, a combination of two different concepts will be used to approach these problems.

Weimerskirch, Paar and Wolf suggested a scheme to cryptographically identify system components in a car [**WPW05**]. They require a RFID chip to be mounted on every component in such a way that it cannot be removed. Their goals are to prevent manipulation of components or replacement with aftermarket products that have not been certified by the manufacturer and to securely derive electronice license plate data from a specific system configuration. Today, it is unreasonable to require all hardware-components to be equipped with smart tags, due to cost and power supply constraints. However, the idea of integrating smart cards or similar devices as part of engine control ECUs in engine blocks in such a way that they cannot be removed without destruction has been around for some time. This indicates that some components will be equipped with non-removable and tamper resistant devices in the near future, which can be used for cross-identification using a scheme such as the one proposed by Weimerskirch et al. [**WPW05**]. For all other ECUs and sensors with computing capabilities, external verification of their memory contents by measuring hardware-specific parameters, such as proposed by Seshadri et al. [**SPvDK04**] will be performed. This technology allows to notice whenever the configuration of embedded devices change, also known as "startup procedure with secure registration". An attacker would therefore have to install an exact copy of the component, the same hardware with exactly the same code. This of course, will render the attack itself useless. The combination of mutual cross-identification among non-removable devices and remote code verification of embedded devices is called imprinting and allows SARI's tamper resistant device to verify the configuration it is operating in.

The whole component imprinting process will be conducted during production, where a controlled and secure environment can be guaranteed. If any updates or changes are made, the imprinting process has to be conducted again. For practical reasons, this will be realized in a maintenance phase (see section 4.2.3 about the lifecycle) at certified dealers' premises.

An imprinting process performed this way significantly raises the bar for attackers to remove or replace a SARI-tamperresistant device.

4.4.3. Secure Positioning

In order for SARI to work, exact positions have to be known. In addition to that, sent information about positions has to be verifiable by receiving nodes. In this section, the concepts of satellite positioning, being the main positioning system this thesis relies on, will be explained. After that, its shortcomings with respect to information security will be pointed out and state-of-the-art countermeasures are given. Then, different approaches to overcome those shortcomings will be presented depending on the attacker model.

4.4.3.1. Satellite Positioning Systems. All currently working or future satellite positioning systems, such as the Global Positioning System (GPS), the GLObal'naya NAvigatsionannaya Sputnikovaya Sistema (GLONASS) or the GALILEO System are based on the same idea: A satellite positioning system consists of a number of satellites that are orbiting the earth. At any given time, the exact position of the satellites can be calculated from a database. Every single satellite has a very precise clock, that is synchronized with all other satellites' clocks. The satellites frequently send out messages including their identifier and the time the message has been sent. A navigation device on earth that receives those signals and knows the exact positions of the satellites can deduce its own position by calculating the distances to the satellites whose messages it received. In theory, receiving messages from at least three satellites would be enough, assuming that the navigation device's internal clock and those of the satellites are perfectly in sync. Since this would require very expensive equipment, in practice a fourth satellite is used to deduce the exact time. See Figure 4.8 for a schematic representation.

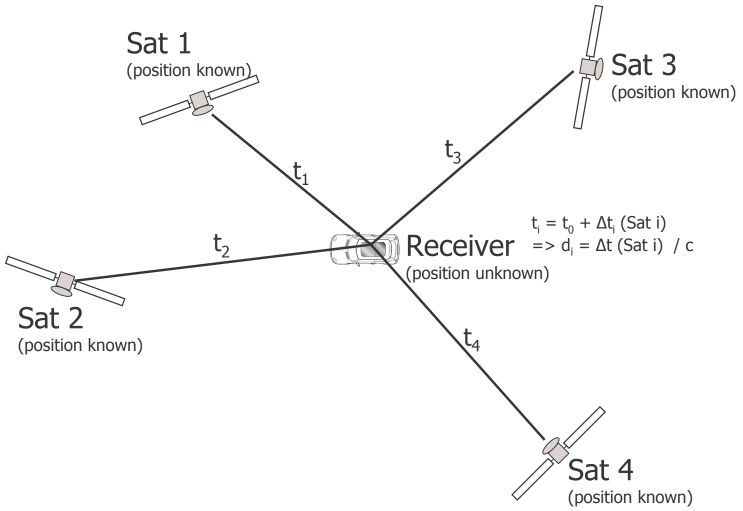
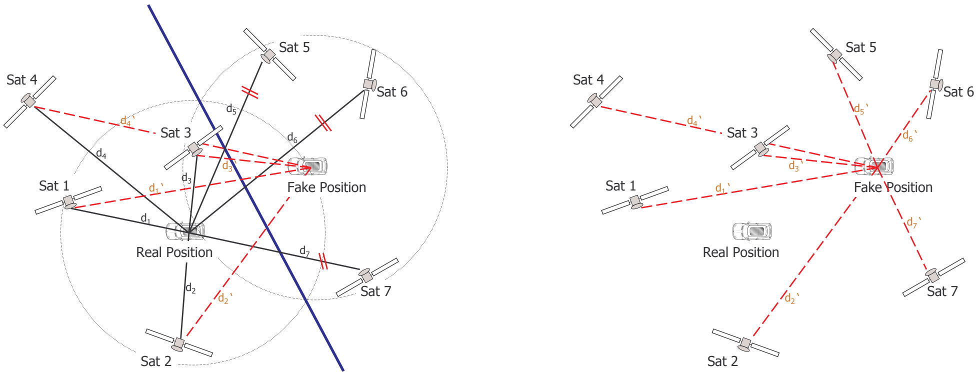


Figure 4.8. Satellite Positioning Principle

4.4.3.2. Satellite Positioning Information Security. *Signal Jamming* The easiest way to affect satellite positioning is to jam the signals. Johnston and Warner pointed out in [**JW04**] that a jammer that works over an area of several hundred square miles can be built for under US$50. Although in section 3.3.5 it was pointed out that jamming in general is out of scope of this work and must be countered by organizational measures, the positioning system may be supported by additional components that help to overcome the effects of jamming, especially in the contect of moving cars where the effects are temporarily limited.

Signal Spoofing Since the satellites' signal have a known format and the contents are known, it is possible to generate these signals. If a receiver is fed with self generated signals, arbitrary positions can be spoofed. There even exists equipment that does exactly that, being used by navigation device developers. The obvious countermeasure is to provide an unpredictable signal derived from the original signal. This is actually done in the military version of GPS, where the signal is multiplied with a pseudo-random spreading sequence. This encrypts the signal similar to a stream cipher and spreads the signal such that its power-spectral density ends below the termal noise density.

Timing Delay Attack But even if the signal is protected like in the case before it can be manipulated. If an attacker can suppress the original signal and replay a delayed copy of the original signal, he can emulate arbitrary positions by replaying the signals with exactly calculated delays.



(a) Simple Timing Delay

(b) Advanced Timing Delay

Figure 4.9. Attacks on Satellite Positioning

The simple notion of the attack is that all signals coming from satellites that are closer to the fake position (Sat 5-7) are jammed / suppressed (see Figure 4.9(a)). The other signals are delayed so that the fake position will be calculated. However, there is obviously one problem. Since the distances to the satellites' known positions can only be enlarged, a clever receiver might notice that all satellites he receives are behind a certain border plane in space (the blue line in Figure 4.9(a)). Also, if there are not enough satellites that fulfill these requirements a fake position cannot be simulated.

But remember that one satellite's signal is used as time reference. If this signal is delayed, so will the overall time perception at the receiver. This finally makes it possible to shift all satellite signals in such a way that they are interpreted as representing an arbitrary fake position (see Figure 4.9(b)).

There are three effective countermeasures to these attacks:

(1) A highly accurate secure timer that is synchronized with the satellites' time-reference in a secure way would deny attackers the ability to shift the time-reference. This would significantly limit the gain of an attack, especially if combined with other measures. Even if a less accurate internal timer is used, this would still prevent ”position hops” and reduce the problem to slow ”position drifts”.

(2) The attack can only be executed if the signals from each satellite are properly isolated. The application of a pseudo-random spreading sequence that is unknown to the attackers leaves directional antennas the only realistic option. However, in practice it is quite difficult to filter out a relatively weak signal from a fast moving satellite and requires sophisticated equipment.

(3) In general, if other reliable and trustworthy sources of positioning data are available, they can be used to verify and/or update the actual position.

The conclusion is that it is highly desirable to have an internal timer inside the tamper seal that provides an independent time reference. And additionally, that a secured satellite positioning signal is required to prevent spoofing.

4.4.3.3. Supportive Positioning Concepts. The primary source of positioning data is satellite positioning. However, other means of getting reliable positioning data may add enough input to create a sufficient basis for positioning even in absence or in suspicion of manipulated satellite positioning data. In many cases this supportive information does not have to be very precise, it is more a matter of whether the calculated high-precision position is reasonable or not.

Certified RSUs

RSUs are immobile devices that are maintained regularly. The owners / operators of RSUs are in many cases known and trustworthy entities. This leads to the straightforward approach of measuring the exact position of RSUs. This information can then be sent out periodically or upon request. If a receiving node can measure the distance / relative position to such a RSU, by measuring signal strength or message roundtrips it can use this information to recalibrate its internal position information. The authenticity and integrity of such information has of course to be guaranteed, e.g. by applying digital signatures. If the RSU is implemented in a closed, tamper-resistant enclosure, maintained regularly, surveilled remotely and/or secured by other means, this will provide a reasonably secure source of positioning information.

Multinode Triangulation

Another concept that is applicable to high node-density scenarios is to calculate the local position by triangulating positioning information from surrounding nodes. For instance Zangl and Hagenauer presented a way of calculating positions in a ad-hoc sensor network in [**ZH01**].

Local Reasoning

A local reasoning system that will be introduced in the following chapter 5, can also be used improve robustness of a positioning system. On one hand it will improve availability by providing extrapolations in the case of missing positioning data input. On the other hand it can seriously improve the robustness against attacks by monitoring positioning relevant data and looking for unusual patterns. Additional data may come from wheel sensors, engine control data, gearbox, etc. This data and existing positioning data (ideally from multiple sources) can be interpreted to find the most likely actual position. Additionally, a movement history[17] provides useful input for the reasoning process. For instance, jumps in the movement graph may indicate irregularities in the positioning system. In a similar way, timer glitches may be detected.

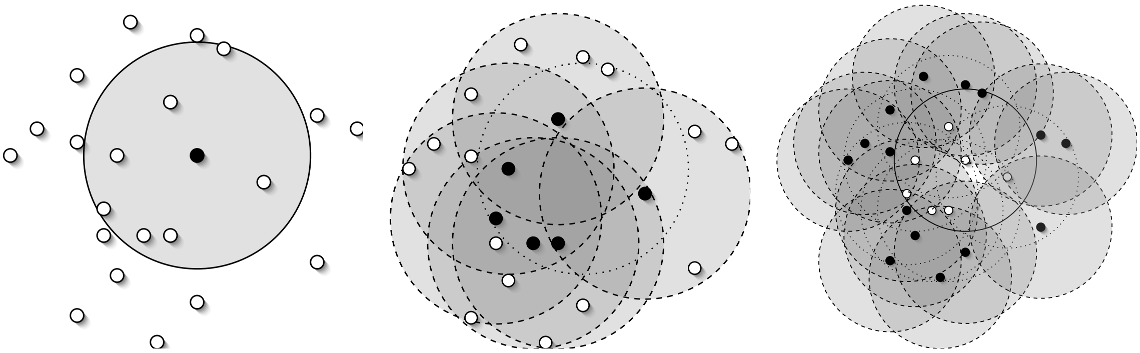
4.4.4. Efficient Message Distribution

Robust and scalable inter-vehicle communications require an efficient way of distributing messages. Bandwidth limitations, denial of service attacks and a growing number of nodes demand an intelligent distribution mechanism. Additionally, the availability of a large number of messages from different parts of the network will support the evaluation and decision mechanisms of type-1 applications. The so-called broadcast storm problem is a phenomenon in this category, where a flexible approach will be provided to address this issue. If the number of messages sent exceeds the network's capacity, messages will be

[17] A movement history will be stored anyway for multiple reasons: identification of traffic scenarios, detection of specific traffic situations such as *evasion*, etc.

priorized accordingly. First, the application type and their priority factor will be considered. Second, the messages will be sorted according to the sending node's behavior, see section 4.4.4.3.

4.4.4.1. Avoiding the Broadcast Storm. In areas with a high node density an effect called broadcast storm occurs if broadcast transmissions are used. A broadcast storm is a networking situation in which messages are broadcast on a network, and each message prompts a receiving node to respond by broadcasting its own messages on the network that in turn prompt further responses, and so on (see Fig 4.10). There are two problems associated with this effect. First, the number of re-transmissions will dramatically increase, due to the high density of nodes in the given area. Second, the number of collisions on the wireless channel will increase, due to a large number of nodes trying to send at the same time. The hidden node problem also adds to the collision issue.



(a) Phase I (b) Phase II (c) Phase III

Figure 4.10. Broadcast Storm

Kosch proposes an approach to solve the broadcast storm problem in [**Kos04a**] by limiting the re-transmission to those nodes that are in an ideal position. However, this requires a digital map in addition to a positioning system. Füßler et al. introduced the concept of contention-based forwarding along a given direction in [**FWK$^+$03**]. They also evaluated their approach according to its usefulness in street scenarios [**FHW$^+$04**]. The idea is that the messages are sent out by the node closest to the destination and this is achieved by calculating back-off times that increase with the proximity of the respective node to the sender. However, this scheme has been designed for routed / geo-cast situations, where a one-dimensional vector between source and destination exists.

4.4.4.2. Re-transmission Optimization. In a generic message dissemination scenario an efficient scheme working in two dimensions without using digital maps or neighborhood information is required. There are three fundamental approaches, each having a distinct disadvantage:

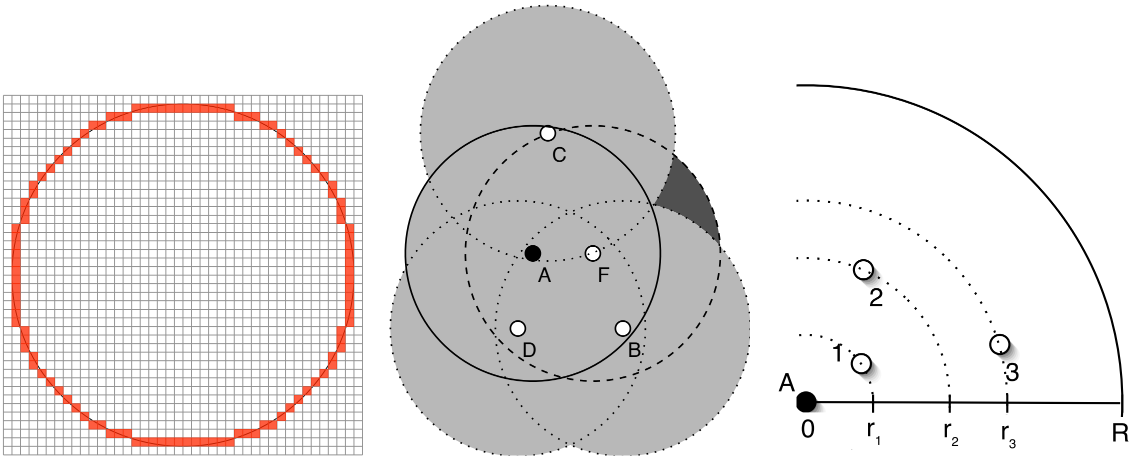
- **Flooding** The simple approach, large number of re-transmissions and collisions.
- **Probabilistic scheme** Frames are re-transmitted with probability p. p can also be changed according to the number of collisions or the traffic load. The disadvantage is that this scheme does not achieve a good area coverage in a network with mobile nodes, since nodes that may be ideally situated for message distribution may decide not to forward the information.
- **Counter-based scheme** A node decides upon re-transmission based on the number of times it has already received the same frame. This scheme has the same disadvantage as the probabilistic scheme, that the nodes within the given distribution area will not be reached in an optimal way.

According to assumption 1 (section 3.1.1) each node knows its position and will include it in its messages. This additional information will be used to calculate the distance to the sender and the relative position to other nodes in the vicinity. The scheme that is proposed here combines two elements.

Virgin-area

The first element reduces the number of re-transmissions by deciding whether a transmission would significantly increase the distribution area of a message / frame[18]. Note that in order to calculate this in the right way, it is crucial to know the communication range of each car. This will either be a preset value, be included in the headers or calculated from communication parameters.

The area around the node is divided in small cells. Then the algorithm determines which cells will be covered by a new re-transmission of the message using the Bresenham algorithm to draw the circle on the grid, see Figure 4.11(a). The next step is to determine what area the previous senders have already covered. The information about the previous forwarders of that message is extracted from the database and the cells of the grid being covered are marked. Only the data about vehicles that have transmitted during a small period T_r is computed, because the network topology changes rapidly and it is expected that the vehicles have changed their positions. The value of T_r will be a fixed value here, but it could also be calculated based on the speed of vehicles. If the percentage of additional coverage (see the dark shaded area in Figure 4.11(b), called the virgin area) by re-transmitting the message exceeds a preset value, re-transmission is initiated. The algorithm achieves a good approximation, is simple and has shown in simulations that it works well. The accuracy of the appoximation depends on the size of the cells, in simulations the cell's sides measured two meters.



(a) Bresenham circle (b) Virgin Area (c) Distance-dependent Backoff

Figure 4.11. Re-transmission Optimization

Distance-dependent Backoff

The second element reduces the probability of collisions by introducing a backoff time that decreases with the distance from the closest sender, see Figure 4.11(c). The formula for the backoff time is: $t_{backoff} = T + (r - d) * f$ where T is a grace backoff used for processing data, r is the communication range of the sender, f is a factor to weigh the backoff-time with respect to the communication range and d is the distance from the sender to the receiver, calculated using the Pythagorean theorem:

[18] It is assumed here that all messages fit into a single frame, therefore both terms can be used synonymously here.

$d = \sqrt{(x_{receiver} - x_{sender})^2 + (y_{receiver} - y_{sender})^2}$ where x and y are the coordinates of the sender or receiver.

Details on the implementation of these two elements can be found in [**Van06**].

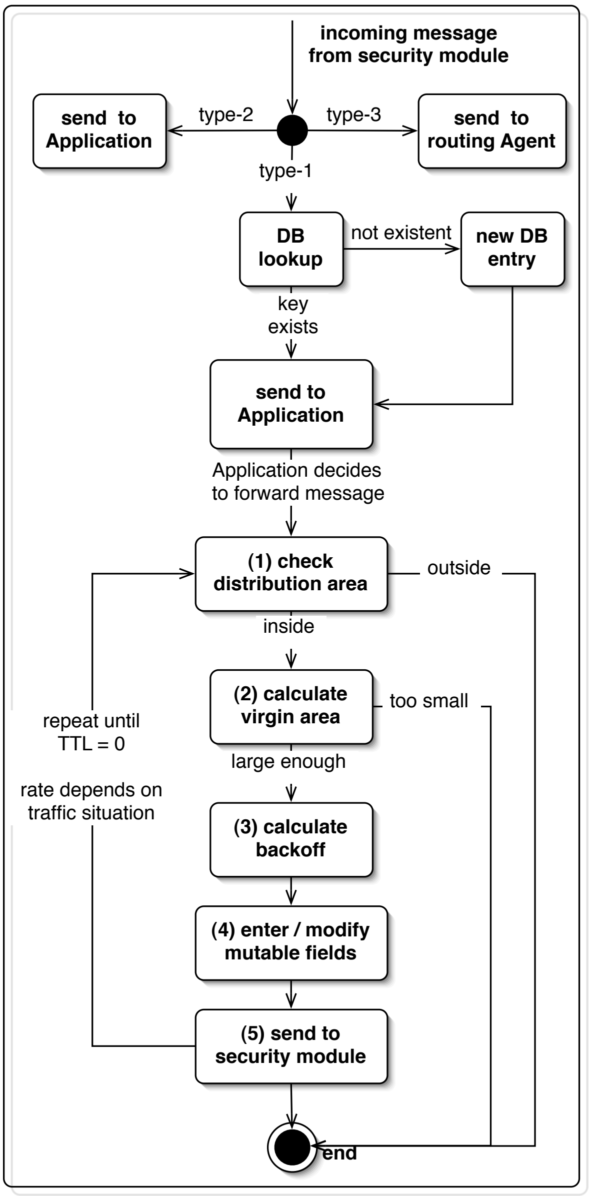


Figure 4.12. Flowchart Message Received

Message Forwarding Process

When a message has been received and passed the security module, the distribution module will take over the message.

From the security module, it is already known whether the message is of type-1, type-2 or type-3. Type-2 messages are directly sent to the **application module**. Type-1 and type-3 are sent to a broadcast agent and a routing agent respectively. The routing agent will be described in detail in chapter 7, a conceptual introduction to the broadcast agent will be presented in the following paragraph. Figure 4.12 shows an UML (Unified Markup Language) action diagram of a typical message forwarding processing in the distribution module's broadcast agent.

If a message reaches the broadcast agent, first a unique identifier of the event contained in the message is derived from its position and time. The database is searched for an existing entry with the same identifier. If found, the position of the relay node, its communication range and the current time are added. Otherwise, a new entry in the database is generated. The message contents are then forwarded to the application module[19].

After a message has been processed by the application module and has been rated as relevant for forwarding, the broadcast agent will perform following logical steps:

(1) **check distribution area**
The distribution area given by the application is compared to the actual position, if inside, the message is passed on, otherwise it will be kept in the database.

(2) **calculate virgin area**
The virgin area, the additional area that will be covered, is calculated according to the algorithm explained above. If it is sufficiently large, the message will be passed on.

(3) **calculate backoff**
The backoff time is calculated depending on the distance to the closest forwarder minus estimated remaining processing (e.g. security module) time so that the message will actually be transmitted at the given time.

(4) **enter / modify mutable header fields**
The mutable header fields have to be updated with current position, time, etc.

(5) **send to security module**
Finally the message will be sent to the security module and then for final transmission. This will be repeated periodically until Time-To-Live (TTL) equals zero, it will then be removed from the database. The rate of sending out the message is one of a set of preset values depending on the traffic scenario.

4.4.4.3. Application Priorization. In an IVC system there are messages of different importance and urgency. Traffic-related messages, especially those affecting road safety, for instance must be delivered with a higher efficiency than messages for personal entertainment. There are cases mostly in the *city* traffic scenario (see discussion on traffic and communication environment categories in section 3.1.5), where the communication channel is saturated[20] for a longer period of time in some areas. Nodes in that area have to make sure that their outgoing messages are ordered according to their priorities. Consequently, in addition to the schemes presented in paragraph 4.4.4.2 a way of priorizing application messages is introduced in this paragraph. This will enable an efficient handling of critical traffic-related messages on a best-effort basis. This scheme can also be used together with the reputation system presented in section 5.5.2 to allow gratifications for cooperative nodes.

All nodes maintain queues for outgoing messages, to be able to send the messages in case of an established connection. The queue is normally organized as FIFO (First-in first-out). This means messages that have been processed by SARI and are scheduled to send will be stored attached to the end of this send-queue. If the wireless channel is available, the wireless module will start to send the first

[19] More exactly, the contents are entered into the database and the application module is notified about a new message.

[20] Saturated channel in this context means that it cannot accommodate all messages that are scheduled for transmission, consequently resulting in the omission of a fraction of the messages and requiring each node to maintain a send-queue for the messages it wants to send.

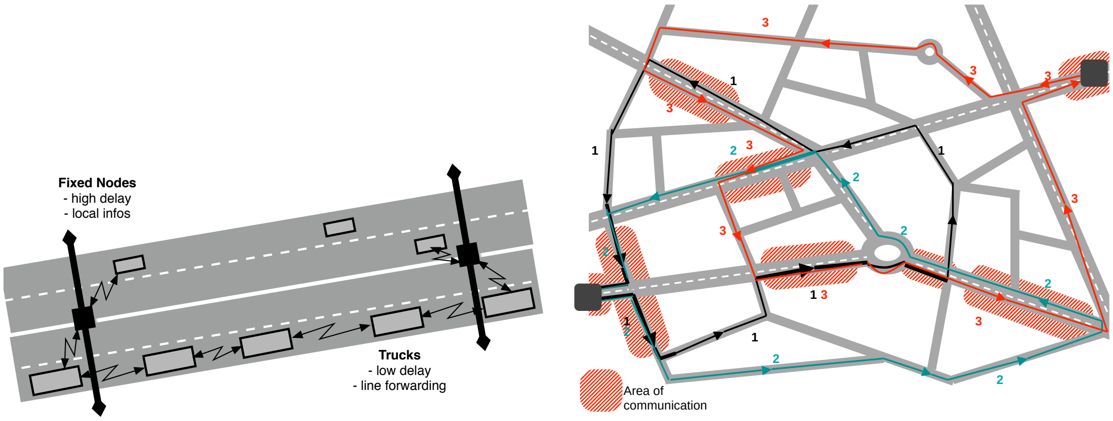
message in the send-queue, then the second and so on. Certainly, in those cases where messages cannot be sent out immediately due to a filled send-queue, this will result in additional delay at respective nodes. The approach followed in this thesis is to consistently re-arrange the order of messages stored in that queue according to priorities calculated from application type, confidence level, etc. By application of this method, the round-trip delays will be significantly reduced and, more importantly, the probability of important traffic messages coming through will be increased. A more detailed view and simulative evaluation will follow in chapter 7, where cooperation issues in connection oriented communication are discussed.

4.4.4.4. Neighborhood Discovery in Traffic Scenario ”Rural”. Assumption 8 in section 4.2.1 requires to use beacons restrictively. This leaves the question open what happens in the rural traffic scenario. Nodes will send out their messages or beacons periodically. The period must not be too long, because it will increase the average time until a response from an encountered car is received. In head-on situations with cars travelling up to 250km/h (resulting in relative speeds of up to 500km/h 1̃50m/s) there are usually no more than 4s being within each other’s reception. Due to protocol overhead, as a rule of thumb, the period must not be longer than about 500ms. On the other hand, sending out messages/beacons too often will increase the risk of interference, once another car is in range. Sending out long and/or multiple messages will additionally increase the probability of interference. Using short beacons as a ”there’s somebody here” notifier[21] can solve this issue.

However, this problem can be mitigated by adjusting the transmission period according to the traffic scenario and speed.

4.4.5. Virtual Backbone

Mobile nodes that move in specific, predictable patterns are ideally suited to form a virtual backbone and support distribution of information. Fixed nodes may also have positive effects, storing information and distributing it to opposing traffic. See Figure 4.13.



(a) Trucks and Fixed Nodes

(b) Public Transportation

Figure 4.13. Virtual Backbone

4.4.5.1. Trucks. Trucks moving on highways are usually moving at a constant speed or at least at some speed almost similar to each other for example if they move uphill. They rarely overtake each

[21] Note that for this purpose the beacons do not need to carry other information than time and wireless channel parameters.

other and even if they do so, this is happening relatively slow in terms of a communication system. These properties make them ideal for the establishment of a backbone network, with a stable network topology. Maybe in the distant future this could even result in an business model for trucks to provide high-speed communications.

4.4.5.2. Public Transportation. Public transportation vehicles moving on the surface, such as buses and trams (one might also consider taxis for some scenarios) are useful for distribution of data in large areas. They may not be able to provide constant connectivity as in the truck example, but they can provide physical transport of data. Their great advantage is that they cover a large area and are distributed properly (according to a timetable). They may also be used to connect separate parts of a fragmented network. Especially for message based applications that accept large latency, such as mail services, distribution of software patches and upgrades, information about events / offers / weather, the physical transport is an useful option.

4.4.5.3. Fixed Nodes. A special case is an IVC communication device that is mounted at a fixed location instead of a vehicle. Technologically, this is identical to a standing car, except that it remains at this position for an infinite time. Such devices could serve as important information hubs when placed at critical locations such as large intersections, places with frequent road hazards, etc. In addition to that, RSUs can also log traffic density and simple movement patterns to calculate long- and midterm traffic statistics. While the term "backbone" seems inappropriate for single devices, it makes sense if one thinks about placing such devices at periodic distances alongside a highway[22]. Cars travelling between those bridges could deliver information about the section that they passed. This information could in turn be used by cars approaching on the opposite direction, travelling to this section.

4.5. Conclusion

This chapter introduces a general system architecture called SARI and its functional structure. An important detail is the use of tamper resistant technology to encapsulate critical components and provide a higher degree of security. This allows the definition of distinct operational states, with different access and control options, which in turn supports a secure but expandable application. Privacy issues are major user requirements which are met by introduction of a privacy architecture. A concept using trusted third parties addresses both, the desire for pseudonymity on one hand and the wish of being able to identify misbehaving nodes on the other hand. Modified zero knowledge proof protocols allow the use of digital credentials and signatures, while preserving pseudonymity. These methods allow a high degree of privacy while providing a high level of security by exploiting properties of the automotive environment such as availability of (relatively) large memory capacity. In section 4.4 SARI Building Blocks building blocks are introduced that are fundamental for a realization of the general architecture. They represent core functionalities that would otherwise not be available or are of special imporance in this context. Following chapters will use this functionality and refer to the sections accordingly.

[22] The German system "Toll Collect" for instance uses toll bridges that would be well suited for that.

CHAPTER 5

Vehicle to Vehicle Scenario

This chapter deals with type-1 traffic-related applications. First terminology, requirements and fundamental concepts will be presented in section one. Then in section two, a classification of traffic-related applications is given and arguments are presented, why detection of traffic incidents is difficult. This leads to a selection and classfication of event. The third section will explain the difficulties of dealing with area events in a distributed environment and show how this can be approached. Section four is about local reasoning, while section five elaborates on reputation systems in general and presents a customized implementation called VARS - Vehicle Ad-hoc Reputation System. Section six presents voting approaches to solve the same tasks exploiting group effects. Finally, section seven combines these approaches to make forwarding decisions and presentation decisions.

5.1. Terminology, Requirements and Concepts

5.1.1. Terminology

Definition 20 (Confidence)**.** *Confidence is a value assigned to traffic-related information that relates to the the degree of calculated information authenticity. The confidence-level is used in the presentation decision process and in the forwarding decision process.*

It will be distinguished between specific situations and events:

Definition 21 (Situation)**.** *Situations describe the environment over a period of time. They usually reflect the occurrence of special circumstances. A real-life example would be a lateral wind area or a construction site. There are also short-lived situations such as a car doing an emergency braking.*

Definition 22 (Event)**.** *An event is generated after detection of one of a set of predefined situations and represents information about that specific situation at a distinct point in time. A local danger warning system for example detects the appearance of specific combinations of settings and generates an event. The information about an event may be sent to other nodes. Event will sometimes be used to denote the sum of all events related to one situation. There are area events, such as rain, fog, traffic jams and point events, such as icy sections, obstacles, etc.*

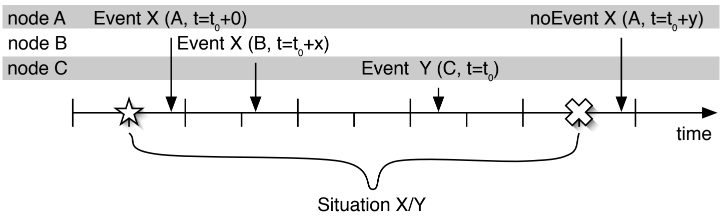


Figure 5.1. Event Situation Timeline

Figure 5.1 gives an impression about the timeline regarding situation and event definitions. This figure also shows, that in a decentralized system, it is difficult to identify whether events detected by two different nodes are the same (indicated by different names "X" and "Y"). This is especially difficult for area events, which will be discussed in section 5.3 Distributed Data Aggregation of Area Events on page 87. Another point that is shown is that once the situation is no more existing, it has to be decided whether the system is actively acknowledged of this fact (based on a non-existing detection where there should be one) or not.

Note that an event does not only denote the moment of an event generation, but the information gathered during that moment. The distinction between situation and event(s) is that situation is used to denote occurrences in general, while event(s) is used whenever the detection of a situation by a node has taken place and therefore digital information about the situation is available somewhere in the system.

5.1.2. Requirements

In addition to the general requirements that have been introduced in section 2.5 there are specific requirements for type-1 applications:

Requirement 5.1 (No Complete Authentication). *Traffic-related applications and messages are in general more important than multimedia messages. And some traffic events are more urgent than others. Priorizing the distribution of messages is required to ensure maximum availability of critical application data.*

Requirement 5.2 (Exploit Large Scale Effects). *VANETs are only working if there is a sufficiently large number of nodes. It seems logical to assume that once in use, the number of nodes will further increase. In any case, due to the number-of-nodes requirements for communication, most situation will be recognized by a number of nodes passing this area. The availability of multiple experiences shall be exploited to the degree possible to improve the system's robustness and security.*

Requirement 5.3 (Robustness). *The system shall be designed as robust as possible, since this will effectively increase the system's efficiency for traffic-related messaging as experienced by drivers.*

Requirement 5.4 (Machine-based Trust). *However, since requirement 2.5 demands that direct user interaction should be avoided, trust is only regarded between system components, usually network nodes, excluding any human interaction.*

Requirement 5.5 (Non-interactive Protocols). *In all cases where more than one receiving node is in range, interactive protocols would be intolerably complex to execute. Additionally, the connection time will in many cases not be sufficient to perform interactive protocols. Therefore, for type-1 applications, non-interactive protocols shall be used.*

Requirement 5.6 (Physical Transport). *A feature of cars as nodes is that, when in operation, they move. This fact, which is true in all but very few cases such as traffic jams shall be used to transport messages whenever a direct connection is not available. This concept of physically transporting messages and sending them to new forwarding nodes as soon as available will be called "store-and-forward".*

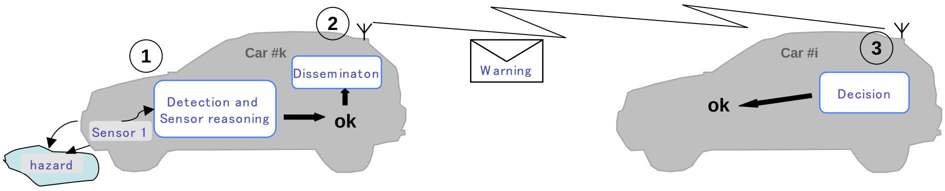


Figure 5.2. Phases Traffic-Related Messaging

5.1.3. Phases of Traffic-Related Messaging

In short, traffic-related messaging basically comprises three main steps as shown in Figure 5.2:

(1) **Detection and Sensor Reasoning**

An integral requirement of traffic-related messaging is the autonomous detection of hazards on the road. Within the scope of this thesis, it is assumed that vehicles are able to individually detect certain hazards without the need of cooperation with other vehicles. In the following, the process of detecting the presence or absence of a hazard is referred to as *experience* of the vehicle. The next step is to reason about the data that lead to the detection of an event. It means to evaluate the data according to previously recorded data or given heuristics, part of a local world-model. Reasoning is an important part of detecting sensor manipulation or mitigating the effects of malfunctioning components.

Section 5.4 will discuss the local reasoning process as part of decision processes, including a paragraph on event detection: paragraph 5.4.1.

(2) **Dissemination**

After an experience has been made, a corresponding warning message is created by the detecting vehicle and broadcasted accordingly. Receiving vehicles store the remote information and relay the message to other vehicles as long as they travel within a dedicated area. By combining all available information from remote experiences, each vehicle is able to conclude a picture of the road situation ahead of its estimated route. Two kinds of messages are considered: warning messages and revocation messages. Whenever a vehicle detects a hazard, a corresponding warning message is created. On the other hand, when a vehicle passes the location of a hazard which has been previously reported by other cars, and no hazard is detected, a revocation message is created, informing other cars that the potential hazard has possibly disappeared.

Please refer to the general discussion on efficient message distribution, section 4.4.4 and virtual backbones, section 4.4.5, as well as the type-1 specific distributed aggregation problem elaborated in section 5.3.

(3) **Decision**

Messages received from other cars need not be evaluated continuously [1]. Instead, this is only necessary if a vehicle has been approaching a potentially hazardous area, which has been experienced and reported before, up to a critical distance. In this case, it has to be decided whether to take action and notify the driver or not. This kind of decision will be called presentation

[1] However, all messages are stored and kept for evaluation at a later time.

decision. Additionally, when messages are received, a forwarding decision is computed to distinguish whether the message will be transmitted immediately or not, depending on current conditions and settings.

Please refer to paragraph 5.1.4 for an overview and to sections 5.4, 5.6, 5.5, 5.7 for elaboration of the components.

5.1.4. Information-centric trust and Decision Processes

Conventional trust models and security concepts cannot fulfill all requirements given in previous chapters. Solutions either focus on securing the communication network or on restricting the access to vehicle components by utilizing conventional security measures, such as digital signatures and trusted hardware. This way, attackers are prevented from manipulating the network or certain parts of the vehicle. But even assuming that attackers can neither manipulate the VANET itself, e.g. by injecting fabricated messages nor have direct access to vehicle components, they can still alter the physical environment around vehicle sensors or drive in a certain style in such a way that the detection process is influenced. A message being generated as a result of such an action fulfills all requirements of conventional security measures, the car will process the information in the specified way, a valid digital signature (or other form of digital credential) will be applied and the message will be distributed as usual. This cannot be detected reliably or prevented using conventional security measures. Even if detected and reported to a credential revocation authority, revocation of a digital credential is difficult due to the decentralized nature of VANETs and the problem of automated on-time verification of information in dynamically changing environments. It is also a drastic measure, one of last resort, because it is costly and requires a lot of effort. Also, it prevents node from participating in the network until a new cretential is issued again, which may degrade the network more strictly than required in every case. Ultimately, conventional measures cannot protect against the threat, because cryptographic protection concepts cannot verify information itself. Therefore, in addition to conventional (but modified) measures introduced in previous sections and chapters, information-centric approaches are presented in this chapter. They are based on some basic assumptions: that traffic-related events share characteristics with other events of the same kind, are sometimes related to other environmental factors that can be measured, that the existence of a large number of nodes supports cooperative evaluation approaches and that the information that is ultimately included in the messages is taken from a closed set, thus messages are comparable, which leads to the approach of voting schemes.

Three approaches on application-level are presented in this chapter:

(1) **Local Reasoning**

(2) **Voting Schemes**

(3) **Reputation System**

Finally, a value has to be calculated that expresses how likely it is that the data about a reported event is true. This value is called confidence-level or simply confidence. It is the degree of calculated information authenticity. The confidence-level is used in the presentation decision and forwarding decision processes. Confidence-levels for different messages about the same event have to be combined to get an event-oriented confidence-level (which obviously is the goal of this procedure). Relay nodes outside the decision area have to calculate confidence and opinions on other relay nodes immediately for the forwarding decision. Nodes reaching the decision area have to calculate final (event) confidence and opinions again for the presentation decision.

5.1.5. Geographic Areas

In a traffic related message distribution system, there are three geographical boundaries of interest, that have to be defined, see Figure 5.3.

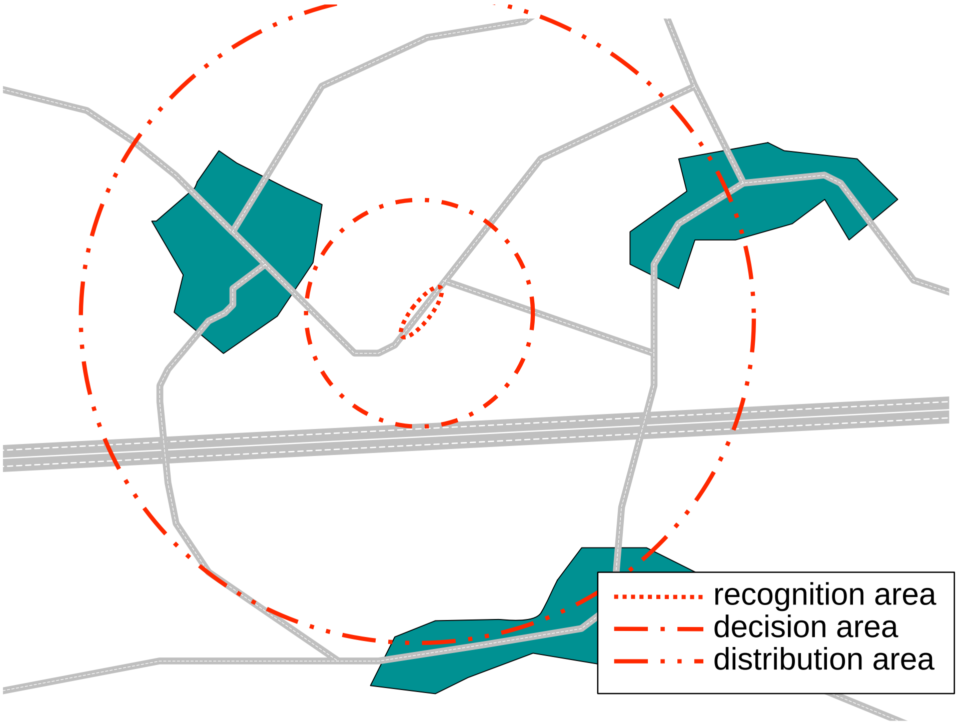


Figure 5.3. Geographic Areas

5.1.5.1. Recognition Area. The *recognition area* is the geographical area within which an event can be experienced or measured. Its dimensions depend on the event class and it is set by the source node.

5.1.5.2. Distribution Area. In the given traffic-related vehicle to vehicle communication scenario, messages are forwarded as long as they are relevant for nodes within the network (see [**KSA02**]). To simplify matters it will be assumed that messages about one event will be distributed within a circular area, which is called *distribution area*. In a more general case, it can be thought of more complex shapes of a distribution area. The standard size and shape for each event class is defined as a preset, discussed in section 5.2.8. That means that the messages of source nodes define the distribution area.

In addition to that, it can be thought of various levels of distribution area size and information depth depending on the distance to the recognition area. See concept of distance-dependant information depth in subsection 5.1.6.

Note that every relay node inside the distribution area have to perform a forwarding decision to decide whether the information in question shall be forwarded or not.

5.1.5.3. Decision Area. It is important to realize that a car's system will usually not display a received warning message at once if it is still far from the recognition area. Instead, it will wait for the appropriate position and time, which is useful for accumulating as much information about an event as possible and is called delayed decision. Therefore, there is an area around the event, upon entering (or already being inside) which, the nodes' systems will have to decide about presenting a message to the driver. The process is called presentation decision, the area is called decision area.

The decision area's dimension depend on the event class's preset definitions and the receiving node's traffic scenario[2] and speed. Driver's preferences, such as simple distinction between "warn early, give all information" and "warn only if close to event and high confidence", reflect personal preferences and will be part of the HMI component.

5.1.6. Distance Dependant Information Degradation

Frank Kargl introduced the concept of distance dependant information depth in [**Kar03**]. It is the idea that - in some cases - the closer nodes get to an event, the more detailed information they need. Or stated the other way around, nodes that are still far away require only little information such as a rough direction and distance and the type of event. Transferred to the architecture presented in this thesis, this would require different information levels for different categories of the distribution area. However, this approach is problematic for instance in rural areas or during nighttime, where only few nodes are available. In this case, cars get vague information far away, but would receive no updates as they get closer to the recognition area, due to unavailability of information-carrying nodes. To avoid this problem, traffic scenarios can be used as indicator, whether messages carrying traffic-related information can be simplified on their route away from the recognition area or not. To be more specific, distance dependant information degradation is only allowed in city and highway scenarios and can be optionally defined using event class definitions (see section 5.2.7).

5.2. Classification of Traffic-Related Applications

The whole variety of applications has been presented in section 2.4 and a rough typization was given in section 2.4.2. However, the first type of applications, traffic-related messaging, includes a whole variety of scenarios and communication situations. Therefore, type-1 applications will be classified in this paragraph in order to be able to analyze weaknesses with respect to security and design the system accordingly.

5.2.1. Dynamics

The real world is constantly changing and so are detected events. However, there are only three dimensions in which a specific event may change over time: its position, its shape / dimensions and its intensity.[3] Moreover, the dynamics in which those dimensions of a specific event change are typically similar for different complex situations. While an oil spot on the road will typically neither move, nor change its intensity or shape, a fog bank may change in all these dimensions over time.

The following three generic classes are therefore distinguished, see also Figure 5.4:

- **Static.** A class of events that has a constant size, no movement and no changes in intensity. (An accident for example, once happened, will not change its location, size or intensity.)
- **Continuous.** A class of events that has varying size, continuous movement and/or continuous change of intensity.

[2] Note that if digital maps are available, information about road structure, intersections, villages, exits, etc. can support the definition of the decision area

[3] Note that some weather events have additional parameters such as wind-direction. For simplicity, this will be included in the intensity dimension for instance using only a (measurable) lateral wind component.

- **Chaotic.** Events that behave chaotic (uncontinous) in terms of changing their size, movement and/or intensity, but stay within well-defined limits.

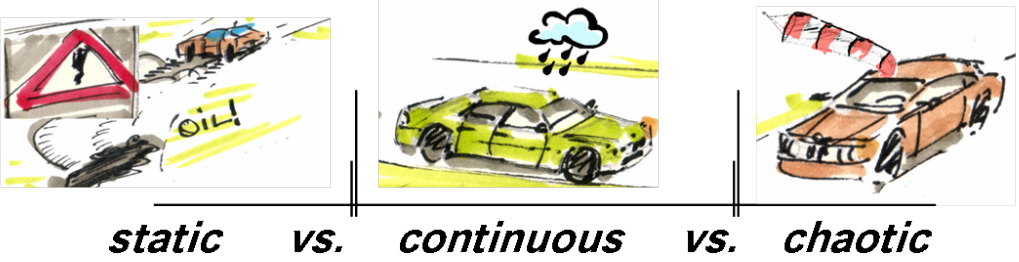


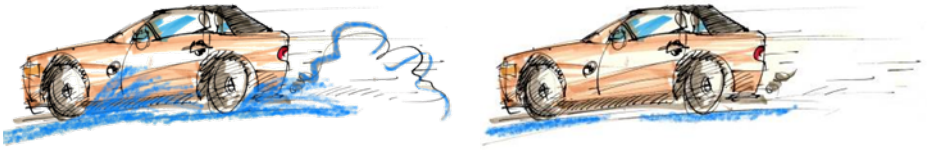
Figure 5.4. Dynamics

Whether an event is static, continuous or chaotic has great impact for the complexity of calculating the level of confidence in a received event. Specifically whether an event has an intensity or just a binary state (yes/no) is critical in a system based on evaluation of multiple messages. Another question is how large local diversity of an events state is, for instance aquaplaning may only happen in lane grooves. A car driving only centimeters beside those grooves may experience significantly different data.

5.2.2. Driver Related Determinism

In a vehicle messaging system, especially in the local danger warning scenario, messages will most likely have an effect on driving behavior since most of the messages contain driving related information. Unfortunately, the altered behavior of the driver can also have an effect on a situation's detectability, see also Figure 5.5. A prominent example is the detection of aquaplaning: A car at normal speed will experience some slippery road condition and can, together with other environmental data, conclude aquaplaning and generate a warning message accordingly. A following car receiving this warning however, may eventually reduce its speed and would therefore not be able to experience slippery conditions. This case is called *non-deterministic detection*.

- **Deterministic Detection** Events that can be detected independently from driving behavior[4].
- **Non-deterministic Detection** Events that may be experienced differently, according to the specific behaviour or action of a driver upon receiving warning messages.



(a) Fast: Aquaplaning!

(b) Slow: Uncritical!

Figure 5.5. Driver Related Determinism

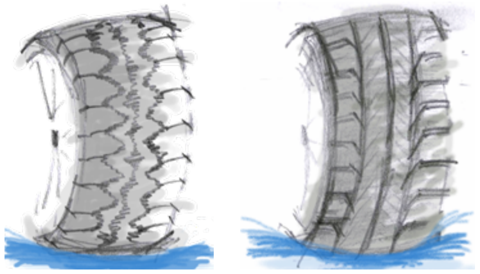
[4] Note that in a mathematical sense this is not exactly true, because a car that does not move will probably fail to detect some events.

5.2.3. Configuration Dependency

In some cases, the relavance of a situation to cars receiving a message depends on a number of settings. For instance, if a car with specific properties, such as the model of the car, the type and make of tires, speed, yaw rate, sends out an icy section warning message, what information would a receiving node need in order to assess the effects properly? Other messages, such as a traffic accident warning may be more trivial, because the receiving car does not have to know under which conditions the message was generated. It is therefore distinguished between:

- **Objective Events** Events that are characterized by a specific situation, but do not depend on specific vehicle parameters.
- **Subjective Events** Events that are depending on specific individual vehicle parameters in addition to situative settings.

See also Figure 5.6.



(a) "Bad" Tires: Slippery!

(b) "Good" Tires: Uncritical!

Figure 5.6. Configuration Dependency

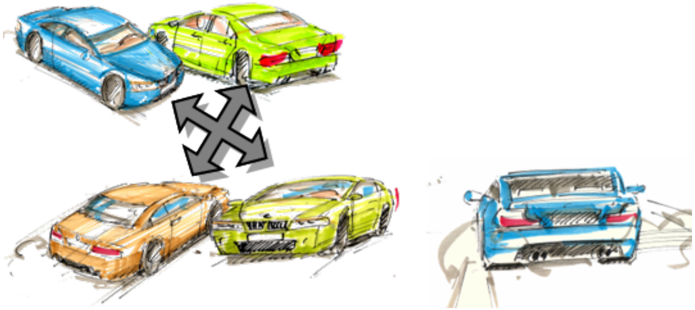
When receiving a message it is important to know whether an event is objective or subjective in order to calculate its relevance and confidance properly. In the case of a subjective event, information about the sender's configuration improves the quality of a receiver's relevance and confidence decisions.

5.2.4. Aggregation Locality

Two ways of event detection are distinguished. The first class of events can be detected by every single car that meets the system's specifications, for instance an icy section on the tarmac. However, there are some types of events, such as an accident where airbags have been fired, that cannot be verified by other nodes with the same procedure as the originating node. The second class of events requires the interworking of different cars in order to detect events, such as traffic jams. See also Figure 5.7.

- **Single Node Detection** Vehicles are able to detect events using their basic sensors[5]. A special case are vehicle originating events which are only directly detectable by the originating node.
- **Multi Node Detection** Events that can only be detected by aggregating information from more than one vehicle.

[5] Basic sensors have to be defined for every event class.



(a) Cooperative (b) Independent

Figure 5.7. Aggregation Locality

5.2.5. Data Collection

How is the data collected? Out of the theoretical sources of data mentioned in paragraph 3.1.3 only the following will be regarded in the following sections [6]. The associated trust varies significantly between different kinds.

- **Onboard Sensors:** Readings of various sensors are monitored and if defined patterns are detected or specific thresholds are exceeded, an event is generated. There is a special case: in some cases, such as approaching emergency vehicles or post-crash warnings, the input comes from a single sensor or button. Therefore, sensor reasoning is not possible and it cannot be directly measured or verified by others. This case will be referred to as single source event.
- **Driving Dynamics:** Cars are equipped with a number of sensors that detect the movement and acceleration of a car along its longitudinal, lateral and vertical axes. The combination of these values indicates certain maneuvers.
- **Location Traces:** Since positions at given times are known, cars can record location traces. From these traces, traffic situations, such as reaching a traffic jam can be concluded or evasion maneuvers/unusual driving may be detected. Note that these conclusions are much more precise if a digital map is available and information such as the road type (highway, city road, etc.) can be used for situation assessment.

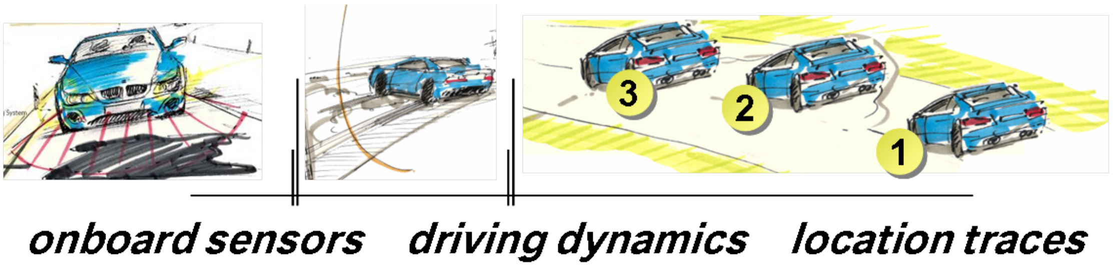


Figure 5.8. Data Collection

[6] Also regarded, but not specially mentioned here is ”supportive data” such as position data, time, security context information and keying material

5.2.6. Impact

The impact a specific message may have on driving conditions or the driver's behaviour is related to how fast the driver must respond according to the information (timing), how safety-relevant the information is (content) and how important this information is in the current situation (situation).

These factors may be clustered into four different categories, see also Figure 5.9:

- **Autonomous** The vehicle acts autonomously. Examples are an automatic full break to prevent the car from a red-light violation or cooperative driving applications.
- **Action** The service provides information to the driver about critical situations (e.g. imminent dangers) that require the driver to intervene immediately in order to avoid an accident. Examples are an accident of a leading vehicle or the end of a traffic jam.
- **Attention** The service provides information to the driver about situations that require particular driver attention, but no immediate intervention. Examples are reduced friction or visibility due to bad weather conditions or certain hazards on the route.
- **Awareness** The service provides information to the driver about situations situations that are not very dangerous, but should be recognized by the driver in order to be aware of traffic and/or road related information. Examples are high traffic or areas without warning capabilities on the route.

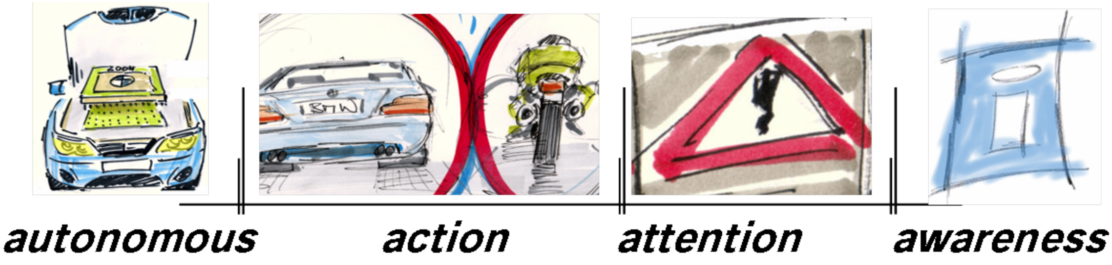


Figure 5.9. Impact

[**DKS05**] gives a more detailed view on this topic.

5.2.7. Event Classes

After presenting classification criteria for type-1 applications, table 5.1 shows how events are grouped together to form practical categories with like properties. Some classes however are excluded in this work, due to their complexity or problematic detection:

- **Non-deterministic Detection** If the detection depends on a large degree on external, not directly measurable factors.
- **Subjective Events** Since, at this stage, there is no input configuration specific information, these events are excluded here.

The event classes are grouped building on aforementioned classifications and additional criteria for presentation decision and forwarding decision:

- **Typical Persistence** How long is the typical average duration of a situation: e.g. for fog may be days, for emergency braking events it may be milliseconds.

- **Typical Dimensions** The typical average size of the recognition area, for simplicity it will only be distinguished between area situations, line situations (a special form of area situations), and point situations.
- **Dynamics** How volatile is the situation, how fast does it move if it moves at all? Is the shape changing?
- **Aggregation Locality** Are events usually generated by a single node or do multiple nodes improve detection?
- **Data Collection** How is the data for event generation typically gathered?
- **Impact** What effects can such a situation have on cars? Is it serious or just of minor relevance?

	area-type weather	road condition	small hazard	stationary traffic	vehicle condition	situational traffic
Typical Persistence	hours	min - days	min - hours	seconds	situational	seconds
Typical Dimensions	kilometers	kilometers	meters	meters	situational	situational
Dynamics	chaotic	static	static	static	continuous	chaotic
Aggregation Locality	multiple	multiple	multiple	multiple	single	multiple
Data Collection	sensors traces	sensors traces	sensors dynamics	sensors dynamics	sensors (button)	sensors dynamics
Impact (highest)	attention	attention	action	action	action	action
Example (see table 2.1 on page 20)	Low Visibility Warning	Work Zone Warning	Obstacle Warning	Intersection Collision Warning	Approaching Emergency Vehicle	Cooperative Glare Reduction

Table 5.1. Event Classes

5.2.8. Event Class Presets

After grouping together events on a theoretical basis during system design, for each group an implementation should define system pre-sets. Pre-sets are used to assign useful values for presentation decisions and forwarding decisions. With pre-sets defined for each event class, it is sufficient to transmit event class and actual parameters (location, time, etc.), thus reducing transmission size and improving confidence calculation / security. In order to reflect the dependancy of meaningful presets on different traffic and communication situations, presets have to be defined for each traffic scenario. Table 5.2 gives an example of presets for the event classes defined above in a city traffic scenario.

	area-type weather	road condition	small hazard	stationary traffic	vehicle condition	situational traffic
Recognition Area	area event	area event	point event	point event	point or area event	point or area event
Persistance Time	1-24 hrs.	0-72 hrs.	0-72 hrs.	0-5 min.	0-5 hrs.	0-5 min.
Priority (1 highest)	2	1	1	1	1	2
Decision Area Size[7]	2 km	1 km	1 km	500 m	500 m	200 m
Distribution Area Size	20 km	10 km	10 km	5 km	5 km	2 km

Table 5.2. Event Class Presets

Table 5.3 shows how different presets may be defined with respect to different traffic scenarios.

5.3. Distributed Data Aggregation of Area Events

Large events, such as fog or rain, can only be detected on their own route by single cars. In order to establish a realistic image of the actual hazard, multiple messages concerning the same event have to

	highway scenario	city scenario	rural scenario
Recognition Area	point event	point event	point event
Persistance Time	0-72 hrs.	0-72 hrs.	0-72 hrs.
Priority (1 highest)	1	1	1
Decision Area Size	5 km	1 km	2 km
Distribution Area Size	50 km	10 km	50 km

Table 5.3. Example: Event Presets for Small Hazard Class

be combined. This section will discuss this problem and provide an approach for distributed area event definition.

5.3.1. Data Aggregation Problem for Area Events

The effects of area events can be measured by multiple nodes at the same time. And thinking of the large number of nodes that will in most cases be able to detect such an event and exchange information about it, it seems easy to handle this kind of events. However, following factors may negatively affect the quality of information about an area event:

- **Distributed way of information gathering:** The information cannot be accumulated and stored at a central database[8]. Instead, receiving nodes will gather information from a variety of other nodes they receive messages from. This effectively means that the information base is different from node to node.
- **The problem of identifying events uniquely:** If multiple messages about the same type of hazard in different - but close - locations are received and if this hazard belongs to an area event class, the question arises whether the messages are related to a single large event or not.
- **Delay between different data collections and dynamics of events:** Some area events, especially weather situations, have a chaotic character, which means that they change significantly over time. The less nodes passing through the recognition area, the larger the delay between their messages. While some messages provide information about a growing recognition area, others may indicate that the situation is no more existing. This may result in an inconsistent image of the situation. Note that due to the distributed character, race conditions between different messages may occur[9].
- **The limitation of measurements to streets:** Since data is only taken on streets, information about the full geographic extension and movement of an area event over time can only be gathered uncompletely.

Additionally, requirement 6 demands that no digital maps should be used. This limits the availability of registered information usable for an aggregation algorithm.

How should the data volume included in the messages be limited in case of a large number of nodes passing through the recognition area? If nodes within the recognition area record data periodically and every of these nodes sends all the information, this could lead to large quantities of data to be transferred.

[8] This might be true for if the VANET is connected to a centralized floating car data system, thus resulting in better results. In this thesis however, this will not assumed to be the case.

[9] This is the case only for aggregated area events because there is no single event detection time, but only a mix of different margin added times.

If nodes only record where they entered and where they left an recognition area, it is difficult to combine different messages about the same event.

5.3.2. Rectangle Aggregation Algorithm

The approach presented in this paragraph provides a way to combine information about an area event in such a way, that shape and position changes are considered and also shrinking is possible. The amount of data included in the messages is limited to smallest size possible, without sacrificing essential information. This approach is based on the idea, that the geographic shape of area events can be represented in a simplified form as rectangle.

5.3.2.1. Generating an Area Event.

(1) **Detection** The goal is to detect a hazard that spans over a specific area, called the recognition area. That is the area where nodes are able to detect the hazard. However, since cars move on streets, the measurements can only be taken along the road. See Figure 5.10

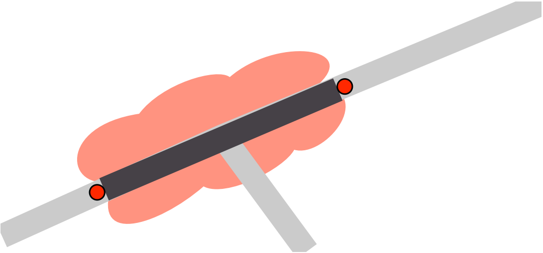
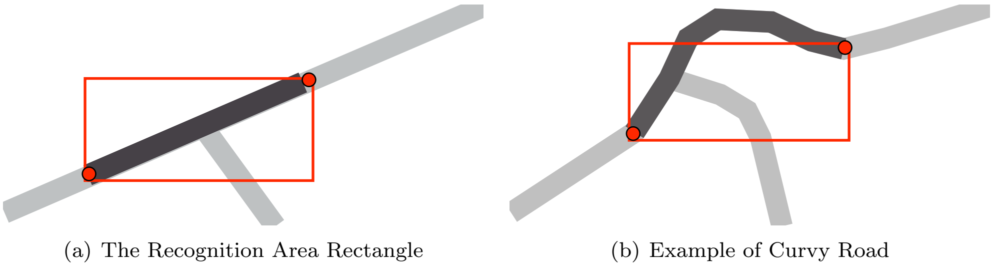


Figure 5.10. Detecting a Hazard

(2) **Basic Principle** The first node that detects an event marks his ingress and egress points. These two positions span a rectangle with sides parallel to the geographic directions. See Figure 5.11(a). However, in some cases such as a curvy road, the rectangle does not include all points on the road where the event has been detected, see Figure 5.11(b).



(a) The Recognition Area Rectangle

(b) Example of Curvy Road

Figure 5.11. Recognition Rectangle and Curvy Road

(3) **Generation** Therefore, the rectangle is not defined by ingress and egress points, but by the largest span during a complete event generation run. Note that an event is generated and the corresponding message is sent immediately after it has been detected to ensure timely warning of other nodes (such as following vehicles). But update messages are sent periodically as long as the car is still experiencing the situation. Update messages continue to enlarge the rectangle during this process as shown in Figure 5.12. Note that the rectangle is only enlarged, never shrinked as the curvy part of the road shows.

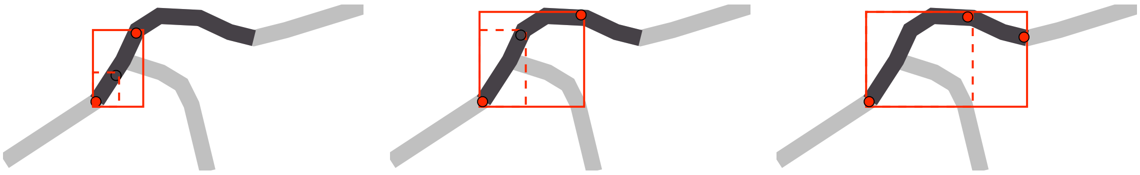


Figure 5.12. Generating an Recognition Area

Those points that effectively define the size of the rectangular recognition area are called margin points[10]. New nodes entering the recognition area compare their ingress and egress points to the existing rectangle. The rectangle will be enlarged, if the combined points span a larger rectangle. See Figure 5.12.

(4) **Adding Margin Point** Margin points from other directions can be added to support event messages, but in the case as shown in Figure 5.13 the rectangle would not be increased and in those cases points are dropped.

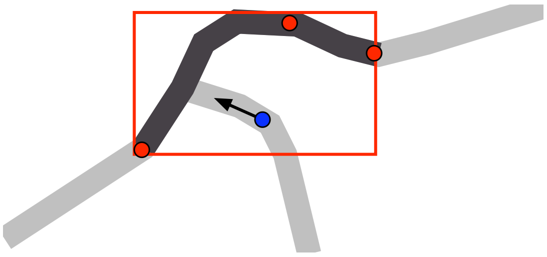


Figure 5.13. Adding a Margin Point

5.3.2.2. Updating an Area Event. There are three cases what happens if a new car detects a known event (thus having received one or more messages about it before):

- the area is enlarged and margin points are added to the list
- one or both of the points are closely situated to existing points and the older ones are therefore updated
- the new points mark a gap and the original rectangle will be splitted in two

(1) **Enlarging** Enlarging a rectangle is straightforward and follows the same rules as described in the paragraph above. Thus the rectangle will be expanded if data indicates that the situation is actually larger than stated in a received message.

(2) **Shrinking** The rectangle will only be reduced, if a another node moves on an inbound street (where a margin point exists) and detects the situation at a later point, thus resulting in a smaller recognition area. See Figure 5.14.

Or it is shrinked if a node passing through the recognition area does not detect the situation any more before leaving it. This is shown in Figure 5.15.

(3) **Combining and Splitting** Two rectangles will be combined if there is there is an intersection. A rectangle will be splitted if a gap is detected in between. See Figure 5.16.

A message position/shape part consists of the most actual ingress and egress points that are known - the margin points. A message may optionally contain information about how many independent source

[10] The starting point is also a margin point. In rare cases there may be up to four margin points (one for every direction).

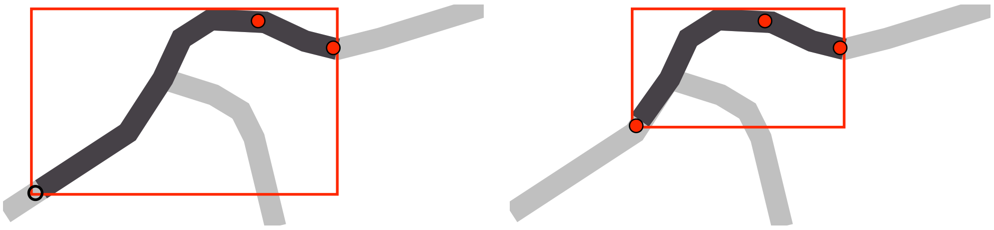


Figure 5.14. Shrinking the Recognition Area - Inbound

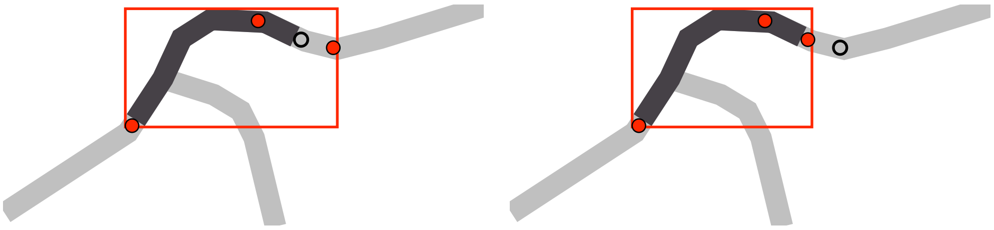


Figure 5.15. Shrinking the Recognition Area - Leaving

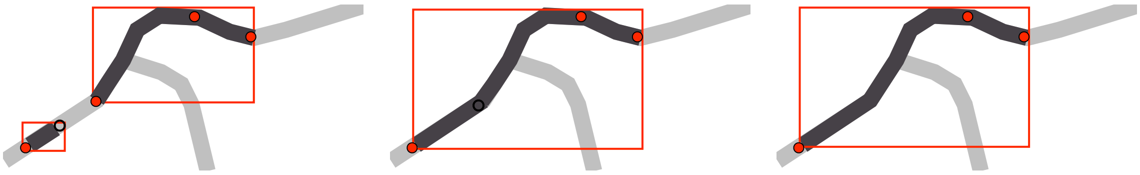


Figure 5.16. Combining Recognition Areas

nodes (=detectors) the sender has received the message from. This value will most of the time be zero for nodes that are further away, if the message has only been forwarded.

A more thoroughly discussion on this topic and an implementation of this algorithm can be found in [**Här05**].

5.4. Local Reasoning

5.4.1. Event Detection

For every event class, as defined in section 5.2.7 sensor data occurring in a specific combination will trigger an event. Therefore, for every event class that is defined for the system, a sensor filter profile exists. A sensor filter as shown in figure 4.1 on page 49 will gather data from sensors and other inputs in the specified way and send the results to a sensor reasoning component. Therefore the sensor filter collects only the kind and amount[11] of data necessary, making an analysis by a sensor reasoning unit possible.

Figure 5.17 shows that event detection depends on filtered sensor information, which can lead to the detection of an event. If an event is detected (thus data indicates that a predefined situation has occured) and the methods in the sensor reasoning unit using the local world model indicate that the data is consistent and plausible, an event is generated, meaning that relevant data is stored and further actions are started. The other reason that an event is detected is that an external trigger causes the

[11] Since some of the data is taken from industrial field buses, such as CAN or FlexRay, it is not always necessary to retrieve the data every interval it is available on the bus.

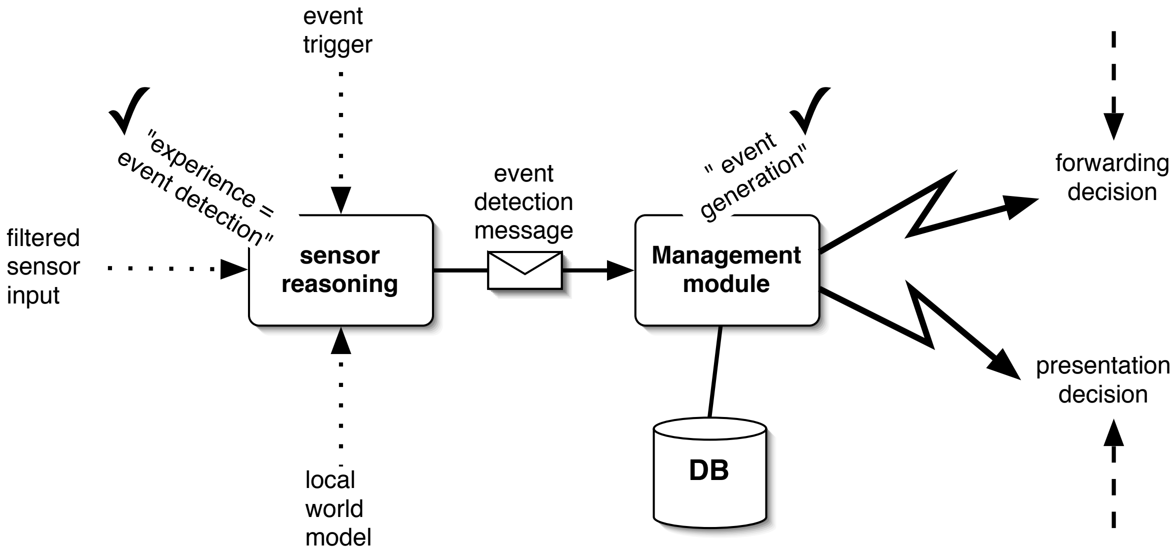


Figure 5.17. Event Generation

sensor reasoning component to check stored data, compare it to the local world model and then, after successful consistency and plausibility checks, generate an event. The act of detecting an event is also called ”experience” in this thesis. Event generation includes all activities necessary to compose a warning message and transmit it.

5.4.2. Plausibility Checks (Reasoning)

Plausibility checks will be used whenever possible because it is the most effective method to distinguish between reasonable data and abnormal patterns. It is used to analyze local sensor data as well as plausibility of received messages. However, the efficiency of plausibility checking is rather limited if the situation cannot be experienced directly, or put differently, if the locally measurable data correlates only little with the data in the recognition area in question. There are two cases plausibility checks are used for. First, internal sensors will be checked for consistency and plausibility. Second, incoming messages will be checked for plausibility. In the solution, there is no qualitative difference between both cases. For example, some input data, such as temperatur naturally vary only few over time. Temperature outside a car can be measured and according values be stored over time. If there is a huge change in temperature, the plausibility of this data is low. (Note that this is a simplified example and plausibility is the result of a whole set of values that are compared, also taking environmental conditions into account.) The reason can be a malfunctioning sensor or one that is tampered with, as well as an incoming message warning about an icy section on the road, when the temperature sensor has not dropped below 30°C over the last hour. Other examples for simple data comparison are comparing actual positioning information from the satellite positioning system to location traces or the received actual time to an internal clock. If there are large jumps between the recorded data and the actual data, this may indicate that there is something wrong. The strength of this approach however, is to regard a multitude of values within a context. Taking the data from above examples: Assuming the actual data (either received or measured) does not deviate from stored data in an unusual way (note: sometimes data jumps are allowed!) and a message about an icy section on the road is received. Then this information could be compared to the temperature curve (has there been a temperature below 0°C lately?), the time (is it winter?) and position (did the car drive to higher altitudes, into a tunnel or other areas where temperature drops quickly?). Finally, the correlated outcome of these questions results in a degree of plausibility. Note also that this becomes more efficient the more extensive the local world-model is.

5.4.3. Event Scenario Building

Optionally, this concept can be extended through long-term oriented reasoning inside RSUs. RSUs have a fixed position and are in long-term use. Thus, they are in the position to record information about the probability of certain events in their geographic vicinity and to correlate these events with respect to time and geographic location. They can ultimately calculate an improved world model. If fixed nodes, such as RSUs are assigned a higher trust level this can improve the decision process for all nodes in the region.

5.5. Reputation System

This section deals with reputation systems as an approach to bolster other measures for establishing authentic traffic-related information. Reputation systems exist in many different variations and have been published under alternative terms, such as buddy system [**FOKR04**] or recommendation system [**MG04**]. VARS - Vehicle Ad-hoc Reputation System, the system introduced in section 5.5.2 suggests a distributed reputation system which establishes individual trust relationships between the participants of a VANET, applying the ideas of reputation in ad-hoc network routing, such as in [**MGLB00**] in combination with reputation systems in social contexts as in [**FOKR04**] to the field of IVC.

In this thesis reputation is defined as follows:

Definition 23 (Reputation)**.** *Reputation is a term that denotes expectations on the behavior of an entity in the future based on its behavior in the past.*

Many explanations and fundamental material on reputation systems can be found in [**MHM03**] and [**Mui03**].

Following system design criteria determine the functioning of reputation systems in general. The choices taken for this thesis are discussed for each point:

Centralization: Some reputation systems have a centralized architecture, such as the well-known rating systems at ebay.com or amazon.com. Others calculate reputations on a local basis with support from central backends using certified reputation levels. Finally there exist systems that maintain local reputation databases for local assessment of peers. This thesis will focus on fully decentralized systems with local reputation databases.

User Interaction: Some reputation systems are based on human ratings that require users to assess other users. The process of reputation generation is fully delegated to human users, while the reputation system itself only stores, organizes and presents those assessments accordingly. Other reputation systems are fully machine-based, meaning that assessments are made without human interaction. Requirement 2.5 prohibits direct user interaction, therefore this thesis will only deal with systems without human interaction for determining reputation values.

Motivation: A key question in reputation systems is why do participants strive for good reputation? In many human-assessed systems, this increases the user's standing in a social environment. However, in machine-based systems motivation seems to be the wrong term, reputation systems help to categorize nodes according to their behavior and resulting actions are mandatory. In the latter case, a good node reputation usually grants either a preferred treatment or additionally available services.

Scalability: Reputation systems require the storage and evaluation of past experiences. Both, required memory to store reputation data and processing power for evaluation, increase with the number of participating nodes and the age-range of experiences being regarded. For this work, reputation systems must scale up to a large number of nodes, that is usually limited by the number of vehicles in a sufficiently large area and can reach the number of a couple of millions. The age where reputation data will start to expire ranges theoretically from a couple of months to only a few minutes. The ideal setup will be examined in following sections.

Strong Identities: New nodes entering a reputation system may either be assigned the lowest absolute reputation value or a neutral value, with nodes having a lower reputation than that. To simplify understanding the first case is defined as a reputation system that only has positive reputation values and new nodes start with zero. In the latter case, nodes may degrade over time and have negative values as well. This distinction is important, because it affects the required persistance of identifiers. In the case with only positive values, the identity of participants does not necessarily have to be a strong identity, since a new identity is always associated with an initial reputation value being the lowest possible value. The other case however requires strong identities, because a node being assigned a negative reputation value could anytime pick a new identity and start over with a neutral value. Therefore, due to requirement 3.9 demanding pseudonym flexibility, only positive absolute reputation values will be used here.

Reputation degradation: Assuming that nodes in a reputation system strive to increase their standing means that there exist procedures to increase their reputation (based on opinions). Reputation values shall not increase infinitely, so there has to be a procedure to decrease reputation. Two possibilities are to decrease reputation values over time, called ageing, or to allow negative opinions. Both will be discussed in this thesis.

Direct Reputation: Direct reputation is based on direct observations and experiences, indirect reputation relates to information about observations and experiences from other nodes, transitive trust (refer to [**JGK03**]). While direct reputation is more reliable in respect to false accusations, indirect reputation represents a global reputation and delivers reputation values even for unknown nodes, which is indispensable for the envisioned use cases.

Figure 5.18 shows how a reputation system may operate. It shows three nodes A, B, and C. Each of them maintains its own reputation database which stores the reputation values for known nodes in the system. Nodes A and C interact with each other (e.g. through communication). This event is shown as interaction X. In this case, both of them are able to make direct observations of each other. In this example, A has a negative perception of C and generates an opinion with the value (-10). Node C on the other hand has a positive perception of A and therefore assigns an opinion of (+2). After this event, both, A and C, will update their reputation databases accordingly. There is also another interaction between nodes C and B resulting in an opinion of (+5) that B generates on C. In this case, the reputation system is transitive and allows indirect reputation, resulting in an increase of A's reputation (+1) in B's database. The reputation system in this example uses reputation values between $R = [0; 100]$ with a

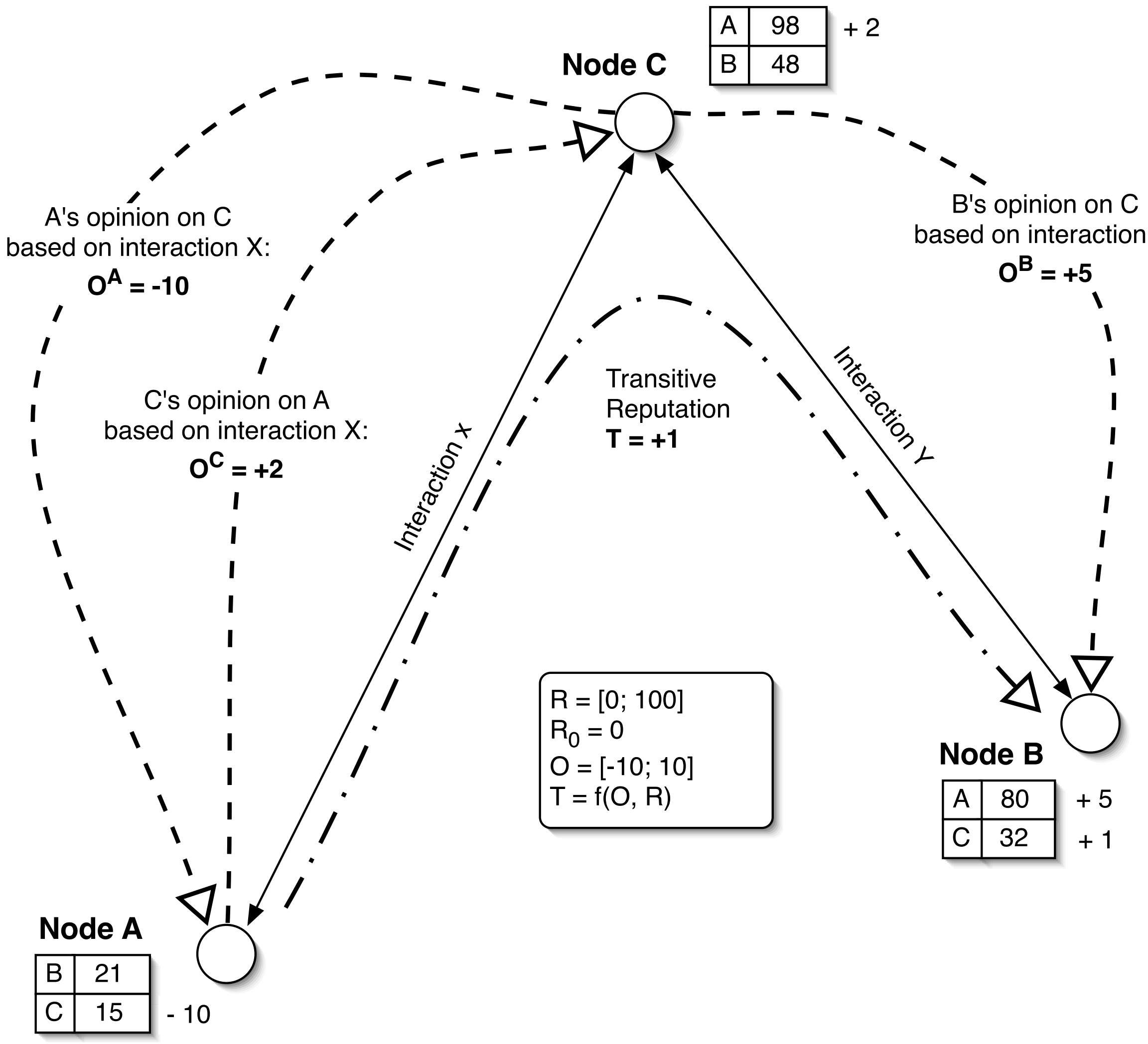


Figure 5.18. Sample Reputation System

starting value for new nodes of $R_0 = 0$. Opinions have values ranging from $O = [-10; +10]$. Transitive reputation is a function of opinions and reputation of respective nodes.

5.5.1. State-of-the-Art

Reputation has been studied in a variety of scientific areas, such as economics, anthropology, sociology, biology and computer science.

A significant amout of research effort has been put into file-sharing systems and how cooperative behavior can be stimulated by introducing reputation systems, see [**CDDCdV$^+$02**], [**LFC03**], and [**DDCdVP$^+$02**]. Kamvar et al. [**KSGM03**] have developed an algorithm for reputation management in Peer-to-peer networks. Additionally, application related reputation has been investigated by Moloney and Ginzboorg [**MG04**], but they require user interactions to establish reputations.

Another application of reputation systems with similar objectives are anonymity services, such as MIX systems [**Cha81**]. Dingledine proposes to rate MIXes with reputation systems in order to improve privacy [**DFH01, DMS03**]

Marti et al. propose to use reputation in support of routing decisions in [**MGLB00**]. This concept has been picked up by some others, like [**OKRK03**] and [**DDB04**]. Others have extended this idea and proposed remuneration systems to stimulate cooperativity in ad-hoc networks. Sprite [**ZCY03, ZCY02**] employs a centralized server, while [**BH00**] builds on tamper-resistant security modules. [**Fäh03, FOKR04**] introduce a concept based on a social model. Boudec and Buchegger have proposed a reputation infrastructure named CONFIDANT [**BB01, BB02b, BB02a, BLB03**] to penalize malicious nodes. Liu and Yang [**LY03**] analyzed methods to distribute reputation information in Mobile Ad-hoc NETworks. Michiardi and Molva have proposed a generic reputation infrastructure called CORE [**MM02, Mic04**]. CORE can be used either in the application or the network layers, although the design has been optimized for network layer.

Jøsang et al. [**JGK03**] analyse trust topologies and provide a notation to express trust relationships. This is especially interesting for reputation systems that use indirect reputation and require transitive trust models.

5.5.2. VARS - Vehicle Ad-hoc network Reputation System

Given the requirements from section 5.1.2 VARS - a Vehicle Ad-hoc network Reputation System - has been designed as a distributed approach to the improve security and reliability of traffic-related messaging. VARS is built with the assumptions and constraints in mind that have been presented in section 3.1, notably the availability of a tamper resistant device in each car. This is important because the core-algorithms of VARS are assumed to be executed on a secure computing platform running on such a device.

5.5.2.1. Architecture and Concepts of VARS. Within this paragraph the architecture of VARS will be presented. The main idea is that reputation values are appended to the messages as *partial opinions* by forwarding nodes. At some point in time an entity makes a decision on the confidence in the messages that announce a specific event. Based on the partial opinions it calculates a confidence level. Ideally, integration of this confidence value with other concepts such as local reasoning and voting schemes leads to a final confidence level, refer to section 5.7.

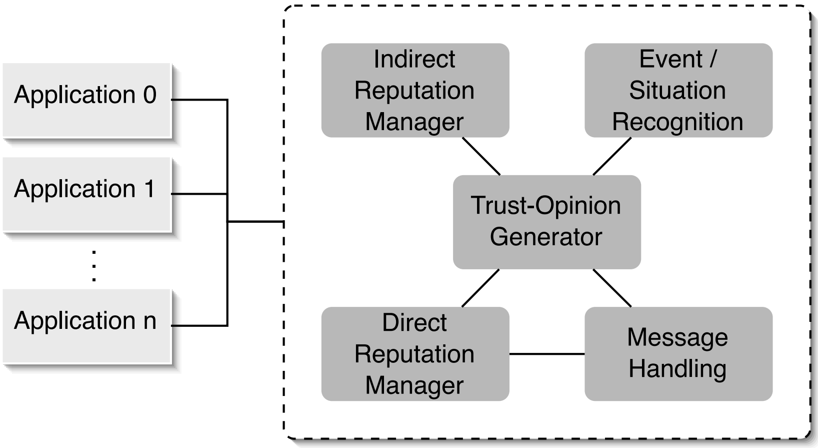

Application 0
Application 1
Application n
Indirect Reputation Manager
Event / Situation Recognition
Trust-Opinion Generator
Direct Reputation Manager
Message Handling

Figure 5.19. VARS Architecture

VARS is a modular approach (see Figure 5.19) that strictly separates direct, indirect reputation handling and trust-opinion generation. Further modules are needed for message handling and situation recognition:

Direct Reputation Manager: Management of reputation values that are generated from first hand experiences. Experiences are recognized by applications and then put in relation to messages and the message's sources to generate direct reputation. The Direct Reputation Manager has a subjective view on node reputation.

Indirect Reputation Manager: Management of reputation values from transitive second-hand reputation. The Indirect Reputation Manager calculates a global reputation value, its goal is to provide reputation values that are shared throughout the network.

Trust-Opinion Generator: Generates opinions on messages at evaluation time. Additionally, the Trust-Opinion Generator communicates trust decisions to the applications and gathers event notifications to justify direct reputation values.

Message Handling:

Collects reputation messages concerning specific events until the time for evaluation has come.

Situation Recognition:

Calculates the *situation dependent reputation level* (see section 5.5.2.1).

In order to exploit the properties of a car-based network that deals with traffic-related information, following new concepts have been integrated in VARS:

Delayed Decision: In the type-1 application scenarios a message is forwarded as long as it may be relevant for nodes within the network. What "relevant" means in this context depends on a multitude of parameters. For further details on this topic please refer to [**KSA02**]. To simplify matters it will be assumed that messages about one event will be distributed within a circular area, which will be called distribution area, see Figure 5.3 on page 81. It is important to understand, that a car's system will not display a received warning message at once if the car is still far from the recognition area within which an event can be recognized. This means that a car's reputation system will gather information about that specific event until the car reaches a certain distance from the recognition area, where the driver has to be informed if the message is found to be trustworthy. At this point, a final decision has to be made about whether the information received about an event is trustworthy enough to be presented to the driver. The term decision area refers to that area. Nodes that enter or are already within the decision area decide upon the trustworthiness of messages related to the according event. The size of this area again depends upon many parameters such as the type of event and speed of the car and topography of streets but these issues will not be addressed in this thesis. Nevertheless the size of event-, decision- and distribution area depends on the type of the event. The decision process of a node being within the decision area is straightforward: it will calculate the confidence in a received message at once. For a node outside the decision area (but inside the distribution area), this will be handled differently: it will store the message and collect more information about the event until it enters the decision area. This delayed decision process offers more time to gather information relevant for making decisions, but on the other side doesn't affect the node's ability to present critical information to its user as timely as needed.

It should be noted here that only nodes that are located within the recognition area can directly verify messages about that event and therefore put this information into their Direct Reputation Manager (see Figure 5.19). If nodes are outside the recognition area, they can only gather information regarding indirect reputation. This also means that non-trivial decisions about presenting an event to the driver can only be based on indirect reputation, because the information about an event is, in the vast majority of cases, only useful if not already experienced.

Opinion Piggybacking: A major challenge in designing a reputation system for large networks is to keep the overhead as low as possible. In the given context properties of communication patterns will be exploited to achieve this. The envisioned car messaging system will flood the packets in the distribution area.

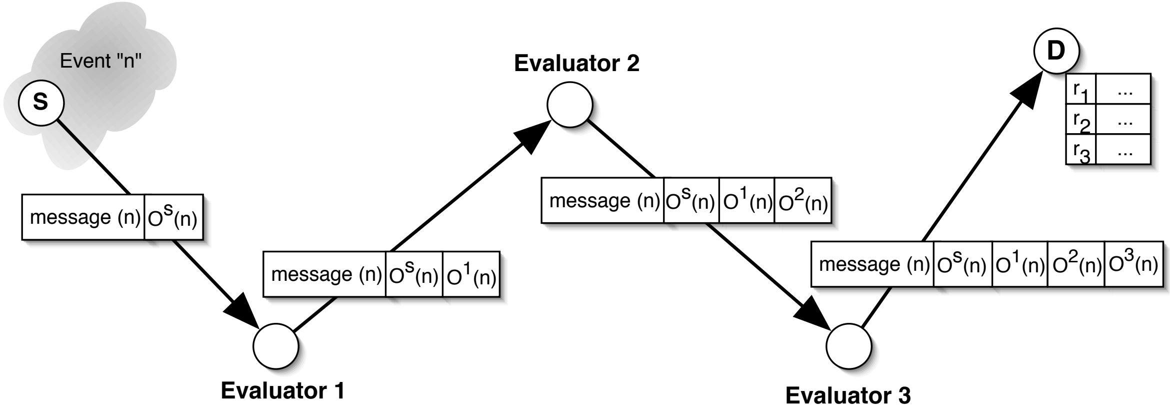

Event "n"
S
message (n) | $O^s(n)$
Evaluator 1
message (n) | $O^s(n)$ | $O^1(n)$
Evaluator 2
message (n) | $O^s(n)$ | $O^1(n)$ | $O^2(n)$
Evaluator 3
message (n) | $O^s(n)$ | $O^1(n)$ | $O^2(n)$ | $O^3(n)$
D
r_1 ...
r_2 ...
r_3 ...

Figure 5.20. Piggybacking of Reputation Information

The proposal is to append reputation related information to the message accordingly. See Figure 5.20: after the source node S sent its message, forwarding nodes (nodes 1,2 and 3) will add their own opinion about the message's creator and/or the message content (o^1, o^2 and o^3) to the message itself. Every node can use the information of its predecessor(s) and update its database. Finally the message reaches its destination (D)[12].

Assuming that there are enough messages, the overall reputation system will exchange sufficient information about participating nodes in a given area. Additional information might be sent using special control packets in times of very low communication traffic.

Geo-/Situation-Oriented Reputation Levels: A car is most of the time moving within a considerably small area, such as the hometown of its driver or the way to his working place. In general, other nodes - serving as sources of information within this area - are met often and thus reputation values of these nodes are stored. Considering this, one would like to set rather strict reputation thresholds to make it harder for an attacker to gain a reputation that is sufficient for an attack. On the other hand, such strict reputation thresholds would suppress all messages if the car is driven on a long distance through "unknown territory" where no familiar sources can be found. Thus it is crucial to adjust thresholds relative to the context of source and car.

An approach to overcome this problem is to use different kinds of geographic- and situation-oriented reputation levels. They reflect the context in which a sender is in relation to a receiving node. In the evaluated implementation four classes of geo situations have been defined:

[12] Note that there is no "final destination". Every forwarding node is a destination and then decides if the message has to be further distributed.

L**:** local areas: very small areas where most nodes know each other

C**:** city area: roughly a node's home zone where many nodes are known from first hand experience and almost every other source can be mapped to a reputation value through transitive reputation

R**:** highways: connections between cities, many nodes that are only connected for a short while and can't gather much reputation values within this time

F**:** unknown territory: same situation as "C" only without the receiver participating under normal conditions

Every geo-situation sets different thresholds for the confidence decision algorithm.

Sender Based Reputation Level: Evaluation of partial opinions is done with respect to three different types of sources of information. A node distinguishes between sources that are known from direct or indirect reputation or have not been previously known (only transitive trust). These source types are assigned sender-based reputation levels 1 to 3 respectively.

Pseudonym Flexibility: The need for privacy has been expressed as requirement 3.9 in section 3.4. In order to address this requirement of preventing a direct mapping between a user and its device, pseudonyms may be used instead of direct identifiers such as license plate numbers, names, etc. The frequency of change is subject to individual privacy preferences and manufacturer defaults. However, changes of pseudonyms must fulfill certain criteria to be effective, as discussed in section 4.3.4.

This is indeed a difficult trade off to make since a reputation system heavily depends on mapping observations to specific nodes. Picking a new identity / pseudonym means that all gained positive reputation is lost. As indicated before, this means that the system cannot use strong identities.

5.5.2.2. Algorithms / Realization. This section gives a short introduction to the most relevant algorithms used in the work. The major tasks to be performed are generating opinions, deciding about trust relations and updating the local databases.

On arrival of a message notifying an event every forwarding node generates an opinion on the trustworthiness of the message. An opinion is calculated either from direct experience if the event is detected, from direct trust if the sender is known, from partial opinions attached to the message or a combination thereof. This opinion is attached as another partial opinion to the message before forwarding it. This process is called *opinion generation.* It will be described in the following section.

If a node enters or already is within the decision area associated with the event in question, it has to calculate the confidence about this event required for a presentation decision.

The reputation system's memory space residing in a node is limited. Therefore, the information that is gathered cannot be stored forever, so there is a *replacement strategy* for obsolete entries. The strategy and the algorithms will be presented in paragraph 5.5.2.2.

Opinion Generation

Every forwarding node calculates a partial opinion and appends it to the message. A single partial opinion $o^{ID} = (o^{ID}_{val}, s^{ID}, ID)$ consists of an opinion value o_{val}, a sender based reputation level s and the identifier ID of the evaluator that generated this opinion.

The calculation of forwarding node i's opinion $o^i = (o^i_{val}, s^i, i)$ is calculated as follows. First, the opinion values of all previous partial opinions o^{ID}_{val} are combined with the reputation values of the corresponding partial opinion's sources r_{ID}. If there is no reputation value r_k for a partial opinion source k stored locally, neither indirect nor direct, the combined opinion value $o^i_{val,k}$ equals the opinion value o^k_{val} from k's partial opinion. In this case, the sender based reputation level of this combined opinion s^i_k is set to 3. If the forwarding node i has either a direct or an indirect reputation value r_k for the source k of the partial opinion, $o^i_{val,k}$ is calculated according to equation (5.1).

$o^i(n)$	Opinion on event n, calculated by node i
$o^i_{val}(n)$	Partial opinion, generated by evaluator i
$o^i_{val,k}(n)$	Partial opinion value generated by k on object n
s^i	Sender-based reputation level from evaluator i
r	Reputation
r_k	Direct or indirect reputation value of node k
r_{max}	Global maximum reputation value of node k
$sbrl()$	Sender-based reputation level
$\mathbb{PR}_x$	Set of all combined opinions with sender-based reputation level x
α	Factor for sender based reputation level 1
β	Factor for sender based reputation level 2
γ	Factor for sender based reputation level 3
$\mathbb{M}_n$	Set of messages concerning event n
T	Threshold of minimum message number referring to an even
q	Quality of a node
g_t	Number of good experiences during probation period t
t	Probation period
$\sum SET$	Sum of the elements' values in set SET
$\|SET\|$	Number of elements in set SET

Table 5.4. Variables and Parameters

$$o^i_{val,k} = r_k o^k_{val} + (r_{max} - o^k_{val})(r_{max} - r_k) \tag{5.1}$$

$$s^i_k = max(s^k, sbrl(k)) \tag{5.2}$$

r_k is the direct or indirect reputation value of node k. r_{max} is the global maximum reputation value.

The combined sender based reputation level for k is calculated as shown in equation (5.2). $sbrl(k)$ is 1 if the partial opinions source l is found within the direct reputation manager or 2 if found within the indirect reputation manager.

Depending on the combined sender based reputation level s^i_k the combined reputation values are weighted differently with the computation of the forwarding nodes opinion o^i_{val} which is shown in equation (5.3).

$$o^i_{val} = \alpha \frac{\sum \mathbb{PR}_1}{|\mathbb{PR}_1|} + \beta \frac{\sum \mathbb{PR}_2}{|\mathbb{PR}_2|} + \gamma \frac{\sum \mathbb{PR}_3}{|\mathbb{PR}_3|} \tag{5.3}$$

$$\alpha + \beta + \gamma = 1 \tag{5.4}$$

Where $\mathbb{PR}_x$ is the set of all combined opinions with sender based reputation level x. The sender based reputation level of this opinion is the lowest combined sender based reputation level that went into the calculation. The weights α, β and γ are constants that have to be set before the calculations. The sum of α, β and γ must be one, see equation (5.4). If the forwarding node has an direct or indirect reputation value on the sender of the message, this value is inserted into the calculation, either into $\mathbb{PR}_1$ or $\mathbb{PR}_2$.

The resulting partial opinion is then appended to the message and its previous opinions. The message may then be transmitted.

Presentation Decision

At the time a node has to evaluate all messages related to a distinct event (see section 5.5.2.1), it calculates a weighted middle value of all partial opinions according to equation 5.3, called the final reputation r_{fin}. The decision process consists of two parts. First, constraints based on the geo-/situation based reputation levels are checked. If the final reputation doesn't exceed a defined threshold or the number of messages announcing the concerned event is below a minimum number, the event is considered not to be true. Otherwise, another check is performed that discards an event if f as shown in equation (5.5) is equal or less than 0.

$$f = |\mathbb{M}_n| - \left\lfloor \frac{r_{max} - r_{fin}}{a_v} \right\rfloor - T \tag{5.5}$$

Where $\mathbb{M}_n$ is the set of messages announcing the event, $|\mathbb{M}_n|$ the number of these messages, a_v is a constant factor that reflects the importance of reputation values and $k > 0$ is a threshold of minimum messages that have to announce an event.

If the event is thought to be prevalent, the trust-opinion generator announces this event to the applications. It is in the application's discretion to inform the user or take actions.

Replacement Strategy

In section 4.2, the architecture has been introduced. There are two databases stored locally, one for direct reputations and one for indirect reputation. In order to keep only those entries, that are most valuable, for direct and indirect reputation the following schemes are used:

- **Direct Reputation**

  Every entry in the direct reputation database has a field called activity field. This field is an indicator for the number of experiences being made during a certain time frame, called probation period. It is called activity field since experiences can only be made with nodes that are actively sending out opinions or data packets. This field is used for the first operation of the replacement algorithm. The algorithm will look for inactive nodes. If such nodes are found, the one with the lowest direct reputation will be replaced. If no inactive node is found, each node's quality will be calculated:

$$q = \frac{r_{dir} * good_t}{t} \tag{5.6}$$

  Where q is quality, r_{dir} is direct reputation and $good_t$ is the number of good experiences during probation period t.

  New nodes have the following quality:

$$q = \frac{r_{max}}{t} \tag{5.7}$$

Where r_{max} is the maximum reputation value.

Then, the new node's quality is compared to those in the database. If the new node has a higher quality than any node in the database, it will replace the one with the least quality.

- **Indirect Reputation**

  For indirect reputation the situation is different, because there is no probation period and no activity field is stored here. The goal is to keep information about those nodes that provide the best opinions and get evaluated a lot themselves. In order to achieve this, each entry in the indirect reputation database has an evaluation field. This field is basically a counter that is incremented by one each time an opinion on the given node is received. But only those opinions are counted that are based on direct reputation or, in other words, the source-oriented reputation level is 3.

  First of all, the algorithm is searching for nodes with a zero in their evaluation field. If such nodes are found, the one with the lowest indirect reputation value will be replaced. If no such node is found, the algorithm will look for nodes that have been evaluated by nodes with a lower source-oriented reputation level than the source-oriented reputation level of the opinions appended to the message. Again if any such nodes are found, the one with the lowest indirect reputation value is replaced. Otherwise, no one will be replaced, the new node will be discarded.

5.5.3. Evaluation of VARS

During the evaluation of VARS some important points about the feasibility of such a system have become clearer. This paragraph is a short summary on the major strengths and weaknesses of this approach and provides a justification for the research of voting schemes that are introduced in the following section 5.6.

VARS has been designed to provide a self-organizing method for evaluation of events that relies on reputation values / opinions on nodes and messages they send. A much more differentiated evaluation is possible than with conventional signature schemes and it also overcomes the shortcomings of these methods by providing an information-oriented approach. The proposed system does not require a central entity, but requires only local algorithms, thus realizing a distributed system. However, even extensions and modifications of the basic concept could not overcome fundamental shortcomings of a reputation system and during simulations of the VARS implementation some additional issues have been identified.

5.5.3.1. Fundamental Shortcomings. Those are shortcomings of the basic principles and the architecture of a reputation system such as VARS.

(1) **Statements Based on Past Behavior**

Making statements about the current behavior by looking at past behavior is not always justified. There are two sides of this. The first one is something called "legacy hunter attack", referring to persons who maintain a high social status and sophisticated image of themselves to be able to attack their victims using their privileges. This can also happen in reputation systems: an attacking node deliberately showing good behavior and therefore receiving a higher reputation level to be able to attack the system more effectively. The second one is that even if a node showed negative behavior in the past (willingly or not), the message created by it may still be accurate and very important. There are cases where important messages may be ignored, because of the bad reputation of its originator.

(2) **Updates only during Hazards**
Updates are only possible when hazard occurs, there is no other way to generate opinions. This may lead to outdated reputation values (worsened by fading of reputation values over time).

(3) **Lack of Adaptivity**
Disappearance of hazards leads to bad reputation for last nodes (that recognized it).

(4) **Bandwidth Overhead**
The bandwidth overhead for transport of opinions has to be seen, especially in situations of high communication loads.

(5) **Scalability**
Scalability is a problem largely due to memory and computing power of nodes. Even if these factors have been defined as minor factors due to fast technical evolution, this places large requirements on used hardware thus increasing cost.

(6) **Problematic Pseudonym Changes**
Pseudonym Changes are affecting reputation negatively.

(7) **Publishing Reputation Ratings**
Publishing of reputation ratings offers attackers better opportunities than only distributing experiences, see [**Buc04**].

5.5.3.2. VARS Specific Shortcomings. Problems encountered during simulation or with specific VARS implementation:

(1) **Nodes in Unknown Areas**
Nodes that often move to unknown areas may pose a problem. If the fraction of those nodes becomes to high, the reputation system may not work in a satisfactory way.

(2) **Stability of Global Reputation**
Simulation shows that the proposed implementation does not necessarily reach a stable state.

(3) **Node Reputation Dominates Message Reputation**
Opinions on messages are only added if experience is made. Effectively VARS generating more node reputations than event confidence. There is also no differentiation of node and message-based reputation (direct reputation ↔ experiences cannot be distinguished).

(4) **Complexity of Simulation**
Collusion is hard to simulate, sophisticated attacker models would be required. No simulation of adaptivity = reaction to disappearing events and different models of reputation fading.

(5) **Connection Reputation and Priorization**
Motivation is based on an "unchangeable" system on TRM - an extension to provide higher priorities (see discussion in section 4.4.4.3) has not been evaluated so far.

5.6. Voting Schemes

If the majority of nodes is assumed to be known to each other at most of the time, reputation systems such as VARS may be used, see section 5.5.2 above. In VARS, the decision, whether to consider a reported hazard or not is mainly based on these individual trust relationships. The problems and shortcomings mentioned above have been substantiated in [**Ost05**] and lead to the evaluation of voting schemes. The main advantages of reputation systems of self-organization and decentralization are also valid for voting schemes. But voting schemes offer the additional advantages of simplicity, scalability and fast adaptivity to changing situations. VANETs are expected to become very large (see also [**BE04**]), which should allow for multiple experiences of the same hazard by different nodes, and therefore enable the compensation of false messages, provided that the number of attackers is sufficiently smaller than the number of honest nodes.[13] The main objective here is to minimize false decisions with respect to the existence of a traffic hazard. Applying voting schemes on the closed set of remote hazard information categories thereby enables the assessment of messages' plausibility. Since the correctness of traffic-related message contents cannot be guaranteed, the system should be designed to be robust against manipulations of those fields. To be more specific, *Byzantine Robustness* or at least *Byzantine Detection* according to the definitions in [**Per88**] is aspired, exploiting the large average number of available experiences.

5.6.1. Related Work

Some work on voting schemes has been done in the field of sensor networks, such as in [**KRH05**] to improve energy efficiency for sensor nodes. But even as sensor networks share some characteristics with VANETs, the objectives of voting schemes are quite different from the ones given in the context of this work. Golle et al. presented a theoretical framework in [**GGS04**] to collect and evaluate the information of received messages. They proposed to validate received data in VANETs, based on the most plausible explanation. However, the authors do not provide methods to assess the plausibility of local danger messages and furthermore assume perfect communication, neglecting delays, transmission losses and collisions in practical networks.

5.6.2. Security and Attack Scenarios

5.6.2.1. Sybil Attacks. The most obvious attack on voting schemes are *sybil attacks*, where one node in the network impersonates multiple nodes. For voting the effect is clear: a single node can overrule many other nodes simply by representing a multitude of non-existing nodes. Douceur named it sybil attack in [**Dou02**]. The effects of sybil attack can be prevented using credentials that cannot be generated by a node itself. However, this approach should be avoided in the type-1 scenario, for the reasons given above. The conceptual / abstract countermeasure is resource probing. A node is challenged to prove that it has some kind of resources, whose availability is physically limited for each node. This approach does not prevent that a node impersonates non-existing virtual nodes completely, but limits the number of faked virtual nodes to a small amount. This is sufficient for the context here. Usually nodes are asked to perform tasks that consume one of their limited resources such as computational power, memory, battery power, physical extensions, etc. for a given period. If implemented correctly, it is not feasible to perform this task for multiple identities within a given time. However, the performance of the equipment in a car may soon be topped by commercial equipment in such a way, that this countermeasure is not feasible. Therefore, the approach taken here is based on the two assumptions, that

(1) Each car has physical dimensions and consumes a certain amount of space. Since cars are not available infinitesimally small, there is a lower bound on physical dimensions.
(2) The position and time pairs' authenticity is fundamental for the architecture presented in this thesis, refer to section 4.4.3. The position of each sender can be determined for the time it sent the message.

[13] For completeness it has to be mentioned that there have to be countermeasures against *Sybil Attacks* (see section 5.6.2.1) where an attacking node tries represent multiple identities to manipulate voting schemes.

Following these assumptions, a node cannot represent multiple nodes at different places at the same time. Representing different nodes at different times at different places (which it must have been visited itself) does not lead to a privileged attack setting. The remaining option for an attacking node is to represent multiple nodes at the same time at the same place. This however will be automatically be detected by receiving nodes' reasoning unit and discarded. Therefore, if the assumptions hold, this provides an effective measure against sybil attacks in combination with voting schemes.

5.6.2.2. Attack Scenarios. The assumption is that a subset of the active nodes misbehaves, some experiencing technical errors and others performing some kind of attack. Since the only (allowed) possibility to manipulate is the message item containing the type of experience, only malicious data attacks are considered. Two different attacks have been identified in this context, namely the *fake attack* and the *flip attack*, which will be explained subsequently. For both attacks, the attackers do only cooperate implicitly, by acting equally in identical situations, and do not explicitly collude, e.g. by driving in convoys. During the *fake attack*, every attacker is creating a warning message whenever he enters the recognition area of a fictitious hazard. All attackers thereby share the same fictitious hazard. Since that hazard cannot be detected by honest participants, these will create a revocation message once they enter the recognition area. The goal for this attack is to trigger a false positive decision within the attacked vehicles. When conducting a *flip attack*, attackers invert the type of experience included in their messages created, whenever entering the recognition area of an actual hazard. Thus, when attackers actually detect the presence of a hazard, they will send a revocation message, while detecting the absence of a previously existing hazard will result in a warning message. The goal of this attack is to reach false negative decisions at the victims whenever a hazard is present and to reach false positive decisions when that hazard has disappeared.

5.6.3. Criteria for Voting Methods

The three main criteria considered for the choice of voting methods are:

(1) **Reactivity**
Communication environment and traffic situation are highly dynamic. Traffic-related situations are continuously changing, new ones appear and others disappear. The voting method should should be able to react quickly to those changes. For example, the time it takes until all approaching cars detect the disappearance of a previously reported hazard should be as short as possible, thus minimizing the amount of wrong decisions.

(2) **Robustness**
Received messages may not always reflect the current state of the environment, independently if caused accidentally or deliberately. It is crucial that voting methods are as robust as possible against messages with wrong / obsolete information to enable correct decisions.

(3) **Scalability**
Depending on the specifics of a certain hazard and the traffic scenario, the number of event detections with respect to a given hazard may vary significantly. This in turn directly corresponds to the number of messages which can be utilized by receiving nodes' voting process. The voting method should perform equally well for most situations.

5.6.3.1. Voting Methods. The idea is having lightweight voting methods which estimate the plausibility of a reported hazard solely by performing voting schemes on the corresponding received local danger messages.

Four basic voting methods have been implemented and simulated, see also [**Ost05**]:

(1) **Freshest Message:** When a decision has to be reached, only the most recent message of a hazard is considered. If it is a warning message, then a positive decision is reached, if it is a revocation message, a negative decision is made. It is assumed that this decision method will not provide protection against adversaries, however, it should achieve a high adaptivity in attacker-free scenarios, resulting in only few false decisions.

(2) **Majority Wins:** This decision method performs a local voting over all received messages regarding a certain hazard. Duplicates are not considered, hence only distinct messages are counted. If the majority of the messages are warnings, then a positive decision will be reached, otherwise a negative decision is made. It is assumed that this decision method provides a high robustness against attacks.

(3) **Majority of Freshest X:** This decision method is a combination of the previous two methods. To reach a decision, a vehicle will perform a voting, considering only the recent x distinct messages, regarding the hazard in question.

(4) **Majority of Freshest X with Threshold:** Finally, extending the previous decision method with a threshold check results in this decision method. Thereby, it is checked if the distinct messages received so far exceed a certain threshold. If this is not the case, a negative decision is reached for the hazard in question, otherwise the result of the decision process is determined by *Majority of Freshest X.*

5.6.4. Tiered Trust

Nodes may be assigned different trust levels (defined separately for each application-class). Examples are fixed nodes that are maintained and supervised by public authorities or emergency vehicles that use daily updated digital certificates - provided that access to infrastructure exists and location privacy is no concern. This kind of nodes can therefore qualify as special information carriers or event detectors with an elevated trust level. Additionally, information from centralized trustworthy services, such as TMC or broadcasted digital information can be integrated into voting schemes by assigment of a higher trust factor. Benjamin Weyl has developed a more sophisticated model based on the same idea in [**Wey07**].

5.6.5. Evaluation of Voting Schemes

The simulative analysis and its key results will be presented in section 8.7.

In general, simulation results indicate that even simple voting methods, based on the number of received distinct messages, may provide sufficient protection against attackers, even colluding ones. The voting method *Majority of Freshest X with Threshold* achieves the best overall results. However, the optimal setting of its two parameters X and threshold has to be investigated for different traffic situations. A distinction between traffic scenarios and adaptation to the actual one would be beneficial for the voting mechanism.

One important result from simulation is that false decisions mostly appear at the adaption phase, the time between a traffic-related situation has disappeared and the majority of nodes has recognized this. The duration of the adaption phase independent from the overall attack time. This leads to the conclusion, that attacks are most effective during the adaption phase or put differently, attacks which last longer produce a lower percentage of false decisions, while shorter attacks increase the percentage of false decisions.

With respect to scalability, a key advantage of voting schemes, it seems likely that all decision methods can be utilized with an increasing number of vehicles, since the number of relevant messages is limited by an adaptive message lifetime.

5.7. Confidence Level and Decisions

After elaborating different methods of ”trust establishment” - or better confidence establishment - in previous sections, the factors for confidence level calculation and decision making are presented in this section. Remember that an overall confidence level is calculated for each event separately upon receiving a new message. It is used for the forwarding decision of every corresponding message. The value is stored in the local database and updated each time a new message is received, except if the new message does not provide additional information. The confidence level is also calculated when entering the decision area for the presentation decision.

Both decisions are a comparison of the calculated overall confidence level with a calculated threshold. If the confidence level is higher than the threshold, the decision is positive and message will be forwarded or presented to the driver via HMI, depending on the kind of decision to be made.

5.7.1. Factors for Conficence Calculation

Effectively there are four factors to be considered for confidence calculation:

(1) **Anonymous Credentials**
The issuance of anonymous credentials and the check of message signatures limits the number of nodes able to participate in the system, see also node authentication. The key question here is whether messages from unauthorized nodes should be considered or not. If node authentication can be omitted, this offers the advantage of allowing non-authorized nodes to provide additional information about traffic situations. Since node authentication is required for type-2 and type-3 scenarios anyway and will therefore be available, it can be applied when necessary. Note however, that this is a system-wide decision.

(2) **Local Reasoning**
Local reasoning and plausibility checking will be used whenever possible, because it is the most effective method to distinguish between reasonable data and abnormal patterns. It is used to analyze local sensor data as well as plausibility of received messages. On one hand, the efficiency of local reasoning is rather limited if locally measurable data correlates only little with the data at the recognition area in question and stored data does not provide significant for an effective judgement. But on the other hand, local reasoning is independent from the number of nodes involved in the process.

(3) **Reputation - VARS**
VARS as an implementation of reputation systems provides opinions on nodes and messages

that can be used to support confidence decisions. Considering the pro's and con's of VARS it can be helpful in some occasions while others are problematic. The key is to know when reputation data is valuable and when it is not.[14]

(4) **Voting Schemes**
Voting schemes are an approach to exploit the multiplicity of data retrieval and compare them. If a sufficiently large number of nodes provides data on a given even, this provides a relatively robust and flexible way of judging information. Also, many attacks are very hard to execute and have a limited impact on the overall system in practical use-cases.

Different schemes of combining different factors for confidence calculation have been discussed. When looking at simulation results, it seems that the best approach is to implement a mandatory check of credentials. After credentials and message signatures have been verified positively, a combination of local reasoning and voting schemes should be used to calculate a confidence level. Unfortunately, this could not be tested due to a lack of working reasoning implementations.

5.7.2. Threshold Factors

The threshold that is calculated for each decision depends on different factors. First, it depends on the impact that a positive decision has on the behavior of the driver or on the message's distribution. Second, the current situation is another factor, e.g. in critical situations information has to have a higher level of confidence to be presented to the driver. Third, there should be a possibility for drivers to define their personal preferences with respect to a confidence level threshold. For instance one type of driver may want to have as much information as possible, neglecting the trustworthiness, while another type of driver does not want to be bothered by a messaging system except for information with a high degree of confidence.

5.8. Conclusion

This chapter starts with a description of the traffic-related messaging process and a classification of traffic-related applications. The classification is the basis for the concept of event classes. It allows to address the large diversity of events using a uniform methodology, while retaining the flexibility to integrate new classes. An important topic in this context is the distributed aggregation of information about an area event, solved by a newly designed algorithm that has been chosen for its simplicity and efficiency. The main part of this section however, is dedicated to the question of how to decide whether information about traffic-related situations is correct or not. Table 5.5 presents a qualitative summary of strengths and weaknesses of different approaches.

As the overview shows, no approach covers all criteria, making a combination of different approaches according to the importance of each criterion the best choice.

[14] One approach would be to look at reputation level convergence, that is to see how reputation values for a given node or event converge towards a certain value. Clearly, in that case future research on this topic would be required.

	Anonymous Credentials	Local Reasoning	Reputation System	Voting Scheme
Scalability	o	++	o	+
Required Penetration Rate	++	++	o	o
Susceptible to Collusion Attacks	+	+	-	o
Reactivity	++	+	o	+
Pseudonymity	+	++	o	++
Organizational Complexity	-	+	+	+
Significance outside recognition area	+	-	+	+

Table 5.5. Comparison of "Confidence Evaluation" Schemes

CHAPTER 6

Vehicle to Infrastructure Scenario

This chapter deals with vehicle to infrastructure communication and associated applications. The generic case is a communication between a mobile node and a fixed node. Fixed nodes that also offer gateway functionality, that is access to other networks, will be called hotspots. Fixed nodes will be seen here as "infrastructure" due to its immobility and distinct operation / funding from mobile nodes.[1] The chapter is organized as follows: in the first section typical use-cases will be described, while the second section defines communication patterns, presents requirements and reviews V2I-specific security. The third section introduces a modified handshake protocol to establish connections between mobile and fixed nodes that meets given security requirements and presents the traffic light example to discusses practical implications. The last section presents an overview of the results and a conclusion of the work done in this field.

6.1. Use case and Applications

What differentiates this type from type-1 traffic-related applications is the following:

(1) **Other Trust Model** Fixed nodes may be better protected against manipulation, can be checked regularly if they are professionally operated and do not have privacy limitations. Conventional security approaches using digital credentials can be used to their full scope. Therefore, depending on these circumstances, specific fixed nodes may be assigned a higher trust level than ordinary mobile nodes.

(2) **Direct Communication** Only direct communication will be regarded here. Looking at store-and-forward communication between a mobile node and a fixed node does not add new insights, since at some point there has to be a mobile node communicating directly to a fixed node. Also multi-hop connections will not be regarded in the basic scenario, since the multi-hop scenario is straight-forward if an appropriate type-3 protocol is available, which will be discussed in the following chapter 7.

(3) **Gateway Functionality** Some fixed nodes may also offer connection to other networks, such as the internet. This opens up a huge field of applications, which will only be treated here briefly.

(4) **Fixed Location** Their fixed location makes fixed nodes ideally suited to gather location specific information to support type-1 applications[2].

In a VANET environment, the mobile node is typically represented by a car and the fixed node by a RSU or hotspot. A traffic light scenario has been chosen as an example, because it provides all major

[1] Distinct operation and funding means that entities operating mobile nodes will usually assume fixed nodes "to be there", without specific investment. Clearly, this depends on the business model, but it seems likely that mobile nodes and fixed nodes will mostly be funded and operated by different entities.

[2] Simple types of fixed nodes may be located on purpose at dangerous areas or areas with a high probability of local danger situations.

characteristics for research. Additionally, most requirements for other infrastructures such as public hotspots apply here as well and the protocol presented in section 6.3 will cover these scenarios.

To give an idea which use cases exist for vehicle to infrastructure communication within the scope of this work, some examples will be given. They are reasonably grouped according to their connectivity:

(1) **Gateway Type (Hotspot)**
  - Gas Station Hotspot: Typical hotspots that provide a connection to the internet and offer a variety of connectivity services. The use case is mostly focused on internet connectivity and services building on that.
  - Traffic Infrastructure: Connected road-side infrastructure, such as toll stations and traffic management systems. In addition to coordinating functions they offer centralized traffic information, access to toll accounting and other dedicated services that require connected units. Traffic-lights connected to central control systems are a special case, since this type of connection is normally not directly linked to mobile node / fixed node transactions.

(2) **Isolated Type (Road-Side Unit)**
  - Typical Road-Side Unit (RSU): A "typical" RSU provides either an information service or some kind of message storage/exchange support. If placed in critical locations, simple RSUs' purpose can be simply warning approaching nodes about static conditions such as low bridges, curve speeds, etc. or about dynamically changing situations such as lateral wind, ice, etc. They can also assist in critical traffic situations, such as pedestrian crossing, left turns, lane merging, etc.
  - Infobox: Basically providing the same functionality as the aforementioned RSU box, the focus is on non-traffic information. Examples are commercial advertising, city-information boxes, message-storage devices for local or geo-tagged messaging and so on. Note that updating the information on such low-cost units without backend-connectivity could ideally be achieved by exploiting the store-and-forward feature of mobile nodes.

The gateway functionality itself is not focus of this thesis and will in general be omitted in the following sections. However, gateway-type infrastructure may provide access to a security backend. In this case, connection and relevant data transfer is assumed to function without additional steps to be performed. Note that this functionality will mainly be used in the following chapter 7.

6.2. Communication Patterns, Requirements and Security

6.2.1. Communication Patterns

In contrast to traffic-related communication, where only asynchronous broadcasts[3] are used, in vehicle to infrastructure communication following communication patterns are used:

(1) **Asynchronous Broadcast** from fixed node to all mobile nodes within communication range or within a destination area, such as a traffic light broadcasting status information or a HotSpot broadcasting a list of services being offered.

(2) **Asynchronous Unicast** from fixed node to one specific mobile node, such as mid-air messaging.

[3] On a logical level, as it has been pointed out in definition 5

(3) **Asynchronous Unicast** from mobile node to fixed node, such as traffic information push, etc.

(4) **Synchronous Unicast** between mobile and fixed node, such as emergency vehicle controlling traffic light phases.

6.2.2. Requirements

In addition to the general requirements that have been introduced in section 2.5 there are specific requirements for type-2 applications:

Requirement 6.1 (Authentication of Fixed Nodes)**.** *Fixed nodes are, as indicated above, in a position to distribute critical information due to their fixed location and a, sometimes, higher trust-level. Trusting such information, such as traffic-light status information given as an example in section 6.3.3 requires proper authentication of those nodes.*

Requirement 6.2 (Time Synchronisation)**.** *The absence of direct satellite positioning and timing information for most fixed nodes (largely due to cost reasons) demands a protocol that ensures time synchronisation and freshness between mobile and fixed nodes for synchronous communication.*

Requirement 6.3 (Fast Connection Establishment)**.** *A handshake for synchronous communication has to be designed in such a way that it is sufficiently fast for mobile nodes passing by a fixed node in high speed. The time available has to suffice for secure connection establishment (as discussed in section 6.3), payload-data transfer and connection break-down (if required). This means the less time is required for connection establishment, the more time is available for application data to be transferred.*

6.2.3. Security

A general formulation of security goals is that the sender of asynchronous information must be authenticated and the messages' integrity has to be checked. Nodes that have been successfully authenticated may be assigned certain attributes or rights. One of the requirements demands that freshness is provided. Without using a globally synchronized time, only an interactive protocol can achieve this, which will be pointed out in the following section. For synchronous communication privacy becomes relevant, because fixed infrastructure (or at least the property of a fixed / known location) may be abused to gather information about mobile nodes. It is desirable that communication partners' identities remain hidden from all surrounding nodes and that message contents can only be read by authorized receiver(s).

The approach presented in the next section achieves node authenticity, availability and freshness, as well as authorization (using node authentication) for asynchronous broadcast from fixed node to mobile nodes. Pseudonymity may also be provided, but this requires a more comprehensive approach avoiding identifyable information. All given requirements are fulfilled for synchronous unicast between mobile and fixed node. Asynchrouous unicast from mobile node to fixed node and vice versa are similar to type-1 messaging and will therefore not be discussed specifically in this chapter.

6.3. V2I Protocol

Following the objectives in previous sections, an interactive protocol has been developed for asynchronous broadcast and synchronous unicast cases. In order to establish a secure communication channel, a session key has to be established (such protocols are called key exchange protocols) which is then used to protect the vehicle to infrastructure connection.

6.3.1. Fundamentals Key Exchange Protocols

Key exchange protocols are used to create a shared, symmetric, secret, cryptographic key that can be used by communication partners to secure their connection. For an overview on existing key exchange protocols and mathematical background refer to [**Sch96**] or [**MOV96**]. The Diffie-Hellman protocol is a key exchange protocol that is well suited for the given use cases, because:

(1) The ingenious approach of distributed key generation (only parameters are transmitted) allows the establishment of a protocol meeting the important security objective of link anonymity 3.10.
(2) It requires[4] the use of credentials for full node authentication, refer to requirement 6.1
(3) It is an interactive protocol, allowing to guarantee freshness without assuming a secure time base available to all participants, see requirement 6.2
(4) The key exchange mechanism itself relies on algorithms that can be executed quickly, thus meeting requirement 6.3

6.3.1.1. Diffie-Hellman Key Exchange Protocol. The Diffie-Hellman key exchange is a cryptographic protocol that allows two parties that have no prior knowledge of each other to establish a shared secret key over an insecure communication channel. Such a key can be used to encrypt subsequent communications using symmetric key encryption. The key will typically be used as a session key, that is for a single session. The protocol has been introduced in [**DH76**]. The protocol's security depends on the discrete logarithm problem, assuming that it is computationally infeasible to calculate a shared secret key $K = g^{xy} \pmod{p}$ given the two public values $X = g^x \pmod{p}$ and $Y = g^y \pmod{p}$ when prime p is sufficiently large. p and g are public system parameters, where p is a prime number and g is a generator such that $\forall \quad 1 < n \leq p-1 \quad \exists \quad k : n = g^k \pmod{p}$. Suppose Bob and Alice are two communication partners that want to establish a secret key. First, Alice generates a random private value x and Bob generates a random private value y (both integers). Then they derive their public values using the public system parameter p and g, calculating: $X = g^x \pmod{p}$ (Alice) and $Y = g^y \pmod{p}$ (Bob). Then they exchange the public values X and Y. Alice computes $K = Y^x \pmod{p} = (g^y)^x \pmod{p} = g^{yx} \pmod{p}$ and Bob computes $K = X^y \pmod{p} = (g^x)^y \pmod{p} = g^{xy} \pmod{p}$, which gives both the same shared secret key K.

6.3.1.2. Station to Station Protocol. The station-to-station protocol, which has been introduced by Diffie, van Oorschot and Wiener to establish a shared key [**DVOW92**] is an extension to the Diffie-Hellman key exchange protocol introduced in the previous paragraph. The Diffie-Hellman protocol is susceptible to identity theft and man-in-the-middle attacks (please refer to [**Eck03**], [**Sch96**] or [**MOV96**] for a description). This is why the station-to-station protocol extends the Diffie-Hellman protocol with node authentication. The public Diffie-Hellman parameters are signed with asymmetric keys from each side and corresponding credentials are provided.

[4] To be exact, that is not true for the Diffie-Hellman protocol, but for its extension station-to-station protocol. This will be discussed in succeeding paragraphs.

6.3.2. Communication Protocol

Following requirement 3.1 given in section 3.2 the solution must be public. Interestingly, some other developments, not reflecting this principle, have shown that keeping the system design secret does not make the system secure, e.g. [**Upt03**].

The communication between a mobile node (car) and a fixed node (traffic light) introduces a couple of security related requirements as shown above. In the following paragraphs, two cases will be regarded: first, sending of status information from the fixed node to unspecified mobile nodes (asynchronous broadcast) and second, the set-up of a secure communication channel between a fixed and a mobile node (synchronous unicast).

In the first case, where fixed nodes send their status information, it is necessary to guarantee the integrity of the data packet containing the information and to ensure the authenticity of the sender. Since symmetric key approaches are difficult to handle in terms of key management, asymmetric cryptography with digital signatures has been used. In addition to that, credentials are employed to prove that the sender is indeed the correct traffic light. In order to provide pseudonymity, pseudonyms using a strawman-authority as presented in section 4.3.2 are used.

The second case, such as emergency vehicles sending control messages to traffic lights, is more difficult. First, authenticity and integrity of control messages have to be checked and the authorisation of the sender has to be verified. Second, all identification information provided by the sender has to be hidden from nodes other than the authenticated traffic light. Third, the freshness of the control message must be ensured, in order to prevent replay attacks.

M_s	Periodic status message
$Sig_{Car}\{x\}$	Signature of ”x”, created by ”Car”
$Cred_Lights$	Credential of ”Lights”
X, Y, p, g	Public Diffie-Hellman key parameters
x, y	Secret Diffie-Hellman key parameters
$K = g^xy \pmod p$	Symmetric session key
$Crypt_K\{x\}$	Encryption of ”x” using key ”K”
M_C	Control message
$M_C - ACK$	Acknowledgement for control message M_C

Table 6.1. Variables and Parameters for Protocol

For this case, a solution according to Figure 6.1 is proposed. Every mobile node (e.g. an emergency vehicle) carries an identification tag[5] and a credential which identifies it as a node with specific rights and privileges. An example would be being authorized to control traffic lights. The credential provides authorization of the mobile node sending a message, (e.g. a control message), while signatures ensure authentication and integrity.

Prior to sending such a message, fixed node (e.g. traffic light) and mobile node (e.g. emergency vehicle) agree on a session key using the Diffie Hellman protocol [**DH76**]. By encrypting all identity related information with the session key, the mobile node can maintain its privacy to all listening nodes, except for the fixed node it is addressing.

[5] This may as well be a pseudonym.

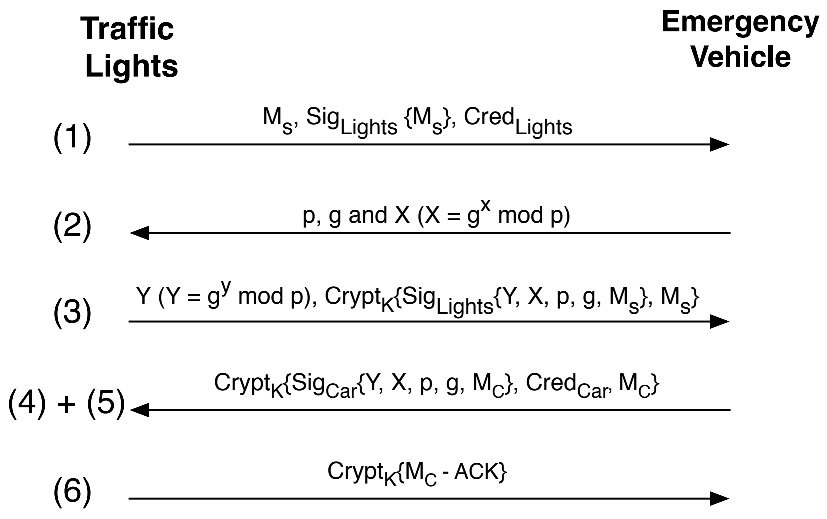

Traffic Lights
Emergency Vehicle
(1)
M$_s$, Sig$_{Lights}$ {M$_s$}, Cred$_{Lights}$
(2)
p, g and X (X = g^x mod p)
(3)
Y (Y = g^y mod p), Crypt$_K${Sig$_{Lights}${Y, X, p, g, M$_s$}, M$_s$}
(4) + (5)
Crypt$_K${Sig$_{Car}${Y, X, p, g, M$_C$}, Cred$_{Car}$, M$_C$}
(6)
Crypt$_K${M$_C$ - ACK}

Figure 6.1. Vehicle to Infrastructure Communication Protocol

6.3.3. Traffic Light Example and Practical Considerations

One of the main goals of IVC is to significantly reduce the number of traffic accidents. At intersections, the trajectories of traffic participants are crossing each other and therefore need special rules to prevent accidents. In Germany in 2001, more than 30% of all accidents, i.e. more than 800.000 accidents happened at intersections[6]. By intelligently controlling signalling at intersections, providing additional information to the driver and warning the driver in critical situations, there is a potential to reduce the number of accidents at intersections.

The approach for a communication protocol will be exemplarily demonstrated by interaction functions between (emergency) vehicles and traffic lights. Two cases will be regarded. First, the traffic lights broadcast their current state and switching times, providing the driver with up-to-date status information and assist the driver with red light / right of way violation warnings. Using an authentication process, every receiving vehicle is able to verify the information before presenting it to the driver, thus avoiding manipulations of the traffic flow through manipulated messages.

The second case assumes that emergency vehicles (or other publicly authorized vehicles) send control messages to the traffic lights in order to actively influence their current state, especially for traffic light preemption. To reduce the number of vehicles potentially hindering a free way for the emergency vehicle, this communication allows each intersection to optimize the traffic flow before an emergency vehicle arrives, according to the expected arrival time of the emergency vehicle and the current local traffic situation. Consequently, emergency vehicles can reach their destination quicker and safer.

6.3.4. Protocol Modifications

Based on the station to station protocol from section 6.3.1.2 a commmunication protocol has been designed to meet the specific needs of the given scenario. It has been modified with respect to following points:

- The Diffie-Hellman parameters p and q are included in the signature to be independent from retrieving those parameters from credentials.

[6] according to the German Federal Accident Statistics (*Bundesunfallstatistik*, right of way violation plus turning accidents)

- After establishment of a shared session key, all messages are encrypted, which protects privacy-relevant data, such as mobile nodes' identities. The only message that is sent from a mobile node before that are Diffie-Hellman key parameters, which contain no critical information with respect to privacy concerns.
- A dedicated credential is not required in the protocol, because it is received together with periodic status messages from the fixed node.

Note that the way it is designed, the modified protocol is ready-for-use in multihop connections.

6.3.5. Protocol Walk-through

As indicated before, the protocol has been designed for a generic case of symmetric unicast between mobile and fixed nodes. For a better understanding the example of emergency vehicles being able to control traffic lights using IVC will be used here, where necessary.

The protocol according to Figure 6.1 works as follows:

(1) The traffic lights are transmitting periodic status messages M_s. Such a message contains information about the current status of a traffic light system at a given crossing. The message is digitally signed. In addition to that, a credential will be sent together with a message, proving that the sender really is an authentic and valid traffic signal system. However, to reduce overhead, the credential may be sent together only with every n-th message, where n depends on the settings of the intersection.

(2) An emergency vehicle that wants to send a control message to a traffic light will at first generate some random values and generate the first half of the Diffie-Hellman key parameters to the traffic light.

(3) The traffic light system answers with a message consisting of two parts. The first part is the second half of the Diffie-Hellman key parameters. After the first part of the message, which is unencrypted, the second part and all following messages are encrypted with the newly generated Diffie-Hellman session key. The second part contains a fresh status message plus a signature of this status message and the public Diffie-Hellman parameters. There are two reasons for including a signature in the encrypted message. First, it is part of the mutual authentication process, proving that the Diffie Hellman parameters are the ones used by the traffic lights. Second, encryption alone does not provide integrity [**Sch96**], this can be achieved by a signature[7].

(4) After having received the second half of the Diffie-Hellman key parameters, the emergency car can also calculate the session key and decrypt the rest of the received message. Now it sends a message that is encrypted with the session key, containing the control message, the emergency car's credential and a signature of the control message and, once more, the public Diffie-Hellman parameters.

(5) The traffic light system can now finally decrypt the message upon reception, verify the signature and check the credential. The credential contains some form of identifier for the emergency

[7] Note that signing ciphertext may open up some attacks. Therefore it is important to sign first and then encrypt subsequently.

vehicle as well as its permission to send control messages. This permission might also be very fine-graded, depending on what the vehicle is allowed to do[8].

(6) Finally the traffic light system sends out an encrypted acknowledgment.

6.4. Conclusion and Security Evaluation

In this chapter, after determining specifc requirements, a secure key exchange protocol has been designed, based on the station-to-station protocol. Freshness (SO-9 requirement 3.13) is provided through the interactive character of this protocol. This assures link anonymity (SO-6 requirement 3.10) towards third party observers while providing node authenticity (SO-2 requirement 3.6) (or pseudonymity (SO-5 requirement 3.9) when using anonymous credentials) towards the intended communication partner. After establishment of a session key, link confidentiality (SO-7 requirement 3.11) is guaranteed. However, denial-of-service attacks could potentially affect availability (SO-3 requirement 3.7). As a countermeasure, intrusion detection methods could be evaluated in a future research effort.

An implementation of the presented protocol in the context of a traffic-light scenario will be described in section 8.3.

There are a some possible improvements to the concepts presented above:

(1) Forwarded individual route information can help to increase traffic efficiency. Also, sensors buried under the tarmac in the vicinity of traffic lights may be avoided if this information is transmitted using V2I communications.

(2) The concept of an ad-hoc network backbone can be used to deliver large amounts of data to isolated-type fixed nodes using store-and-forward.

(3) Timo Kosch presented the idea of using traffic lights as message forwarding nodes in [**Kos05**]. Especially in highly obstructed city environments, traffic lights would be ideally positioned at the intersection of road traffic.

[8] One may think about limiting the privileges geographically, etc.

CHAPTER 7

Connection-Oriented Communication Scenario

This chapter introduces the security methods for Connection Oriented Communication – COC in VANETs and presents a protocol, AODV-SEC that provides security measures to achieve the security objectives defined in section 3.4. Connection oriented communication implies that the destination node is known to a sender (source node) and therefore a (non-pseudonymous) certificate-based approach has been chosen for this type of communication. However, the limited access to a security back end during the operational phase a VANET requires some modifications to conventional approaches and will be discussed in this chapter. As pointed out in section 2.4, typical applications for this communication type are comfort and traffic efficiency applications (Platooning, Cooperative Cruise Control, Adaptive Chassis Control, Adaptive Drivetrain Management, etc.) and infotainment communications (Dynamic Grouping, Information Services, Teleconferencing, File Subscription, Instant Messaging, Multihop Cellphone Forwarding, etc.), mainly for groups of cars moving in the same direction.

This chapter is structured as follows: In the first section, a short introduction to Mobile Ad-hoc NETwork routing in general is given and two protocols that are well suited for use in Vehicle Ad-hoc NETworks are introduced. Then, security implications are explained, state-of-the art secure routing protocols are discussed in detail and related work is presented. In the second section, the results of a performance evaluation of chosen secure routing protocols are given, based on simulations testing those protocols in an automotive environment. Section three introduces a specially designed protocol - AODV-SEC. Section four concludes this chapter.

Note that unlike traffic-related messages, messages in the connection-oriented scenario may be larger and therefore be broken into smaller pieces, called packets. Descriptions in this chapter will therefore use the term packets to reflect this.

7.1. Secure Routing in Ad-hoc Networks

Ad-hoc networks have found diverse fields of applications such as interconnecting electronic devices at home or army units on the battlefield. If nodes have a high degree of mobility, routing protocols must accommodate frequent topology changes and rely on potentially unknown intermediate nodes.

7.1.1. Mobile Ad-hoc Network Routing

For messages to reach their destination in a multi-hop scenario, a routing protocol is necessary. A routing protocol has to create, maintain and re-create routes which should be as stable as possible in a dynamically changing environment. It basically constists of two parts: finding routes between source-destination pairs and delivery of messages to their destination. The main issue of mobile ad-hoc network routing protocols is to find the right balance between the ability to quickly establish routes based on actual information and to reduce signalling overhead to prevent collisions on the wireless channel and minimize consumption of limited bandwidth.

Existing routing protocols do not work very well in such an environment. More specifically, the Routing Information Protocol (RIP) [**Mal98**] suffers from slow convergence and the well-known count-to-infinity problem. The Open Shortest Path First (OSPF) [**Moy98**] converges only slightly faster than RIP and also consumes a lot of bandwidth.

There are a number of choices to be made when designing a Mobile Ad-hoc NETwork routing protocol. The protocol has to be adaptive (versus static) and distributed (versus centralized.) to meet the specific requirements of this kind of network. The remaining design choices are between reactive and pro-active protocols, the method of distributing routing information and the way routing decisions are taken. The most important design goals in the context of this work are: low overhead, quick adaptation, reliability, security and scalability. Existing routing protocols will be grouped to either pro-active or reactive routing protocols, since this has the strongest effect on the tradeoff between routing overhead and responsiveness in the given context. In general, pro-active protocols work well in environments where many new routes are created, but network topologies change rarely compared to that. Reactive protocols therefore are better for environments with frequent topology changes and relatively few routes to be established. This thesis will focus on the latter ones, however both variants are explained for better understanding.

7.1.1.1. Pro-active Routing. The concept of proactive routing means that all nodes exchange routing information periodically, or whenever the network topology changes, in order to maintain a consistent, complete and up-to-date view of the network. Each node uses the exchanged route information to calculate the costs to reach all possible destinations. This strategy avoids delays associated with finding routes on-demand. Pro-active routing protocols can be divided into distance vector and link-state routing.

- **Distance vector routing protocols** Each node maintains a vector with the distance to all nodes and periodically broadcasts this vector to each of its neighbors. Each node updates its own routing table by calculating the shortest path to each node using the distance vectors received from others. Examples are: Routing Information Protocol (RIP) [**Mal98**], Destination-Sequenced Distance Vector (DSDV) [**PB94**] and Wireless Routing Protocol (WRP) [**MGLA99**].
- **Link-state routing protocols** Each node maintains the state of its links to its neighbors only and broadcasts this information to all other nodes. Using the link-state information from all other nodes, each node computes a complete picture of the network and calculates the shortest path to all nodes. Examples are: Open Shortest Path First (OSPF) [**Moy98**], Optimised Link State Routing (OLSR) [**CJ03**], Topology Broadcast Reverse Path Forwarding (TBPRF) [**BO99**], Fisheye State Routing (FSR) [**GH01**] and Landmark Routing Protocol (LANMAR) [**GHMP03**].

The main advantage with proactive routing schemes is that there is no initial delay when a route is needed. On the other hand, they typically have higher overhead and longer route convergence time than the reactive schemes presented in the next section, especially in the case of high mobility. For better performance in ad-hoc networks both distance vector and link-state algorithms have been modified.

7.1.1.2. Reactive Routing. In general, reactive routing is not dependent on periodic route information and route calculations. Instead, whenever a route is needed the source node performs a route discovery (disseminates a route request throughout the network and wait for a route reply) before sending packets to the destination node. The route is maintained thereafter until the destination becomes inaccessible or is no longer needed. The route discovery process adds a significant initial delay to a transmission and consumes many resources during that phase. However, reactive protocols do not spend resources on exchanging route information or maintaining inactive routes, which is an advantage in highly dynamic environments. Examples of reactive protocols are Ad-hoc On-demand Distance Vector (AODV) [**PBR99**], Dynamic Source Routing (DSR) [**JM96**], Temporally Ordered Routing Algorithm (TORA) [**PC97**] and Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) [**AT99**].

7.1.2. AODV and DSR

Out of the abundance of MANET routing protocols, two protocols are sticking out due to their performance and their wide acceptance: Ad-hoc On-demand Distance Vector (AODV) routing protocol [**PBR99**] and Dynamic Source Routing (DSR) [**JM96**]. Both are also ideally suited for the given environment and will therefore be seen as a potential basis for a secure routing approach.

7.1.2.1. AODV. AODV - Ad-hoc On Demand distance Vector routing - is a reactive (on-demand) routing protocol using a routing table as in conventional routing protocols. Each node keeps a record of each active route it belongs to: the main entries of such a record are the destination IP address of the route, the node to which the packet has to be forwarded, the number of hops required to reach the destination, a list of the neighboring nodes that use this route, and a sequence number. Using destination IP-address and associated next intermediate node, any packet arriving to a node will be forwarded to the next hop of the route. The remaining entries are used to mitigate errors, discard obsolete, and to avoid loops.

The records are generated during the route discovery procedure. In order to learn a new route, a node creates a route request (RREQ) packet containing a destination address and floods it through the network. Each intermediate node receiving such a packet will temporarily store a partial record to create a backroute. Each node knowing a fresh route between source and destination (including the destination node) will unicast a route reply (RREP) packet to the source node. Finally, any node receiving and forwarding the RREP will store a record for the route, only keeping the shortest and freshest one in case it receives several RREP packets. Each node on the route of a RREP packet is then able to send or relay packets to the destination. With respect to route maintenance, obsolete routes are deleted if their expiration time is over and connectivity is checked periodically using hello messages. These hello messages are broadcasted and allow any link breakage to be reported immediately to neighboring nodes. All nodes using a breaking link are immediately noticed of the error using route error (RERR) packets.

7.1.2.2. DSR. DSR - Dynamic Source Routing - is a source routing protocol that uses a route cache storing complete routes to known destinations instead of a routing table. Each packet carries the complete route to its destination in its header. The forwarding operation is easier to perform, but packet headers are larger. The route discovery procedure consists of broadcasting a route request (RREQ) to the destination node. Each intermediate node that relays the RREQ packet the route request adds its own address in the header. Therefore, upon reaching the destination, a packet will contain the complete route. A route reply (RREP) consequently consists of returning the route back to the source node. The source node will save each route in its route cache, allowing the storage of multiple routes to the same destination. Route failures are discovered through encountered link failures. When a link between two nodes breaks, while transmitting a packet, a route error packet (RERR) is sent to the source node. A source node can then choose another route in its route cache if available or initiate another route discovery. In contrast to AODV, only nodes on the route are noticed of the error. Some optimizations to DSR have been proposed, such as authorizing intermediate nodes to issue replies during a route request if they own a route to the destination in their route cache or allowing intermediate nodes knowing another route to the destination to re-route a packet by sending it on an alternative route.

7.1.3. MANET Routing and Security

In the literature, a couple of attacks dedicated to routing protocols have been identified:

(1) **Impersonation / Spoofing** Impersonation attacks are also called spoofing attacks. The attacker feigns the identity of another node in the network, thus receiving messages directed to that node. Usually this would be one of the first steps to intrude a network with the aim of carrying out further attacks to disrupt operation. A compromised node may gain access to encryption keys or authentification information. In many networks, a malicious node could obstruct proper routing by injecting false routing packets into the network or by modifying routing information. Attackers might see an advantage in selectively forwarding packets that pass them.

(2) **The Sybil attack** This kind of attack has been described in section 5.6.2.1. Malicious nodes in a network may not only impersonate one node, they could assume the identity of several nodes, by doing so undermining the redundancy of many routing protocols. If a single malicious node is able to represent several other nodes, the effectiveness of measures distributing information pieces via different nodes is significantly degraded. An attacker may get access to all pieces of the fragmented information or may alter all packets in the same transmission so that the destination node(s) cannot detect tampering anymore. In trust-based routing environments, representing multiple identitities can be abused to deliver fake recommendations about the trustworthiness of a certain party, hereby attracting more traffic to it.

(3) **Wormhole** In a wormhole attack, two or more malicious node collaborate to generate a path outside the network to route messages between each other. Wormholes are hard to detect because the path that is used to pass on information is usually not part of the actual network. A wormhole itself may not be harmful, since it usually lowers the time it takes for a package to reach its destination (which made the route more interesting in the first place). But due to this shorter route, a wormhole attracts a significant amount of traffic. Wormholes are an entry point for further attacks, to alter large amounts of packets, eavesdrop on message contents or filter specific packets. The last attack is also known as blackhole attack, where specific packets (usually all packets of a chosen source, hence the name blackhole) are removed from the stream of packets passing through a wormhole. Removing or altering packets is especially harmful when applied to routing packets such as route requests (RREQ), route response (RREP) or other.

(4) **Sleep deprivation torture** This kind of attack is most typical for wireless ad-hoc networks consisting of battery-powered mobile devices. The strategy of this attack is to repeatedly request services from an attacked node, so that it is prevented from reaching an idle or power preserving state, thus depriving it of its sleep (hence the name). If carried out with nodes at key (routing-) positions or in a distributed denial of service (DDoS) style, this can have tremendous effects on the availability of networks based on nodes that have limited resources,

for example battery power. In the case of paid services, this may also affect business models as services become unaccessible or attacked nodes generate large (but useless) traffic. A general countermeasure to be taken is priorization of nodes and services.

(5) **Rushing attack** This type of attack is primarily directed against reactive routing protocols based on the Dynamic Source Routing protocol [**JM96**]. A malicious node will attempt to tamper with ROUTE REQUEST (RREQ) packets, modifying the node list, and hurrying this packet to the next node. Since in basic DSR only one RREQ packet of each route request is forwarded, the malicious node can route subsequent packets through itself if its RREQ manages to reach the next node in the route before any other neighboring nodes can. Rushing attacks can be detected by evaluating the Route Discovery. A solution has been proposed by Hu et al. through packet leashes [**HPJ02**].

(6) **Routing table overflow** In a routing table overflow attack the attacker attempts to create routes to nonexistent nodes. The goal is to create enough routes to prevent new routes from being created or to overload the protocol implementation.

7.1.4. State-of-the-Art Secure Routing Protocols

7.1.4.1. SAODV. SAODV [**Zap02**] is an extension of the AODV protocol, see section 7.1.2.1. It provides authentication and integrity of the routing data. The protocol requires each node of the network to possess a public and private key pair and to know the public keys of the other nodes. There are two extensions to AODV: a signature is added in order to authenticate the source node, and a method based on a hash function is used to prevent a malicious node from changing the value of the hop counter by more than one hop. To protect RREQ packet's hop counter, each sending node chooses a random number r, the maximal number m of forwarding operations the packet is allowed to undergo (max hop count), and a hash function H. The data H and $H^m(r)$ are appended to the packet, as well as a mutable hash field $H^{<hopcount>}(r)$, starting with $H(r)$ for the source node. The sending node signs its packet, hereby omitting the mutable hop count and hash fields that change at each intermediate node and appends the signature. Each intermediate node verifies the signature and keeps the packet in case of positive outcome. Only then it computes $H^{m-<hopcount>}(hashfield)$ and discards the packet if this value is not equal to $H^m(r)$. Then, it calculates the hash field by applying the hash function once more, forwards the packet, and sets up a reverse route. The same process is applied to RREP packets. However, by adding a second signature to the RREQ to identify a route and its validity, intermediate nodes can be allowed to answer to RREQ by sending RREP including a copy of the second signature. Although authentication is provided, this does not prevent incrementation of the hop count. But a node can only "lie" about itself.

7.1.4.2. Tesla and Hash Chains. Tesla is a broadcast authentication protocol introduced by Perrig et al. in [**PCTS02**]. Although this protocol is not applicable alone for the requirements of type-3 communication within the context of this thesis, it forms the basis of Ariadne, which is (together with SAODV) one of the protocols that will be evaluated more thoroughly in this thesis. Several broadcasting authentication schemes have been proposed, but TESLA claims to be efficient, scalable with respect to the number of receivers, and loss-tolerant. Loss-tolerancy is a strong requirement due to the potentially lossy wireless channel in the automotive environment. TESLA requires asymmetric public key cryptography for bootstrapping security associations and for authentication to prevent impersonation. The first main concept is to use time as a factor in asymmetric authentication. This assumes that the nodes are loosely time synchronized, meaning that each receiver needs to know an upper bound of the sender's time. The authors propose a protocol to achieve such a synchronisation: Using one-way hash chains is the second main concept of the protocol.

One-way Hash Chains

One-way hash chains have been introduced by Lamport for one-time passwords [**Lam81**]. A one-way hash chain is a series of integers $k_1, \ldots, k_m$ such that for each suitable i, j for $j < i$ each k_j is easily computable from k_i and if $j > i$, it is hard to compute k_j from $k_1, \ldots, k_i$. Such a chain can be generated by choosing a one-way function[1] f, a random integer r of suitable size, and by defining $s_m = f(r), s_{m-1} = f(s_m) = f^2(r), \ldots, s_1 = f(s_2) = f^m(r)$.

The protocol starts by clustering the time into time units and assigning keys of key-chains to time intervals composed of several time units. A source node needs to bootstrap configuration parameters: it distributes a key of its key chain that has not yet been used, the time interval to which the key corresponds and the number of time units of its intervals. Keys are now disclosed periodically, after their usage time has ended. When sending a packet, the node uses a key that has not yet been disclosed, calculates a message authentication code (MAC) for the packet using this key, and discloses the most recent key it can disclose by appending it to the packet. When receiving a packet, a node stores (otherwise discards) the packet if it has verified that the corresponding key has not been disclosed yet. It will then use the provided disclosed key to authenticate the MAC of previously stored packets and waits for the actual key to be disclosed. Delaying the disclosure of keys is the asymetric process that enforces security: once a key is disclosed, any receiver can forge packets with it. But since each key is disclosed after the end of the time interval during which it is allowed to be used, such packets will be discarded by receiving nodes. TESLA introduces a delay due to the key diclosure process, which needs to be optimized for the working environment.

7.1.4.3. ARIADNE. Hu, Perrig and Johnson introduced a secure ad-hoc routing protocol based on DSR in [**HPJ01**], that withstands node compromise and relies only on efficient symmetric cryptography. ARIADNE guarantees that the target node of a route discovery process can authenticate the initiator, that the initiator can authenticate each intermediate node on the route to the destination present in the RREP message and that no intermediate node can remove a previous node in the node list within RREQ or RREP messages. ARIADNE needs a mechanism to bootstrap keys required by the protocol. More specifically, each node needs a shared secret key ($K_{S,D}$ is the shared key between a source S and a destination D) with each node it communicates with at a higher layer, an authentic TESLA (see paragraph 7.1.4.2 above) key for each node in the network and an authentic Route Discovery chain element for each node for which this node will forward RREQ messages. ARIADNE provides point-to-point authentication of a routing message using a message authentication code (MAC) and a shared key between the two parties. However, for authentication of a broadcast packet such as RREQ, ARIADNE uses the TESLA broadcast authentication protocol. ARIADNE is able to cope with attacks performed by malicious nodes that modify and fabricate routing information, with attacks using impersonation and, in an advanced version, with the wormhole attack. Selfish nodes are not taken into account.

[1] A one-way function is an easily computable function whose inverse is hard to compute, see also section 3.2.3.

In ARIADNE, the basic RREQ mechanism is modified using eight fields, used to provide authentication and integrity to the routing protocol: RREQ, initiator, target, id, time interval, hash chain, node list, MAC list. The initiator and target are set to the address of the initiator and target nodes, respectively. As in DSR, the initiator sets the ID to an identifier that it has not recently used in initiating a Route Discovery. The time interval is the TESLA time interval at the pessimistic expected arrival time of the request at the target, accounting for clock skew. The initiator of the request then initializes the hash chain to $MAC_{KS,D}(initiator, target, id, timeinterval)$ and the node list and MAC list to empty lists. When any node A receives a RREQ for which it is not the target, the node checks its local table of $< initiator, id >$ values from recent requests it has received, to determine if it has already seen a request from this same Route Discovery. If it has, the node discards the packet, as in DSR. The node also checks whether the time interval in the request is valid: that time interval must not be too far in the future, and the key corresponding to it must not have been disclosed yet. If the time interval is not valid, the node discards the packet. Otherwise, the node modifies the request by appending its own address (A) to the node list in the request, replacing the hash chain field with $H[A, hashchain]$, and appending a MAC of the entire REQUEST to the MAC list. The node uses the TESLA key K_{A_i} to compute the MAC, where i is the index for the time interval specified in the request. Finally, the node rebroadcasts the modified RREQ, as in DSR. When the target node receives the RREQ, it checks the validity of the request by determining that the keys from the time interval specified have not been disclosed yet, and that the hash chain field is equal to: $H[\eta_n, H[\eta_{n-1}, H[\ldots, H[\eta_1, MAC_{KS,D}(initiator, target, id, timeinterval)] \ldots]]]$ where η_1 is the node address at position i of the node list in the request, and where n is the number of nodes in the node list. If the target node determines that the request is valid, it returns a RREP to the initiator, containing eight fields: RREP, target, initiator, time interval, node list, MAC list, target MAC, key list. The target, initiator, time interval, node list, and MAC list fields are set to the corresponding values from the RREQ, the target MAC is set to a MAC computed on the preceding fields in the reply with the key KDS, and the key list is initialized to the empty list. The RREP is then returned to the initiator of the request along the source route obtained by reversing the sequence of hops in the node list of the request. A node forwarding a RREP waits until it is able to disclose its key from the time interval specified, then it appends its key from that time interval to the key list field in the reply and forwards the packet according to the source route indicated in the packet. Waiting delays the return of the RREP but does not consume extra computational power. When the initiator receives a RREP, it verifies that each key in the key list is valid, that the target MAC is valid, and that each MAC in the MAC list is valid. If all of these tests succeed, the node accepts the RREP; otherwise, it discards it. In order to prevent the injection of invalid route errors into the network fabricated by any node other than the one on the sending end of the link specified in the error message, each node that encounters a broken link adds TESLA authentication information to the route error message, such that all nodes on the return path can authenticate the error. However, TESLA authentication is delayed, so all the nodes on the return path buffer the error but do not consider it until it is authenticated. Later, the node that encountered the broken link discloses the key and sends it over the return path, which enables nodes on that path to authenticate the buffered error messages. ARIADNE is protected also from a flood of RREQ packets that could lead to the cache poisoning attack. Benign nodes can filter out forged or excessive RREQ packets using Route Discovery chains, a mechanism for authenticating route discovery, allowing each node to rate-limit discoveries initiated by any other node. The authors present two different approaches.

ARIADNE is immune to the wormhole attack only in its advanced version: using the TIK (TESLA with Instant Key disclosure, references) protocol that allows for very precise time synchronisation between the nodes of the network, it is possible to detect anomalies in routing traffic flows in the network.

7.1.4.4. SEAD. Johnson, Hu and Perrig presented a proactive secure routing protocol based on the Destination-Sequenced Distance Vector protocol (DSDV) in [**JHP02**]. SEAD deals with attackers that modify routing information broadcasted during the update phase of the DSDV protocol: in particular, routing can be disrupted if the attacker modifies the sequence number and the metric field of a routing table update message. Replay attacks are also taken into account. SEAD makes use of efficient one-way hash chains rather than relying on (computationally-) expensive asymmetric cryptography operations. However, SEAD assumes some mechanism for a node to distribute an authentic element of the hash chain that can be used to authenticate all the other elements of the chain. The authors suggest to ensure the key distribution relying on a trusted entity that signs public key certificates for each node; each node can then use its public key to sign a hash chain element and distribute it. The basic idea of SEAD is to authenticate the sequence number and metric of a routing table update message using hash chains elements. In addition, the receiver of SEAD routing information also authenticates the sender, ensuring that the routing information originates from the correct node.

SEAD does not cope with wormhole attacks though the authors propose, as in the ARIADNE protocol, to use the TIK protocol to detect the threat.

7.1.4.5. ARAN. ARAN, introduced by Sanzgiri et al. in [**SDL$^+$01**], is a routing protocol that relies on digital certificates and public key cryptography. It is a table-driven protocol similar to AODV. It provides authentication for end-to-end connections as well as for intermediate nodes, integrity, and non-repudiation. ARAN requires the availability of a trusted server, which issues certificates. A route discovery works the following way: a source node S seeking a route to a destination D broadcasts a route discovery packet (RDP) which contains the following fields:

$< RDP, IP(D), certificate(S), nonce(S), timestamp, MAC(S) >$.

The packet is signed with a MAC and any node can authenticate using the certificate. The certificate and MAC of the last intermediate node are appended to the packet for hop by hop authentication. The nonce ensures freshness. Each intermediate node authenticates the previous node, stores a temporary reverse route, and retransmits the packet after replacing the MAC and certificate of the previous intermediate node with its own MAC and certificate. The destination D replies to the first RDP it receives with a response message (REP) consisting of following fields

$< REP, IP(A), certificate(D), nonce(S), timestamp, MAC(D) >$

which is delivered to the source in reverse order applying the same authentication process. Similar to SAODV, ARAN does not provide any mechanism against the wormhole attack. The route maintenance is achieved by the way of authenticated error messages, that are sent back to the source of active routes. In contrast to SAODV, nodes need not store all node public keys, but can retrieve certificates from other nodes or servers. However, the question of certificate revocation has not been solved.

7.1.4.6. SRT & SMT. SRT and SMT, presented by Papadimitratos and Haas in [**PH02**] are extensions for existing routing protocols such as DSR. The goal of SRP is to secure the route discovery mecanism, while SMT protects the forwarding operation. In contrast to the aforementioned protocols SRP requires only a security association between the source node and the destination node. It deals with malicious nodes which modify, forge or resend packets, and try to minimize the acceptance of false routing information. The packet extension contains mainly three fields: a request identifier, a request sequence number, and a message authentication code (MAC). The sequence number identifies the route request and enables the forwarding nodes to discard already seen packets. In order to prevent malicious nodes from forging impersonated packets that would lead genuine packets with the same sequence numbers to be dropped, a random identifier is provided with the sequence number in order to identify the request. Finally, to authenticate and prove the integrity of the packet a MAC of the whole packet (excluding mutable fields) and of the shared private key is computed. The destination node replies the same way. Note that in case of DSR the MAC includes the whole route. To prevent flooding, intermediate nodes rank their neighbors as a function of their transmission rates. The nodes with the highest rates are assigned with the lowest priorities with respect to the forwarding process, so that malicious or selfish nodes are served last. Concerning route maintenance, error packets are not authenticated, thus leaving a gap for potential attacks. However, since the error packets have to be source routed back to the source, only a node on the route (or a node colluding with such a node) can forge such messages.

7.1.5. Other Work on Secure Routing

In addition to the works on secure routing protocols that have been analyzed above, Zhou and Haas give a general overview on security of ad-hoc networks in [**ZH99**], including routing security. Wagner and Karlof discuss secure routing for sensor networks in [**WK03**]. Hu, Perrig and Johnson discuss the specifics of rushing attacks and offer a solution called packet leashes [**HPJ03**]. Bobba et al. elaborate the problem of bootstrapping keys in a decentralized system [**BEGA03**].

7.2. Performance of Secure Routing Protocols

The goal of this section is to compare the performance of DSR and AODV routing protocols and their secure counterparts, Ariadne and SAODV, in order to choose the best protocol for application in an automotive environment. This section has been included in this chapter (in contrast to other implementation, simulation and evaluation sections) because it is fundamental for the development of a secure routing protocol and not an evaluation of the results thereof. Ariadne and SAODV have been chose due to their wide acceptance in the scientific community, availability of source code and simulation code, performance and security features. Coincidentally, DSR and AODV have been evaluated in [**DPR00**], but the simulations were based on other mobility models. Nevertheless, in principle, the results given in the paper support the interpretation of simulation results given in this thesis.

7.2.1. Simulation Setup

Simulations have been conducted using NS2[2] version 2.26, a discrete event network simulator and the mobility extension developed by the CMU Monarch project[3]. It allows to simulate the physical, link, network, transport and application layers and includes a broad range of protocols. DSR and AODV are available for NS2 version 2.26, however Ariadne had to be modified, since it had only been available for NS2 version 2.1b3 on the CMU Monarch website. SAODV is simulated by increasing the sizes of the RREQ, RREP and RERR packets in order to include the security overhead, and by delaying

[2] `http://www.isi.edu/nsnam/ns/`
[3] `http://www.monarch.cs.rice.edu/cmu-ns.html`

the transmission of signed packets in order to approximate the use of cryptographic algorithms. The verification time of the signature is neglected.

Simulation parameters:

Parameter	Value
radio propagation model	two-ray ground reflection
medium access protocol	IEEE 802.11
nominal radio range	250m
mobility model	random waypoint / BMW trace file
simulation area	1000m x 1000m
maximum speed	20m/s
number of nodes	50, 100
simulation time	100s, 200s
network traffic	CBR over UDP
average packet size	512 bytes
percentage of source nodes	30% to 60%
routing layer send queue size	64 packets
max. send queue hold time	30s
interface layer queue size	50 packets

Table 7.1. Simulation Parameters

The main input parameters / data for the simulations consist of a routing protocol, a scenario file, and a trace file. To simulate an automotive environment, trace files generated by a BMW mobility file generator (called GenMobTrace) using a map of Munich to simulate cars' movements are used. The simulations are restricted to submaps of 1000m x 1000m and the mean value of resulting values obtained for each square is used. The trace files generated by GenMobTrace provide only poor connectivity, as seen in simulation results. For comparison, trace files generated by the CMU mobility generator setdest are used as well. All results are calculated by taking the mean value of 10 different trace files.

7.2.2. Simulation Results

The NS2 output is analyzed to extract the following values:

- **packet delivery ratio**: The fraction of data (CBR) packets delivered to the destination node.
- **average end-to-end delay**: The mean of the delivery time of the received data packets.
- **maximum end-to-end delay**: The maximal delivery time of the received data packets
- **fraction of end-to-end delays better than 0.5s**: The ratio of data packets delivered with a delivery time better than 0.5s.
- **routing overhead**: The number of packets received at the routing layer.
- **MAC overhead**: The number of packets received at the MAC layer.

The results are presented separately for simulations carried out with BMW's GenMobTrace and CMU's setdest mobility trace file generators. The graphs displaying the most relevant results such as the delivey ratio, the end to end average delay and the amount of routing packets are given in this section, the remainder of graphs is available in Appendix C, section C on page 189.

7.2.2.1. BMW GenMobTrace results. The simulation results in Figure 7.1 show a very low delivery ratio (between 7 and 18 %). This results from the poor connectivity of the scenario files, not from mobility. Nevertheless, it resembles a low penetration rate of cars with ad-hoc technology. DSR has the best behavior concerning the delivery ratio, Ariadne the worst. AODV and SAODV are in

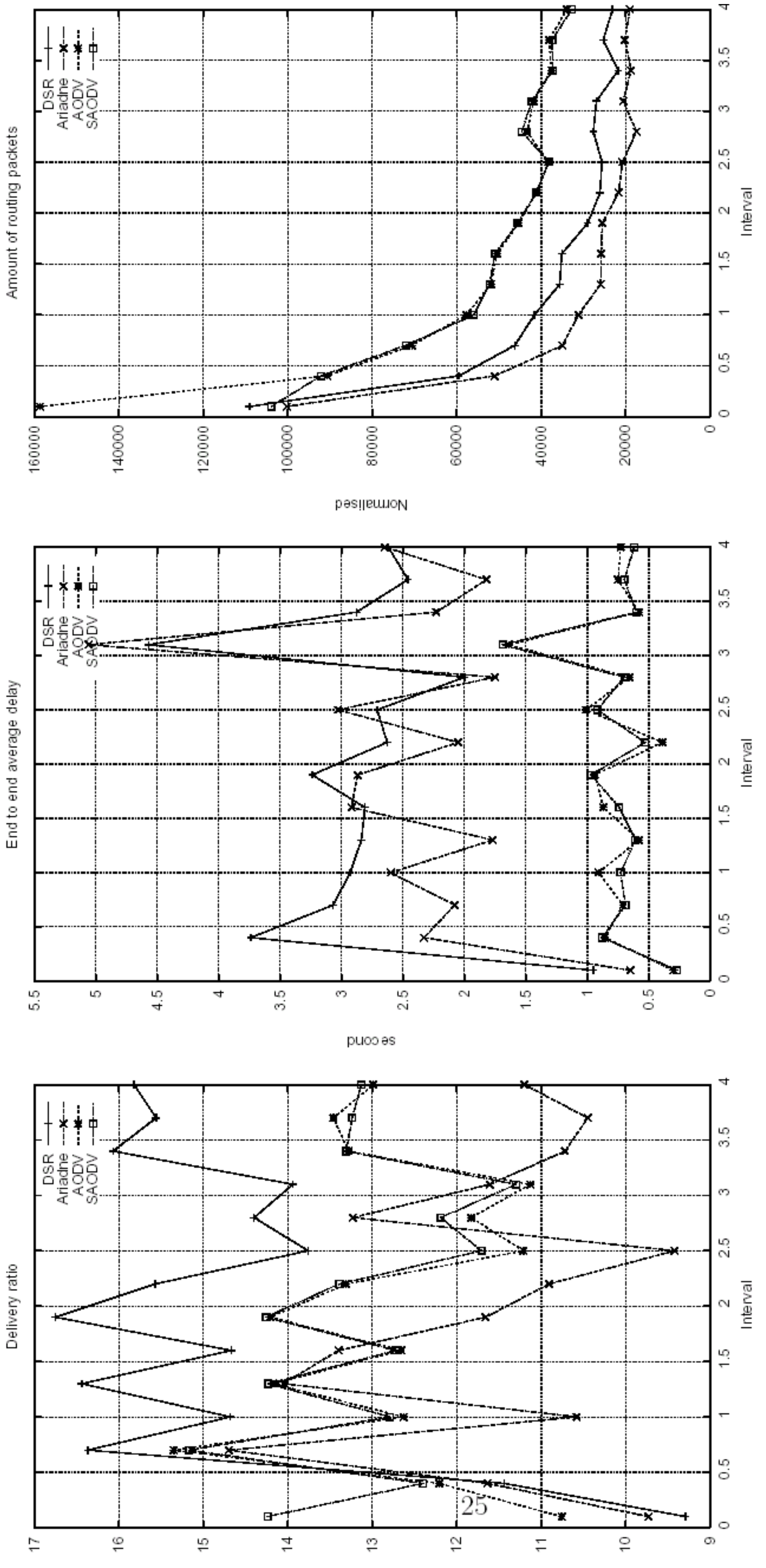


Figure 7.1. 50 nodes, 200s, 60% CBR, BMW GenMobTrace

between, without much difference. Ariadne relies on a global synchronisation and the replies are scheduled according to a global clock with the tesla mecanism, which explains the significant difference to DSR. Ariadne achieves up to a 5% lower delivery ratio than DSR. There are two reasons: first, since the reply and error messages are delayed, topology changes are taken into consideration later than in DSR and connections are missed. Second, Ariadne does not benefit from gratuitous replies which would accelerate the route discovery process. Note that DSR and Ariadne perform worse than (S)AODV in the case t = 0.1s. This confirms one result of [**DPR00**] that DSR becomes worse than AODV in stressing situations. Concerning the end to end delays, DSR and Ariadne are worse than (S)AODV. The network seems to be jammed by route discoveries because of the poor connectivity, additionally link failures are only propagated backwards to the source in DSR, instead of all the nodes using the link as in AODV, thus delaying the error report. Finally, DSR and Ariadne use a lower amount of routing packets than (S)AODV. One reason is that source routing provides more information, since a route provides routes to all the intermediate nodes. Other simulation settings support theses interpretations.

7.2.2.2. CMU setdest results. Figure 7.2 results have been obtained with scenarios generated by the CMU mobility generator with a higher rate of connectivity than in the previous paragraph. With a high network traffic of 60%, a large number of nodes (100) and the simulation time to 200s in Figure 7.2, AODV and SAODV are significantely better than DSR and Ariadne. Interestingly, DSR achieves a very poor result in case of low rate: DSR achieves a delivery ratio lower than 30%, while AODV achieves more than 70%. This effect has already been identified in already noted in [**DPR00**] when the number of sources increases. SAODV remains slower than AODV, since SAODV has a more complex packet treatment and can therefore process fewer packets per second, which sometimes results in a topology mismatch in case of high mobility.

7.2.3. SAODV as basis

The superior performance of (S)AODV in the given scenario has lead to the decision to use AODV as a basis for a secure routing protocol in the context of this work. However, SAODV requires the distribution of certificates of all nodes in a previous setup stage. Additionally, SAODV does not provide security against trusted nodes that become malicious. Both shortcomings will be addressed in the new protocol AODV-SEC presented in the following section 7.3.

7.3. AODV-SEC

Due to the limitations of SAODV, the preferred secure ad-hoc network routing protocol, AODV-SEC has been developed in cooperation with the Institute of Communication Networks (LKN), Munich University of Technology, Germany. The results have been published in [**EDS$^+$04**]. AODV-SEC uses hash chains and digital signatures similar to SAODV. The bootstrapping problem, which has been elaborated in [**BEGA03**] is approached by building on the LKN-ASF (LKN-ad-hoc Security Framework), which has been introduced in [**SE04**].

7.3.1. AODV-SEC Trust Model

Assuming that certificates are common in routed connections and that a certificate handling infrastructure is available at some sporadically, AODV-SEC has been designed to operate using certificates.

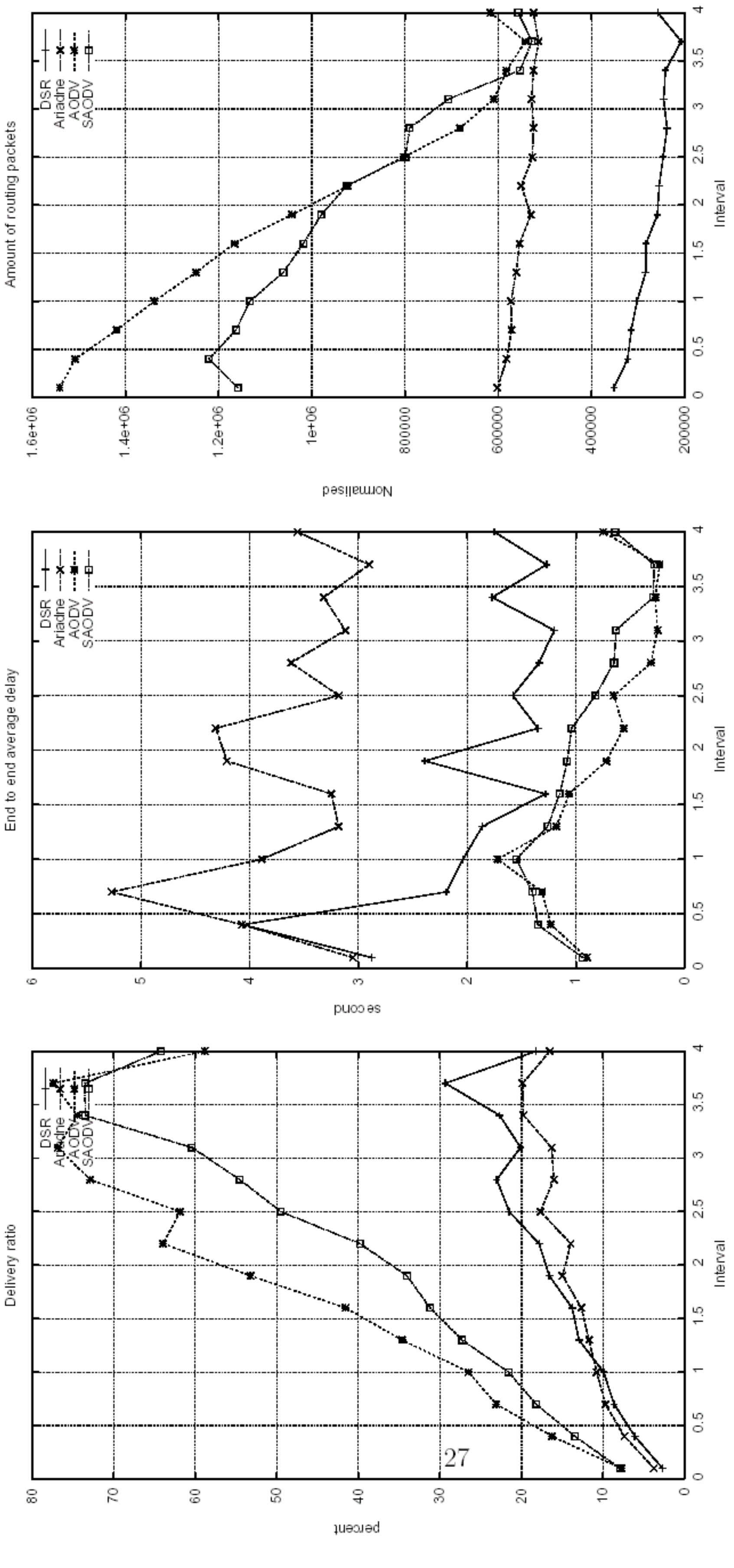


Figure 7.2. 100 nodes, 200s, 60% CBR, CMU Setdest

LKN-ASF, introduced by Christian Schwingenschlögl in [**SE04**], provides a protocol for certificate handling in MANETs which enables the use of a centralized public key infrastructure (PKI) in the distributed mobile ad-hoc environment. The security is based on public key certificates certified by a trust center. The protocol manages and validates certificates exchanged between nodes. The use of a PKI to secure communication remains secure only if expired and untrustworthy certificates can be revoked. This is done by the trust center. In the MANET environment the the server infrastructure's constant availability is not guaranteed. LKN-ASF manages the timely distribution of revocation messages throughout the network. In addition the protocol ensures that an expired certificate gets replaced in time and therefore nodes are not excluded from the network unintentionally. These properties make this framework ideally suited for use in Vehicle Ad-hoc NETworks.

7.3.2. AODV-SEC

The protocol is an extension of AODV that protects the route discovery and maintenance mechanisms. The goal is to enable nodes receiving routing information to decide whether to import this information into their routing database or not, called import authorization. In order to achieve this, some other security services are needed, such as source authentication and neigbor authentication which are used to verify a source node or a neighboring node respectively. Additionally, the integrity of information has to be guaranteed. Integrity and source / neighbor authentication together can provide data authentication.

Two mechanisms are used to secure AODV-SEC:

(1) **One-way Hash Functions**
One-way hash functions have been explained in section 7.1.4.2. They are used to secure the correctness of the received data. They protect the mutable fields in the messages' headers. An attacker cannot spoof a packet because the data is signed using a hash function over the received signature. The only way to verify the correctness of the data is re-calculating the signature and then compare.

(2) **Digital Signatures**
Digital signatures are used to authenticate the non-mutable fields of the messages. Only nodes with valid certificates are able to sign messages in such a way that it can be positively verified by a receiver. Certificates are handled via gateways and through other nodes, thus increasing flexibility in ad-hoc networks. The threat potential is reduced by limited period of validity.

Compared with other secure routing protocols, AODV-SEC does not require certificate installation before the node starts to operate. It can be done on-the-fly, using certificates from neighboring nodes. The last-hop authentication scheme used in AODV-SEC provides full protection on the whole route.

Following operations are performed by AODV-SEC: Headers of RREQ, RREP and RERR packets include non-mutable fields that are set by the node creating the packet and mutable fields that have to be altered by some or all intermediate nodes. In AODV-SEC, non-mutable fields such as the source address are protected by digital signatures, which provides source authentication, integrity of the packet and data authentication. To be more specific, all fields in the header except hop-count and hash are included in the digital signatures for RREQ and RREP packets. RERR packets are different, because it is not relevant which node started the RERR but the information that a neighboring node is not able to route messages to certain destinations. So in the case of RERR packets, the complete header will be digitally signed by every node again.

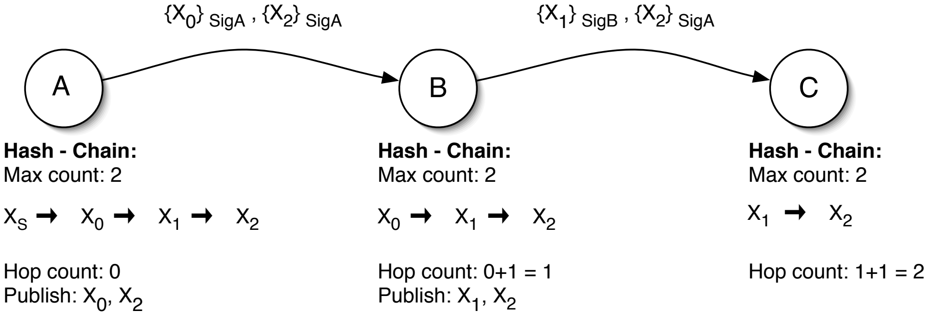


Figure 7.3. Hash chain example (AODV-SEC)

The mutable fields of RREQ and RREP are protected using hash chains as shown in Figure 7.3 to allow every receiving node to verify that the hop count has not been decremented by an attacker.

Source Node
If a (source) node generates a RREQ or RREP message, it first generates a random number (seed), sets the $MaxCount = TTL$ field to the IP header's TTL value, sets the hash field to the seed value and calculates:

$$x_{max} = h^{MaxCount}(seed) \tag{7.1}$$

applying a hash function $h(x)$ $MaxCount$-times to the seed value. x_{max} and

$$x_0 = h(seed) \tag{7.2}$$

are both signed by the source node and sent as part of the package header, x_{max} in the "TopHash" field and x_0 in the "Hash" field.

Intermediate Node
Each intermediate node applies the hash function $h(x)$ $MaxCount - HopCount$ times to the value in the hash field and verifies that the resultant value is equal to the value in the TopHash field, here (variables from above):

$$x_{max} == h^{MaxCount-HopCount}(x_0) \tag{7.3}$$

If both values are equal, the node proceeds to apply the hash function to the value in the Hash field, here:

$$x_1 = h(x_0) \tag{7.4}$$

signs it and puts it into the Hash field.

In future implementations, forward secure signatures could be used for this task. This would provide non-repudiation and help to track misbehaving nodes. Cryptographic operations securing mutable fields have to be relatively cheap, since they are performed by each intermediate node.

The trust between nodes is based on certificates that are issued by a trusted authority. If a node doesn't know another nodes' certificate, it can ask the node for its certificate itself. In the given implementation it is also possible, that intermediate nodes store certificates from other nodes in the route, such that requests for certificates are limited to the next node caching the demanded certificate. This greatly reduces overhead as can be seen in the simulation results as well, see section 8.8.

7.4. Conclusion

In this section, it has been shown, that some routing protocols are better suited for use in a inter-vehicle communication environment than others. More importantly, a thorough investigation of secure MANET routing protocols using network simulation resulted in the choice of a protocol that has been modified for use in automotive environment.

Table 7.2 gives a simple overview on the features / results of the routing protocols that have been examined.

	DSR	ARIADNE	AODV	SAODV	AODV-SEC
Delivery Ratio	++	-	+	+	o
End-to-end Delay	o	o	+	+	-
Throughput	o	o	++	++	+
Scalability	-	-	++	+	+
Credential Handling	-	o	-	o	++

Table 7.2. Comparison of Routing Protocols

AODV-SEC, the protocol introduced in section 7.3 provides last-hop authentication and source node authentication combined on flexible credential handling, based on the excellent overall performance of AODV.

A future step is to evaluate whether pseudonym credentials can be used together with AODV-SEC. While the use of pseudonyms is typically not required for use in type-3 scenarios as pointed out before, it would harmonize the overall architecture.

CHAPTER 8

Prototyping and Simulation

SARI has been modeled and described in previous chapters. A large portion of this design has been the result of prototypical implementations, tests and simulations. Prototyping and simulation are also the chosen approaches to verify feasibility of components or modules and to evaluate the efficiency of theoretical models. However, due to the diversity of the questions to be solved and the complexity of the overall system, most testing and verification tools have been created specifically for the component in question. Therefore, an integrated system test including all components was not feasible. Nevertheless, an elaborate simulation environment has been created, combining communication network simulation, traffic / mobility simulation and application simulation.

This chapter shows which components / modules have been evaluated and whether prototypical implementations on a small scale (including a small number of nodes) or simulations have been performed or both. The main results and/or significant findings will be described. The sections are organized according to the component or module in question, except for section 8.4, which introduces the simulation environment architecture that has been used whenever possible, but not for all simulations.

8.1. Development and Verification using Prototyping and Simulation

Prototypes and simulation have been used for development, but also to verify concepts introduced in this thesis, where neither theoretical analysis has been sufficient nor estimations about efficiency in an automotive environment have been available using secondary literature.

Prototypes provided a proof of concept at an early stage and helped to identify shortcomings or point out improvement opportunities. More specifically, prototypical implementations such as the pseudonym credential architecture and methods (see section 8.2 on smart cards have delivered important performance data in practical systems. This allowed to check the feasibility of the idea in principle. Later improvements and measurements have then be dedicated to improve the implementation. However, the development of seperate proof-of-concept prototypes complicates the integration into a homogeneous overall system.

Prototypical implementations in hard- and software (including software related implementations on computers or in test vehicles) are looking at a single node in the network. In order to evaluate large scale effects, which are an important part of this thesis, simulations have been used. An introduction into the complexity of the used simulation environment will be given in section 8.4.

8.2. Prototyping of Privacy Concepts

The pseudonym credential architecture and related cryptographic algorithms presented in section 4.3.3 has been implemented on a smart-card, with organization O and authority A implemented on a PC. This implementation is a proof-of-concept and shows the algorithms' performance in such a configuration.

The standard JavaCard API does not provide a class similar to *java.math.BigInteger* as known from a standard Java SDK such as 1.4. Therefore it is not possible to use the cryptographic co-processor functionality for the scheme introduced in section 4.3.3 with numbers larger than 16 Bit. The extension of the standard-API to be able to use large numbers is technically possible, but requires a large development effort. For proof-of-concept, an implementation with small numbers has been made. The implementation has been integrated in a BMW 7 series (E65) prototype to interact with the onboard systems and communication system.

In order to make qualitative statements about the proposed zero-knowledge system, a software implementation with large numbers has been made on PCs.

Each of the CAs, Authority A and Organization O, defines following system parameters for its domain:

(1) Its own CA-ID X
(2) The module $N_{X-CA} = p_{X-CA} \cdot q_{X-CA}$
(3) The number of rounds t
(4) The number of key parameters k
(5) Its public key $pubKey_{X-CA}$

Because of the knowledge of the factors of the modules N_{A-CA} and N_{O-CA} by Authority A and Organization O respectively, they are able to calculate each "private key" in their domain. The signatures do not fulfill the requirement of non-repudiation, because the CAs are theoretically able to send messages in the name of a node in their domain. However both, Authority A and Organization O, are assumed to be trustworthy entities.[1]

The distributed core-element of the privacy architecture is the TRM, in the prototypical implementation represented by a JavaCard smart card. To reflect the process as presented in section 4.3.2 a state machine has been implemented as shown in Figure B.1in Appendix B.1.

The system parameters t (number of rounds) and k (number of key parameters) that are defined for every CA are affecting the performance of operations on a smart card. There is a tradeoff between memory consumption and execution speed. Execution speed is increased if t is small, but this increases memory consumption. To achieve sufficient level of security (i.e. comparable to RSA with 1024 bit keys), the combination of both parameters' values must be large enough.

Parameter k	16	24	32	40	80	160
t_{max}	10	6	5	4	2	1
Memory consumption in Byte	2368	3552	4736	5920	11840	23680
Max key-pairs in 64 MByte	28339	18893	14169	11335	5667	2833
Signature length in Byte	1329	847	737	625	449	481

Table 8.1. Memory Consumption for Key Length 1024 bit

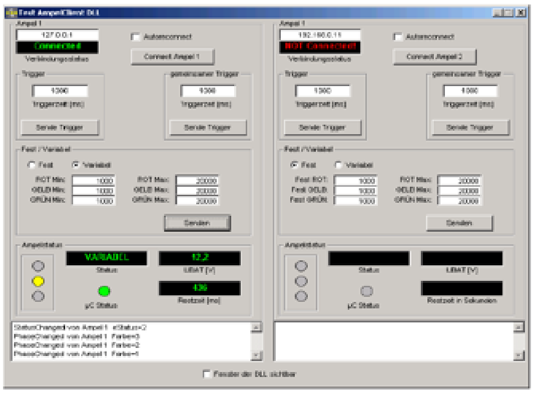
Table 8.1 shows how different values of t and k affect memory requirements. A detailed analysis of the correlation between system parameters and system performance can be found in [**Wim04**].

8.3. Traffic Light Lab Setup

An example for the vehicle to infrastructure scenario as discussed in chapter 6, including the communication protocol (see section 6.3), has been implemented in Java and tested on PCs, a prototype

[1] Note that this is basically the same level as root-servers of PKIs.

car and traffic lights equipped with a wireless communication unit. Figure 8.1(a) shows the GUI of the traffic light control software, Figure 8.1(b) shows the lab setup for a traffic light control unit with wireless communication.



Test AmpelClient DLL

Ampel 1

127.0.0.1

Connected

Verbindungsstatus

□ Autoconnect

Connect Ampel 1

Trigger: 1300 Triggerzeit [ms]

gemeinsamer Trigger: 1300 Triggerzeit [ms]

Fest / Variabel: ○ Fest ● Variabel

ROT Min: 1000 ROT Max: 20000

GELB Min: 1000 GELB Max: 20000

GRÜN Min: 1000 GRÜN Max: 20000

Ampelstatus

Status: VARIABEL

UBAT [V]: 12,2

Restzeit [ms]: 436

```
StatusChanged von Ampel 1  eStatus=2
PhaseChanged von Ampel 1  Farbe=3
PhaseChanged von Ampel 1  Farbe=2
PhaseChanged von Ampel 1  Farbe=1
```

Ampel 1

192.168.0.11

NOT Connected!

Verbindungsstatus

□ Autoconnect

Connect Ampel 2

Trigger: 1300 Triggerzeit [ms]

gemeinsamer Trigger: 1300 Triggerzeit [ms]

Fest / Variabel: ● Fest ○ Variabel

Fest ROT: 1000 ROT Max: 20000

Fest GELB: 1000 GELB Max: 20000

Fest GRÜN: 1000 GRÜN Max: 20000

Ampelstatus

Status

UBAT [V]

Restzeit in Sekunden

□ Fenster der DLL sichtbar

(a) GUI Traffic Light Control Software

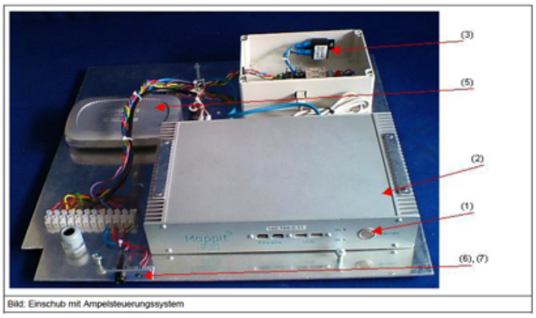


Bild: Einschub mit Ampelsteuerungssystem

(b) Lab Setup Control Unit

The traffic light example has been chosen because it offers a simple, but plausible application that requires solutions to the major problems of type-2 scenarios. The proposed communication protocol as well as a traffic light status and control application have been integrated into the general telematics architecture running inside a BMW 7 series experimental vehicle based on the OSGi service framework[2] (refer also to [**ZWH04**]). For communication between the traffic lights and the vehicle IEEE 802.11b hardware and protocols were used. The use-case of the traffic light example is twofold. First, the traffic lights broadcast their current state and switching times, providing drivers with status information and assist the driver with red light / right of way violation warnings. Using an authentication process, every receiving vehicle is able to verify the information before presenting it to the driver. Second, emergency vehicles (or other publicly authorized vehicles) send control messages to the traffic lights in order to actively influence their current state, especially for traffic light preemption. To reduce the number of vehicles potentially hindering a free way for the emergency vehicle, this communication allows each intersection to optimize the traffic flow before an emergency vehicle arrives, according to the expected arrival time of the emergency vehicle and the current local traffic situation. Consequently, emergency vehicles can reach their destination quicker and safer.

8.3.1. Test Setup

The test setup consists of two portable construction site signaling installations that are connected with a central control unit. The control unit ensures an adequate switching mechanism of all attached signaling installations and provides wireless access for authorized vehicles, in the given setup a 7 series experimental vehicle. Each traffic light, being part of a more or less complex combination of traffic lights for each direction / lane and a central control unit, has to be unambiguously addressed. A possible solution, is to use the geographic position of the central control unit to address the whole traffic light system of an intersection and then additionally address a distinct traffic light by the (discretized) angle of its geographic direction towards the intersection center. Additionally, the lane of has to be addressed, see Figure 8.1 for an example showing a large intersection.

[2] `http://www.osgi.org`

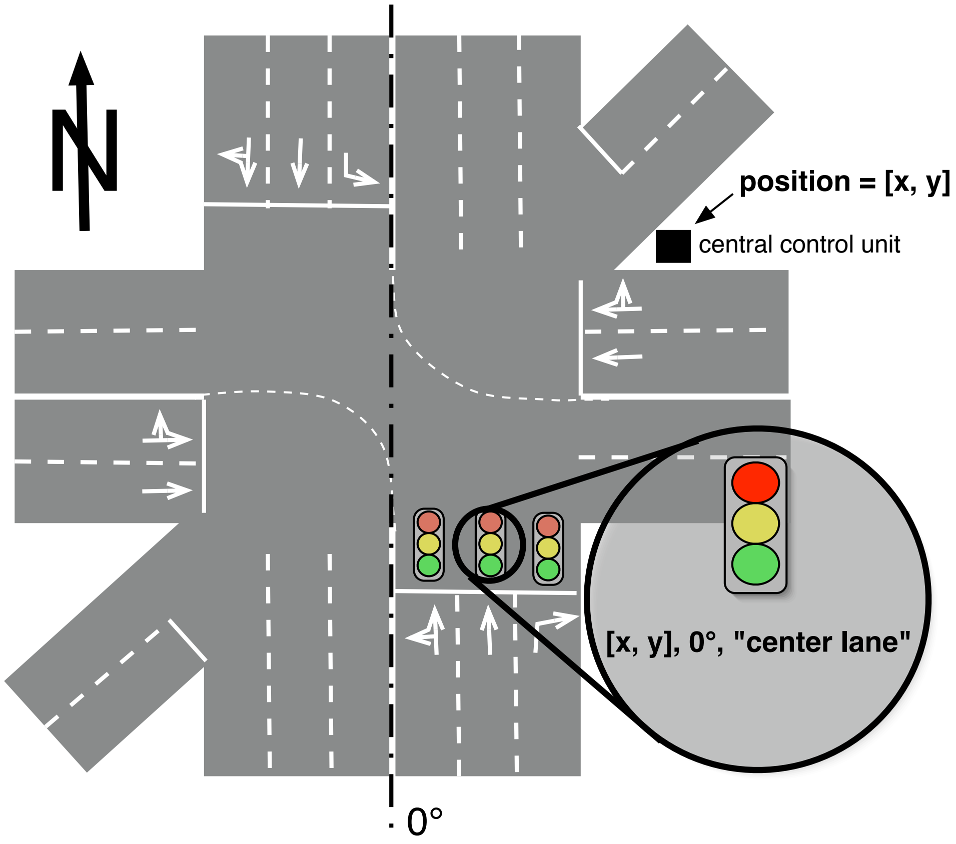


Figure 8.1. Setup of a Large Intersection

XML-messages have been used for the exchange of status and switching information between the vehicles and the traffic lights. The messages sent to the traffic light (M_c) consist of:

- Identification of the traffic light intersection system, hence the geographical coordinates of the control unit
- Identification of the lane / direction, hence the discretized angle
- Control message payload, such as an updated switching time for a green phase request

The periodic status message from the traffic light (M_s) contains:

- Identification of the traffic light (i.e. its geographical coordinates)
- Status and switching time for each traffic light lane / direction that affects vehicle traffic (not pedestrians), including those for dedicated left- or right-turning.

8.3.2. Performance

The results of tests with the lab setup and performance tests of the software provided a deeper understanding of the system behavior and helped identifying crucial parameters. The most important factors limiting the system's performance from a user perspective are the range of the radio communication and the bandwidth required for the protocol. Assuming that only single-hop communication is possible in a first phase of prototyping, the protocol has to be fast enough to handle a control message from a quickly approaching emergency vehicle in spite of the security overhead and the time that a traffic light control logic needs to switch from red to green (see 8.3.3).

Table 8.2 shows overall protocol performance measured in the test setup.

		step	time
step 1	vehicle	DH-Phase I	65,05 ms
step 2	traffic light	DH-Phase I DH-Phase II signature generation encryption	140,78 ms
step 3	vehicle	DH-Phase II decryption signature verification signature generation encryption	81,74 ms
step 4	traffic light	decryption signature verification encryption	5,79 ms

Table 8.2. V2I Protocol Performance

8.3.3. Signal Switching Time

The calculations for the switching time are based on the german guidelines on traffic light systems [**fSuV98**]. Switching time is the time to switch a traffic light in a given lane / direction from red to green. In the trivial case if it is already green on the desired route, the switching time is zero. So the upper bound for the switching time has to be calculated, which is the minimum time possible (considering all constraints) to switch the signal in a worst case scenario. The worst case is that the road crossing the desired path just got a green signal recently and there is also a pedestrian signal. Pedestrians are the slowest units here, the guideline's pedestrian section states three relevant things:

(1) Pedestrians move at a speed of 1.2 m/s
(2) Pedestrians must be able to cross at least 2/3 of the distance between two "safe spots"[3] during one phase of green light.
(3) Pedestrians must be able to cross the whole distance between two "safe spots" after their signal switches from green to red [4] before cars crossing them get a green light.

Assuming that the distance between two "safe spots" for pedestrians is not larger than 15 m, a pedestrian takes $15m/1.2m/s = 12.5s$ to cross such a distance. According to the requirements seen before, this adds up to around 20 seconds switching time (adding the time it takes from "red light" for pedestrians to a "green light" for cars).

8.3.4. Protocol Performance and Range Estimations

For security, the largest overhead is calculating the parameters for the Diffie-Hellman key exchange. Exact values largely depend on the used hardware and the software implementation. Measurements with the described setup showed values in the order of $10^2 ms$. As long as the protocol needs to be executed on the infrastructure side only for one emergency vehicle[5], this time is acceptable. The time to switch the traffic lights to green and start clearing the intersection will be in the order of seconds and

[3] Large intersections usually have some central reserve where pedestrians may take a rest while crossing a street.

[4] This assumes that the lights switch to red in the same moment as a pedestrian starts to cross a street.

[5] Note that theoretically two or more emergency vehicles approaching from different directions may lead to a conflict. However, in practice this should be coordinated by regional emergency dispatch centers, etc. For the system, the worst case is that a request to switch lights is rejected.

thus have a much bigger impact. A whole run of an emergency vehicle's request to a clear path takes the following steps:

(1) Time for a communication protocol handshake run (see Table 8.2)
(2) Switching time (see section 8.3.3)
(3) Clearance time

The time to switch the traffic lights is the biggest chunk of a whole green-light switching process. Some time for the first cars in the lane to start moving after they got a green light has to be added, called "clearance time". The whole process may take up to around 25 seconds, while a emergency vehicle may travel with an average speed of up to 100 km/h (27 m/s) in the worst case. This would require a communication range of about 650 m, to be able to react in time. This may seem a large distance for 802.11 technology. However, the demonstrator using a boosted antenna provided a range up to 1000m. Even if this is impractical for serial implementations, the protocol has been designed for multi-hop forwarding of control messages to extend the range of an emergency vehicle.

For many other applications in the vehicle to infrastructure scenario these limitations will not be relevant, since the requirements for establishing a secure connection are more relaxed. The transmission of large files or usage of an established connection have been tested and there are no practical limitations other than for usual 802.11 usage. Effectively the protocol is ready to use for a wide range of applictions, with multi-hop connections remaining to be investigated.

8.4. Simulation Environment

Gödel's incompleteness theorem [**Göd31**] states that any logical model of reality is incomplete (and possibly inconsistent) and must be continuously refined/adapted in the face of new observations. This is especially true in a field such as inter-vehicle communication which has been studied only for a few years. In order to adapt to new scientific results and being able to assess various concepts, the tool of simulation seems adequate to evaluate overall system performance. There are other reasons to use simulation: the multitude of parameters that influence the global system behavior cannot be analyzed theoretically due to their sheer complexity. Simulation is a tool to empirically test different configurations and get a "feeling" for implications on system performance. Another reason is that large-scale effects play an important role in the design of a IVC-network. Testing and evaluating these effects using prototypes means building and maintaining prototypes, managing different device and software versions, using public roads, etc. This is too expensive during the early stage of system design.

However, simulation cannot reproduce reality in arbitrary detail, since it is based on a model of the actual world. Therefore, simulation results have to be interpreted very carefully. In most cases absolute conclusions cannot be drawn. Instead, results can only be compared to prior results to make relative statements. Nevertheless, simulations are a powerful tool for all kinds of evaluation, especially with respect to large-scale or long-term effects.

This section describes the simulation enviroment that has been created in a joint effort at BMW Research and Technology and used for inter-vehicle communication research. For this work it is a key instrument to support the design-phase of SARI, evaluate various components, and the system as a whole.

8.4.1. Existing Simulation Software

Different kinds of simulation software, that have been developed in independently, are relevant for this work. In telecommunications, network simulation is used to evaluate communications systems, including various communication protocol stacks and physical distribution models. In the field of transportation, diverse simulation tools have been established to study traffic flows on microscopic and macroscopic levels. In this work this will be referred to as mobility simulation. Finally, there is the application running on a "bare" vehicle communication system itself. As said before, many aspects of distributing messages in such a network are closely linked to application specific modules. In order to simulate a complete IVC-system, the algorithms and protocols of the applications being examined have to be realized. As a whole, this will be denoted as C2C Simulation.

8.4.1.1. Network Simulators. During past decades, a large number of network simulation software has been written. All have special features or advantages, but most are limited to fixed networks or simply extend fixed networks by wireless connections. However, for the ad-hoc network envisioned in this paper, only two network simulators are currently well-known among the research community: GloMoSim[6], which has been developed at the UCLA, and NS-2[7], which has been developed at UC Berkeley.

GloMoSim is a specialized simulation software for wireless ad-hoc networks. It allows a large number of nodes to be simulated, but lacks flexibility and documentation is rather bad.

NS-2 is a powerful and well-known network simulator with extensions such as Rice University's Monarch project [8] that cover also wireless ad-hoc networks. Due to its large user-base, there are many physical and mathematical models, network protocols and special extensions available. But it requires a long familiarization period because it is very complex and extensions will sometimes run only on specific versions of the simulator.

In addition to that, Kosch [**Kos05**] developed CARISMA, a simulation environment that also includes simple line-of-sight path models based map topologies and generates connectivity trace-files. This is done by taking into account the transmission range and a line-of-sight model, which simulates the obstruction of the communications caused by buildings. These pre-generated trace-files can serve as an input for V2V simulations, but all input parameters that depend on node actions can only be estimated in advance by applying statistics. For instance, CARISMA calculates all neighbors that can be reached by a specific node at every simulation interval. But if a node actually communicates, it will use a fraction (or all) of the available communication bandwidth. Consequently, not all of the node's neighbors listed in the trace file can be reached at the same point of time. If scalability is one of the issues to be examined, the simulation results may deviate significantly from reality.

8.4.1.2. Traffic / Mobility Simulators. Three traffic / mobility simulators have been investigated, VISSIM[9], SUMO[10] and GenMobTrace.

VISSIM is a professional traffic simulator which features the most advanced functionality, a sophisticated traffic model and also includes a visualization tool. However, as the source code is not available,

[6] `http://pcl.cs.ucla.edu/projects/glomosim/`

[7] `http://www.isi.edu/nsnam/ns/`

[8] `http://www.monarch.cs.rice.edu/cmu-ns.html`

[9] `http://www.ptv.de/cgi-bin/traffic/traf_vissim.pl`

[10] `http://sumo.sourceforge.net/`

extensions can only be implemented by utilizing the provided APIs, which offer only limited control, especially in the case of visualization and it does not read ESRI-Shapefiles[11].

SUMO is an open-source traffic simulator, which has been developed by the DLR. This simulator also features an advanced mobility model, which allows traffic lights and multiple lanes, and includes a visualization tool.

Lastly, GenMobTrace is a basic traffic simulator and part of the CARISMA simulation, which has been developed by Pfeifroth within the context of his diploma thesis [**Pfe03**]. It features only a very simple mobility model, which does not take into account multiple lanes, takeover maneuvers and traffic signs, for example. A peculiarity of this simulator is the additional simulation of the reachability between two arbitrary nodes, as dicussed in the previous section. Its goal is to create a trace-file [12] of the simulation run, which can be further utilized by other programs. There is also a basic visualization tool which reads the trace-file generated by GenMobTrace and visualizes the simulation run. Only this simulator was actually able to utilize the digital maps which were available at BMW Research and Technology in the ESRI-Shapefile format.

8.4.1.3. V2V Simulators. The effects of traffic-related messaging (type-1 applications) have been studied by independent groups, but there is no current standard or a predominating tool for these simulations. Kosch has been studying traffic-related message distribution within his dissertation [**Kos05**] and also conducted simulations using his own CARISMA environment. While this simulation covers some aspects of message distribution, such as the broadcast storm (dicussed in [**Kos04a**]), neither does it provide a generic framework for IVC applications nor is it built on a flexible simulation architecture. However, GenMobTrace, a part of the simulation environment used for mobility and connectivity, will be used because of its simple handling.

In summary, because of the different strengths of simulators and more importantly their restrictions, almost all combinations have been used. This makes quantitative statements about simulation results difficult, but still allows comparison of different implementations running on the same setup.

8.4.2. Simulation Concept

The simulation environment is logically splitted into two parts.

All modules that will later run within a car's OBU or even within the secure environment (shown in Figure 4.1) are the "local" part. Those modules that emulate parts of the car, the car's environment and all other factors that affect the local part are the "outside" part. The local part and the outside part are connected to each other via defined interfaces. These interfaces have been designed in such a way, that the simulation environment's implementation matches the actual implementation of a experimental vehicle as accurately as possible. The great advantage of this architecture is that software that has been designed for use in cars can be tested in the simulation environment and then evaluated in prototypes with only minor changes.

Following modules belong to the local part:

- **Distribution Module**
- **Sensor Reasoning Module**

[11] ESRI is a standard for shapefile format, see [**EEstd98**]

[12] which includes amongst other information the position of every vehicle and its reachable neighbors, at every simulation interval.

- **Management Module**
- **Database Module**
- **Aggregation Module**

Following modules are represented in the outside part:

- **Mobility = Position**
- **Communication Channel**
- **Sensor Data**
- **Driver Reactions**
- **Attack Simulation**

Some of the modules that have been mentioned in section 4.2 will not be part of the simulation, since their effects on the system can be estimated quite well and their operation is not within the focus of this work:

- **Security Module** The actions of the security module are affecting the system by the time it takes to perform cryptographic functions. The security relevant aspects are covered by the attacker simulation. In other words, all attacks on a global scale are known in the simulation environment, since they have been explicitly included in the attacker model[13].
- **Secure Time & Position** Secure Time & Position is not implemented for the same reasons as the security module: the implications on the system are known because they are derived from a system model. Please refer to section 4.4.3 for details on possible implementations.

8.4.3. Simulator Coupling

As mentioned before, standard simulation software has not been sufficient to simulate inter-vehicle communications, in addition to that, different simulation topics have to be integrated to model a automotive communication environment. The ”simulation environment” is therefore an ensemble of independent software, originally designed to simulate wireless communication of mobile nodes, microscopic traffic flows, weather and traffic situations and intervehicle communication applications.

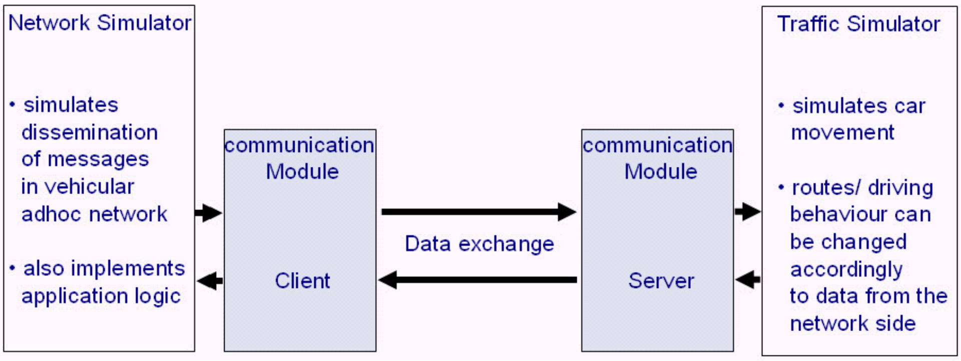
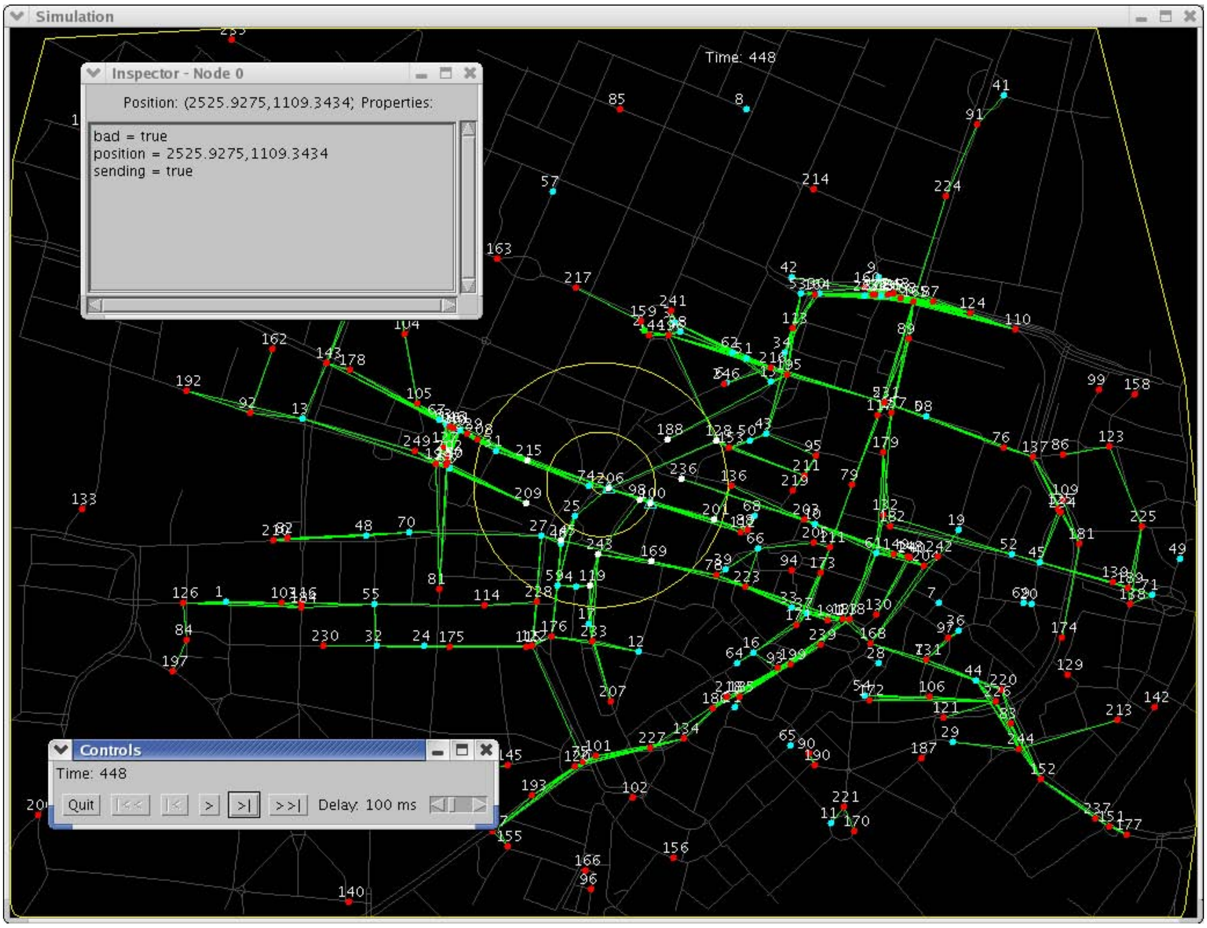

Network Simulator
• simulates dissemination of messages in vehicular adhoc network
• also implements application logic
communication Module
Client
Data exchange
communication Module
Server
Traffic Simulator
• simulates car movement
• routes/ driving behaviour can be changed accordingly to data from the network side

Figure 8.2. Simulator Coupling

Network simulation and traffic simulation are directly coupled using a client-server model as shown in Figure 8.2. A more detailed description of the simulation environment has been published in [**SDK+05**].

[13] Note that this is a significant problem in security engineering: if the system is tested according to the attacker models of the system designers, this can never cover all possible attacks

8.4.4. Visualization Module



Position: (2525.9275,1109.3434) Properties:

bad = true
position = 2525.9275,1109.3434
sending = true

Time: 448

Figure 8.3. A Screenshot of a VANET Simulation Run Playback

The visualization tool is able to playback a simulation run while displaying simulation results computed by the simulation environment. A picture of the visualization tool is shown in Figure 8.3. Its utilization significantly supports the analysis of the simulation results.

Besides showing the road network, hazard areas, simulated vehicles and their reachability, simulation specific events can be visualized, such as the point in time when a decision or detection are made and the outcomes of these actions. Furthermore, the dissemination of hazard information is visualized and attackers are distinguished from normal participants.

8.5. Information Distribution Related Implementations

8.5.1. Efficient Message Forwarding Implementation

The efficient message forwarding algorithm presented in section 4.4.4 has been implemented in Java and analyzed in the simulation environment shown later. The basic algorithm is shown in Figure 8.4.

After receiving a message, the unique event-identifier is taken from the message's header and the database is searched for an existing entry with the same identifier. If such an entry exists, the coordinates of the sender and the current time are added. Otherwise, a new entry is generated.

If a message with the same identifier is already scheduled for transmission, it is removed from the send queue, because it has to be checked whether the message has become redundant. For the case

```
Receive_Message M:

Check_In_Database(M)
Update_Database(M)
Check_Send_Queue(M)
if(Retransmission of M is redundant):
 Save_In_Database(M)
else:
 T = Compute_Distance_Defer_Time(M)
 Schedule(M , T)
```

Figure 8.4. Simplified Algorithm for Efficient Message Distribution Algorithm

it is not redundant, a new time for rebroadcast is calculated. The area around the node is divided in small cells and the algorithm determines which cells will be covered by a new retransmission of the message. If the percentage of additional coverage is sufficient, the message will be retransmitted. The area covered by retransmission is assumed to be a circle in the cell grid as calculated by the Bresenham algorithm (see section 4.4.4.2). After the coverage of the vehicle is determined, the coverage of previous senders is calculated based on data in the database. Only data from nodes that have been transmitting within a small period T_r is used, due to the rapidly changing network topology. T_r is a fixed value, determined from simulation results. In future implementations it could be determined dynamically depending on node speed, etc. The additional coverage (calculated in cells) provided by the actual node is calculated and, if exceeding a limit, scheduled for retransmission. The retransmissions are distributed within a certain time frame, according to the "distance-defer time" model. The idea of this model is that the nodes that are furthest from the last sender should retransmit first to achieve the most efficient distribution.

This algorithm has been tested in the simulation environment in chosen communication scenarios. Its feasibility has been shown for those cases. Although the algorithm is not able to compute the intersection of the coverages exactly, it provides a good approximation.[14] This is sufficient to improve message distribution in densely populated areas with high communication load compared to other methods.

8.5.2. Distributed Data Aggretation Module

The distributed data aggregation module introduced in section 5.3 has been implemented in Java and examined in the simulation environment. This implementation represents the core part of the aggregation module, according to SARI as presented in 4.2.

The module consists of three major methods:

(1) **addMessage** Called whenever new messages are received, this method adds external information about events to the database.

(2) **addDetection** This method is called periodically if the node is within the recognition area and sensor data is added to the database.

(3) **cleanUp** This method deletes obsolete hazards from the database and is called periodically (in the order of minutes).

[14] The accuracy of the computation depends on the cells' size.

addMessage and addDetection are shown as simplified UML action diagrams in Appendix B.2, Figure B.2 and Figure B.3. cleanUp is a simple method that is not explained in detail, refer to [**Här05**] for full description.

As a result can be said that the performance depends on parameters such as the period between two addDetection() calls (which in turn depends on the output of the sensor reasoning module). In those simple scenarios which the module has been simulated in, it showed that it was working as planned. However, for more complex situations, the algorithm's feasibility has not been tested. The implementation also provides basic support for sender classification, which can be used to weigh information according to sender's reputation, see VARS section 5.5.2.1, or different trust level classes, see section 5.6.4. Only a method on how rank information according to a sender's class has to be implemented and tested. The algorithm has one major drawback for a certain kind of events: it does not differentiate between the lanes / directions on a road where an event is detected. Traffic jams for instance, usually only affect one driving direction significantly. A workaround is to add a directional indicator to the messages for those event classes where required.

8.6. VARS Simulation

The simulation's goal is to measure robustness of VARS, presented in section 5.5.2, against nodes that maliciously announce non existing events and to measure how many reputation entries have to be stored.

Two separate experiments are performed. Every experiment consists of simulations which differ in a single simulation parameter. Each simulation consists of multiple simulation runs with a random distribution of car starting points. The results of a simulation are the average of all simulation runs.

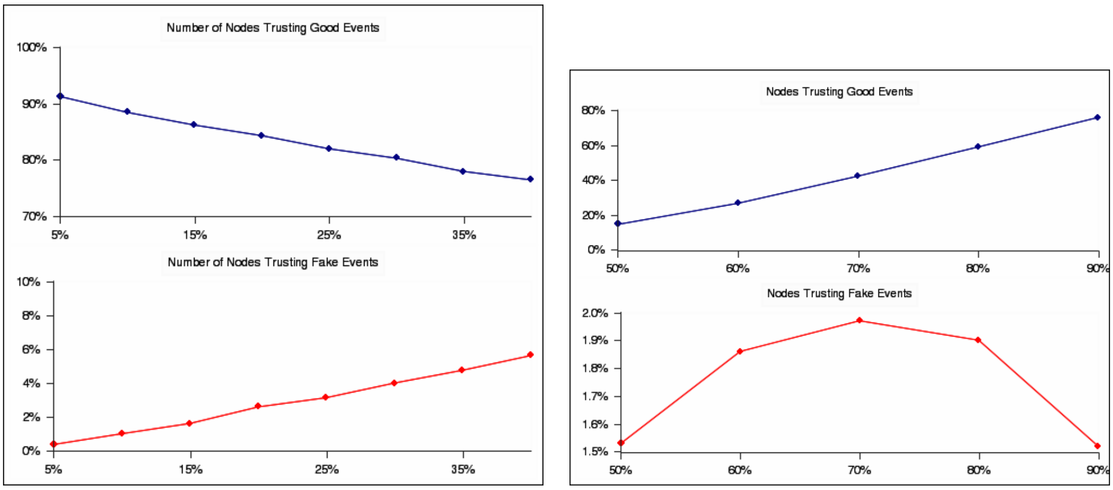
Events are generated every 210 simulation ticks, alternating between good and fake events. A good event is reported by every node, malicious and "good" ones, within recognition range (see paragraph 5.5.2.1). Fake events are only reported by malicious nodes. A simulation runs 10,000 simulation ticks and is performed with 1000 nodes on a 4×6 km city map[15]. The simulation is situated only within the geo-situation "city" with weights of $\alpha = 16/21, \beta = 4/21$ and $\gamma = 1/21$. The number of cars that had confidence that distinct good or fake events are true have been measured.

8.6.1. Robustness Against Malicious Nodes

The problem with good events is, that a number of nodes with good reputation is needed, otherwise the messages wouldn't be trusted. On the other hand, messages on fake events can only be recognized as fake events if the reputation of malicious nodes on other nodes is low. Goal of the experiment is to see the impact of malicious nodes on the confidence decision. The experiment consists of 8 simulations with a percentage of malicious nodes from 5 to 40 percent in steps of 5 percent. The minimum reputation value of messages that would be accepted for evaluation is set to 70 percent.

The results, see Figure 8.5(a), show an almost linear decrement of nodes that trusted good events relative to the increment of malicious nodes. The number of nodes that trusted messages on fake events increased accordingly with approx. 6 percent of all nodes trusting fake events if 40 percent of all nodes are malicious and almost no accepted fake events for 5 percent malicios nodes this seems to be acceptable behavior.

[15] A part of a precise map of Munich.



(a) Results of Robustness Experiment

(b) Results of Capacity Experiment

Figure 8.5. VARS Simulation

8.6.2. Performance in Relation to Memory Usage

This experiment is supposed to show how many reputation values have to be stored for acceptable results. As the targeted networks are huge, it has to be accepted that reputation values cannot be managed for all participants. The question is how well the system works if only a fraction of all node's reputation values can be stored.

The experiment consists of five simulations with memory capacities for 50 to 90 percent of all nodes in steps of 10 percent. The minimum reputation for accepted messages is set to 50 percent to allow for more messages to be accepted.

However, the result are disenchanting. With a storage capacity of 50 percent only 14.5 percent of good events would have been trusted by the system. On the other hand, only 1.53 percent of all nodes would have believed in fake events. See Figure 8.5(b).

8.7. Voting Schemes

The voting scheme introduced in section 5.6 has been simulated in the simulation environment presented above.

8.7.1. Prerequisites and Assumptions

Following assumptions have been made for the simulations:

- Hazards can be deterministically detected by every vehicle.
- Experiences and messages can be assigned unambiguously to a hazard.
- Hazards are static, meaning that they do not change their position, size or intensity. The only change considered is their appearance and disappearance.
- An optimal message dissemination algorithm is utilized, thus whenever two vehicles are within communication range and their communication is not obstructed by buildings, they will exchange all relevant messages. Issues like bandwidth limitations or channel access are not considered.
- Every vehicle uses the same decision and dissemination algorithms.

- The decision if a message is relevant for the dissemination process is based only on its creation time and on the current position of a vehicle.

These prerequisites and assumptions have been formulated by Benedikt Ostermaier in [**Ost05**].

8.7.2. Simulation Setup

For the simulation of voting schemes, connectivity data and mobility data are precomputed using GenMobTrace and written in trace-files. The trace-files are read by an implementation representing the voting architecture, attack simulation and application logic, called **AppSim**. AppSim was developed in order to enable rapid prototyping of applications for VANETs. The simulator itself and the simulated applications are written in Java. Besides the application logic, message forwarding algorithms can be simulated, based on the reachability information computed by GenMobTrace. AppSim is also a time-discrete simulator, its resolution is up to one millisecond.

Recapitulating the toolchain, the basic workflow of a simulation run consists of three consecutive steps. First, the vehicle movements and their reachability information are simulated with the help of GenMobTrace. Subsequently, the trace-file is utilized by AppSim which simulates not only the traffic-related messaging application but also the virtual vehicles the application resides in and the message exchange necessary for communications. AppSim outputs additional trace-files, which together with the results from GenMobTrace, form the basis for the evaluation of the simulation results. This evaluation can be done by manually inspecting the files, generating charts and visualizing the simulation run.

8.7.3. Simulation Scenarios

In all simulation scenarios, 250 vehicles have been simulated for 1200 seconds on a 8 km^2 section of a digital map of Munich. The choice of an urban scenario was based on the fact that GenMobTrace was designed to work best in such environments. Vehicles were placed randomly across the street network and move according to a model after Krauss [**Kra98**], while choosing their destinations according to the random waypoint model. The maximum communication range has been set to 400 meters, and messages will expire 200 seconds after they have been created. A simplified message forwarding algorithm is utilized, which transmits relevant messages to every node which is directly reachable, at every second of simulation time. A hazard has been placed at an intersection in the center of the simulation area, appearing after 100 seconds of simulation time. For fake attacks, this hazard is fictitious and disappears after 1100 seconds of total simulation time, thus resulting in an attack time of 1000 seconds. For flip attacks, this is a real hazard, which is present for 500 seconds. In this case, attackers start their flip attack as soon as the hazard appears, and continue for 500 seconds of simulation time after the hazard has disappeared, hence also resulting in an attack time of 1000 seconds.

A summary of the simulation parameters is shown in Tab. 8.3.

An important but implementation specific detail is the handling of vehicles which leave the dissemination area. Normally, such vehicles would keep messages as long as they are relevant, since the drivers may enter the dissemination area again. However, the utilized mobility simulator does neither allow for vehicles to leave the simulation area nor does it enable vehicle sources and sinks. Therefore, all simulated vehicles will stay within the simulation area for the whole simulation time. This leads to the situation that simulated vehicles will repeatedly enter and leave the areas of a hazard for which they already

Simulation area:	8 km^2
Number of vehicles:	250
Simulation time:	1200 sec
Lifetime of a message:	200 sec
Communication range:	400 m
Hazard appearance time:	100 sec
Hazard disappearance time for fake attack:	1100 sec
Hazard disappearance time for flip attack:	600 sec
Start of fake/flip attack:	100 sec
Stop of fake/flip attack:	1100 sec
Diameter of recognition area:	50 m
Diameter of decision area:	300 m
Diameter of dissemination area:	700 m

Table 8.3. Simulation Parameters

possess relevant messages, which is denoted as **taxi effect**. In order to avoid this effect, which affects the simulation results especially in low density VANETs, all vehicles which leave the dissemination area of a hazard drop all messages concerned, thus erasing all records of this danger.

The number of messages being considered by **Majority of Freshest X** has been set to 22, and the message threshold necessary for **Majority of Freshest X with Threshold** has been set to 2. Hence, the latter voting method requires at least three received messages for a hazard in order to be able to reach a positive decision. These values have been determined by the analysis of various traffic patterns, considering the above mentioned parameters.

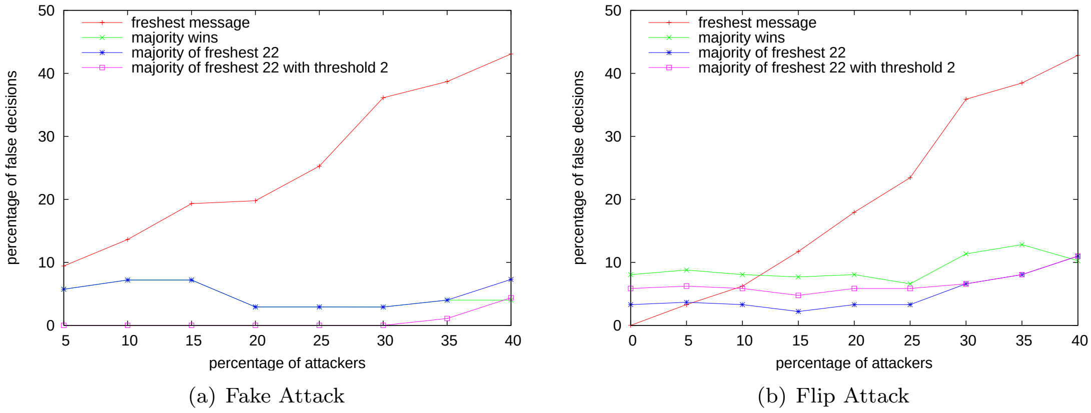
In order to evaluate the four introduced voting methods, they have been simulated at first without any attackers, in order to be able to investigate their basic performance. Subsequently, both the **fake attack** and the **flip attack** have been simulated with increasing fractions of attackers, ranging from 5% to 40% in steps of 5%, to investigate the robustness of the four voting methods. Hence, 68 simulation runs[16] have been conducted in total for the succeeding analysis, which are all based on the same trace-file generated by GenMobTrace. The decision, whether a simulated vehicle is a well-behaving node or an attacker is based on the internal node number generated by GenMobTrace. For a fraction of 5% of attackers, the first 5% of the nodes are considered to be attackers, thus making every set of attackers a superset of the smaller sets of attackers. As initial vehicle positions are assigned randomly, so is the initial distribution of attackers.

8.7.4. Results

The simulation results are visualized in Figure 8.6 and analyzed in following sections. In order to measure the performance of the voting methods, the percentage of false decisions with respect to the total number of decisions reached within a simulation run has been utilized. A decision is considered false whenever its result does not match the status of the actual hazard at the time when the decision is reached. Both diagrams show the performance of each voting method with respect to an increasing number of attackers, and for the fake attack, also in an attacker-free scenario.

Besides the visualization tool and the manual inspection of trace-files, detailed progression diagrams have been utilized, in order to analyze the simulation results. These diagrams show the development of

[16] The total number of 68 simulation runs results from 4 voting methods $*$ (1 scenario without attackers + 8 fake attack scenarios + 8 flip attack scenarios).



(a) Fake Attack

(b) Flip Attack

Figure 8.6. Comparison of the Four Voting Methods

important key figures during the course of simulation time, and are generated from trace-files of AppSim, for each simulation run. An exemplary progression diagram is shown in Figure 8.7.

8.7.4.1. Freshest Message. Without any attackers, no false decisions at all have been made. This can be explained as follows: When the hazard appears, there are already some vehicles inside its recognition area, which detect the danger at once and disseminate warning messages. Thus, subsequent vehicles are warned and will make a correct decision. In case the hazard disappears, there has to be a vehicle which is inside the decision area but outside the recognition area, and which is informed about the hazard, thus having already reached a positive decision. This vehicle has to enter the recognition area of the hazard after it disappeared and before any other vehicle makes a decision. In this way, a revocation message is created and distributed, informing other vehicles about the disappearance of the hazard.

Looking at the fake attack, it reveals that the number of false decisions increases almost linearly with the number of attackers. The fraction of false decisions thereby is near the fraction of attackers. As expected, a protection against this type of attack cannot be identified.

Regarding the protection against the flip attack, the situation seems similar. However, as long as there are less than 20% of attackers, the results are better than those of the fake attack. This is because at the beginning of the fake attack, there are a lot of false decisions, which can be explained as follows: As soon as the first warning message is disseminated (which is faked), all vehicles inside the decision area have to reach a late decision, solely resulting in false decisions. In contrast thereto, at the beginning of the flip attack, there are most likely already some messages around, if the hazard has been recognized by an honest vehicle at first. So, for this attack, attackers do not necessarily possess the starting advantage that they have for the fake attack. With an increasing number of attackers, the starting advantage of the fake attack becomes negligible and the number of false decisions for the fake attack and the flip attack converge.

8.7.4.2. Majority Wins. It reveals that Majority Wins produces a not negligible number of false decisions even when there are no attackers around. Analyzing the corresponding progression diagram, it shows that all of these false decisions take place after the hazard has disappeared, and persist as long as there are more warning messages than revocation messages around. This period of time is denoted as the **second adaption phase**, it is influenced by the lifetime of the messages. It lasts from the disappearance time of the hazard until about $(disappearance\ time) + (message\ lifetime)/2$. Unlike the fake and the flip attack, there is no first adaption phase when there are no attackers around.

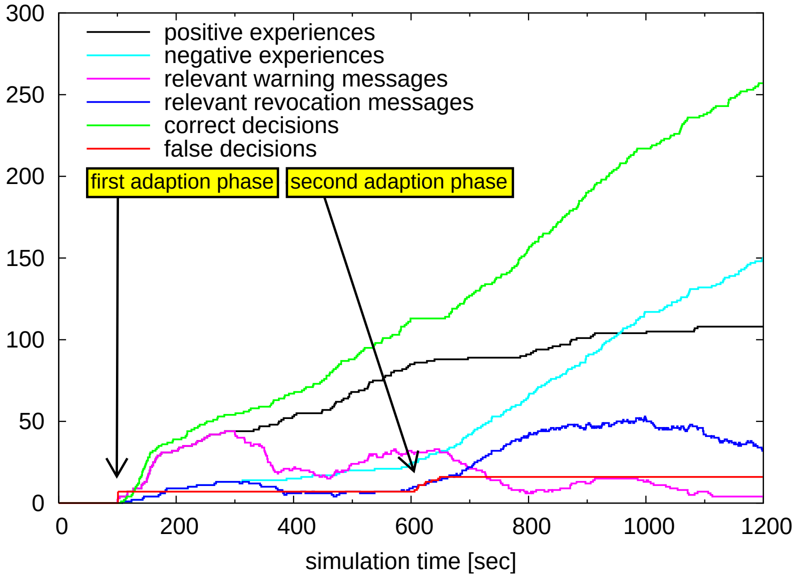


Figure 8.7. Progression diagram visualizing the development of important key figures of the simulation run: Majority of Freshest 22 with Threshold 2, Flip Attack, 20% Attackers

Looking at the fake attack, the number of false decisions is limited and does not significantly rise with an increasing number of attackers. This is because all false decisions are reached at a short period of time, after the faked hazard is announced. This period of time is called **first adaption phase**, it is usually much shorter than the second adaption phase. After this period, there are continuously more revocation messages than warning messages, thus resulting in exclusively correct decisions within the remaining simulation time. The decline of false decisions between 15% and 20% of attackers (Figure 8.6(a)) is an interesting aspect. In the former case, a lot of vehicles are within the decision area when the first attacker is sending his faked warning message, so all of these vehicles have to reach a late decision, resulting in false decisions. In the latter case, since there are more attackers around, the attack is launched at a sooner point in time. Because there are less vehicles in the decision area at that time, fewer false decisions are made.

Considering the flip attack, the number of false decisions remains almost constant and even decreases marginally until a fraction of 25% of attackers is reached. Looking at the corresponding progression diagrams, it reveals that up to this point, false decisions are still reached only within the second adaption phase, thus providing robustness against false messages. The effect of a decrease of false decisions with an increasing fraction of attackers can be explained by an increasing number of (false) revocation messages before the hazard disappears. Therefore, the difference between warning and revocation messages is smaller, which shortens the second adaption phase. The reverse effect, namely an extension of the second adaption phase due to more false warning messages after the hazard has disappeared occurs with 30% of attackers, resulting in more false decisions. Additionally, there are now some false decisions for a short period of time after the hazard appears. Just like for the fake attack, this period of time is denoted as the **first adaption phase**, and is usually much shorter than the second adaption phase. The decrease of false decisions at 40% of attackers can again be explained by a shortening of the second adaption phase.

Comparing both attacks, it reveals that the fake attack is substantially better handled. This is because there is no second adaption phase for this attack. Since for both attacks, all false decisions are reached within the adaption phases, a shortening of these two phases may help to decrease the number of false decisions. This is aspired by the next voting method.

8.7.4.3. Majority of Freshest X. By considering only the recent x messages for the determination of the majority, a shortening of the adaption phases is aspired. As noted before, x has been set to 22 for the conducted simulation runs.

The false decisions of Majority of Freshest X in a scenario without attackers are substantially lower than those for Majority Wins. This is because the length of the second adaption phase is successfully reduced (as before, there is no first adaption phase in such a scenario).

The performance with regard to the fake attack is almost identical with that from Majority Wins. This is because the first adaption phase is very short, and there is no second adaption phase for this attack, so this voting method does not improve the protection against the fake attack. Furthermore, the result for 40% of attackers is worse than that of Majority Wins. An analysis of the progression diagram reveals that in this case, there are false decisions after the first adaption phase, which are caused by considering only the recent 22 messages received.

Looking at the flip attack, there are generally less false decisions than with Majority Wins, except for a fraction of 40% of attackers. The reason for the latter is similar to the corresponding scenario with the fake attack, in such a way that there is a considerable number of false decisions between the two adaption phases. An analysis of the progression diagrams reveals that this kind of false decisions can be first observed with 30% of attackers.

Comparing the results of both attacks, it shows that the flip attack produces fewer false decisions than the fake attack, until about 25% of attackers are reached. This can be explained by the shortening of the second adaption phase, which then produces only little false decisions compared to the first adaption phase of the respective fake attack. A further increase of attackers leads to false decisions before the second adaption phase, thus outnumbering the false decisions of the fake attack.

In summary, this voting method significantly improves the protection against the flip attack. However, the performance of the fake attack is not enhanced. As almost all false decisions are reached at the beginning of this attack, a shortening of the first adaption phase would generally improve the performance of this voting method with respect to fake attacks.

8.7.4.4. Majority of Freshest X with Threshold. The last voting method tested is an enhancement of Majority of Freshest X. The goal is to reduce the number of false decisions regarding fake attacks, by utilizing a message threshold. In order to be able to reach a positive decision at all, a minimum number of messages has to be received at the time a decision is made. As noted before, the threshold has been set to 2, thus requiring at least 3 messages in oder to invoke Majority of Freshest X.

The performance in a scenario free of attackers is slightly worse than that of Majority of Freshest X. This is because the threshold adds a first adaption phase, in which all vehicles reach a false decision. As soon as there are at least 3 warning messages disseminated, correct decisions are reached until the second adaption phase.

Looking at the results of the fake attack, it reveals that this voting method provides a significant protection against this type of attack. Up to a fraction of 30% of attackers, no false decisions are reached at all. With 35% of attackers, a small first adaption phase emerges, and with 40% of attackers, false decisions are reached in the middle of the attack.

Considering the protection against flip attacks, it shows that it is comparable to the one provided by Majority of Freshest X. However, the number of false decisions is generally higher for fractions of attackers below 35%. This is a direct effect of the utilized threshold, which results in false decisions after the hazard appears, even when there are no attackers around at that time.

Altogether, this voting method almost completely avoids false decisions with regard to fake attacks, while its performance on flip attacks slightly decreases, when compared with Majority of Freshest X (Figure 8.6(b)).

8.8. AODV-SEC

In order to assess the performance of the modified routing protocol AODV-SEC, that has been described in section 4.4.4, simulations have been conducted using the GloMoSim network simulator because of its high scalability and the availability of AODV protocol source code. Node mobility has been included by employing pre-calculated CARISMA trace-files, based on a digital map of munich.

All simulations are based on the wireless LAN standard 802.11b with a nominal bitrate of 2 $\frac{Mbit}{s}$. The radio-range has been set to 250m. For the simulations no real cryptographic functions have been implemented to be able to simulate large scenarios with several hundred nodes. Wait times have been used to simulate the delays usually generated by the cryptographic functions. The wait times used in the simulation can be seen in table 8.4.

Operation	**Time Delay**
Create Hash-Value	$1ms$
Create Signature	$800ms$
Verify Signature	$400ms$

Table 8.4. Time delays due to cryptographic operations

The first simulations were used to show the protocol's operability and to get a first impression on protocol performance. Several simulations have been done in a static scenario (no topology changes) with different number of hops. The times for the route discovery can be seen in table 8.5.

Simulation Data	**2 Hops**	**3 Hops**	**4 Hops**	**5 Hops**
Cached Keys	$4,046s$	$6,453s$	$8,878s$	$11,298s$
Non-cached Keys	$8,051s$	$12,862s$	$17,699s$	$22,492s$
Regular AODV	$45ms$	$62ms$	$93ms$	$113ms$

Table 8.5. Route discovery times of different multihop connections

At first sight a route discovery time of several second seems to be too high compared to the milliseconds of plain AODV. Unfortunately these values show the real performance taking into account the delay times for the cryptographic functions introduced in table 8.4. The route discovery can also be described with analytical equations.

The time for a route discovery to a direct neighbor can be calculated by:

$$t_{RDisc} \approx 2(T_{CheckHash} + t_{VerifyKey} + t_{VerifyPacket}) + t_{SignPacket} + t_{CreateHash} + \sigma \tag{8.1}$$

The value σ stands for any extra delay, e.g. transmission delay and packet generation. Taking into account this equation it becomes clear why the route discovery process takes several seconds. Therefore, if the delay times shown in table 8.4 can be reduced significantly the protocol will perform much faster.

If a node detects a hostile node in the route, it will be exluded from the route to the destination. This process was also validated in the simulations.

8.8.1. Mobile Scenarios with High Network Load

In a second step the protocol was evaluated with two different scenarios:

- **Scenario 1**: $1500m \times 300m$ simulation area with 50 mobile nodes
- **Scenario 2**: $2200m \times 600m$ simulation area with 100 mobile nodes

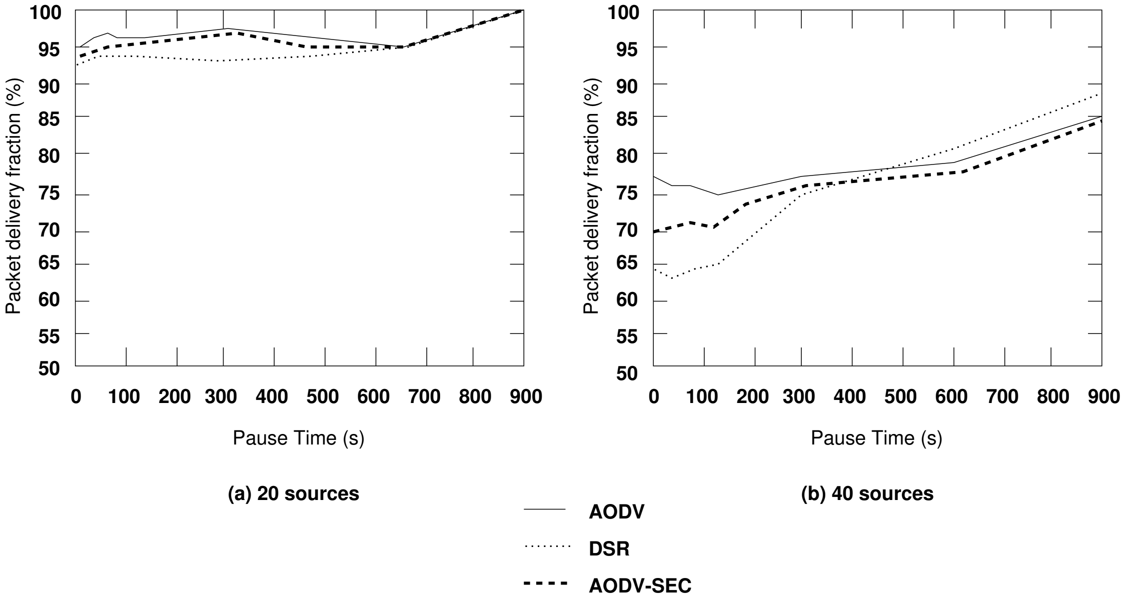


Figure 8.8. Packet Delivery for Scenario 1

For the mobility model the *random waypoint model* has been used. The number of traffic sources has been set to either 10 or 40 sources sending data-packets of 512-bytes as a constant bit rate source. The simulations for these scenarios have been done for plain AODV, DSR and AODV-SEC to be able to compare AODV-SEC to conventional ad-hoc routing protocols. Several issues have been analyzed, such as packet delivery, routing load, MAC load, throughput as well as delays resulting from the route discovery process.

In Figure 8.8 the packet delivery fraction which has been simulated for scenario 1 is plotted. AODV-SEC performs nearly as good as the plain AODV. For scenario 2 the results are equivalent. Especially for pause times smaller than $300s$ both plain AODV and AODV-SEC outperform DSR.

In Figure 8.9 the delays caused by AODV-SEC are plotted. Depending on the number of hops the delay increases. If the number of connections increases, hence, the load increases, the delay increases but aproaches a certain limit value, as can be seen in the graphs.

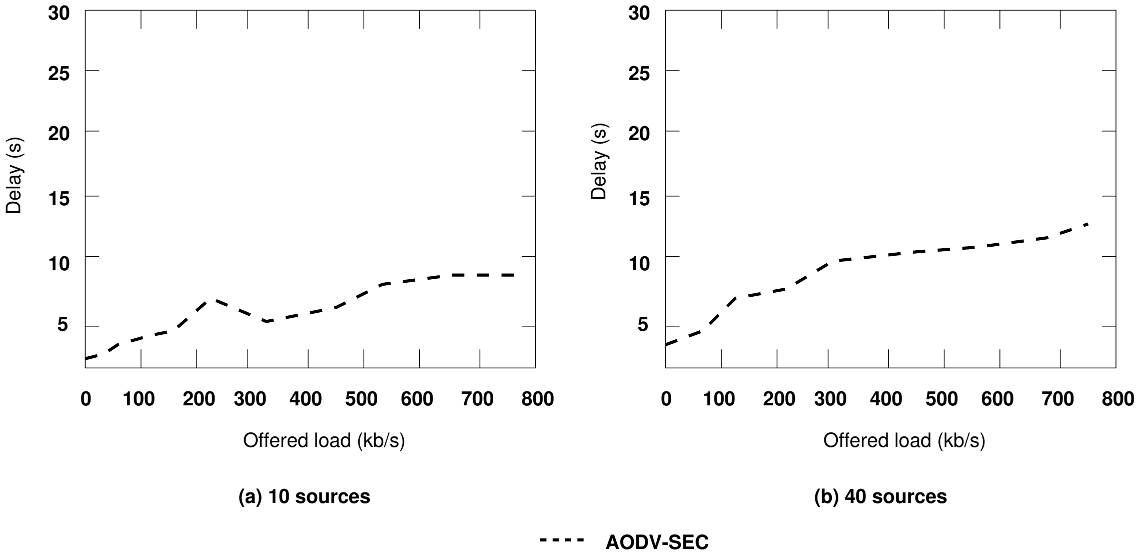


Figure 8.9. Route Discovery Delays for Scenario 2 (no pause time)

The performance of AODV-SEC has been evaluated by simulating the throughput that can be delivered successfully. In Figure 8.10 the corresponding results are shown. The result prove that AODV-SEC performs almost as good as plain AODV and again performs better as DSR.

Generally, AODV-SEC performs almost as good as plain AODV. It is suitable for the use in mobile ad hoc networks. However, the delay added by cryptographic operations is a major drawback. Future work should concentrate on finding alternatives to those methods or seek optimized operation.

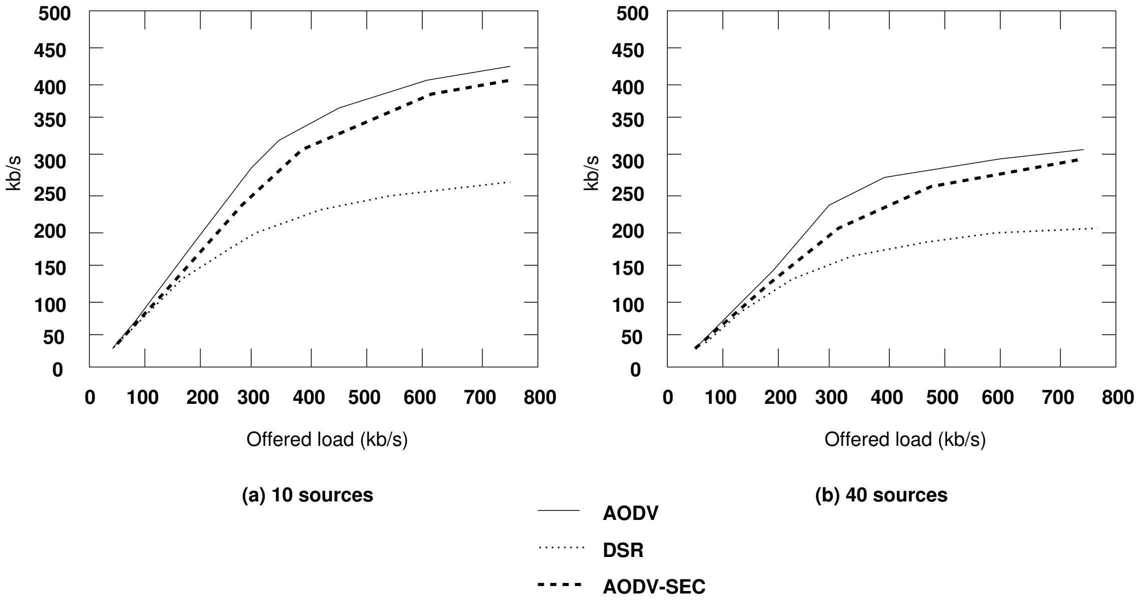


Figure 8.10. Data-Throughput Results for Scenario 2 with increasing load

8.9. Summary

Apart from the advantage of getting hard results, real-world hardware offers the advantage that one can get real hands-on experiences with the equipment, which stimulates improvements. Seeing a system from a product oriented perspective leads to new questions, intuitive solutions and definitely a better understanding. Especially dealing with in-car systems and automotive standards lead to a number of decisions that cannot all be discussed in this work. Many design decisions concerning this work have been made after insightful experiences with prototypes.

Although not part of this work, an interface between the simulation environment and a real BMW 7 series experimental vehicle has been implemented, providing a way to evaluate the effectiveness of proposed solutions in a realistic environment as the findings confirm simulation results. This interface has been done in such a way that from the experimental vehicle's perspective, the other nodes within the simulation environment are experienced as if connected to other cars in the real world. Using this link in the opposite direction, the experimental vehicle's actions are included in the simulation as one of many other network nodes. This configuration has been used to examine some of the approaches in addition to simulation only and has stimulated valuable feedback from fellow researchers.

CHAPTER 9

Conclusion

This chapter concludes the thesis and will summarize the approach taken. After recapitulating the topics covered in previous chapters and indicating focal points, major contributions and results will be presented in the first section This includes a list of publications connected to this thesis. The second section is dedicated to identify directions for future work and indicate major unsolved problems / questions emerging from this work. Finally, the last section gives a short estimation about possible developments in the field of inter-vehicle communications and alternative usage scenarios.

9.1. Conclusion

9.1.1. Summary

The first two chapters provided an understanding of the problem setting for this thesis and its specific challenges. The main points in this context have been determining the specifics of an automotive environment and their influence on system architecture. An important part has been dedicated to traffic-related applications and their categorization into three scenarios. In chapter three a lightweight system model has been introduced as basis for the following threat analysis. Attack trees and DRED model have been used as tools to achieve a detailed and substantiated picture, allowing to define security objectives for system design. Before being able to build an overall architecture, constraints and assumptions have been complemented with points affecting architecture and implementation, in chapter four. Building the resulting Secure Architecture for Robust Inter-vehicle communication - SARI, fundamental building blocks have been designed, developed, and described to provide a common basis for all application scenarios. Privacy has been thoroughly discussed and an anonymous credential architecture has been developed, including cryptographic protocols for credential generation, signatures and key exchange. Chapter five to seven are focused on the tree application scenarios, traffic-related vehicle to vehicle communication discussed in chapter five being the the most important one. In this scenario the diversity of traffic-related applications has been investigated, leading to a classication proposal based on resulting criteria. The main question however was how information can be validated or rated according to its correctness in a distributed system. Three concepts have been proposed, with two of them implemented and evaluated through simulation. Vehicle to infrastructure communication has been discussed and specific requirements have been formulated in chapter six. These needs have been met by designing a new cryptographic protocol, allowing the establishment secure communication, while preserving privacy. This protocol and its performance has been evaluated with a lab setup using traffic-lights and test vehicles. In order to find a suitable basis for connection-oriented communication, different secure Mobile Ad-hoc NETwork routing algorithms have been analyzed, including performance comparison by simulation in chapter seven. A new secure routing algorithm has been designed because of decisive shortcomings of those algorithms with respect to credential handling. Simulation and implementation efforts have been grouped together in chapter eight and present the most important results thereof.

9.1.2. Major Contributions and Results

The intention of this section is to point out major achievements or scientific contributions of this thesis.

- **Categorization of Application Scenarios**
  The first fundamental contribution is the categorization into three application scenarios stemming from the realization that the functional requirements are too diverse to cover all in a single, monolithic sytem. It is important, because it allowed to address the constraints, requirements and security objectives of each scenario purposefully and provides a basic structure for ongoing work.
- **Traffic Scenarios**
  The idea of defining traffic scenarios (here: "highway", "city", "rural") is to be able to adapt the system setup to different traffic and communication environments and therefore be able to achieve better system performance. Locally available data, such as velocity patterns, link duration or wireless channel saturation are used by each node to identify traffic scenario it is in. The concept of traffic scenarios is fundamental for parametrization of algorithms and methods that are introduced throughout this thesis. The availability of global parameters in simulations allows an evaluation of the nodes' effectiveness in recognizing their traffic scenario. Simple trials have shown promising results.
- **Security Analysis Inter-vehicle Communication**
  The threat analysis and risk estimation is based on long term analysis and frequent discussions with other research groups. It provides a structured overview on security issues and identifies important security objectives in the new security field of inter-vehicle communications.
- **Privacy Architecture and Anonymous Credentials**
  As pointed out before, privacy is one of the main concerns with IVC. The design of a privacy architecture with anonymous credentials provides an adjustable degree of privacy, while offering the possibility of identifying single nodes under special circumstances, thus increasing maintainability and security. The development of dedicated protocols adapted to this architecture allowed to achieve the challenging privacy requirements.
- **Efficient Message Distribution**
  The message distribution scheme presented in this thesis is a fundamental concept, that uses cross-layer interaction to improve the distribution mechanism. It extends existing schemes from directed, one-dimensional to a omni-directional approach. Although adding some delay through distance-dependant backoff, it showed that significant improvement towards a more scalable and more robust network is feasible.
- **Distributed Data Aggregation**
  The aggregation of data concerning a hazard with large geographic dimensions using a distributed algorithm is a new problem introduced through traffic-related applications. The provided algorithm is a first proposal towards a reliable solution.
- **Information-centric Trust**
  The idea is that on one hand digital credentials do not solve the question of verifying traffic-related information alone, on the other hand, the traffic-related messaging scenario offers some criteria that support other approaches. Starting from this perception, a number of decentralized

trust schemes have been compared, leading to three preferred approaches, where two have been investigated in depth, reputation systems and voting systems.

- **Reputation System**

  The development of a reputation system for use in Vehicle Ad-hoc NETworks has lead to some specific concepts such as delayed decision, opinion piggybacking, geo-/situation-oriented reputation levels, sender based reputation level and pseudonym flexibility. Those concepts have been developed one-by-one as answers to specific shortcomings. The reputation system as presented offers many opportunities for adjustment and customization. However, the achieved degree of complexity is very high and an effective use remains limited to advantageous scenarios.

- **Voting Schemes**

  The existing idea of using voting schemes in Vehicle Ad-hoc NETworks has been improved with respect to practical einschränkungen of wireless ad-hoc networks. The main contribution is the work about developing the most robust and efficient voting mechanisms in large-scale networks. Simulations have shown that although such a mechanism does not guarantee 100 % authenticity, it works sufficiently well and attacks are much harder in practice than expected.

- **Vehicle to Infrastructure Protocol**

  The key establishment protocol developed for vehicle to infrastructure communication fulfills the requirements link confidentiality and freshness without reliable clocks in fixed nodes. This keeps a degree of security known from communication with hotspots, while meeting above requirements at the same time. The protocol is able to be executed on a smart card with respect to computational effort and memory consumption, thus fitting into the overall architecture with tamper resistant devices.

- **Secure Routing Mechanism and related Framework**

  The requirements for connection oriented communication, especially node authentication and flexible credential handling have led to the development of a secure routing variant of existing AODV protocol. The protocol shows reasonable performance in comparison with established protocols and therefore provides an important part of inter-vehicle communication.

In addition to the topics presented in the paragraph above, there are other topics, which either represent only minor contributions with respect to the problem outline of this work or only minor efforts have been made. Secure Positioning for instance is a fundamental building block for this thesis, existing solutions have been evaluated and a combination thereof have been proposed to offer a practical solution. The concept of a virtual backbone has great potential to increase the performance of IVC and further research is encouraged. Local reasoning and plausibility checks can dramatically improve the ability to present reliable and correct information. The distinction between recognition area, decision area and distribution area reflects the processes and methods defined within SARI and offers crucial advantages, such as late decisions. The simulation environment and OSGi framework that has been used for prototyping and simulations has been developed in a joint effort and represents a unique development, verification and demonstration tool. Although much work has been done to establish this platform and it has greatly improved much of the development, it is not the kind of work directly answering the quesions of the problem outline.

Concerning implementation efforts and prototyping, all major protocols or methods, complementing the theoretical concepts, have been implemented in software. Topics with a strong correlation to hardware technologies have been realized with prototypes: the privacy architecture and credential generation / distribution methods have been implemented on smart cards interacting with on-board system of a BMW 7 series prototype. The protocol developed for vehicle to infrastructure communication has been evaluated in a traffic-light scenario with implementations on the same prototype car and a full traffic light setup, certified for road traffic. Application related methods, algorithms and protocols, such as the scheme for efficient message distribution, the distributed data aggregation algorithm and voting schemes, have been designed to run in a OSGi framework and therefore able to run on prototype cars, lab equipment and the simulation environment.

9.1.3. Publications

Publications that have been made in context with this thesis:

(1) Alexander Buchmann, Florian Dötzer, Harald Görl, Sven Lachmund: Lösen TCPA und Palladium die Sicherheitsprobleme von heute?, DACH Security, Basel, Switzerland, March 30-31, 2004 [**BDGL04**]

(2) Stephan Eichler, Florian Dötzer, Christian Schwingenschlögl, Javier Fabra, Jörg Ebers-pächer: Secure Routing in a Vehicular Ad Hoc Network, IEEE VTC 2004 Fall, Los Angeles, USA, September 26-29, 2004 [**EDS$^+$04**]

(3) Florian Dötzer, Florian Kohlmayer, Timo Kosch, Markus Strassberger: Secure Communication for Intersection Assistance, WIT 2005: 2nd International Workshop on Intelligent Transportation, Hamburg, Germany, March 15-16, 2005 [**DKKS05**]

(4) Florian Dötzer: Privacy Issues in Vehicular Ad Hoc Networks, Workshop on Privacy Enhancing Technologies, Dubrovnik (Cavtat), Croatia, May 30 - June 1, 2005 [**Doe05**]

(5) Florian Dötzer, Lars Fischer, Przemyslaw Magiera: VARS: A Vehicle Ad-Hoc Network Reputation System, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Taormina, Italy, June 13-16, 2005 [**DFM05**]

(6) Florian Dötzer, Timo Kosch, Markus Strassberger: Classification for traffic related inter-vehicle messaging, 5th IEEE International Conference on ITS Telecommunications, Brest, France, June 27-29, 2005 [**DKS05**]

(7) Christoph Schroth, Florian Dötzer, Timo Kosch, Benedikt Ostermaier, Markus Strassberger: Simulating the traffic effects of vehicle-to-vehicle messaging systems, 5th IEEE International Conference on ITS Telecommunications, Brest, France, June 27-29, 2005 [**SDK$^+$05**]

(8) Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh and Tim Leinmüller: Attacks on Inter-Vehicle Communication Systems - An Analysis, Technical Report, NOW-Project, October 2005 [**ABD$^+$05**]

(9) Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh and Tim Leinmüller: Attacks on Inter-Vehicle Communication Systems - An Analysis, WIT2006: 3rd International Workshop on Intelligent Transportation, Hamburg, Germany, March 14-15, 2006 [**ABD$^+$06**]

(10) Benedikt Ostermaier, Florian Dötzer, Markus Strassberger: Enhancing the Security of Local Danger Warnings in VANETs - A Simulative Analysis of Voting Schemes, The Second International Conference on Availability, Reliability and Security (ARES 07), Vienna, Austria, 2007, 432-442 [**ODS07**]

9.2. Directions for Future Work

The feasibility of the concepts presented in this work has been shown in lab tests, functional prototypes and simulations. For large field tests, such as SIM-TD (e.g. see [**dA07**]), components have to be improved and the system integration has to be ensured.

Event class definitions and event class presets presented in this work are only examples. More thoroughly investigated proposals and an extended set of event classes is necessary to achieve an effective system.

Attack simulations on the reasoning and confidence decision system should be extended in order to examinate the dynamic behavior of such complex systems. For instance the effects of a traffic-related messaging system on traffic itself and potential feedback loops between messaging system and traffic flow should be investigated. To give an example: first, detection of hazards that slow down the traffic may lead to cars changing the route. Second, if most of the cars are re-routed, the hazard will not be detected any more, leading to a new stream of cars ending up in a traffic jam before the hazard. This can lead to a wave-like traffic density distribution. Additionally, this weakens the argument to buy such a system, since cars that are not equipped with such a system enjoy being relieved of those cars that are, in situations of heavy traffic. On the other hand cooperative attackers have not been simulated. In reality however, colluding attackers may be very likely depending on the attack target. Game theoretical approaches should be integrated in attack simulations to reflect this.

On the practical side, the political feasibility of the virtual backbone idea (presented in section 4.4.5) should be evaluated and the technical opportunities should be investigated. There is a large potential for improvement, especially in densely populated city environments.

9.3. Outlook

The wide-spread usage of a system such as the one proposed in this thesis is still far away. Many questions have to be answered to achieve the full operability of such a system. However, for some specific use-cases a direct benefit is much easier to reach and market introduction may be sooner than most researchers realize. Apart from a car-based approach, other types of network nodes are to be investigated. Considering commercial aircraft, some of them are already equipped with 802.11 technologies to connect to airport gates on ground. Thinking about densely populated aeronautic routes where aircraft are often within line of sight (therefore theoretically providing 802.11 communication during flight) this is not too distant from the problem outline in vehicle based networks. Other examples are trains or subways where some of the concepts may be applied. Both, commercial aircraft and public transportation have in common, that traffic patterns are much more predictable than in individual transport systems. Another direction is towards integration of (personal) mobile devices, such as mobile phones, PDAs, laptops, etc. and an inter-vehicle communication network. This offers advantages in both directions: IVC networks benefit from the integration of personal (and personalized) data while mobile device users benefit from information available in IVC networks, i. e. actual positioning data or traffic information.

References

[ABD$^+$05] Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh, and Tim Leinmüller. Attacks on intervehicle communication systems - an analysis. Technical report, NOW-Project, 2005.

[ABD$^+$06] Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh, and Tim Leinmüller. Attacks on inter-vehicle communication systems - an analysis. In *Proceedings of 3rd International Workshop on Intelligent Transportation*, Hamburg, Germany, March 2006.

[AG00] N. Asokan and Philip Ginzboorg. Key-agreement in ad-hoc networks. *Computer Communications*, 23(17):1627–1637, 2000.

[ALRL04] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.

[And01a] R. Anderson. Why information security is hard - an economic perspective. In *Proceedings of the 17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, USA, 2001.

[And01b] Ross Anderson. *Security Engineering*. Wiley Computer Publishing, 2001.

[AT99] George Aggelou and Rahim Tafazolli. Relative distance micro-discovery ad hoc routing (rdmar) protocol. online, September 1999.

[Bak05] Humayun Bakht. The history of mobile ad-hoc networks. *Computing Unplugged*, 200508, August 2005.

[BB01] Sonia Buchegger and Jean-Yves Boudec. The selfish node: Increasing routing security for mobile ad hoc networks, May 2001.

[BB02a] S. Buchegger and J.-Y. L. Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403–410, January 2002.

[BB02b] Sonja Buchegger and Jean-Yves Le Boudec. Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pages 403 – 410, Canary Islands, Spain, January 2002. IEEE Computer Society.

[BDGL04] Alexander Buchmann, Florian Doetzer, Harald Görl, and Sven Lachmund. Lösen tcpa und palladium die sicherheitsprobleme von heute? In *Proceedings of DACH Security*, Basel, Switzerland, March 2004.

[BE04] J. Blum and A. Eskandarian. The threat of intelligent collisions. *IEEE IT Professional*, 6(1):24–29, 2004.

[BEGA03] Rakesh Babu Bobba, Laurent Eschenauer, Virgil Gligor, and William Arbaugh. Bootstrapping security associations for routing in mobile ad-hoc networks. In *Proceedings of GLOBECOM 2003*, December 2003.

[Bey02] Dave Beyer. Wireless mesh networks for residential broadband. In *Proceedings of National Wireless Engineering Conference*, San Diego, USA, 2002.

[BFIK99] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: security issues for mobile and distributed objects*, pages 185–210, 1999. Springer Verlag London, UK ISBN: 3-540-66130-1.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of IEEE Symposium on Security and Privacy*, number 96-17, pages 164–173, 1996.

[BH00] L. Buttyán and J. Hubaux. Enforcing service availability in mobile adhoc wans. In *ACM international symposium on Mobile ad hoc networking and computíng*, pages 87–96, Boston, Massachusetts, 2000. ACM Press.

[Bis02] Matt Bishop. *Computer Security*. Pearson Education, 2002.

[BLB03] S. Buchegger and J. Y. Le Boudec. The effect of rumor spreading in reputation systems for mobile ad-hoc networks. In *Proceedings of WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, Sophia-Antipolis, France, March 2003.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the international Congress of Mathematicians*, pages 1444–1451, Berkeley, CA, USA, 1986.

[BO99] B. Bellur and R. G. Ogier. A reliable, efficient topology broadcast protocol for dynamic networks. In *IEEE INFOCOMM*, pages 179–186, 1999.

[Buc04] Sonja Buchegger. *Coping with Misbehavior in Mobile Ad-hoc Networks*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2004.

[CBH02] Srdjan Capkun, Levente Buttyan, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. In *ACM International Workshop on Wireless Security*, 2002.

[CDDCdV$^+$02] F. Cornelli, E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Implementing a reputation-aware gnutella servent. In *Proceedings of the International Workshop on Peer-to-Peer Computing*, May 2002.

[Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.

[CHB03] Srdjan Capkun, Jean-Pierre Hubaux, and Levente Buttyan. Mobility helps security in ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 46–56, New York, NY, USA, 2003. ACM Press.

[CJ03] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). The Internet Society, October 2003. Network Working Group.

[CUBV05] Romit Roy Choudhury, Tetsuro Ueda, Jacir Bordim, and Nitin Vaidya. Beamnet: Ad hoc networking testbed using beamforming antennas. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 2005.

[dA07] VDA Verband der Automobilindustrie. Datenaustausch für mehr sicherheit. online, 2007. Jahresbericht 2007, Sicherheit und Technik.

[DDB04] Prashant Dewan, Partha Dasgupta, and Amiya Bhattacharya. On using reputations in ad hoc networks to counter malicious nodes. In *Proceedings of QoS and Dynamic Systems (in conjunction with IEEE ICPADS)*, 2004.

[DDCdVP$^+$02] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, November 2002.

[DFH01] R. Dingledine, M. J. Freedman, and D. Hopwood, D.and Molnar. A reputation system to increase mix-net reliability. In *Proceedings of 4th International Information Hiding Workshop*, April 2001.

[DFM05] Florian Doetzer, Lars Fischer, and Przemyslaw Magiera. 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, Italy, June 2005.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[DKKS05] Florian Doetzer, Florian Kohlmayer, Timo Kosch, and Markus Strassberger. Secure communication for intersection assistance. In *Proceedings of the 2nd International Workshop on Intelligent Transportation*, Hamburg, Germany, March 2005.

[DKS05] Florian Doetzer, Timo Kosch, and Markus Strassberger. Classification for traffic related inter-vehicle messaging. In *Proceedings of the 5th IEEE International Conference on ITS Telecommunications*, Brest, France, June 2005.

[DMS03] R. Dingledine, N. Mathewson, and P. Syverson. Reputation in p2p anonymity systems. In *Proceedings of 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[Doe02] Florian Doetzer. Aspects of multi-application smart card management systems. Master's thesis, Technical University Munich, 2002.

[Doe05] Florian Doetzer. Privacy issues in vehicular ad hoc networks. In *Workshop on Privacy Enhancing Technologies*, Cavtat, Croatia, May 2005.

[Dou02] J. Douceur. The sybil attack. In *Proceedings of the IPTPS02 Workshop*, March 2002.

[DPR00] Samir Ranjan Das, Charles E. Perkins, and Elizabeth E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of INFOCOMM*, pages 3–12, 2000.

[DVOW92] Whitfield Diffie, Paul C. Van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.

[Eck03] Claudia Eckert. *IT-Security: Konzepte, Verfahren, Protokolle*. R. Oldenbourg Verlag, 2003.

[EDS$^+$04] Stephan Eichler, Florian Doetzer, Christian Schwingenschlögl, Javier Fabra, and Jörg Eberspächer. Secure routing in a vehicular ad hoc network. In *Proceedings of IEEE VTC 2004 Fall*, Los Angeles, USA, September 2004.

[EEstd98] 1998 ESRI. ESRI shapefile technical description. Esri shapefile technical description, 1998.

[EGB02] Laurent Eschenauer, Virgil Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *Proceedings of the Security Protocols Workshop*, April 2002.

[FEL01] Walter Franz, Reinhold Eberhardt, and Thomas Luckenbach. Fleetnet - internet on the road. In *Proceedings of the 8th World Congress on Intelligent Transportation Systems*, October 2001.

[Fes05] Andreas Festag. Wifi für autos. *Funkschau*, 13:45–46, 2005. url last accessed: 9.10.2005.

[FFS87] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing*, pages 210–217, 1987.

[Fäh03] S. Fähnrich. Entwurf eines vollständig verteilten reputationssystems für ad-hoc netze, November 2003.

[FHW$^+$04] Holger Füßler, Hannes Hartenstein, Jörg Widmer, Martin Mauve, and Wolfgang Effelsberg. Contention-based forwarding for street scenarios. In *Proceedings of the 1st International Workshop in Intelligent Transportation (WIT 2004)*, pages 155–160, Mar 2004.

[FOKR04] Stefan Fähnrich, Philipp Obreiter, and Brigitta Konig-Ries. The buddy system : A distributed reputation system based on social structure. In *Technical Report 2004-1, University of Karlsruhe*, pages 293–297. University of Karlsruhe, March 2004.

[fPP03] Independent Centre for Privacy Protection. First partial success for an.on, 2003.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO 86 Proceedings*, pages 186–194. Springer Verlag, 1987.

[fS89] International Organization for Standardization. Information technology - osi - basic reference model - part 2: Security architecture. ISO IS 7498-2, 1989.

[fSuV98] Forschungsgesellschaft für Strassen-und Verkehrswesen. Rilsa - richtlinien für lichtsignalanlagen, 1998.

[fW05] Bundesministerium für Wirtschaft. Ordination concerning the technical and organisational implementation of measuresures for the interception of telecommunications - tkÜv, November 2005.

[FWK$^+$03] Holger Füßler, Jörg Widmer, Michael Käsemann, Martin Mauve, and Hannes Hartenstein. Contention-based forwarding for mobile ad-hoc networks. *Elsevier's Ad Hoc Networks*, 1(4):351–369, Nov 2003.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[GGS04] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In *VANET '04: Proceedings of the first ACM workshop on Vehicular ad hoc networks*, pages 29–37. ACM Press, 2004.

[GH01] Mario Gerla and Xiaoyan Hong. Fisheye state routing protocol for ad-hoc networks. IETF MANET Working Group, May 2001.

[GHMP03] Mario Gerla, Xiaoyan Hong, Li Ma, and Guangyu Pei. Landmark routing protocol for large scale ad-hoc networks. Internet, May 2003.

[GK00] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, march 2000.

[GMW86] O. Goldreich, S. Micali, and A. Widgerson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Proceedings of the 27th IEEE Symposium on the Foundation of Computer Science*, pages 174–187, 1986.

[Gör05] Harald Görl. *Systematische Analyse und Konstruktion integrierter Sicherheitsarchitekturen für mobile verteilte Systeme*. PhD thesis, Technische Universität München, Munich, Germany, June 2005.

[HDL$^+$02] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitatos, and S. Sajama. Wireless ad hoc networks, December 2002. Encyclopedia of Telecommunications.

[HPJ01] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad-hoc networks. Technical report, Department of Computer Science, Rice University, Tech. Rep. TR01-384, December 2001.

[HPJ02] Yih-Chun Hu, Adrian Perrig, and David Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. Technical report, Dept. of Computer Science, Rice University, 2002.

[HPJ03] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2003 ACM workshop on Wireless security*, pages 30–40. ACM Press, 2003.

[Här05] Christian Härdt. Implementation of algorithms for information aggregation in a vanet environment. Technical report, Technical University Munich, 2005.

[Hub04] Jean-Pierre Hubaux. New research challenges for the security of ad hoc and sensor networks. Keynote at 1st European Workshop on Security in Ad-Hoc and Sensor Networks, August 2004.

[ISO99] ISO. Is 15408, 1999.

[JGK03] Audun Jøsang, Elizabeth Gray, and Michael Kinateder. Analysing topologies of transitive trust. In Theo Dimitrakos and Fabio Martinelli, editors, *Proceedings of the First International Workshop on Formal Aspects in Security & Trust (FAST2003)*, pages 9–22, Pisa, Italy, September 2003.

[JHP02] D. Johnson, Y. C. Hu, and A. Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 02)*, pages 3–13, June 2002.

[JM96] David Johnson and David Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353, pages 153–181. Kluwer Academic Publishers, 1996.

[JW04] Roger G. Johnston and Jon S. Warner. Think gps offers high security? think again! In *Proceedings of the Business Contingency Planning Conference*, Las Vegas, May 2004.

[Kar03] Frank Kargl. *Sicherheit in Mobilen Ad hoc Netzwerken*. PhD thesis, Universität Ulm, 2003.

[Ker83] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5—-83, January 1883.

[KK99] Oliver Kömmerling and Markus Kuhn. Design principles for tamper-resistant smartcard processors. In *Proceedings of the USENIX Workshop on Smartcard Technology*, Chicago, USA, May 1999.

[KKP99] J. Kahn, R. Katz, and K. Pister. Next century challenges: Mobile networking for ”smart dust”. In *Proceedings of MobiCom 99*, pages 271–278, Seattle, WA, August 1999.

[KMSB01] Timo Kosch, Hans-Ulrich Michel, Karl-Ernst Steinberg, and Hermann Bonenberger. Multimediatechnologie im automobil oder das bewusste automobil in einer vernetzten welt. In *Tagungsband Informationstage ”Mobile Computing” 2001*, 2001.

[Kos04a] Timo Kosch. Efficient message dissemination in vehicle ad-hoc networks. In *Proceedings of the 11th World Congress on Intelligent Transportation Systems*, October 2004.

[Kos04b] Timo Kosch. Local danger warning based on vehicle ad-hoc networks: Prototype and simulation. In *Proceedings of 1st International Workshop on Intelligent Transportation (WIT 2004)*, March 2004.

[Kos05] Timo Kosch. *Situationsadaptive Kommunikation in Automotiven Ad-hoc Netzen.* PhD thesis, Technical University Munich, 2005.

[Kra98] S. Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics.* PhD thesis, Universität zu Köln, 1998.

[KRH05] K. Kwiat, K. Ravindran, and P. Hurley. Energy-efficient replica voting mechanisms for secure real-time embedded systems. In *Proceedings of the Sixth IEEE International Symposiumon a World of Wireless Mobile and Multimedia Networks*, Taormina, Italy, June 2005.

[KSA02] Timo Kosch, Christian Schwingenschlögl, and Li Ai. Information dissemination in multihop inter-vehicle networks - adapting the ad-hoc on-demand distance vector routing protocol (aodv). In *Proceedings of the 5th International Conference on Intelligent Transportation Systems*, Singapore, September 2002.

[KSGM03] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the twelfth international conference on World Wide Web*, pages 640–651. ACM Press, 2003.

[Lam81] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[LBC$^+$01] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 61–69, Rome, Italy, July 2001.

[LFC03] Kevin Lai, M. Feldman, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[LY03] Y. Liu and Y. R. Yang. Reputation propagation and agreement in mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, March 2003.

[Mal98] G. Malkin. Rip version 2. Technical report, IETF RFC 2453, 1998.

[MEL01] Andrew P. Moore, Robert J. Ellison, and Richard C. Linger. Attack modeling for information security and survivability. Technical Report CMU/SEI-2001-TN-001, Carnegie Mellon University, 2001.

[MG04] Seamus Moloney and Philip Ginzboorg. Security for interactions in pervasive networks. In *Proceedings of 1st European Workshop on Security in Ad-Hoc and Sensor Networks.* Springer, August 2004.

[MGLA99] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications*, pages 183—211, Oct. 1999. Special Issue on Routing in Mobile Communication Networks.

[MGLB00] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, 2000.

[MHM03] Lik Mui, Ari Halberstadt, and Mojdeh Mohtashemi. *Evaluating Reputation in Multiagents Systems*, volume 2631, pages 183–194. Springer, Berlin, 2003.

[Mic04] Pietro Michiardi. *Cooperation enforcement and network security mechanisms for mobile ad hoc networks*. PhD thesis, Eurecom, 2004.

[MM02] Pietro Michiardi and Refik Molva. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proceedings of Communication and Multimedia Security 2002 Conference*, September 2002.

[MMP$^+$05] Kirsten Matheus, Rolf Morich, Ingrid Paulus, Cornelius Menig, Andreas Lübke, Bernd Rech, and Will Specks. Car-to-car communication - market introduction and success factors. In *Proceedings of the 5th European Congress and Exhibition on Intelligent Transport Systems and Services*, Hanover, June 2005.

[MMV$^+$03] J.D. Meier, Alex Mackman, Srinath Vasireddy, Michael Dunner, Ray Escamilla, and Anandha Murukan. Improving web application security, threats and countermeasures, June 2003. last accessed: 09.05.2006.

[Mok05] Abdel Kader Mokaddem. Its car to car communications at 5,9 ghz. Presentation at the Workshop on Spectrum Requirement For Road Safety, February 2005.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38:114–117, April 1965.

[MOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, USA, 1996.

[Moy98] J. Moy. Ospf version 2. Technical report, IETF RFC 2328, 1998.

[Mui03] L. Mui. *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*. PhD thesis, Massachusetts Institute of Technology, 2003.

[NSSP04] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The sybil attack in sensor networks: analysis & defenses. In *IPSN'04: Proceedings of the third international symposium on Information processing in sensor networks*, pages 259–268. ACM Press, 2004.

[ODS07] Benedikt Ostermaier, Florian Dötzer, and Markus Strassberger. Enhancing the security of local danger warnings in vanets - a simulative analysis of voting schemes. In *Proceedings of The Second International Conference on Availability, Reliability and Security (ARES 07)*, pages 432–442, Vienna, Austria, 2007.

[OKRK03] P. Obreiter, B. Koenig-Ries, and M. Klein. Stimulating cooperative behavior of autonomous devices - an analysis of requirements and existing approaches. In *Second International Workshop on Wireless Information Systems (WIS2003)*, Angers, France, 2003.

[oSN98] National Institute of Standards and Technologies (NIST). Common criteria: Launching the international standard. *ITL Bulletin*, 00:1–7, 1998.

[Ost05] Benedikt Ostermaier. Analysis and improvement of inter-vehicle communication security by simulation of attacks. Master's thesis, Technical University Munich, 2005.

[oT] U.S. Department of Transportation. Vehicle infrastructure integration (vii). http://www.its.dot.gov/vii.

[oWP04] The Network on Wheels Project. Now website, http://www.network-on-wheels.de, 2004.

[PB94] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIG-COMM'94 Conference on Communications Architectures, Protocols and Applications, pp. 234-244*, 1994.

[PBR99] Charles E. Perkins and Elizabeth M. Belding-Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[PC97] Vincent D. Park and M. Scott Corson. A high adaptive distributed routing algorithm for mobile wireless networks. In *IEEE Conference on Computer Communications, INFOCOM'97, April 7-11, 1997, Kobe, Japan*, volume 3, pages 1405–1413. IEEE, April 1997.

[PCTS02] Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Song. The tesla broadcast authentication protocol. *RSA Cryptobytes*, 5(3):2–13, 2002.

[Per88] R. Perlman. *Network Layer Protocols with Byzantine Robustness.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1988.

[Per03] Gordon Peredo. Brake-ing news – technologies for inter-vehicle communication, May 2003.

[Pfe03] Johannes Pfeifroth. Entwicklung und implementierung eines fahrzeug-mobilitätsmodells zur ad hoc netzwerksimulation. Master's thesis, TU Munich, 2003.

[PH02] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 27–31, January 2002.

[PK01] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In Hannes Federrath, editor, *Proceedings of Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9. Springer, 2001.

[Pla01] Global Platform. Global platform card specification (version 2.1), 2001.

[PP05] Bryan Parno and Adrian Perrig. Challenges in securing vehicular networks. In *Proceedings of Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[Pro00] John Proakis. *Digital Communications.* Mc Graw Hill, 2000.

[Pro01a] The Cartalk 2000 Project. Cartalk project website, http://www.cartalk2000.net/, 2001.

[Pro01b] The Invent Project. Invent project website, http://www.invent-online.de/, 2001.

[Pro03a] The PReVENT Project. Prevent project website, http://www.prevent-ip.org, 2003.

[Pro03b] The Willwarn Project. The willwarn project website, http://www.prevent-ip.org/willwarn, 2003.

[Pro04] The FleetNet Project. The fleetnet website, http://www.et2.tu-harburg.de/fleetnet/english/vision.html, 2004.

[Pro05a] The PATH Project. Path project website, http://www-path.eecs.berkeley.edu/, 2005.

[Pro05b] The Vehicle Safety Communication Project. Vsc website, http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/camp3/pages/vscc.htm, 2005.

[Rap99] Theodore Rappaport. *Wireless Communications.* Prentice Hall Inc., 1999.

[RBP$^+$04] M. Renaudin, F. Bouesse, Ph. Proust, J. P. Tual, L. Sourgen, and F. Germain. High security smartcards. In *Design, Automation and Test in Europe Conference and Exhibition Volume I (DATE'04)*, volume 1, page 10228, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[RT99] Elizabeth Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(1):46–55, April 1999.

[SA99] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *7th International Workshop Proceedings, Lecture Notes in Computer Science*, pages 172–194, 1999.

[Sch96] Bruce Schneier. *Applied Cryptography*. John Wiley and Sons, New York City, 1996.

[Sch99] Bruce Schneier. Attack trees: Modeling security threats. Dr. Dobb's Journal, 1999.

[Sch00] Bruce Schneier. *Secrets & Lies*. John Wiley and Sons, 2000.

[SD02] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In P. Syverson and R. Dingledine, editors, *Proceedings of Privacy Enhancing Technologies*, San Francisco, CA, 2002. LNCS.

[SDK$^+$05] Christoph Schroth, Florian Doetzer, Timo Kosch, Benedikt Ostermaier, and Markus Strassberger. Simulating the traffic effects of vehicle-to-vehicle messaging systems. In *Proceedings of the 5th IEEE International Conference on ITS Telecommunications*, Brest, France, June 2005.

[SDL$^+$01] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth M. Belding-Royer. A secure routing protocol for ad hoc networks. Technical report, Electrical Engineering and Computer Science, University of Michigan, Tech. Rep. UM-CS-2001-037, August 2001.

[SE04] Christian Schwingenschlögl and Stephan Eichler. Certificate-based key management for secure communications in ad hoc networks. In Olga Casals, Jorge Garcia-Vidal, José M. Barceló, and Llorenc Cerdià, editors, *Proceedings of 5th European Wireless Conference: Mobile and Wireless Systems beyond 3G*, pages 498–504, February 2004.

[SH02] Christian Schwingenschlögl and Marc-Philipp Horn. Building blocks for secure communication in ad-hoc networks. In *Proceedings European Wireless*, 2002.

[Shi00] R. Shirey. Rfc 2828 - internet security glossary. http://www.ietf.org/rfc/rfc2828.txt, 2000.

[SPvDK04] Arvind Seshadri, Adrian Perrig, Leendert van Doorn, and Pradeep Khosla. Swatt: Software-based attestation for embedded devices. In *IEEE Symposium on Security and Privacy*, 2004.

[SS03] Harkirat Singh and Suresh Singh. A mac protocol based on adaptive beamforming for ad hoc networks. In *Proceedings of IEEE Pimrc'03*, September 2003.

[Sta96] Multiple States. Wassenaar arrangement on export controls for conventional arms and dual-use goods and technologies, May 1996.

[tCCC04] The Car to Car Communication Consortium. C2c-cc website, 2004.

[Tel94] G. Tel. *Introduction to Distributed Algorithms*. Cambridge University Press, 1994. Includes good definition for distributed system.

[Tie00] Veith Tiemann. Asymmetrische / moderne kryptographie - ein interaktiver Überblick, 2000. http:
www.wiwi.uni-bielefeld.de/StatCompSci/lehre.

[Uni81] European Union. Directive 95/46/ec on the protection of personal data, 1981.

[Upt03] Jodi Upton. Gadget may wreak traffic havoc. *The Detroit News*, 00, October 2003.

[Van06] Borislav Vangelov. Analysis and implementation of forwarding algorithms in vehicle ad hoc networks. Technical report, Technical University Munich, 2006.

[Wey07] Benjamin Weyl. *On Interdomain Security: Trust Establishment in Loosely Coupled Federated Environments*. PhD thesis, TU Darmstadt, 2007.

[Wim04] Richard Wimmer. Ein verteiltes nicht-interaktives authentisierungskonzept für mobile ad-hoc netze. Master's thesis, Munich University of Technology, April 2004.

[WK03] David Wagner and Chris Karlof. Secure routing in wireless sensor networks: Attacks and countermeasures. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, May 2003.

[WPW05] André Weimerskirch, Christof Paar, and Marko Wolf. Cryptographic component identification: Enabler for secure vehicles. In *Proceedings of Vehicular Technology Conference 2005*, Dallas, USA, September 2005.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[Zap02] Manuel Guerrero Zapata. Secure ad hoc on-demand distance vector routing. *SIGMOBILE Mobile Computer Communication Review*, 6(3):106–107, 2002.

[ZCY02] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A simple cheat-proof credit-based system for mobile ad-hoc networks. Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, July 2002.

[ZCY03] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of Infocomm: 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2003.

[ZH99] Lidong Zhou and Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13(6):24–30, 1999.

[ZH01] Johannes Zangl and Joachim Hagenauer. Large ad hoc sensor networks with position estimation. In *Proceedings of the 10th Aachen Symposium on Signal Theory*, pages 115–118, Aachen, Germany, September 2001.

[ZWH04] Daqing Zhang, Xiaohang Wang, and Kai Hackbarth. Osgi based service infrastructure for context aware automotive telematics. In *IEEE Vehicular Technology Conference (VTC Spring 2004), Milan, Italy*, May 2004.

Note that all definitions in the glossary without references have been created as part of this thesis.

Glossary

A

ad-hoc network An ad-hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. – [**RT99**].

adaption phase Adaption phase is the time between a traffic-related situation has disappeared and the majority of nodes has recognized this.

API Application Progammer's Interface.

area event An area event is, in contrast to a point event, related to situations that exist over an extended geographical area and cannot be easily described as a combination of coordinates and radius.

availability (readiness for correct service) The property of being accessible and usable upon demand by an authorized entity. – RFC 2828.

B

broadcast Broadcast is the distribution of data to a number of recipients. Since the wireless channel is a shared medium, broadcast may be achieved on MAC layer where all nodes listening on a specific channel will receive the data. However, a logical broadcast may also be achieved on higher layers in the OSI-model, where nodes distribute information, while using unicast mechanisms at the MAC layer.

broadcast storm A broadcast storm is a networking situation in which messages are broadcast on a network, and each message prompts a receiving node to respond by broadcasting its own messages on the network that in turn prompt further responses, and so on. – Wikipedia.

C

C2C Simulation (C2C Application Simulation) A C2C simulation or better C2C application simulation is a software to provide: first an emulation of sensor data that is generated according to a situation script and second integrate car-to-car (C2C) applications into an overall simulation environment so that realistic messages are generated, forwarded and evaluated by a number of nodes.

communication environment Communication environment is a term for the combination of global (clustering coefficient, number of neighbor distribution, etc.) or local (average hop count, link duration, wireless channel saturation, etc.) communication parameters.

confidence Confidence is a value assigned to traffic-related information that relates to the the degree of calculated information authenticity. The confidence-level is used in the presentation decision and forwarding decision processes.

D

decentralized A decentralized system is a distributed system which functions without an organized center or authority. All nodes have the same capabilities, while they can perform different roles at a specific point of time, such as cluster-head and cluster-node.

decision area The decision area is the geographic region within which a final decision has to be made about whether the information received about an event is trustworthy enough to be presented to the driver.

delayed decision The process of collecting messages about an event and only making a decision upon entering the the decision area is called delayed decision.

distributed A distributed system consists of physically distributed computing devices, each one performing a specific specialized task, for coordinated use.

distribution area The distribution area is the geographic region within which the respective traffic-related messages will be distributed.

E

ECU An Electronic Control Unit (ECU) is an embedded system that controls one or more electrical subsystems in a vehicle.

evaluator Evaluator is a relay node generating opinions, as part of a reputation system.

event An event is generated after detection of one of a set of predefined situations and represents information about that specific situation at a distinct point in time.

event class Event classes are a categorization of traffic-related situations according to characteristics that are relevant for distribution and confidence calculation.

event detection message If unusual patterns are detected and the sensor reasoning module generates an event, an event detection message will be sent to the management module within SARI.

exposed node An exposed node is prevented from sending packets to other nodes due to a neighboring transmitter.

F

forwarding Forwarding is the act of distributing network packets according to network metrics. In this work the term forwarding is specifically denoted to describe the process of deciding about message distribution using application layer information. In contrast, if the message distribution is executed without application layer knowledge it is called routing.

forwarding decision The forwarding decision relates to traffic-related applications and denotes the decision about how information about a traffic event is forwarded to other nodes.

G

GenMobTrace GenMobTrace is a basic traffic simulator and part of the CARISMA simulation, which has been developed by Pfeifroth within the context of his diploma thesis [**Pfe03**].

geo situation VARS, the Vehicle Adhoc network Reputation System defines four different geographic- and situation-oriented reputation levels - local, city, highways, unknown territory - that reflect the situation a receiving node is in with respect to the availability of its reputation network.

geocast Geocast is a form of multicast, where the multicast group is not specified through address-ranges or node IDs but through geographical positions of nodes at a given point in time.

GloMoSim GloMoSim is a specialized simulation software for wireless ad-hoc networks.

H

hidden node A hidden node is visible from a node A, but not from other nodes communicating with node A. This leads to collisions on the wireless channel.

hotspot A fixed node that provides gateway functionality and may also support VANET technologies.

I

imprinting The imprinting process is performed during the car's production to create a fingerprint of a car's system configuration.

intermediate node Within the context of this thesis intermediate node denotes a node that is part of a route, but neither source nor destination node in type-3 scenarios.

IVC (Inter-Vehicle Communication) Inter-Vehicle Communication (IVC) denotes the combination of all wireless communication where road vehicles are involved, either as source, destination or forwarding node. This includes VANET communication, connections to cellular networks or hotspots, as well as the reception of broadcast media.

K

Kerckhoffs' law "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge." – [**Ker83**].

L

late decision Late decision is the concept that presentation decisions are only made when entering (or already being inside) decision areas. The time between the first message about an event and the time a presentation decision has to be made, is be used to gather additional information.

local world-model How a node perceives the world, consisting of two parts:

1. a static model that has been burnt into the system on production (reference)
2. a dynamic model that takes actual messages from others, sensor readings and other information into account (actual value).

logical broadcast Logical broadcast is a term to denote a broadcast-like distribution on higher - mostly application- layers, while using unicast mechanisms at the MAC layer.

M

margin point Margin points are geographic locations that define the size of an recognition area with rectangular shape in the *rectangle aggregation* algorithm.

mesh network A mesh network is a special type of ad-hoc network, where nodes are immobile.

message A message is an application layer data container construct.

Mobile Ad-hoc NETwork 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. – IETF.

mobile device A mobile device is a machine that is either portable or self-propelled and that can be operated while in motion. Such a machine necessarily includes a computation device and is able to establish wireless communications. A machine that is neither portable nor self-propelled is called immobile device.

mobility simulation A collection of tools that is used to generate or simulate movement behavior of mobile nodes. Within the context of this work, this relates mainly to traffic simulation and different models of node behavior.

multicast Multicast is the distribution of data to a number of recipients that are members of a multicast group. A multicast group usually refers to address-ranges or groups of node IDs.

N

network simulation A network simulation is a software that is used to analyse networking protocols and physical communication channel environments. In this work a network simulation is one part of an simulation environment consisting of network simulation, mobility simulation and C2C application simulation.

NS-2 NS-2 is a well-known network simulator for fixed and wireless networks.

O

OBU (On-Board Unit) An On-Board Unit (OBU) is the computation system used for IVC applications of a mobile node such as a car.

opinion An opinion is a piece of information generated by an observer that either increases or decreases the reputation of a specific object resulting from direct experiences or observations that the observer made about the target.

P

PKI (Public Key Infrastructure) A PKI is an organizational arrangement that binds public keys with respective user identities by means of digital certificates.

point event Point events are traffic-related situations that have been detected (thus "event") and have limited spatial dimensions. Their dimensions can be described simply as a combination of coordinates and radius.

presentation decision The presentation decision relates to traffic-related applications and denotes the decision about whether information about a traffic event is presented to the human driver or not.

pseudonym credential Pseudonym credential is a name for digital credentials used in the privacy architecture presented in this thesis that are issued by authority A.

R

race condition A race condition is a flaw in a system or process, where the output of the process depends on the sequence or timing of events within the process.

recognition area The recongition area is the geographic region within which the respective situation can be recognized by nodes.

registered information Registered information is long-term data that is available from central databases, such as road-type, urban boundary, curve radius, slope, terrain formation, etc.

relay node A relay node is a network node that distributes a received message according to network-wide principles, independently whether it is using the message itself or not. This applies to both, forwarding and routing.

reliability (continuity of correct service) The ability of a system to perform a required function under stated conditions for a specified period of time. – RFC 2828.

reputation Reputation is a term that denotes expectations on the behavior of an entity in the future based on its behavior in the past.

robustness dependability with respect to external faults – [**ALRL04**].

routing Routing is the act of distributing network packets according to network metrics. In general this consists of exchanging routing relevant information, such as hello messages, establishment and maintenance of network routes. Network routes are usually used for a group of messages that have the same destination or are otherwise correlated. In this work the term routing is specifically used to describe the process of distributing messages without using application layer information. In contrast, if the message distribution process is called forwarding if it uses application specific information.

RSU (Road Side Unit) A Road Side Unit (RSU) is the computation system used for IVC applications of a geographically fixed node such as a traffic light.

S

security objective A statement of intent to counter specified threats and/or satisfy specified organizational security policies or assumptions. – [**oSN98**].

security service A processing or communication service that is provided by a system to give a specific kind of protection to system resources. Examples are: availability service, data confidentiality service, data integrity service, data origin authentication service, non-repudiation service, peer entity authentication service. – RFC 2828.

self-organization Self-organization refers to a process in which the internal organization of a system, normally an open system, increases automatically without being guided or managed by an outside source. – Wikipedia In the case of ad-hoc networks self-organization means that all functions that are necessary to establish and maintain data communication is realized without requiring human intervention during normal operation.

sender-based reputation level The sender-based reputation level is a categorization information sources in VARS. It distinguishes between sources that are known from direct or indirect reputation or have not been previously known (only transitive trust).

single source event An event, that is only directly detectable by a single sensor of one specific node.

situation Situations describe the environment over a period of time. They usually reflect the occurrence of special circumstances. A real-life example would be a lateral wind area or a construction site. There are also short-lived situations such as a car doing an emergency braking.

source node A source node is a network node where a message or a stream of messages originates. In case of traffic-related vehicle to vehicle messages, a source node is also called *detector*.

store-and-forward Store-and-forward denotes the concept of physically transporting digital information through mobile nodes.

strong identity An entitiy is said to have a strong identity if it cannot easily choose between multiple identifiers or generate new identifiers itself.

subscription credential Subscription credential is a name for digital credentials used in the privacy architecture presented in this thesis that are issued by organization O. The name refers to the practical viewpoint that organization O issues credentials to authorize service usage (which a user must subscribe for).

SUMO SUMO is an open-source mobility simulator, which has been developed by DLR (Deutsches Zentrum für Luft- und Raumfahrt).

switching time The time a traffic light in a given lane / direction takes to switch from red to green.

T

traffic environment Traffic environment is a term for the combination of global (relative speed diversity, standard deviation of speed, etc.) or local (average speed, top speed, number of turns per time, etc.) traffic parameters.

traffic scenario In this thesis traffic scenarios are a subset of scenarios that categorize combinations of traffic environment parameters and communication environment parameters with respect to their actual values. They are used to distinguish between different situations that a car may be in and that affect the behavior of SARI significantly.

TRM (tamper resistant module) A tamper resistant module is an embedded computing device that has been protected against physical tampering, such as implemented in smart cards. For this thesis, this includes also mechanisms implemented in the operating system to control input and output of the device and allow credential based installation of software packages.

trust Generally, an entity can be said to 'trust' a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects, that is it does what it claims to do and does not perform unwanted functions. - RFC 2828.

U

unicast Unicast is the act of transmitting data to a single node with a distinct destination address.

V

Vehicle Ad-hoc NETwork (VANET) A Vehicle Ad-hoc NETwork (VANET) is a special kind of MANET, where nodes are road vehicles including passenger cars, trucks, buses, motorcycles, etc.

vehicle originating events Vehicle originating events are a special case of single node events that cannot be directly verified by nodes other than the first detecting node. An example are exploding airbags, that may be detected by the respective car and trigger an event, but cannot be directly verified by others.

VISSIM VISSIM is a professional traffic simulator which features a sophisticated traffic model and also includes a visualization tool.

APPENDIX A

Pseudonymous Credentials: Key Encryption Protocol

Key encryption is used to transfer subscription credentials over unsecure communication channels. The key encryption and decryption algorithm is straight-forward using the approach introduced in section 4.3.3. However, the private key parameters s_i have to be chosen carefully, see following paragraph.

Table A.1 gives an overview on used variables and parameters.

s_i	Private key parameters
s_i^*	Encrypted private key parameters
$\bar{s}$	Private key
$s_{SQR,i}$	Square root of private key parameter
$s_{sCred,i}$	Private key parameter of subscription credential
$v_{sCred,i}$	Public key parameter of subscription credential
$s_{pCred,i}$	Private key parameter of pseudonym credential
$v_{pCred,i}$	Public key parameter of pseudonym credential
m_{sCred}	Encrypted subscription credential
N	Module
p	Prime factor (of N)
q	Prime factor (of N)
$H(x)$	Hash function of x

Table A.1. Variables and Parameters for Key En-/Decryption Algorithm

A.1. Encryption of Credentials

First of all, private key parameters s_i generated at organization O have to be chosen carefully. The general condition is that they must fulfill following equation:

$$s_i \in \mathbb{QR} \pmod{N} \qquad for \quad i = 0, 1, \ldots, k-1 \tag{A.1}$$

and then carefully chosen:

$$s_{SQR,i} = min(\sqrt{s_{sCred}} \pmod{N}) \qquad \forall i = 0, 1, \ldots, k-1 \tag{A.2}$$

The private key part of a credential is composed of its private key parameters: $\bar{s} = [s_0, s_1, \ldots, s_{k-1}]$. For encryption, a matrix consisting of hashed key parameter values (where $h = H(s_{sCred,0}, s_{sCred,1}, \ldots, s_{sCred,i-1})$) is generated:

$$\bar{\bar{e}} = \left\{ \begin{array}{cccc} e_{00} & e_{01} & \cdots & e_{0(k-1)} \\ e_{10} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ e_{(k-1)0} & \cdots & \cdots & e_{(k-1)(k-1)} \end{array} \right\} = \left\{ \begin{array}{cccc} h_0 & h_1 & \cdots & h_{k-1} \\ h_k & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{(k-1)*k} & \cdots & \cdots & b_{k*(k-1)} \end{array} \right\} \tag{A.3}$$

Together with:

$$s_i^* = s_{SQR,i} \cdot \prod_{m=0}^{k-1} (v_{pCred})_m^{e_{i,m}} \pmod{N}) \qquad \forall i = 0, 1, \ldots, k-1 \tag{A.4}$$

this results in an encrypted credential:

$$m_{sCred} = [v_{sCred}, \bar{\bar{e}}, s_0^*, s_1^*, \ldots, s_{k-1}^*] \tag{A.5}$$

A.2. Decryption of Credentials

A receiver decrypts a credential using following formulas:

$$\begin{aligned} s_{SQR,i} &= s_i^* \cdot \prod_{m=0}^{k-1} (s_{pCred})_m^{2e_{i,m}} \pmod{N}) \qquad \forall i = 0, 1, \ldots, k-1 \\ s_{sCred,i} &= s_{SQR,i}^2 \pmod{N} \qquad \forall i = 0, 1, \ldots, k-1 \end{aligned} \tag{A.6}$$

This is true because of:

$$\begin{aligned} & s_i^* \cdot \prod_{m=0}^{k-1} (s_{pCred})_m^{2e_{i,m}} \\ &= s_{SQR,i} \cdot \prod_{m=0}^{k-1} (v_{pCred})_m^{e_{i,m}} \cdot \prod_{m=0}^{k-1} (s_{pCred})_m^{2e_{i,m}} \\ &= s_{SQR,i} \cdot \prod_{m=0}^{k-1} ((v_{pCred})_m \cdot (s_{pCred})_m^2)^{e_{i,m}} \\ &= s_{SQR,i} \pmod{N} \qquad \forall i = 0, 1, \ldots, k-1 \end{aligned} \tag{A.7}$$

The receiver verifies public key and private key using:

$$v_{sCred,i} \cdot s_{sCred,i}^2 \overset{!}{=} 1 \pmod{N} \qquad \forall i = 0, 1, \ldots, k-1 \tag{A.8}$$

APPENDIX B

Statecharts

B.1. Statechart JavaCard Implementation

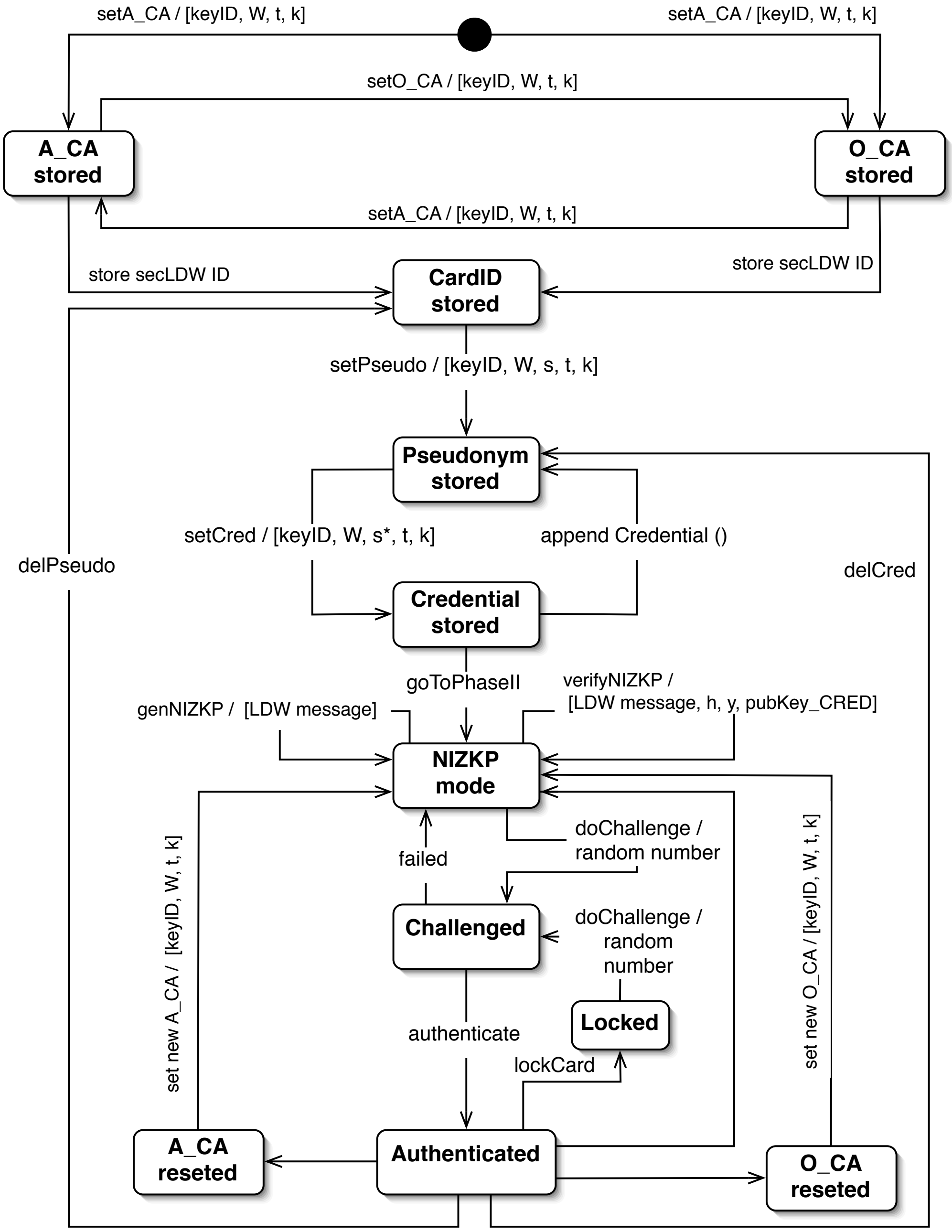

setA_CA / [keyID, W, t, k]
setA_CA / [keyID, W, t, k]
setO_CA / [keyID, W, t, k]
A_CA stored
O_CA stored
setA_CA / [keyID, W, t, k]
store secLDW ID
store secLDW ID
CardID stored
setPseudo / [keyID, W, s, t, k]
Pseudonym stored
setCred / [keyID, W, s*, t, k]
append Credential ()
delPseudo
delCred
Credential stored
goToPhaseII
genNIZKP / [LDW message]
verifyNIZKP / [LDW message, h, y, pubKey_CRED]
NIZKP mode
doChallenge / random number
failed
set new A_CA / [keyID, W, t, k]
set new O_CA / [keyID, W, t, k]
Challenged
doChallenge / random number
Locked
authenticate
lockCard
A_CA reseted
Authenticated
O_CA reseted

Figure B.1. Statechart for JavaCard Implementation

B.2. (UML) Action Diagrams Distributed Data Aggregation

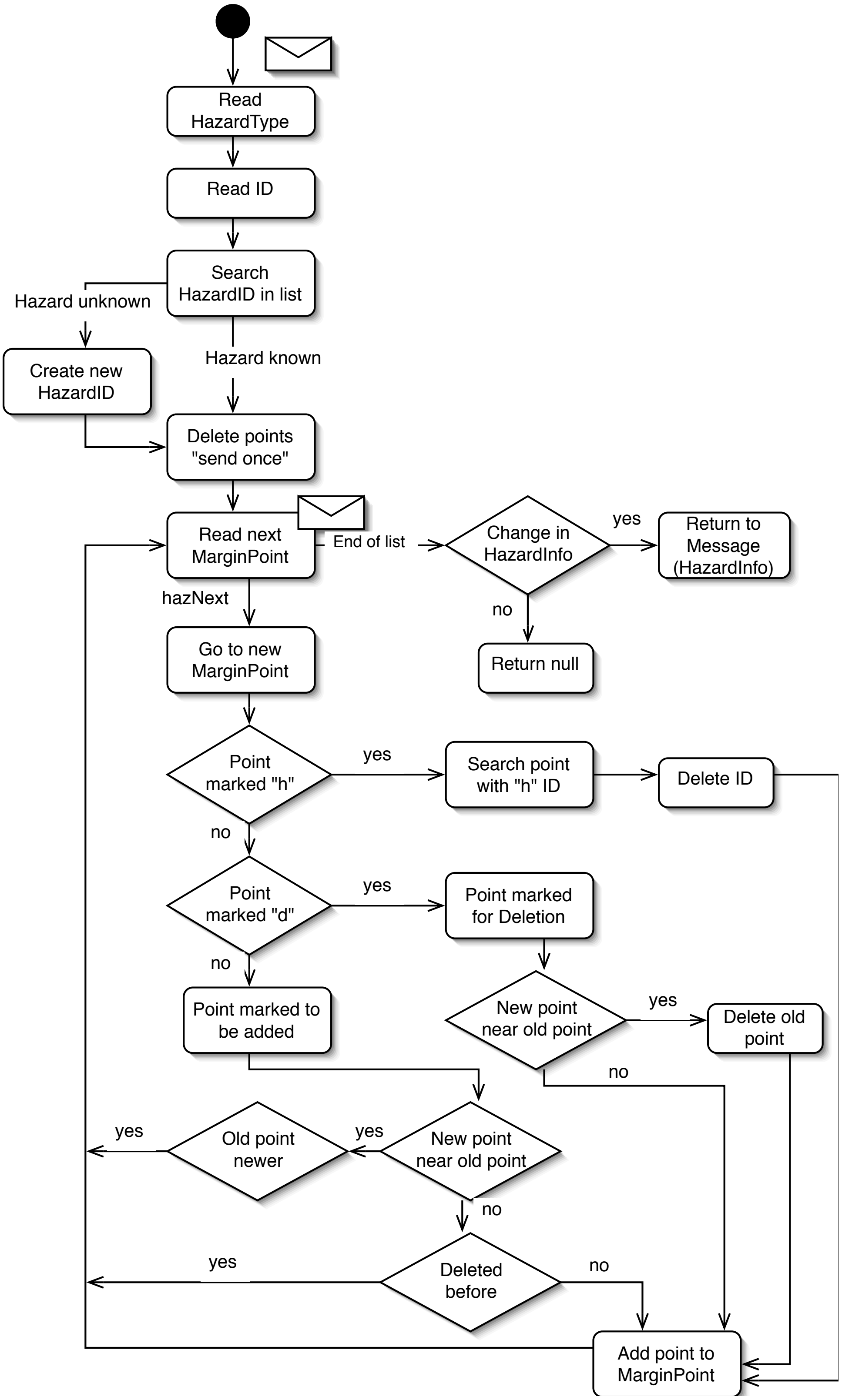


Figure B.2. Distributed Data Aggregation: addMessage(M)

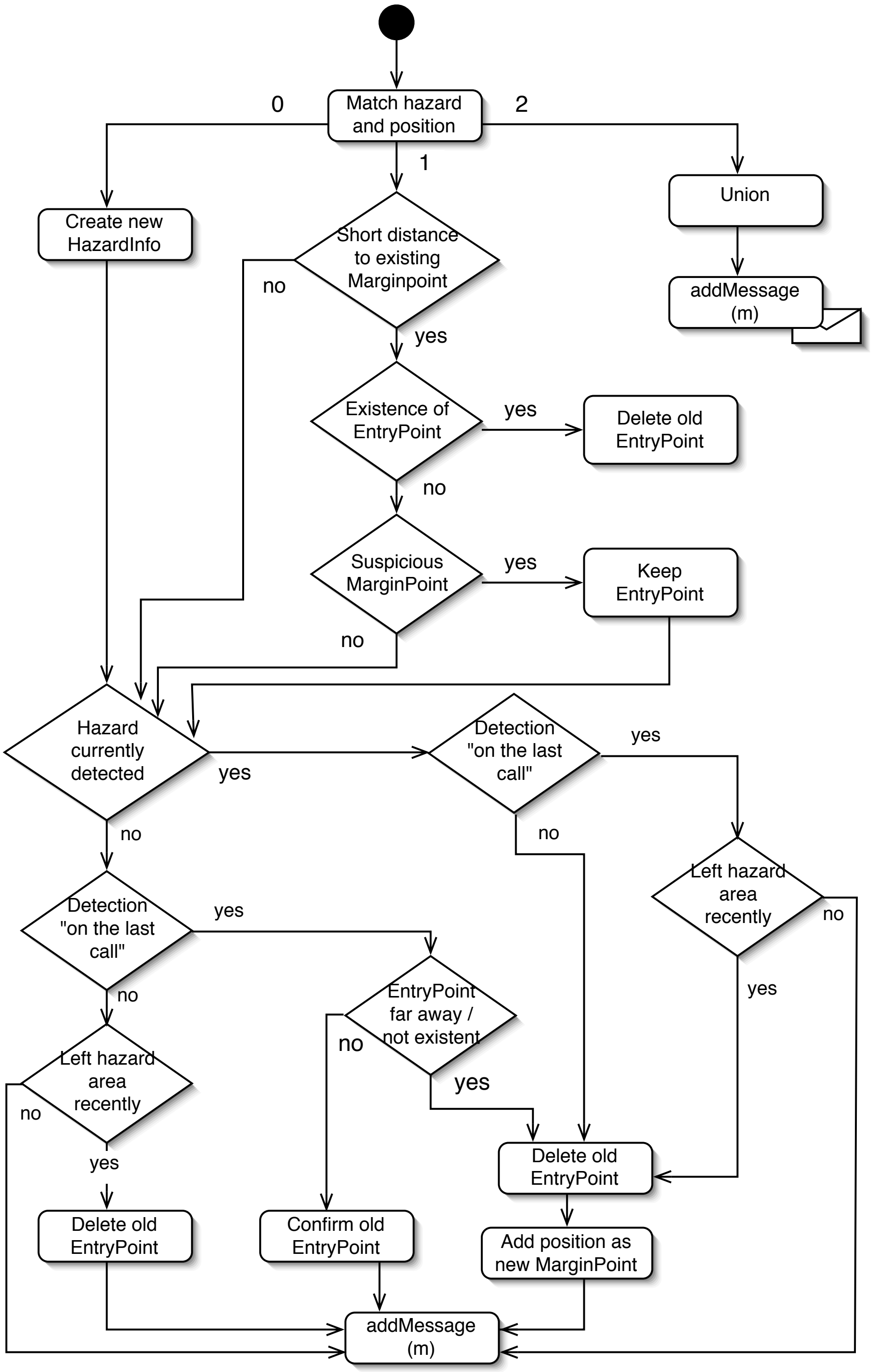

Match hazard and position
0
1
2
Create new HazardInfo
Short distance to existing Marginpoint
no
yes
Union
addMessage (m)
Existence of EntryPoint
yes
Delete old EntryPoint
no
Suspicious MarginPoint
yes
Keep EntryPoint
no
Hazard currently detected
yes
Detection "on the last call"
yes
no
no
Left hazard area recently
no
yes
Detection "on the last call"
yes
no
EntryPoint far away / not existent
no
yes
Left hazard area recently
no
yes
Delete old EntryPoint
Delete old EntryPoint
Confirm old EntryPoint
Add position as new MarginPoint
addMessage (m)

Figure B.3. Distributed Data Aggregation: addDetection(M)

APPENDIX C

Simulation Charts

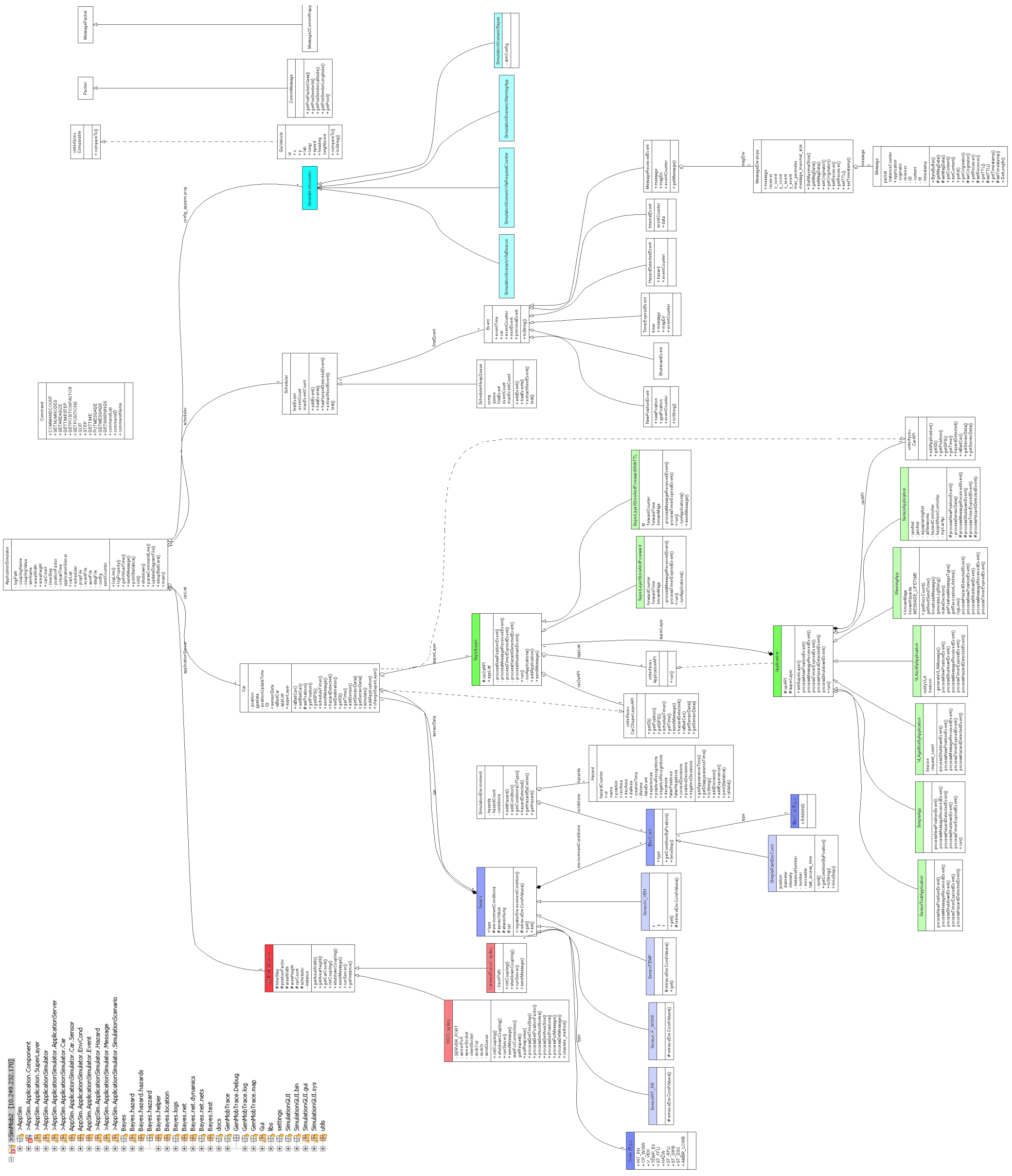
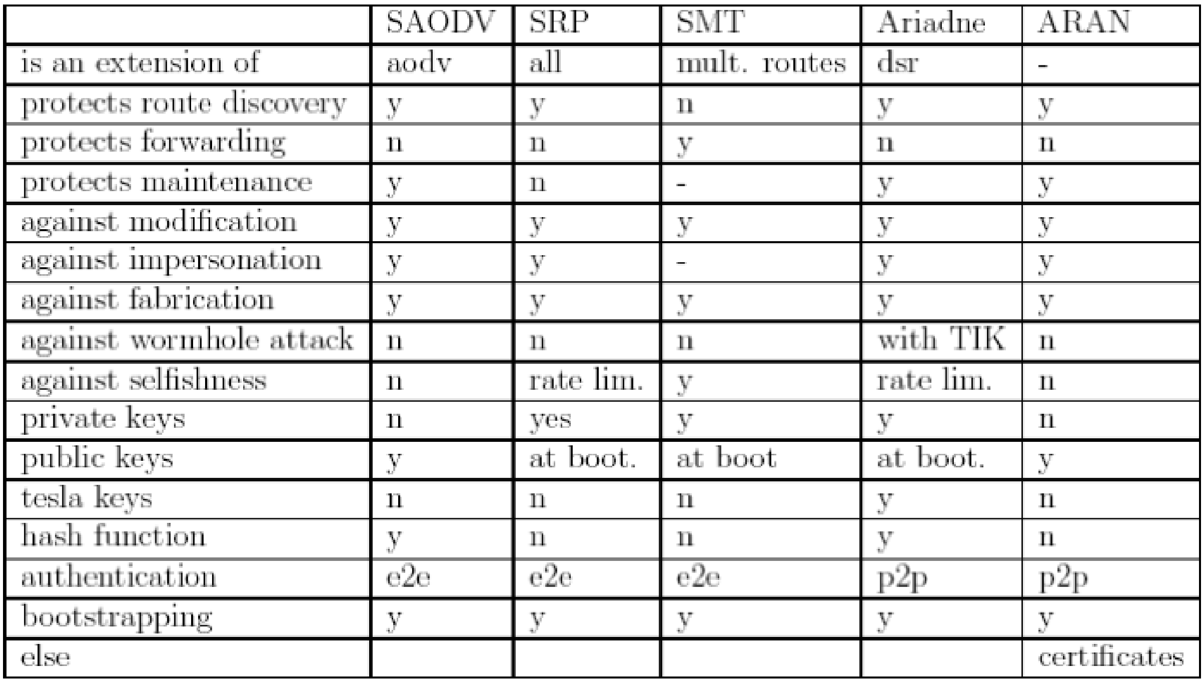


Figure C.1. Overview Application Simulation Implementation



	SAODV	SRP	SMT	Ariadne	ARAN
is an extension of	aodv	all	mult. routes	dsr	-
protects route discovery	y	y	n	y	y
protects forwarding	n	n	y	n	n
protects maintenance	y	n	-	y	y
against modification	y	y	y	y	y
against impersonation	y	y	-	y	y
against fabrication	y	y	y	y	y
against wormhole attack	n	n	n	with TIK	n
against selfishness	n	rate lim.	y	rate lim.	n
private keys	n	yes	y	y	n
public keys	y	at boot.	at boot	at boot.	y
tesla keys	n	n	n	y	n
hash function	y	n	n	y	n
authentication	e2e	e2e	e2e	p2p	p2p
bootstrapping	y	y	y	y	y
else					certificates

Figure C.2. Overview Secure Routing Protocols

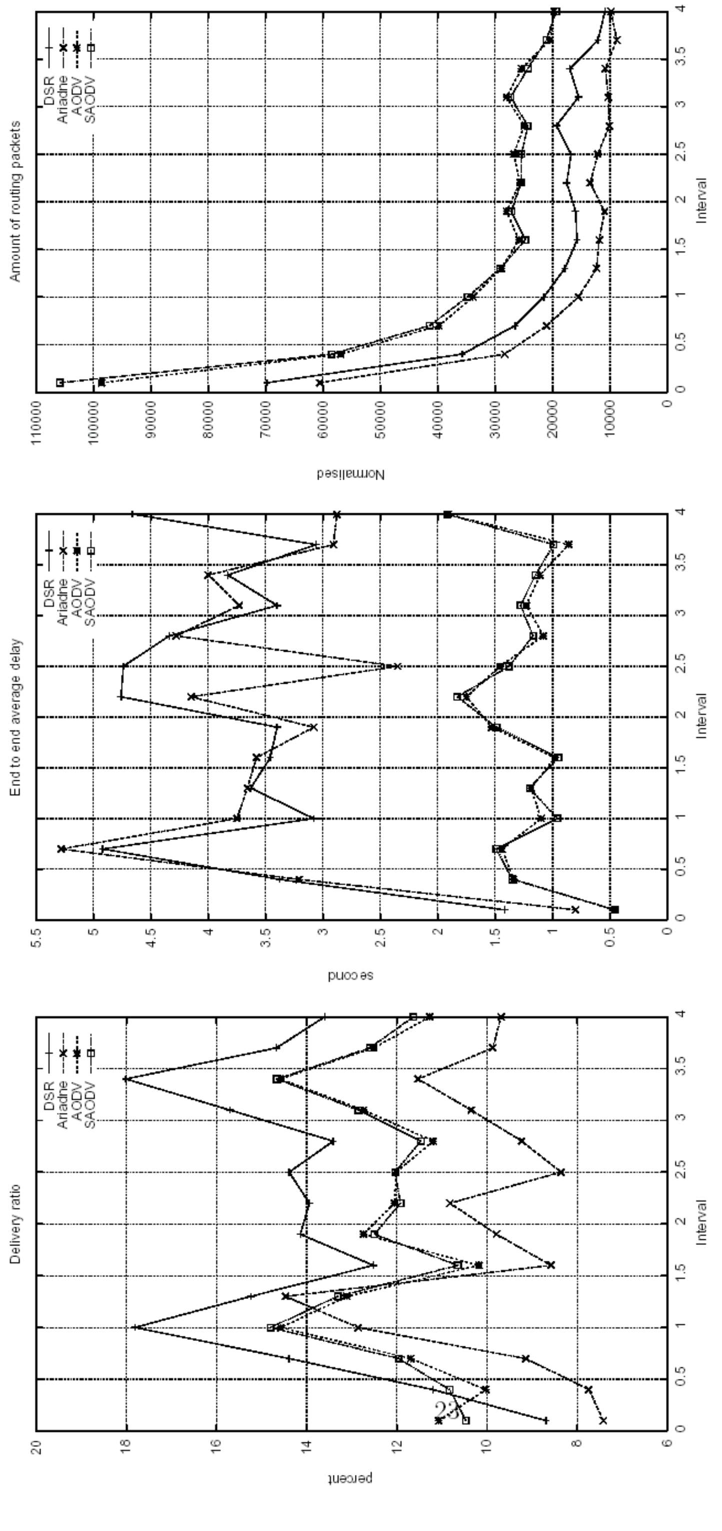


Figure C.3. 50 nodes, 200s, 30% CBR, BMW GenMobTrace

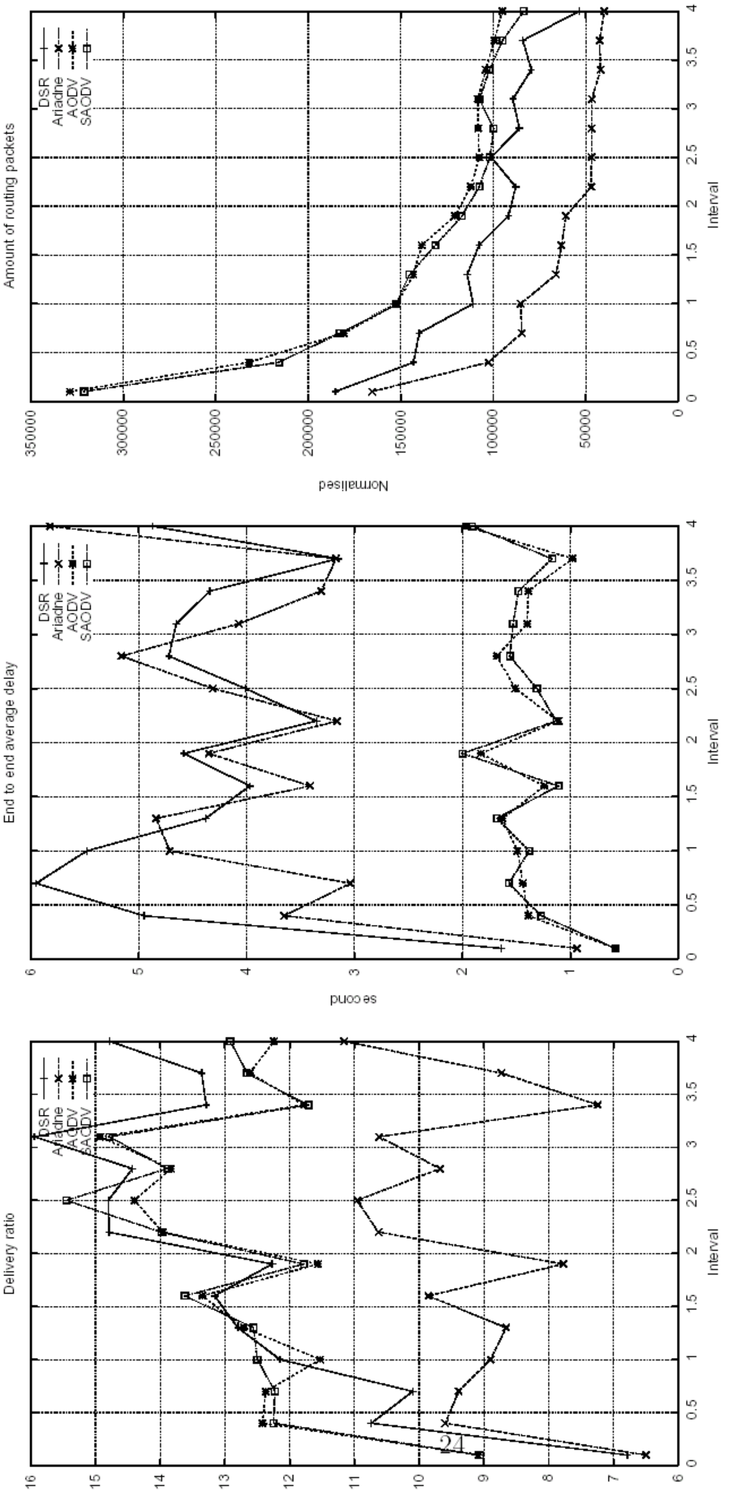


Figure C.4. 100 nodes, 200s, 30% CBR, BMW GenMobTrace

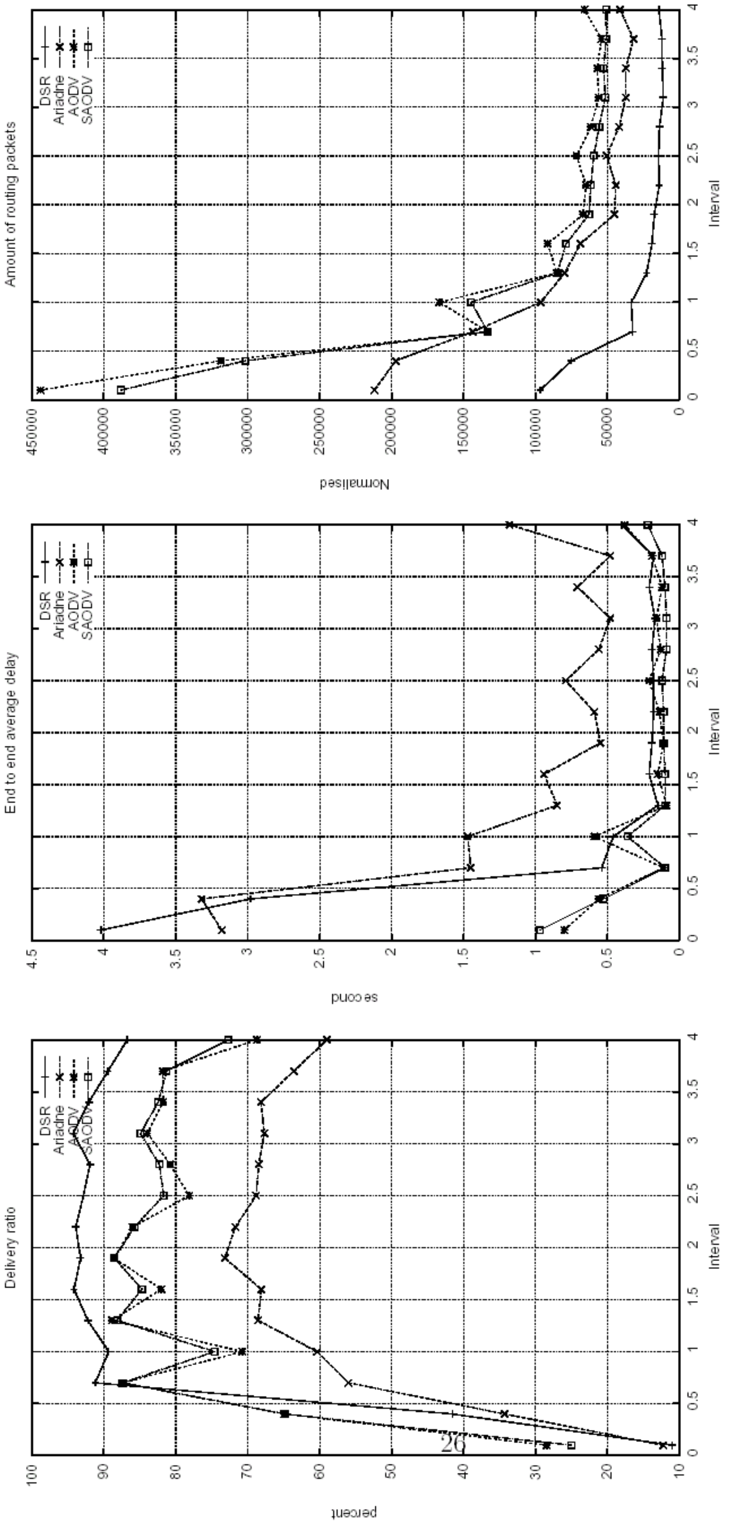
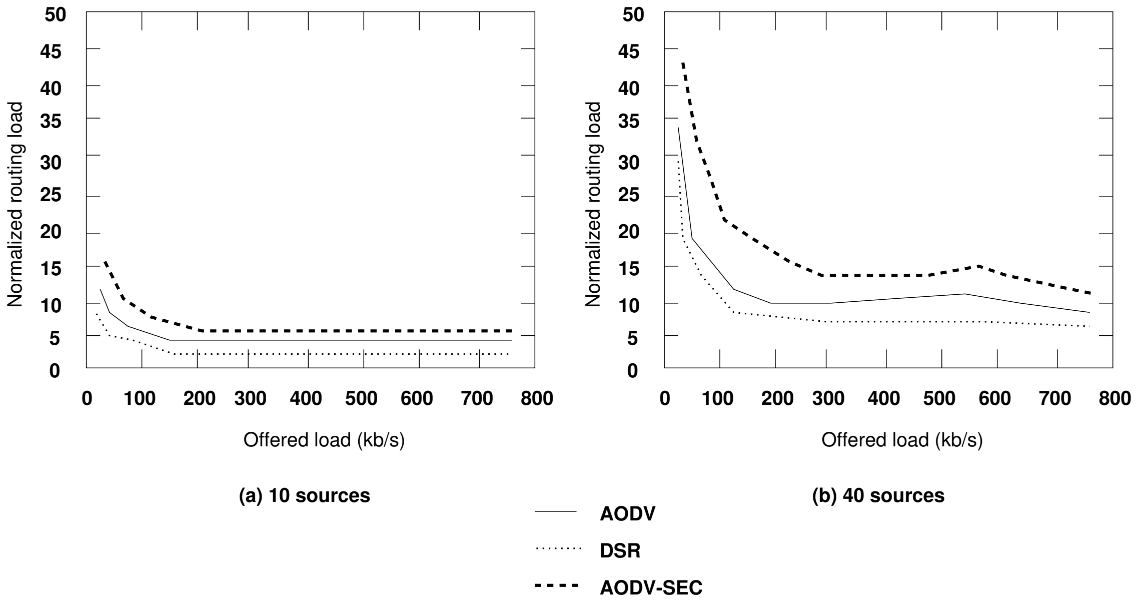


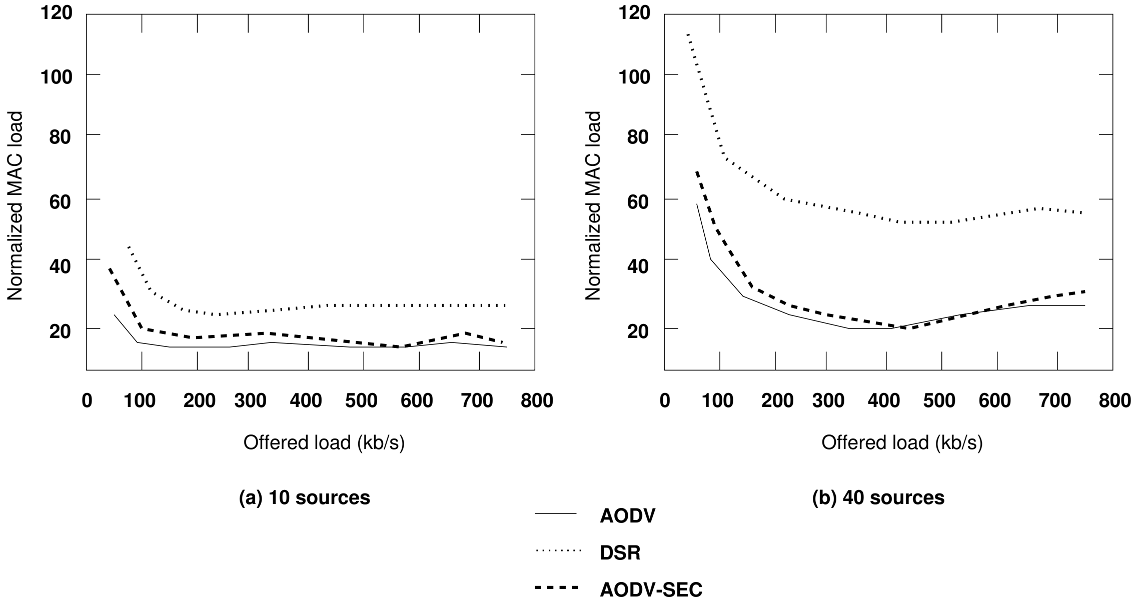
Figure C.5. 50 nodes, 100s, 30% CBR, CMU Setdest



(a) 10 sources

(b) 40 sources

Figure C.6. Normalized Routing Load for Scenario 2



(a) 10 sources

(b) 40 sources

Figure C.7. Normalized MAC Load for Scenario 2