Absicherung von Diagnosefunktionen in E/E-Fahrzeugarchitekturen durch verteilte Zugriffskontrolle und Anomalieerkennung

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Marcel Rumez M.Sc.
geb. in Mühlacker

Tag der mündlichen Prüfung: 08.11.2022

Hauptreferent: Prof. Dr.-Ing. Eric Sax
Korreferent: Prof. Dr.-Ing. Reiner Kriesten

Kurzfassung

Die Automobilindustrie befindet sich derzeit in einer Transformation, die durch Trends wie beispielsweise der Elektromobilität, dem automatisierten und vernetzten Fahren oder der geteilten Mobilität angetrieben wird. Das Fahrzeug sowie insbesondere die zugehörige Elektrik/Elektronik-Architektur ist ein Teil dieser Transformation. Für die Bereitstellung neuer Fahr- und Komfortfunktionen wird zunehmend mehr Software integriert sowie der Vernetzungsgrad durch die Anbindung der Umwelt (z.B. Hersteller-Backend Systeme) über drahtlose Kommunikationstechnologien gesteigert. Dabei sind in den letzten Jahren zunehmend Angriffe auf die Informationssicherheit von Fahrzeugen bekannt geworden, die auf schwache oder fehlende Absicherungsmaßnahmen zurückzuführen sind. Dadurch gelang es beispielsweise im Jahr 2015 unautorisierten Personen über das Mobilfunknetz aktiv in die Fahrzeugkommunikation einzugreifen und darüber Aktuatoren wie die Lenkung oder Bremse aus der Ferne anzusteuern.

Im Rahmen dieser Dissertation werden daher bekannte Angriffe im Zeitraum 2010 - 2019 analysiert sowie bisherige Schwächen bei der Absicherung und Erkennung von Angriffen identifiziert. Im Kern kristallisiert sich auf Basis der untersuchten Angriffe eine bisherige Schwäche im Bereich der Zugriffskontrolle auf Anwendungs- und Netzwerkebene heraus, da kein oder nur ein eingeschränktes Rechtemanagement implementiert war. Darüber hinaus zeigt die Aufarbeitung des aktuellen Stands der Technik und Wissenschaft eine bisherige Forschungslücke auf diesem Gebiet.

Auf der Grundlage dieses Wissensstandes wird im Rahmen dieser Arbeit ein Konzept für eine verteilte automotive attributsbasierte Zugriffskontrolle (A-ABAC) vorgestellt. Diese ermöglicht die Kontrolle sowie Durchsetzung von Diagnose-Berechtigungen in Fahrzeugsteuergeräten sowie den Datenaustausch der beteiligten Module in signal-orientierten Fahrzeugarchitekturen. In Anlehnung an bekannte Angriffe wird das entwickelte Konzept durch verschiedene Use-Cases im Rahmen eines Proof-of-Concepts getestet.

Ein weiterer Schwerpunkt dieser Arbeit liegt auf der Entwicklung eines Konzepts zur Erkennung von Anomalien innerhalb der Diagnosekommunikation, die durch Insider-Angriffe verursacht werden. Dieser Aspekt wurde bisher im Stand der Technik und Wissenschaft noch nicht adressiert. Präventive Maßnahmen sind in ihrer Wirkung eingeschränkt, einen Angreifer mit Insider-Wissen (z.B. legitimierte Zugangsdaten) abzuwehren. Das entwickelte Intrusion Detection System (IDS) besitzt die Fähigkeit, auf Basis eines bekannten Normalverhaltens durch einen Insider verursachte Abweichungen (Anomalien) zu erkennen, indem eine Methode der Computerlinguistik auf die Diagnosekommunikation adaptiert wird. Die prinzipielle Funktionsfähigkeit des Erkennungsansatzes wird im Rahmen von drei unterschiedlichen Anomalietypen gezeigt.

Neben signal-basierten Architekturen werden zunehmend service-orientierte Architekturen (SOA) in Fahrzeuge integriert, um die Updatefähigkeit sowie Anpassungsmöglichkeiten während des Entwicklungsprozesses sowie im Feld durch dynamische Kommunikationsbeziehungen zu steigern. Durch diesen Paradigmenwechsel entstehen jedoch neue Herausforderungen in Bezug auf die Informationssicherheit. Bisherige Absicherungsmaßnahmen sind in das veränderte SOA-Kommunikationsverhalten nur eingeschränkt adaptierbar. Im Ausblick werden zugehörige Unterschiede mit Fokus auf die Maßnahmen der Zugriffskontrolle und Anomalieerkennung diskutiert sowie Potentiale aufgezeigt. Darunter auch die Möglichkeit zur Adaptierung der in dieser Arbeit entwickelten Zugriffskontrolle.

Abstract

The automotive industry is currently undergoing a transformation driven by trends such as electric mobility, automated and connected driving, and shared mobility. The vehicle, and in particular the associated electrics/electronics architecture, is part of this transformation. More and more software is being integrated to provide new driving and comfort functions, and the degree of connectivity is being increased by connecting the environment (e.g., manufacturer backend systems) via wireless communication technologies. In this context, attacks on the information security of vehicles have been increasingly reported in recent years due to weak or missing countermeasures. In 2015, unauthorized persons were able to actively influence vehicle communication via the mobile network and remotely control actuators such as the steering or brakes.

This dissertation analyzes known automotive attacks in the period from 2010 to 2019 and identifies security weaknesses of corresponding systems. A major weakness of the attacks investigated is in the area of access control at the application and network level, since no or only a limited rights management was implemented. In addition, the review of the current state of the art and science reveals a previous research gap in this area.

Based on these results, a concept for a distributed automotive attribute-based access control (A-ABAC) is presented in this thesis. This enables the control and enforcement of privileges in vehicle control units as well as the data exchange of the involved modules within signal-oriented vehicle architectures. According to known attacks, the developed concept is tested by different use cases as a proof-of-concept.

Another focus of this work is the development of a concept for detecting anomalies within diagnostic communication caused by insider attacks. This aspect has not yet been addressed in the state of the art and science. Preventive measures are limited in their ability to defend against an attacker with insider

knowledge (e.g., legitimized credentials). The developed intrusion detection system has the ability to detect deviations (anomalies) caused by an insider based on a known normal behavior by adapting a computational linguistics method to diagnostic communication. Three different types of anomalies are used to demonstrate the detection approach.

In addition to signal-based architectures, more flexible service-oriented architectures (SOAs) are increasingly being integrated into vehicles, e.g., to increase the ability for updates or adaptions during the development process as well as after production due to dynamic communication relations. However, this paradigm shift creates new implications in terms of information security. Existing security measures for signal-oriented communication can only be adapted to a limited extent due to a changed SOA communication behavior. In the outlook, related disparities with a focus on access control and IDS are discussed and potentials are outlined.

Danksagung

Diese Dissertation entstand während der Zeit als externer Doktorand am Institut für Technik der Informationsverarbeitung (ITIV) am Karlsruher Institut für Technologie (KIT). Die darin durchgeführten Forschungsaktivitäten wurden im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts *AUTO-SIMA* auf Basis der Förderlinie *IngenieurNachwuchs - Kooperative Promotion* am Institut für Energieeffiziente Mobilität (IEEM) der Hochschule Karlsruhe (HKA) durchgeführt.

Zunächst bedanke ich mich herzlich bei meinem Doktorvater Prof. Dr.-Ing. Eric Sax für die umfangreiche Betreuung und regelmäßigen Rückmeldungen im Verlauf der Forschungsarbeit. Darüber hinaus auch für die Leitung der sehr wertvollen Doktoranden-Seminare. In diesen Veranstaltungen wurde man durch motivierende Ansprachen, konstruktives Feedback sowie Erfahrungsberichte aus Wissenschaft und Industrie bestens unterstützt.

Des Weiteren bedanke ich mich bei Prof. Dr.-Ing. Reiner Kriesten für die Funktion als Projektleiter und Betreuung der Forschungsarbeit an der HKA sowie die Übernahme des Zweitgutachtens. Außerdem grundlegend für das entgegengebrachte Vertrauen während meiner gesamten Zeit als akademischer Mitarbeiter am IEEM. Ein weiterer Dank geht an die Mitglieder der Prüfungskommission, bestehend aus dem Vorsitzenden Prof. Dr.-Ing. John Jelonnek sowie den Beisitzern Prof. Dr. Ivan Peric und Prof. Dr.-Ing. Thomas Leibfried.

Ein besonderer Dank geht natürlich ebenso an alle Kolleginnen und Kollegen des IEEM und ITIV für den fachlichen und persönlichen Austausch sowie die sehr gute Zusammenarbeit bei gemeinsamen wissenschaftlichen Veröffentlichungen. Insbesondere möchte ich hierbei meine Bürokollegen Dr. Jürgen Dürrwang sowie Florian Sommer hervorheben, die immer konstruktives Feedback zu erstellten Ausarbeitungen gaben und mit denen ich viele gute fachliche Gespräche im Themengebiet *Automotive Security* führen durfte. Außerdem besaßen sie einen intrinsischen Antrieb zum permanenten Erreichen neuer For-

schungserkenntnisse, das mich ebenfalls motivierte. Zugleich geht aber auch ein herzliches Dankeschön an meinen sehr geschätzten Kollegen Felix Müller, der stets ein verlässlicher Ansprechpartner war und mich durch viele unterschiedliche Aktivitäten unterstützte. Des Weiteren bedanke ich mich ebenfalls bei Prof. Dr. Thomas Fuchß für die sehr gute fachliche Zusammenarbeit sowie bei allen Studierenden, die über Projekt- und Abschlussarbeiten wissenschaftliche Untersuchungen in den von dieser Arbeit adressierten Themengebieten durchführten.

In meinem privaten Umfeld danke ich meinen Eltern, die mir immer eine uneingeschränkte Unterstützung auf meinem bisherigen Lebensweg gaben und damit grundlegend zum Erfolg dieser Dissertation beitrugen. An dieser Stelle möchte ich mich aber auch besonders bei meiner Frau Jennifer bedanken, die mir einen bedingungslosen Rückhalt gab und damit Teil dieses Erfolges ist. Nun können wir mehr gemeinsame Zeit mit unserem Sohn Lucas verbringen.

Bretten, im November 2022
Marcel Rumez

Inhaltsverzeichnis

1 Einleitung

Im heutigen Zeitalter der digitalen und vernetzten Welt steigen mit der Verbreitung auch die Risiken für Angriffe (s. Definition 2.2.11) auf die Informationssicherheit (engl. information security, s. Definition 2.2.1). Allein in Deutschland wurden im Bundesamt für Sicherheit in der Informationstechnik (BSI) in den letzten Jahren zunehmend Meldungen über Sicherheitsvorfälle auf kritische Infrastrukturen registriert [1]. Demnach lag die Zahl im Jahr 2018 noch bei 145. Im Jahr 2020 stieg die Anzahl der Vorfälle bereits auf 419. Die betroffenen Betreiber stammen dabei vorwiegend aus den Branchen Energie, Gesundheit, Telekommunikation sowie Transport & Verkehr. Darüber hinaus gibt es eine Vielzahl von Angriffen auf weitere deutsche Unternehmen und Einrichtungen. So waren laut einer Umfrage 88 % der befragten Unternehmen von Cyber-Angriffen betroffen [2]. Der dadurch verursachte Schaden stieg in den Jahren 2019 und 2020 von 103 auf 220 Milliarden Euro.

Vor einigen Jahren waren derartige Angriffe vorwiegend aus dem Personal-Computer Bereich bzw. aus größeren Unternehmen bekannt. Jedoch ist die Vernetzung und dadurch die Verbindung zum Internet sehr stark fortgeschritten wodurch eine Vielzahl an heutigen Produkten die Eigenschaft einer permanenten Internetverbindung besitzen. Dieser technologische Fortschritt bringt neben Vorteilen auch Nachteile mit sich. Ähnlich wie bei der Verbreitung des Internets im klassischen IT-Bereich wurde die Informationssicherheit der neuen Systeme teilweise ausgeklammert. Die Angreifergruppen (s. Definition 2.2.11) machen sich die daraus resultierenden unsicheren Systeme zu Nutze und versuchen über verschiedene Schnittstellen (drahtlos oder drahtgebunden) in die Systeme einzudringen [3]. Die Angreifer haben das Ziel, dem Objekt beispielsweise durch *Ransomware* gezielt Schaden zu zufügen oder sich illegal Daten zu beschaffen (Wirtschaftsspionage). Die zwei jüngsten Ransomware-Angriffe im Sommer 2021 bestätigen erneut die Aktualität dieser Bedrohung sowie das damit verbundene Schadenspotential. So musste eine Supermarktkette durch diese Art von Angriff hunderte Filiale schließen [4]. In einem weiteren

Fall führte der Angriff zum Ausfall der gesamten Verwaltungs- IT-Infrastruktur in einem deutschen Landkreis [5]. Daraufhin wurde der in Deutschland bisher erste Katastrophenfall ausgerufen, der auf einen Cyber-Angriff zurückzuführen ist. Das BSI schätzt derzeit die Ransomware-Angriffe als eine der größten Bedrohungen für IT-Systeme von Unternehmen und Organisationen ein [1]. Aufgrund der fortschreitenden Vernetzung und Digitalisierung ist die Informationssicherheit auch in anderen Industriebereichen wie beispielsweise der Automobilindustrie bei der Entwicklung von neuen Fahrzeugen von entscheidender Bedeutung.

Definition 1.0.1 Ransomware

Als Ransomware bezeichnet man Schadsoftware, die in IT-Systemen den Zugriff auf Daten oder Funktionen einschränkt. Meist verschlüsselt die Software alle Daten auf dem System und fordert das Opfer für eine Entschlüsselung zu einer Lösegeldzahlung auf [6].

1.1 Die Automobilwelt im Wandel

Derzeit befindet sich die Automobilbranche in einer Transformation, da sich bisherige Geschäftsmodelle grundlegend verändern. Die Haupttreiber für diese Veränderung können nach Strategy& [7] in fünf Kategorien (Elektrifizierung, autonomes Fahren, geteilte Mobilität, Vernetzung, jährliche Updates) eingeteilt werden. Im Detail bedeutet dies, dass gerade bei den Antriebstechnologien ein Großteil der Hersteller die Modellpalette an Verbrennungsmotoren durch neue elektrische Fahrzeugplattformen ersetzen, um die Grenzwerte der Gesetzgeber (z.B. Europäische Union (EU)) im Hinblick auf die Schadstoffemissionen auch in Zukunft erfüllen zu können. Daneben verursacht das Thema autonomes Fahren bei den bisherigen Original Equipment Manufacturers (OEMs) ein verändertes Marktumfeld, da bei dieser Technologie neue Konkurrenten in den Markt einsteigen. So investieren große Softwareunternehmen aus dem Silicon Valley viel Geld in die Forschung und Entwicklung von automatisierten Fahrfunktionen [8]. Um dieser Konkurrenz durch die Bündelung von Kompetenzen zur Steigerung der Innovationskraft entgegenzutreten, fusionieren bereits erste große Automobilkonzerne im Bereich des autonomen Fahrens [9], die

eigentlich als Konkurrenten anzusehen sind. Darüber hinaus prägt das Thema Mobilität die zukünftigen Geschäftsmodelle. Die Fahrzeughersteller gehen zunehmend auch in die Rolle eines Dienstleisters für Mobilität über, um auf die veränderten Kundenanforderungen einzugehen. Dadurch wächst die Zahl der Car-Sharing Anbieter und das Fahrzeug wird in Kombination mit anderen Transportmittel, wie der Bahn oder dem Bus ein Teil eines ganzheitlichen Mobilitätskonzepts. Die Basis dieser genannten Trends bildet dabei der zunehmende Vernetzungsgrad des Fahrzeugs intern als auch mit der Außenwelt [10]. Moderne Fahrzeuge sind bereits automatisch über das Mobilfunknetz permanent mit dem Internet verbunden, da eine gesetzliche Bestimmung der EU vorsieht, dass jedes neue Fahrzeug ab April 2018 ein sogenanntes *emergency Call (eCall)* System integriert haben muss [11].

Definition 1.1.1 eCall

Als eCall wird ein automatisches Notrufsystem bezeichnet, das bei einem schweren Unfall automatisch die Notrufnummer kontaktiert und neben Audiodaten auch die genaue Position des verunfallten Fahrzeugs übermitteln kann.

Diese drahtlose Schnittstelle bietet den OEMs zukünftig die Möglichkeit, die Update- und Upgradefähigkeiten ihrer Fahrzeugsysteme zu erhöhen, um damit dynamischer auf Kundenanforderungen und Marktveränderungen reagieren zu können. Konkret bedeutet dies, dass Fahr- und Komfortfunktionen nicht mehr ausschließlich während des Fahrzeugentwicklungsprozesses spezifiziert und implementiert werden, sondern auch dann, wenn sich das Fahrzeug bereits auf der Straße befindet. Insgesamt werden agile und kurze Entwicklungsprozesse angestrebt, um neue Softwarefunktionen schneller in die Fahrzeuge zu bringen [12]. Daneben gibt es Überlegungen eine Art *App-Store* in die Fahrzeuge zu integrieren, damit Kunden jederzeit benötigte Funktionen nachladen können [13]. Um diese Flexibilität umsetzen zu können ist es notwendig, das bisherige Kommunikationsparadigma der signal-orientierten Kommunikation in eine service-orientierte Kommunikation (s. Abschnitt 2.1.1) zu überführen. Statische Abhängigkeiten zwischen den verschiedenen Steuergeräten (engl. Electronic Control Units (ECUs)) entfallen, da die Kommunikationsbeziehungen erst während der Laufzeit dynamisch festgelegt werden [14]. Benötigt eine Applikation beispielsweise eine Sensorinformation, wird diese zukünftig je nach Verfügbarkeit entweder von einem internen Steuergerät des Fahrzeugs

oder über die Luftschnittstelle von einer Daten-Cloud bereitgestellt. Des Weiteren sind Änderungen bzw. die Integration neuer Funktionen bei Fahrzeugen im Feld einfacher möglich, da beispielsweise benötigte Sensorinformation über vorhandene Services domänenübergreifend abonniert werden können [15].

Diese grundlegende Änderung des Kommunikationsverhaltens hat jedoch direkte Auswirkungen auf die elektrisch-elektronische (E/E)-Architektur (s. Abschnitt 2.1.1) der Fahrzeuge [16]. Das über Jahrzehnte eingesetzte und zuverlässige Controller Area Network (CAN) für die interne Fahrzeugkommunikation wird zunehmend durch die Automotive Ethernet Technologie (s. Abschnitt A.1.1) ersetzt, um die veränderten Anforderungen u.a. in den Bereichen Datenrate, Interoperabilität sowie Datensicherheit zu erfüllen. Darauf aufbauend kommen zunehmend Protokolle wie Scalable Service-Oriented Middleware over IP (SOME/IP) (s. Abschnitt A.1.1) zum Einsatz, die die Basis für eine service-orientierte Architektur (SOA) bilden [17]. Da Fahrzeuge eine große Anzahl sicherheitskritischer Funktionen enthalten, die vorwiegend auf dem CAN-Protokoll basieren und sich über Jahrzehnte weiterentwickelt haben, ist davon auszugehen, dass dieser Paradigmenwechsel nicht in einem Schritt vollzogen wird. Vielmehr wird es in kommenden E/E-Architekturen eine hybride Form zwischen signal- und service-orientierter Kommunikation geben (s. Abbildung 2.8), um die Migration von Altsystemen (engl. legacy systems) sukzessive durchzuführen.

Durch die zunehmende Anzahl an implementieren Fahrzeugfunktionen in den nächsten Jahren wird die Informationssicherheit im Fahrzeug neben der funktionalen Sicherheit (s. Abschnitt 1.2) ein entscheidendes Qualitätsmerkmal werden. Heutige Fahrzeuge enthalten bis zu 100 Millionen Zeilen Programmcode, der bis im Jahr 2030 nach aktuellen Schätzungen auf 300 Millionen anwachsen soll [18]. Neben konkreten Absicherungsmaßnahmen in Form von technischen Lösungen rücken auch Prozessthemen in den Vordergrund, die durch regulatorische Vorgaben (z.B. durch die United Nations Economic Commission for Europe (UNECE) [19]), in Entwicklungsprozessen sowie im Feldbetrieb der Fahrzeuge bei den OEMs Anwendung finden müssen (s. Abschnitt 3.2.4). Einer Prognose zu Folge [20] wird sich dadurch der Markt für automotive Cybersecurity in den Jahren 2020 - 2030 von 4,9 auf 9,7 Milliarden USD erhöhen (s. Abbildung 1.1).

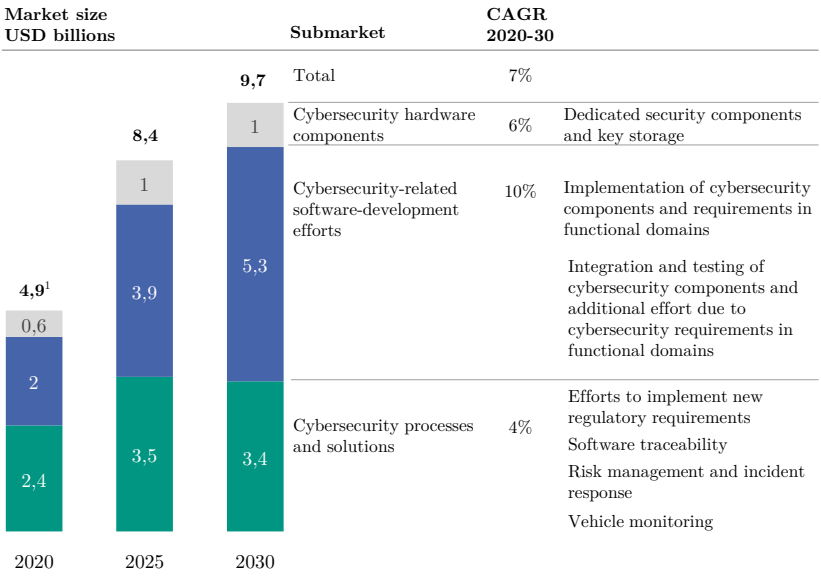


Abbildung 1.1: Prognose für die Entwicklung des automotive Cybersecurity Marktes in den Jahren 2020 - 2030 in Milliarden USD sowie die zugehörigen jährlichen Wachstumsraten (Compound Annual Growth Rate (CAGR)) der Teilmärkte (basierend auf [20]).

1.2 Die Bedeutung der IT-Sicherheit im automotive Kontext

Um in der IT ein verlässliches System entwickeln und betreiben zu können, ist die funktionale Sicherheit (engl. Safety) sowie die Informationssicherheit von Bedeutung (s. Abbildung 1.2). Dabei umfasst die funktionale Sicherheit auch Betriebssicherheit genannt nach der Norm International Electrotechnical Commission (IEC) 61508 [21], die Gewährleistung einer korrekten Funktionsweise einer Funktion innerhalb eines Gerätes oder Systems in Bezug auf die Elektronik bzw. Software. Darüber hinaus definiert der Standard auch die Betrachtung von gefährlichen Situationen und Zuständen des Systems, welche unter Umständen eine Gefahr für einen Menschen darstellen können.

Davon abgeleitet wurden weitere Standards, die unterschiedliche Industriebereiche adressieren. Die International Organization for Standardization (ISO) 26262 [22] wurde aus diesem Grund speziell für die Automobilbranche entwickelt, um detaillierte Vorgehensweisen und Maßnahmen zur Erfüllung der funktionalen Sicherheit zu definieren. Auf der anderen Seite stellt die Informationssicherheit (s. Definition 2.2.1) die zweite Komponente eines verlässlichen Systems dar, die betrachtet werden muss, um im Wesentlichen die vier informationstechnischen Sicherheits-Eigenschaften (Definitionen, s. Abschnitt 2.2.1) Vertraulichkeit (engl. confidentiality), Verfügbarkeit (engl. availability), Integrität (engl. integrity) sowie die Authentizität (engl. authenticity) von Informationen sicherzustellen [23].

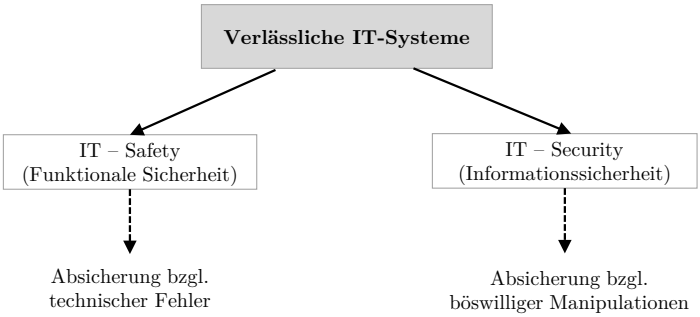


Abbildung 1.2: Ein verlässliches IT-System erfordert eine funktionale- und informationstechnische Sicherheit (basierend auf [24]).

Zur Gewährleistung dieser Eigenschaften können konkrete, herstellerabhängige Maßnahmen in die Informationssysteme integriert werden, um den Schutz der Informationen zu erhöhen und gleichzeitig das Risiko für einen potentiellen Angriff zu minimieren. Um effiziente Maßnahmen für ein System auswählen und integrieren zu können wird zuvor üblicherweise eine Bedrohungsanalyse (engl. threat analysis, s. Abschnitt 2.2.1) durchgeführt [25]. Die Analyse dient der Ermittlung von möglichen Angriffsvektoren (engl. attack vectors), die ein Angreifer nutzen könnte, um in ein System einzudringen. Darauf basierend werden missbräuchliche Anwendungsfälle (engl. misuse cases) definiert, die ein Angreifer vorsätzlich an einem System ausführen könnte. Es handelt sich hierbei grundsätzlich um einen nicht gewünschten Anwendungsfall oder eine Verkettung mehrerer dieser Anwendungsfälle, die ursprünglich nicht

durch den Hersteller bzw. Entwickler des Systems vorgesehen waren. Im Fahrzeug kann dies wiederum verschiedene Konsequenzen zur Folge haben. Die Angreifbarkeit und deren Auswirkungen wurde bereits durch wissenschaftliche Institutionen belegt [26], [27], [28], [29] (s. Abschnitt 3.1). Einerseits war dadurch ein unautorisiertes Auslesen von Daten möglich, da notwendige Security-Eigenschaften wie beispielsweise die Vertraulichkeit innerhalb des betroffenen Fahrzeugsystems nicht gewährleistet wurde. Auf der anderen Seite wurden Angriffe demonstriert, die ein Einschleusen von manipulierten Nachrichten in die fahrzeuginternen Netzwerke möglich machte. Letztlich lassen sich die Ursachen für die erfolgreichen Angriffe wiederum auf nicht ausreichend geschützte Security-Eigenschaften wie die Authentizität und Integrität zurückführen. Es wurde somit nicht überprüft, ob die gesendeten Daten von einem vertrauenswürdigen Sender stammen (s. Abschnitt A.1.3) oder die Daten womöglich auf dem Informationskanal manipuliert wurden. Des Weiteren zeigten Sicherheitsforscher im Jahr 2017 mögliche Angriffsszenarien auf, wodurch Fahrzeugsysteme in Zukunft durch Ransomware ungewollt verschlüsselt werden könnten [30]. In diesem Fall wäre die Security-Eigenschaft der Verfügbarkeit verletzt, da die betroffenen Fahrzeugfunktionen nur noch sehr eingeschränkt funktionieren würden. Der Fahrer müsste dann ein Lösegeld an die entsprechende Angreifergruppe bezahlen, um das Fahrzeug wieder in einen funktionsfähigen Zustand zu versetzen. Darüber hinaus enthält der vom BSI im Jahr 2021 veröffentlichte Bericht [31] zur Lage der Cyber-Sicherheit in der Automobilbranche einen Überblick über aktuelle Bedrohungen sowohl für Fahrzeuge als auch für zugehörige Produktionsanlagen.

Passieren Angriffe auf die Informationssicherheit in bestimmten Fahrzeugzuständen (s. Definition 3.2.2) beispielsweise während der Fahrt, ist durch die Verletzung der informationstechnischen Eigenschaften gleichzeitig auch eine Beeinflussung des Fahrverhaltens möglich und dadurch die funktionale Sicherheit des betroffenen Systems gefährdet. Es entsteht eine gefährliche Wechselwirkung zwischen der Betriebs- und Informationssicherheit, die grundsätzlich bei allen cyber-physischen Systemen (CPS) auftreten kann. Ein modernes Fahrzeug lässt sich nach Definition 1.2.1, auch als CPS bezeichnen, da der Aufbau ebenfalls aus verschiedenen Systemen bzw. Komponenten besteht (s. Abbildung 1.3). Die elektronischen Steuergeräte verarbeiten dabei die eingelesenen Sensorsignale über die implementierten Algorithmen und steuern die entsprechenden Aktoren an. Dabei ist es möglich, dass der Fahrer (Benutzer) als auch die Umwelt das Fahrverhalten beeinflussen können. Dieser Aufbau repräsen-

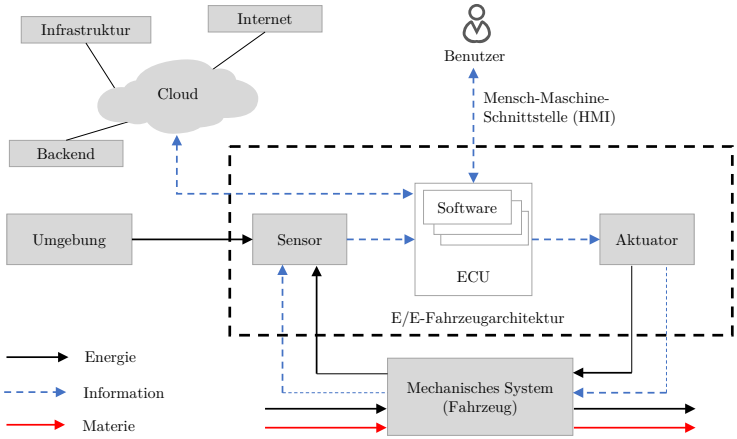


Abbildung 1.3: Schematischer Aufbau eines CPS basierend auf einem Fahrzeug sowie deren Vernetzung zu weiteren Teilsystemen (z.B. Internet) (basierend auf [32]).

tiert schematisch den typischen Systemaufbau in Fahrzeugen, der verschiedene Sensoren, Aktoren und Steuergeräte enthält, die über eine Netzwerktechnologie beispielsweise dem CAN miteinander vernetzt sind. Daneben können andere Teilsysteme (z.B. Hersteller-Cloud) über drahtlose Schnittstellen (z.B. Mobilfunk) ebenfalls einen Einfluss auf das Fahrverhalten verursachen, indem beispielsweise Diagnosekommandos an das Fahrzeug gesendet werden. Durch bestehende Wirkkette der aufgezählten Komponenten eines CPS ist es möglich, dass Angreifer beispielsweise durch manipulierte Sensorinformationen vorsätzlich das Fahrverhalten beeinflussen können. Als Konsequenz würde der implementierte Algorithmus auf dem beteiligten Steuergerät die manipulierten Sensorinformationen verarbeiten und den dazugehörigen Aktor (z.B. Bremse), aus Sicht des Entwicklers, zu einer ungewollten Ansteuerung bewegen.

Definition 1.2.1 Cyber-physisches System (CPS)

Nach Broy [33] ist die Struktur eines CPS über ein Schalenmodell definierbar, indem verschiedene Systeme miteinander interagieren und übergeordnet ein Gesamtsystem bilden. Die Teile der Systeme bestehen dabei aus elektronischen, mechanischen sowie softwaretechnischen Komponenten und sind über eine gemeinsame Datenschnittstelle mit der Umwelt beispielsweise dem Internet verbunden.

Die Angreifbarkeit von CPS in der automotive Domäne wurde mehrfach von verschiedenen Institutionen aus Forschung und Industrie demonstriert und in Publikationen veröffentlicht. Eine umfassende Untersuchung und Klassifizierung bereits bekannter automotive Angriffe wurde von Sommer et al. vorgestellt [34]. Grundlegend können die Angriffe dabei über den genutzten Angriffsvektor unterteilt werden, ob der ausgeführte Angriff auf einem physikalischen Zugriff (z.B. On-Board Diagnostics (OBD)-Schnittstelle, s. Abschnitt A.1.1) oder einer Luftschnittstelle (z.B. Mobilfunknetz) basierte. Dieses Kriterium ist für die Bewertung der Auswirkung entscheidend, wodurch möglicherweise eine Kompromittierung einer ganzen Fahrzeugflotte möglich ist. Ein derartiges Szenario stellte der im Jahr 2015 gezeigte Jeep-Hack [28] dar, indem es den Sicherheitsforschern gelang das Fahrverhalten aus der Ferne (engl. remote) über das Mobilfunknetz zu beeinflussen. Konkret war es den Forschern möglich, die Ansteuerbefehle für Lenkung und Bremsen beliebig zu ändern. Als sehr kritisch kann diese Schwachstelle (engl. vulnerability) aus zwei Gründen eingestuft werden. Zum einen war es möglich die Schwachstelle ohne physikalischen Zugriff auszunutzen (engl. exploit) und zum anderen hätte der Angriff innerhalb der gesamten betroffenen Baureihe funktioniert. Darüber hinaus haben weitere Forschungsaktivitäten gezeigt, dass durch die Ausnutzung mehrerer Schwachstellen eine unautorisierte Auslösung pyrotechnischer Einrichtungen (u.a. der Airbags, s. Abschnitt 3.1.3) über einen Diagnose-Service (s. Abschnitt 2.1.3) in aktuellen Fahrzeugen von verschiedenen OEMs möglich ist [DBRK17]. Aus der beschriebenen Historie von informationstechnischen Angriffen sowie den aktuellen Trends für neue Fahrzeugarchitekturen lässt sich die nachfolgende Problemstellung ableiten sowie zugehörige Forschungsfragen definieren.

1.3 Problemstellung und Forschungsfragen

Seit dem Bekanntwerden einiger medienwirksamer Sicherheitsvorfälle in den letzten Jahren ist in der gesamten Automobilindustrie eine steigende Sensitivität im Bereich der Informationssicherheit zu verzeichnen, wodurch einige Spezifikationen und Standards entstanden sind. Als Beispiel ist hier das Secure Onboard Communication (SecOC) Modul (s. auch Abschnitt A.1.5) oder der Crypto Stack [35] des AUTomotive Open System ARchitecture (AUTOSAR) Gremiums zu nennen, welche beispielsweise die Eigenschaften Authentizität und Integrität von Nachrichten auf dem CAN-Bus sicherstellen. Außerdem wurde im Jahr 2016 von der Society of Automotive Engineers (SAE) das *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems* [36] publiziert, dass die Informationssicherheit von Fahrzeugsystemen adressiert. Darüber hinaus wurde im Jahr 2021 der erste automotive Cybersecurity Standard (ISO/SAE 21434 [37]) publiziert, der Prozesse und Maßnahmen für die Einhaltung und Nachweisbarkeit einer sicheren Entwicklung spezifiziert, um die derzeitigen Lücken im Vergleich zur funktionalen Sicherheit zu schließen (s. auch Abschnitt 3.2.4). Viele der Ansätze adressieren derzeit vorwiegend die Absicherung des Informationskanals und damit die Authentizität/Integrität und/oder Vertraulichkeit der übertragenen Nachrichten. Hingegen fehlt es an Ansätzen für die *Autorisierung*.

Definition 1.3.1 Autorisierung

Die Autorisierung (engl. Authorization) kann als Prozess aufgefasst werden, der einem Subjekt (z.B. Benutzer) bestimmte Berechtigungen (lesen/schreiben) zuweist, um auf ein zugehöriges Objekt (z.B. Daten) zugreifen zu dürfen. Vor jeder Autorisierung muss zwingend eine Überprüfung der Echtheit eines Subjekts (Authentifizierung) erfolgen [38].

Dadurch ist es Angreifern im Fall einer erfolgreichen Authentifizierung möglich, beliebige Funktionen im Fahrzeugnetzwerk ausführen. Insbesondere ergibt sich diese Bedrohung für *Insider Angriffe* (s. Abschnitt 2.2.1), bei denen der Angreifer von einem kompromittierten Steuergerät oder allgemein von einem Netzwerkknoten agiert, der von den beteiligten Empfängern als vertrauenswürdig angesehen wird [39]. Dadurch sind auch *Insider Angriffe* (s.

Abschnitt 2.2.1) möglich, bei denen der Angreifer von einem kompromittierten Steuergerät oder allgemein von einem Netzwerkknoten agiert, der von den beteiligten Empfängern als vertrauenswürdig angesehen wird [39]. Dadurch wäre es Angreifern auch durch einen abgesicherten Informationskanal möglich, ungewollte Manipulationen sowohl auf dem kompromittierten Steuergerät als auch einem Empfängerkonten durchzuführen, da keine Kontrolle der Berechtigungen erfolgt.

Die einzige Ausnahme bilden aktuell Diagnosefunktionen, die je nach Zugriffsart und Funktionsumfang (z.B. Lesezugriff/Schreibzugriff) durch eine zusätzliche Berechtigungsstufe abgesichert sind. Der Wechsel zwischen den beiden Stufen ist aktuell mittels des sogenannten *Security Access* implementiert, der jedoch aufgrund von wissenschaftlichen Untersuchungen [40] [DBRK17] in der Vergangenheit als nicht sicher anzusehen war. Mittlerweile ist die Gewährleistung der Sicherheitseigenschaft *Authentizität* im Rahmen der Diagnose sichergestellt. Es fehlt jedoch weiterhin an Konzepten für die Autorisierung. So werden für die Vergabe von Berechtigungen keine dynamischen Parameter wie z.B. der aktuelle Fahrzeugzustand mit einbezogen. Wichtig wäre im Rahmen der Autorisierung zu prüfen, ob das Fahrzeug möglicherweise gerade in Bewegung ist oder sich in einem Diagnosemodus befindet, um darauf basierend feingranulare Zugriffsentscheidungen zu treffen. Damit wäre beispielsweise eine Reduzierung der Fähigkeiten eines Angreifers möglich, der bereits einen erfolgreichen Zugang zum Fahrzeugnetzwerk erlangt hat. Des Weiteren ergibt sich die Problematik, wie Angreifer bzw. zugehörige Angriffe erkannt werden können, wenn diese eine erfolgreiche Authentifizierung durchlaufen haben und dadurch als Insider agieren.

Definition 1.3.2 Security Access

Der Security Access ist ein Authentifizierungs- und Autorisierungsverfahren für die Diagnose, basierend auf dem Challenge-Response-Prinzip. Der Tester sendet hierzu eine Anfrage an ein Steuergerät. Als Antwort erhält der Tester eine Zufallszahl (Seed) und berechnet durch einen geheimen Algorithmus einen Schlüssel. Diesen Schlüssel sendet er anschließend an das Steuergerät, das die korrekte Berechnung und damit die Gültigkeit der Authentifizierung prüft und anschließend eine Menge an Berechtigungen vergibt [41].

Aus der beschriebenen Problemstellung lässt sich nachfolgende Hauptforschungsfrage ableiten:

Wie kann die Informationssicherheit von Diagnose-Funktionen auf automotive Steuergeräten durch ein Rechtemanagement sowie Intrusion Detection System in signal-orientierten Fahrzeugarchitekturen erhöht werden?

Die Einführung eines Rechtemanagement bietet die Möglichkeit, Berechtigungen von Benutzern präventiv auf die notwendigen Funktionen zur Erfüllung einer bestimmten Aufgabe oder Rolle einzuschränken. Grundlegend müssen jedoch nicht alle Funktionen in jedem Fahrzeugzustand (s. Definition 3.2.2) verfügbar sein, um damit den präventiven Schutz weiter zu erhöhen. Dadurch ergibt sich die nachfolgende Unterforschungsfrage:

Wie können Zugriffe auf Diagnosefunktionen in Abhängigkeit vom jeweiligen Fahrzeugzustand kontrolliert und durchgesetzt werden?

Weiter muss davon ausgegangen werden, dass präventive Schutzmaßnahmen nie einen vollumfänglichen Schutz bieten können [23]. Durch neue Angriffstechniken können bestehende Schutzmaßnahmen umgangen werden. Aus diesem Grund existieren bereits in anderen Anwendungsdomänen z.B. der klassischen IT sogenannte proaktive Absicherungsmechanismen, die Angriffe bzw. Abweichungen zum Normalverhalten (Anomalien) während ihres Auftretens erkennen können. Unter der Annahme, dass es einem Angreifer gelingt im Bereich der Fahrzeugdiagnose beispielsweise eine erfolgreiche Authentifizierung zu erreichen und dadurch als Insider agiert, lässt sich daraus die zweite Unterforschungsfrage ableiten:

Wie können Insider-Angreifer innerhalb der Diagnosekommunikation detektiert werden?

1.4 Zielsetzung und eigener Beitrag

Die bekannt gewordenen Angriffe auf die Informationssicherheit von Fahrzeugen erfordern für zukünftige Entwicklungen geeignete Absicherungsmaß-

nahmen, um das Risiko für neue Angriffe zu minimieren. Die vorliegende Arbeit analysiert daher vergangene Angriffe, um bisherige Schwächen der eingesetzten Systeme zu identifizieren. Auf Basis dieser Erkenntnisse wird ein Konzept für eine Zugriffskontrolle vorgestellt, um ein feingranulares Rechtemanagement für Fahrzeugfunktionen zu integrieren. Dadurch ist es möglich, dynamische Zugriffsentscheidungen in Abhängigkeit bestimmter Fahrzeugzustände zu treffen. Neben dieser präventiven Maßnahme adressiert die Arbeit eine pro-aktive Maßnahme und präsentiert ein Konzept zur Erkennung von kontextbasierten Anomalien in der Diagnosekommunikation, da diese Art von Erkennungsmaßnahme bisher ein Randgebiet im Bereich der Technik und Wissenschaft darstellt. Da in den nächsten Jahren zunehmend eine serviceorientierte Kommunikation Einzug in Fahrzeugarchitekturen hält, werden Unterschiede in Bezug auf die Anwendbarkeit von Absicherungsmaßnahmen aus der signal-orientierten Kommunikation analysiert. Im Ausblick werden derzeit offene, an diese Arbeit angrenzende Forschungsthemen erläutert sowie auf zukünftige Standards und Regularien der Informationssicherheit eingegangen, die für die Fahrzeugentwicklung von Relevanz sind.

1.5 Struktur der Arbeit

Diese Dissertation gliedert sich über insgesamt sieben Kapitel gemäß der Abbildung 1.4. Dabei umfassen die ersten beiden Kapitel (*Einleitung u. Grundlagen*) neben der Motivation und Zielsetzung/Forschungsfragen der Arbeit auch Verfahren und Technologien, die den eigenen Beitrag der Arbeit (Kapitel 3 -7) stützen. Dieser ist dabei an ein etabliertes Cybersecurity-Framework [42] des National Institute of Standards and Technology (NIST) angelehnt, welches die Prozessschritte (*Identify, Protect, Detect, Respond, Recover*) enthält.

Nachfolgend werden die Inhalte der einzelnen Kapitel grobgranular beschrieben:

Kapitel 2: Technische Grundlagen Dieses Kapitel enthält Grundlagenthemen zur Automobilelektronik, der allgemeinen IT-Informationssicherheit sowie ausgewählte Sicherheitstechnologie aus der automotive Domäne. Zu Beginn werden verschiedene Fahrzeugarchitektur-Varianten sowie zugehörige Kommunikationsprotokolle erläutert. Daran anknüpfend wird das Themengebiet der Fahrzeugdiagnose aufgegriffen. Danach folgt eine auf diese Ar-

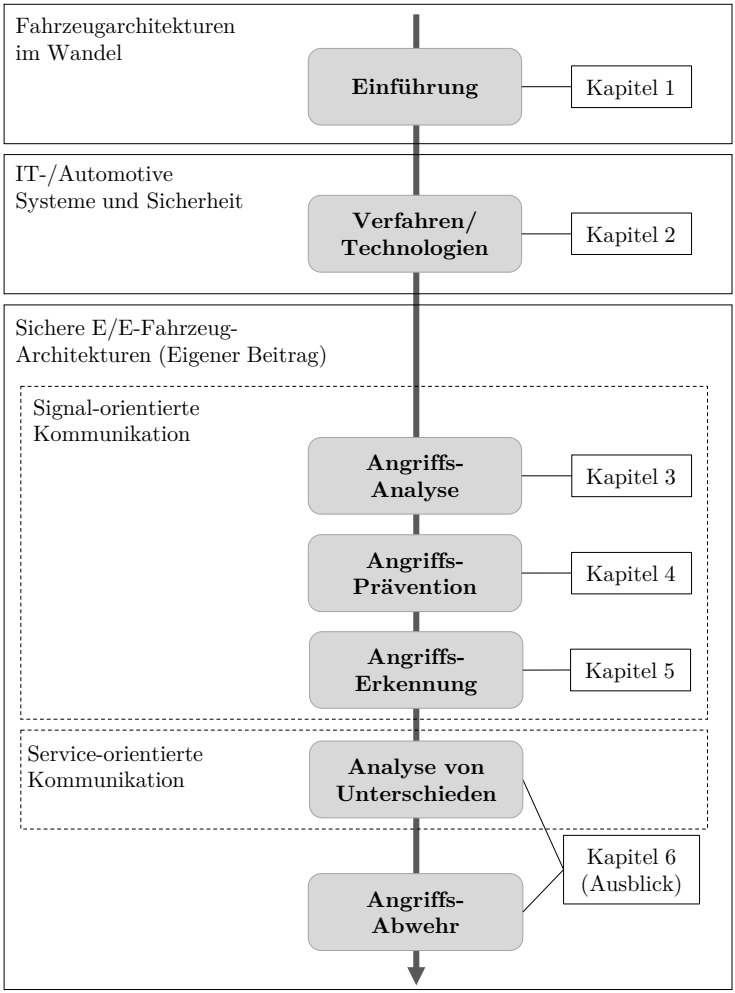


Abbildung 1.4: Struktur der Arbeit.

beit ausgerichtete Auswahl an Technologien und Verfahren der traditionellen IT-Sicherheit. Im dritten Unterkapitel werden etablierte Sicherheitsverfahren (Informationssicherheit) mit Fokus auf Fahrzeugsysteme erläutert.

Kapitel 3: Angriffsanalyse und Stand der Technik u. Wissenschaft Dieses Kapitel analysiert im ersten Teil bisherige publizierte Angriffe auf Fahrzeuge und ordnet diese über verschiedene Auswertungen ein. Der Fokus liegt hierbei auf der Gewährleistung der Security-Eigenschaft *Autorisierung*. Im zweiten Teil wird der aktuelle Stand der Technik und Wissenschaft hinsichtlich den Themenbereichen Angriffs-Prävention mit Fokus auf Firewalls und Verfahren der Zugriffskontrolle aufgearbeitet und erläutert. Zudem werden auch Ansätze im Bereich der Angriffs-Erkennung (IDS) aufgezeigt. Neben den wissenschaftlichen Ansätzen werden auch veröffentlichte Guidelines, Standards und Regularien untersucht, die durch Industriegremien, Standardisierungsorganisationen oder staatlichen Gremien bzgl. automotive Security spezifiziert wurden. Zum Ende des Kapitels werden in einer Zusammenfassung die Kernpunkte der Analyse dargelegt sowie derzeitige Forschungslücken aufgezeigt.

Kapitel 4: Angriffsprävention durch Zugriffskontrolle bei Diagnosefunktionen Zu Beginn werden auf Basis der Erkenntnisse aus Kapitel 3, Anforderungen für eine Zugriffskontrolle bzgl. Diagnosefunktionen in Fahrzeugsystemen definiert. Danach wird der Ansatz ausgehend von der Architektur erläutert, wie Zugriffsberechtigungen beschrieben bzw. kontrolliert und durchgesetzt werden. Ein Kernaspekt ist dabei die Einbeziehung von Fahrzeugzustandsinformationen in Zugriffsentscheidungen, die auf unterschiedlichen Sensorinformationen des Fahrzeugs basieren. Neben der Zugriffskontrolle auf Anwendungsebene wird auch ein Ansatz für eine Netzwerk-Firewall erläutert, die in Fahrzeuggateways integrierbar ist und ebenso Zustandsinformationen für die Filterung von Netzwerkpaketen miteinbezieht.

Kapitel 5: Angriffserkennung durch Anomalieerkennung bei Diagnosefunktionen Neben der Angriffs-Prävention in Form einer Zugriffskontrolle, die in Kapitel 4 erläutert wird, adressiert dieses Kapitel das Gebiet der Angriffserkennung im Bereich der Diagnoseanwendungen. Dafür werden zu Beginn mögliche Erkennungsansätze diskutiert sowie spezifische Diagnose-Anomalien definiert und zugehörige Anforderungen für die Erkennung von Anomalien in Kommunikationsdaten definiert. Danach erfolgt die Beschreibung des entwickelten IDS-Konzepts, das auf einem Sprachmodell der Computer-

linguistik basiert. Anhand von exemplarischen Testdaten wird die prinzipielle Funktionswiese in einer prototypischen Umsetzung gezeigt.

Kapitel 6: Zusammenfassung & Ausblick Die Arbeit wird mit einer Zusammenfassung der vorgestellten Analysen und Konzepten abgeschlossen. Darüber hinaus wird im Ausblick auf zukünftige automotive relevante Cybersecurity Standards und Regularien gegeben sowie die Thematik der Reaktion (engl. response) im Falle eines Angriffs bzw. Sicherheitsvorfalls (engl. security incident) aufgegriffen. Daran anknüpfend wird ein weiterer Forschungsbedarf aufzeigt, der die systematische Analyse von Sicherheitsvorfällen mit geeigneten Methoden und Werkzeugen umfasst, um daraus Erkenntnisse für zukünftige Entwicklungsprozesse abzuleiten. Die in Kapitel 4 und 5 beschriebenen Ansätze basieren auf einer signal-orientierten Kommunikation. Da aktuell sowie in Zukunft das SOA-Paradigma den Einzug in neue Fahrzeugarchitekturen hält, um dadurch eine service-orientierte Kommunikation umzusetzen, werden in diesem Kapitel Unterschiede zu signal-orientierten Architekturen erläutert. Der Fokus richtet sich dabei auf die in dieser Arbeit adressierten Themengebiete (Firewalls, Zugriffskontrolle und IDS).

2 Technische Grundlagen

Die nachfolgenden Kapitel dienen zur Einführung in die Grundlagen informationstechnischer Fahrzeugsysteme und sollen darüber hinaus ein Grundverständnis der Informationssicherheit vermitteln.

2.1 Elektronik im Automobil

In den 90er Jahren begann in der Automobilindustrie eine umfassende Entwicklungsaktivität im Bereich der elektronischen Systeme von Fahrzeugen [43]. Bis zu diesem Zeitpunkt waren die elektronischen Steuerungen bzw. Steuergeräte als einzelne, nicht vernetzte Komponenten in die Fahrzeuge integriert. Durch die Einführung des CAN (s. Abschnitt A.1.1) als dominierende Vernetzungstechnologie in Fahrzeugen ergaben sich völlig neuartige Konzepte im Hinblick auf Steuer- und Komfortfunktionen. Dadurch war es möglich, Funktionen verteilt auf unterschiedliche Steuergeräte zu integrieren und Informationen (z.B. Sensorwerte) über das CAN-Netzwerk auf andere Steuergeräte zu übertragen [44]. Über die nachfolgenden Jahre stieg der Bedarf an Bandbreite sowie weiteren Vernetzungsmöglichkeiten, da kontinuierlich neue Funktionen integriert wurden, sodass bis heute unterschiedliche Netzwerktechnologien (s. Abschnitt A.1.1) für verschiedene Anwendungsbereiche im Fahrzeug verwendet werden [45]. Dabei wird die FlexRay [46] Technologie vorwiegend in sicherheitskritischen Domänen wie Fahrwerksreglung eingesetzt. Hingegen kommen bei Komfortanwendungen, wie z.B. der Klimaregelung, günstigere Technologien wie der Local Interconnect Network (LIN)-Bus [47] zum Einsatz.

Die Betrachtung der evolutionären Entwicklung von Fahrzeugarchitekturen (s. Abbildung 2.1) zeigt auf Basis der Komplexität im Vergleich zur Menge an Funktionen (notwendige Komplexität) in den letzten Jahren immer wieder eine deutliche Abweichung zueinander [48]. Das allgemeine bekannte Phänomen des evolutionären Wachstums im Bereich Software Engineering wurde bereits

von Brooks [49] im Jahr 1987 beschrieben, das zu einer Steigerung der Integrationskosten führt und gleichzeitig die Innovation bremst. Im Fahrzeug wurde diesem Phänomen, wie beispielsweise durch die Einführung des CAN-Busses 1987, kontinuierlich entgegengewirkt. Dadurch folgte eine Approximation der beiden Kenngrößen. Dieser Trend wird sich auch in der Zukunft weiter fortsetzen, wobei gleichzeitig die Menge an Funktionen im Fahrzeug kontinuierlich ansteigen wird.

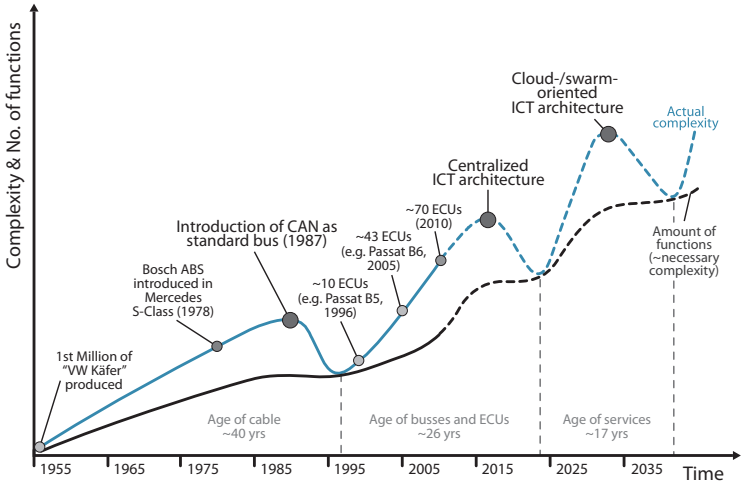


Abbildung 2.1: Verlauf der Komplexität auf Basis integrierter Elektronik im Vergleich zur Menge an implementierten Funktionen in Fahrzeugarchitekturen [48].

2.1.1 E/E-Architekturen

Unter einer Elektrik/Elektronik (E/E)-Architektur wird das Netzwerk-Design bestehend aus verschiedenen Steuergeräten und deren Vernetzung im Fahrzeug verstanden, die hierarchisch in vier Systemebenen (s. Abbildung 2.2) klassifizierbar ist [45]. In der obersten Schicht (Funktionsumfang) werden alle kundenerlebbaren Funktionen und zugehörige Features definiert. Diese Funktionen können wiederum eine Ebene darunter (Funktions-/Softwarearchitektur) in ihre verschiedenen Teilfunktionen aufgegliedert werden. Im Automotive-Bereich

wird diese Ebene auch als logische Architektur bezeichnet, die schematisch auf dem EVA-Prinzip (Eingabe-Verarbeitung-Ausgabe) basiert. Dies bedeutet eine Aufteilung in die Komponenten Sensor (Eingabe), die eigentliche Funktion (Verarbeitung) sowie dem zugehörigen Aktuator (Ausgabe). Ist dieser Schritt vollzogen, wird die logische Architektur auf eine technische Architektur oder auch Vernetzungsarchitektur übertragen. Hierbei werden die einzelnen Software-Komponenten auf reale Hardware-Komponenten (Aktoren, Sensoren, Steuergeräte) verteilt. Gleichzeitig ist wichtig, dass die entsprechende Leistung für die HW-Komponenten zur Verfügung gestellt wird. Als unterste und letzte Ebene folgt die Komponententopologie. Hierbei wird die geometrische Anordnung von Komponenten im zur Verfügung stehenden Bauraum des Fahrzeugs festlegt und die benötigten Leitungssätze für Versorgung und Kommunikation angeordnet.

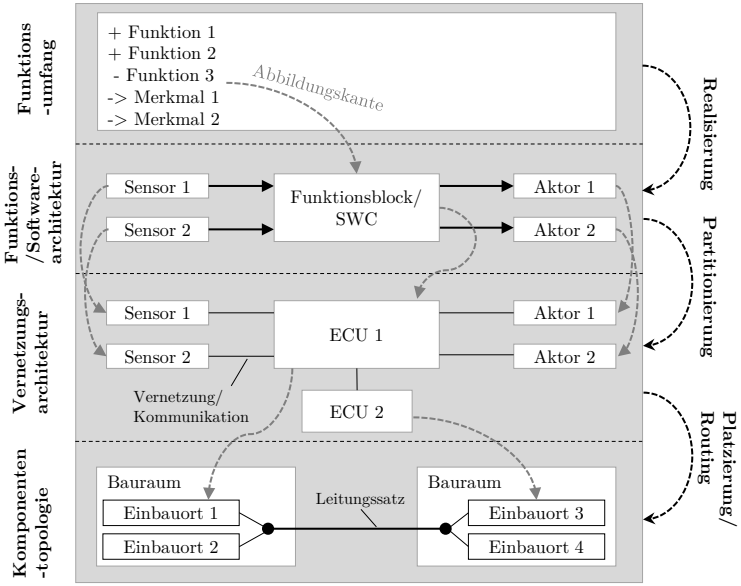


Abbildung 2.2: Übersicht verschiedener Ebenen einer E/E-Architektur (basierend auf [45]).

Da diese Arbeit die Absicherung der Fahrzeugarchitekturen fokussiert, wird nachfolgend auf heutige sowie zukünftige Architekturen und Netzwerktechno-

logien eingegangen. Nach Lock et al. [50] wird sich das E/E-Architekturdesign in den nächsten Jahren grundlegend verändern (s. Abbildung 2.3), um die Komplexität aus aktuell über 120 dezentralen Steuergeräten zu verringern und darüber hinaus eine erhöhte Rechenleistung für vorwiegend automatisierte Fahrfunktionen in unterschiedlichen Ausbaustufen bereitzustellen. Zukünftig werden die bisher physikalisch getrennten Domänen und verteilten Funktionen in leistungsstarken Domänen-unabhängigen Zonen-ECUs (auch Vehicle Computer (VC) genannt) zentralisiert. Eine Vorstufe dieser Entwicklung bildet die Domänen-Zentralisierung, in der bisherige domänenspezifische Funktionen von einzelnen ECUs auf Domänen-Controller übertragen werden. Parallel dazu wird die Interaktion mit Cloud-basierten Funktionen außerhalb des Fahrzeugs über eine Luftschnittstelle weiter zunehmen.

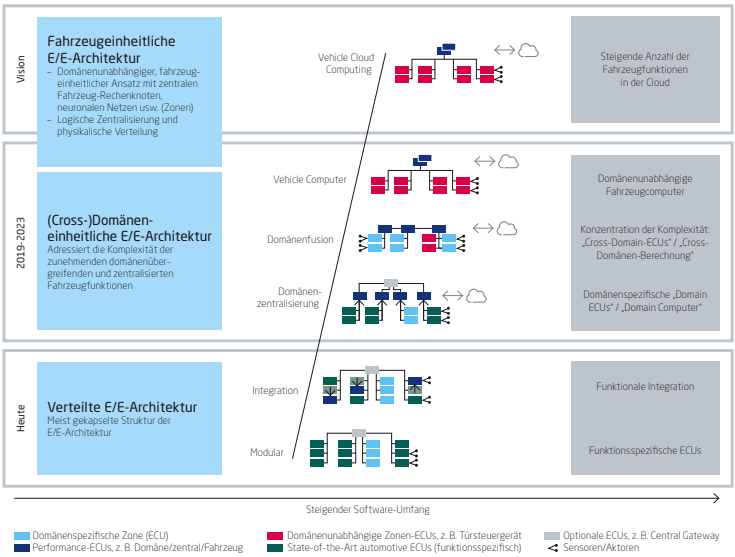


Abbildung 2.3: Veränderung der E/E-Architektur von Fahrzeugen in den nächsten Jahren [50].

Verteilte E/E-Architektur

Aktuell sind in Fahrzeugen E/E-Architekturen integriert (vgl. Abbildung 2.3), in der jede Funktionalität auf einem eigenen Steuergerät implementiert wird (z.B. Tür-ECU oder Motor-ECU) [45]. Je nach Anwendungsdomäne (Antrieb, Karosserie oder Infotainment) kommen dabei unterschiedliche Bustechnologien (z.B. CAN, LIN oder FlexRay) zum Einsatz mit denen die ECUs über ein zentrales Gateway vernetzt werden (s. Abbildung 2.4).

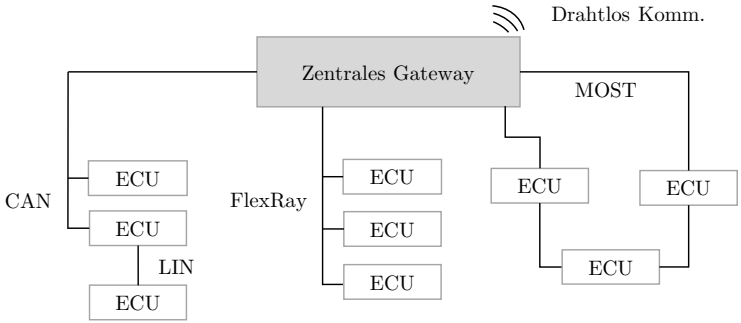


Abbildung 2.4: Darstellung einer E/E-Architektur mit verteilten Steuergeräten und unterschiedlichen Bussystemen (basierend auf [15]).

Domänen-einheitliche E/E-Architektur

Darüber hinaus existieren in aktuellen Baureihen Architekturdesigns (vgl. Abbildung 2.3), die eine Weiterentwicklung des verteilten Ansatzes darstellen [15]. Durch den steigenden Bedarf an Bandbreite bei der Fahrzeug-internen Kommunikation wird als Rückgrat (engl. Backbone) ein Ethernet-basiertes Netzwerk integriert. Das zentrale Gateway fungiert darin als Routingknoten, an den alle untergeordneten Domänen-Controller jeweils über eine eigene Ethernet-Verbindung angebunden sind (s. Abbildung 2.5). Die Domänen-Controller sind dabei leistungsstarke Plattformen, auf die rechenintensive Funktionen der jeweiligen Domäne ausgelagert sind. Die Aufteilung der Domänen erfolgt weiterhin in die bisher verwendeten Anwendungstypen, wie z.B. Fahrwerk, Antrieb oder Infotainment [51].

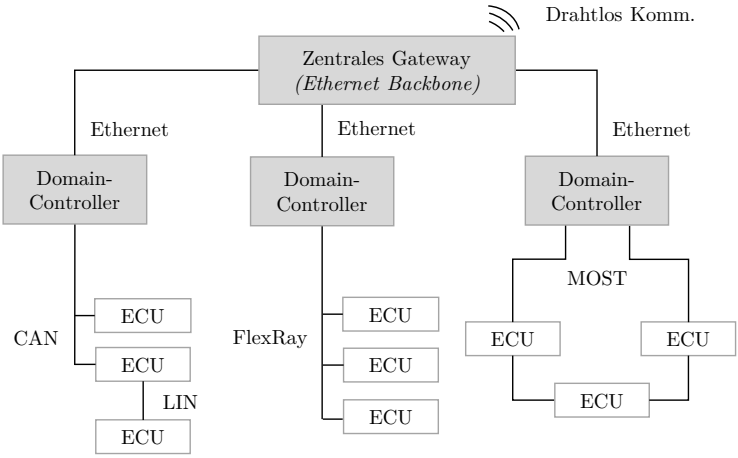


Abbildung 2.5: Darstellung einer verteilten Domänen E/E-Architektur mit zentralem Ethernet-Backbone (basierend auf [15]).

Fahrzeugeinheitliche E/E-Architektur

Diese Art von E/E-Architekturen werden frühestens ab dem Jahr 2024 prognostiziert (vgl. Abbildung 2.3) und bestehen aus domänenunabhängigen Fahrzeug-Rechenknoten (Zone Controller Units (ZCUs)) die mit High-Performance-Computers (HPCs) über automotive Ethernet (s. Abschnitt A.1.1) verbunden sind (s. Abbildung 2.6) und dazu Redundanzen zur Gewährleistung einer zuvor spezifizierten Ausfallsicherheit beinhalten [15]. HPCs sind leistungsstarke Rechencluster, die bisherige physikalisch getrennte Domänen-Controller und zugehörige Funktionen darin zentralisieren. ZCUs sind im Fahrzeug räumlich orientiert ausgerichtet (Front, Heck), um den Verkabelungsaufwand zu minimieren und dienen als Gateways zur Sensor-Aktor-Ebene (z.B. Beschleunigungssensor, elektrische Lenkung), die wiederum über klassische Netzwerktechnologien wie beispielsweise CAN angebunden sind.

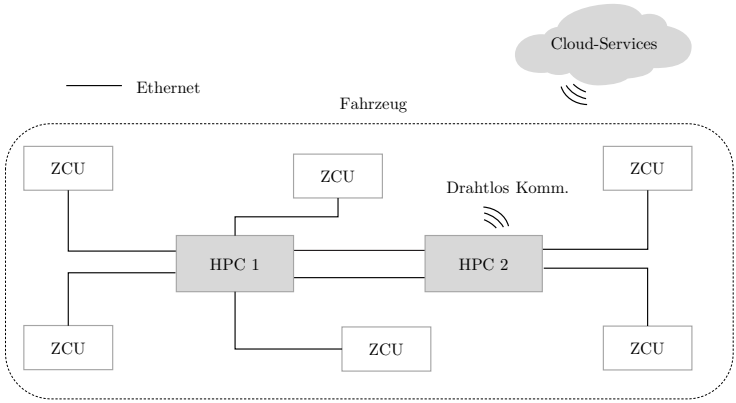


Abbildung 2.6: Grundsätzlicher Aufbau einer Zonen-orientierten E/E-Architektur (basierend auf [17]).

2.1.2 Signal- und service-orientierte Kommunikation

Die im ersten Kapitel beschriebene Transformation sowie die neuen Anforderungen in der Automobilbranche führen derzeit zu einer Änderung des etablierten Kommunikationsparadigmas der signal-orientierten Kommunikation. Bisher wird die Gesamtkommunikation eines Fahrzeugs innerhalb der Entwicklungsphase (Entwurf der technischen E/E-Architektur) definiert. Dabei werden alle benötigten Funktionssignale, die über das Netzwerk versendet bzw. empfangen werden, in einer Kommunikationsmatrix (K-Matrix) erfasst. Im nächsten Schritt erfolgt eine Zuweisung auf entsprechende Botschaften, denen Attribute wie beispielsweise das Sendeverhalten bzw. die Nutzdatenlänge hinzugefügt werden. Zusätzlich sind in Gateways statische Routingtabellen hinterlegt, die Botschaften auf verschiedene interne oder externe Fahrzeugnetzwerke umsetzen [45]. Dadurch kann das Netzwerk-Design als rein statisch betrachtet werden, da nach Abschluss der Entwicklungsphase nur noch sehr begrenzte Änderungen möglich sind.

Um eine spätere Update- und Upgradefähigkeit zu unterstützen und auch während der Entwicklungszeit einfachere Änderungen des Netzwerks zu ermöglichen, verläuft derzeit ein Übergang zu einem service-orientierten Kommu-

nikationsansatz [52]. Basierend auf dem SOA-Paradigma (s. Abbildung 2.7) sollen Kommunikationsverbindungen dynamisch aufgebaut sein, sodass auch Änderungen während des Betriebs möglich sind.

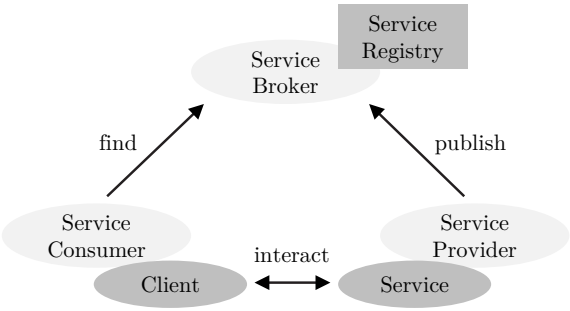


Abbildung 2.7: Übersicht der grundlegenden Funktionsweise des SOA-Paradigmas (basierend auf [53]).

Dazu werden Funktionen bzw. Informationen in möglichst kleine Services unterteilt, die zur Erfüllung einer umfangreicheren Funktionalität orchestriert werden können. Daneben erfolgt eine strikte Entkopplung zwischen Hard- und Software, sodass das Funktionsdesign keine statische Abhängigkeit zur darunterliegenden Architektur besitzt. Es muss beispielsweise einem Funktionsentwickler nicht bekannt sein, in welcher Botschaft ein bestimmter Sensorwert übertragen wird oder auf welchem Steuergerät eine benötigte Funktion integriert ist. Vielmehr ist eine Übersicht an verfügbaren Services in einem Fahrzeug notwendig, die in einem Service-Register (engl. service registry) hinterlegt sind. Benötigt eine Funktion als *Service-Konsument* (engl. service consumer) zur Laufzeit einen bestimmten Service, kann dieser dynamisch über das Service-Register aufgefunden und mit den dort gespeicherten Adressinformationen bei einem *Service-Anbieter* (engl. service provider) abonniert werden. Die Vermittlung zwischen dem Anbieter und Konsument übernimmt dabei eine *Middleware*. Daneben erleichtert dieses Konzept ein Hinzufügen von neuen Funktionen auch nach Abschluss der Entwicklung, da keine statischen Kommunikationsverbindungen in Form von Botschaften und Signalen der E/E-Architektur zugewiesen sind. Benötigte Informationen können dadurch entsprechend über verfügbare Services abonniert werden [12].

Definition 2.1.1 Service-orientierte Architektur

Eine Service-orientierte Architektur (SOA) ist nach dem BSI [54] über verschiedene Schlüsselmerkmale definierbar, da in der Literatur keine einheitliche Definition existiert: Standardisierte Schnittstellen, lose Kopplung, Funktionsabstraktion, Wiederverwendbarkeit, Service-Autonomie, Zustandslosigkeit des Service, Auffindbarkeit des Service, Orchestrierbarkeit des Service. Verschiedene Definitionen und detaillierte Erläuterungen sind in [55] gegeben.

Definition 2.1.2 Service-Anbieter

Als Service-Anbieter (engl. service provider) wird eine Rolle bezeichnet, die Services innerhalb einer SOA bereitstellt [56].

Definition 2.1.3 Service-Konsument

Als Service Konsument (engl. service consumer or service client) wird eine Rolle bezeichnet, die angebotene Services innerhalb einer SOA nutzt [56].

Definition 2.1.4 Middleware

Als Middleware wird eine Software bezeichnet, die im Hintergrund zwischen der Betriebssystemebene sowie der Anwendungsschicht läuft. Diese hat die Aufgabe zwischen verschiedenen Anwendungen bzw. zwischen Betriebssystem und Anwendungen zu vermitteln, um einen einfachen Datenaustausch zu ermöglichen [57].

Übergang zu service-orientierten Architekturen

Bisherige Architekturen bieten durch eine signal-orientierte Kommunikation (s. Abschnitt 2.1.2) nur eingeschränkte Möglichkeiten für Update- bzw. Upgrademöglichkeiten im Feld [15]. Um die Anpassungsfähigkeit zu erhöhen, wird neben strukturellen Veränderungen (s. Abschnitt 2.1.1) das Paradigma der

service-orientierten Kommunikation in E/E-Architekturen eingeführt. Aktuell existieren erste hybride Architekturentwürfe für die nächste E/E-Generation (s. Abbildung 2.8), die signal- und service-orientierte Kommunikation vereinen. Dabei erfolgt eine Unterteilung in drei unterschiedliche Architekturschichten (Sensor/Aktor-Ebene, Rechen- und Off-Board/Cloud-Ebene), die unterschiedliche Aufgaben erfüllen.

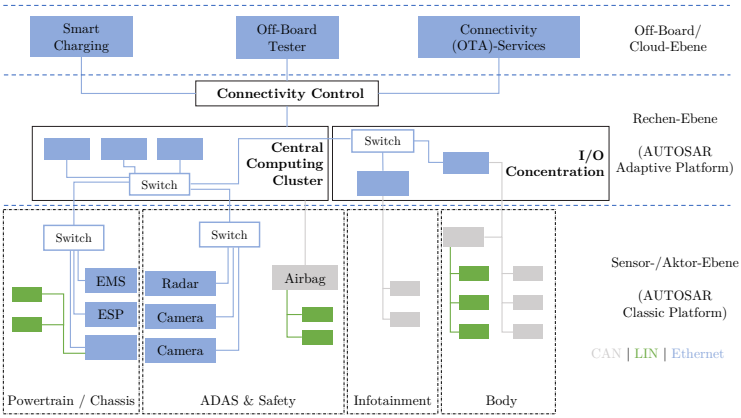


Abbildung 2.8: Aufbau hybriden E/E-Architektur, die aus signal- und service-orientierten Teilen besteht (adaptiert von [58]).

Die unterste Schicht ist dabei für Sensor-Aktor Wirkketten vorgesehen, in der echtzeitkritische Systeme wie z.B. die Motorsteuerung über klassische Bussysteme wie CAN vernetzt sind, die auf der AUTOSAR Classic Plattform [59] für signal-orientierte Kommunikation basieren. Die darüber liegende Ebene wird als Rechen-Ebene bezeichnet, da hier alle rechenintensiven Funktionen mit breitbandigen Ethernet-Verbindungen integriert sind. Für die Umsetzung ist durch AUTOSAR Adaptive [60] ein Standard verfügbar, der diese neuartigen Konzepte unterstützt und ausschließlich auf dem SOA-Paradigma basiert [61]. Beide Plattformen sind dabei in der Lage parallel in Fahrzeuge integriert zu werden, da eine Interoperabilität beider Standards gewährleistet wird. Die oberste Schicht bildet die Off-Board/Cloud-Ebene, in der Funktionen bzw. Geräte außerhalb der Systemgrenze des Fahrzeugs enthalten sind und über eine physikalische oder Luftschnittstelle eine Verbindung zum Fahrzeug herstellen.

2.1.3 Diagnose in Fahrzeugen

Die Fahrzeugdiagnose stellt eine spezielle Disziplin bei Fahrzeugen dar und hat über die vergangenen Jahre stets an Bedeutung gewonnen, um die elektronischen Systeme im Fehlerfall zuverlässig analysieren zu können [51]. Bereits in den 1990er Jahren hat die Automobilbranche diese Notwendigkeit erkannt und begann teilweise eigene aber auch standardisierte Protokolle zu entwickeln. Als eine der bekanntesten Standardisierungen im Bereich der Diagnose gilt der im Jahr 1988 in Teilen der USA eingeführte OBD Standard (s. Abschnitt A.1.1), der hauptsächlich durch gesetzliche Verordnungen zur Prüfung von abgasrelevanten Systemen vorangetrieben wurde. Eine Weiterentwicklung zum OBD-2 Standard [62] wurde für alle Fahrzeuge mit Otto-Motor innerhalb der ECU mit Erstzulassung 2001 (Diesel-Fahrzeuge ab Erstzulassung 2004) über die EU-Verordnung 98/69/EG [63] vorgeschrieben.

Darüber hinaus dienen Diagnosefunktionen nicht nur zur Analyse von Fehlern, sondern auch zum *Flashen* von Updates bzw. komplett neuen Funktionen. Zum Verständnis dieser Arbeit wird nachfolgend nur auf aktuelle Diagnoseprotokolle näher eingegangen.

Definition 2.1.5 Flashen

Im Bereich der Mikrocontroller-Technik wird der Begriff Flashen als Methode für die Programmierung der nicht flüchtigen Speicherbereiche bezeichnet, um mit elektrischen Löschimpulsen bestimmte Speicherinhalte zu löschen und danach einen neuen Programmcode zu übertragen [64].

2.2 Informationssicherheit

Dieses Kapitel gibt einen Überblick über die grundlegenden Definitionen, Prinzipien und Methoden der *IT-Sicherheit* bzw. *Cyber-Sicherheit*, da diese Grundlagen im Rahmen dieser Arbeit auf automotive Systeme übertragen und für Analysen und Ansätze genutzt werden. Wird beispielsweise ein Angriff auf ein IT-System eines Unternehmens oder dessen Produkt in der Öffentlichkeit bekannt, hat dieser neben dem Verlust der eigentlichen Informationen auch eine

Auswirkung auf dessen Reputation, die einen indirekten finanziellen Schaden verursacht. Bestehende oder potentielle Kunden würden durch diesen Vorfall ein gewisses Maß an Vertrauen verlieren.

Definition 2.2.1 Informationssicherheit

Die Informationssicherheit (auch als Cybersicherheit bezeichnet) gemäß ISO 27001 [65] umfasst den Schutz von Unternehmenswerten (engl. Assets) wie beispielsweise Informationen[1] oder immaterielle Werte wie die Reputation, um damit eine Beeinflussung des Geschäftsbetriebs zu verhindern [66].

Definition 2.2.2 Cyber-Sicherheit

Der Begriff Cyber-Sicherheit umfasst alle Aspekte der Sicherheit für die Informations- und Kommunikationstechnik, indem die klassische IT-Sicherheit im gesamten Cyber-Raum betrachtet wird. Dieser umfasst alle mit dem Internet verbundenen oder ähnlichen Netzwerke, die über eine Informationstechnologie vernetzt sind. Der Fokus liegt dabei auf der Kommunikation, Anwendungen, Prozesse sowie den zu verarbeiteten Informationen [67].

Definition 2.2.3 Information

Die Information repräsentiert ein Datum und ergibt sich durch eine zugewiesene Interpretationsvorschrift. So kann beispielsweise ein Datenobjekt (z.B. Datei, Datenbankeintrag) einen numerischen Wert besitzen, der abhängig von der geltenden Interpretationsvorschrift, den aktuellen Kilometerstand des Fahrzeugs oder die aktuelle Geschwindigkeit repräsentiert [68].

2.2.1 Sicherheit und Risikomanagement

Für den Schutz von Informationen ist es notwendig, bestimmte Eigenschaften zu definieren, die beispielsweise in einem Unternehmen, Produkt oder einer

Nachricht durch ein oder mehrere Schutzmaßnahmen gewährleistet werden sollen. In der Informationssicherheit sind dafür die folgenden fundamentalen Sicherheits-Eigenschaften existent [69], [23], [70]:

Definition 2.2.4 Vertraulichkeit

Die Vertraulichkeit (engl. confidentiality) gewährleistet den Schutz gegen unberechtigte Kenntnisnahme von Informationen (Geheimhaltung) an jedem Punkt der Datenverarbeitung. Wird beispielsweise eine Nachricht mit der Security-Eigenschaft der Vertraulichkeit von einem System A über ein Netzwerk zu einem System B versendet, so muss diese von ihrem Ursprung bis zum Ziel durchgehend gewährleistet sein. Zur Umsetzung dieser Eigenschaft werden u.a. kryptografische Verfahren wie Verschlüsselungstechniken eingesetzt. Außerdem sind Maßnahmen zur Zuweisung und Kontrolle von Zugriffsberechtigungen notwendig, um zu verhindern, dass unautorisierte Subjekte (s. Abschnitt 2.2.2) an Informationen gelangen.

Definition 2.2.5 Integrität

Die Integrität (engl. integrity) umfasst einen Schutz gegen ein unautorisiertes Verändern von Informationen auf einem System oder einer Nachricht in einem Netzwerk. Ist bei der Übertragung von derartigen Nachrichten ein Integritätsschutz vorhanden, kann der jeweilige Empfänger prüfen, ob die Nachricht möglicherweise auf dem Transportweg unrechtmäßig manipuliert bzw. verändert wurde. Zur Sicherstellung der Nachrichtenintegrität werden u.a. kryptografische Hashfunktionen eingesetzt. Hingegen kommen für den Integritätsschutz von Informationen auf einem IT-System z.B. Zugriffskontrollen zum Einsatz.

Definition 2.2.6 Authentizität

Die Authentizität (engl. authenticity) ist als Nachrichtenauthentizität sowie Authentizität einer Quelle definierbar. Die Eigenschaft garantiert hierbei die Echtheit einer Nachricht oder einer Quelle, dass es sich tatsächlich um die angegebene Identität handelt oder die versendete Nachricht von der angegebenen Quellenidentität stammt.

Definition 2.2.7 Verfügbarkeit

Die Verfügbarkeit (engl. availability) garantiert die zuverlässige und erwartete Funktionsweise eines IT-Systems und der bereitgestellten Funktionen zu jedem Zeitpunkt. Diese wird auch als Wahrscheinlichkeit über einen definierten Zeitraum angegeben, für diesen das System spezifizierte Anforderungen erfüllen muss. Es ist dadurch eine Kombination von verschiedenen physikalischen, administrativen oder technischen Schutzmaßnahmen erforderlich, um diese Eigenschaft zu erfüllen.

Grundlegende Zusammenhänge

Um die Zusammenhänge (s. Abbildung 2.9) der Informationssicherheit einheitlich beschreiben zu können haben sich die folgenden Definitionen gemäß [23] etabliert:

Definition 2.2.8 Schwachstelle

Eine Schwachstelle (engl. vulnerability) ist eine Sicherheitslücke, die auf einer *Schwäche* (engl. weakness) in der Konzeption, Implementierung, Konfiguration, dem Betrieb einer Organisation bzw. Systems basiert. So kann beispielsweise ein Dienst auf einem *Server* eine unsichere Implementierung beinhalten oder auch ein offener Port in einer Firewall zu einer möglichen Schwachstelle führen.

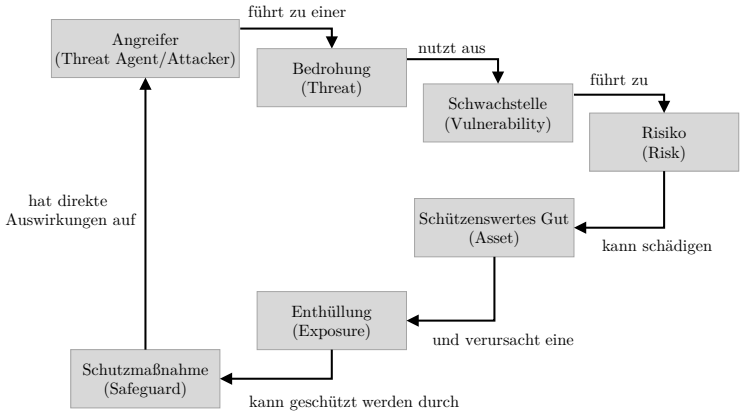


Abbildung 2.9: Zusammenhang verschiedener Security-Definitionen (basierend auf [23]).

Definition 2.2.9 Server

Als Server wird auf Hardware-Ebene ein physisches Gerät (z.B. Computer, ECU) definiert, das Ressourcen über ein Netzwerk zur Verfügung stellt. Alternativ wird ein Hardware-Server auch als *Host* bezeichnet. Auf den Geräten sind neben dem Betriebssystem ein oder mehrere softwarebasierte Server implementiert. Auf der Software-Ebene wird ein Server als Anwendung definiert, die Dienste für Clients lokal oder in einem Netzwerk anbietet [71]. (s. auch Definitionen 2.1.2 u. 2.1.3.)

Definition 2.2.10 Bedrohung

Eine Bedrohung (engl. threats) ist eine Gefahr, die das Potential besitzt einen konkreten Schaden an einer Organisation oder System verursachen zu können.

Definition 2.2.11 Angreifer

Der Angreifer (engl. threat agent/attacker) repräsentiert eine einzelne Person, Gruppe oder auch allgemein ein IT-System, der versucht einen Vorteil durch die Ausnutzung (engl. exploitation) einer Schwachstelle zu erlangen (Eine Klassifikation von Angreifergruppen und deren Motivation ist in [72] enthalten.). Dieses Vorgehen wird als *Angriff* bezeichnet. Dadurch entsteht ein Zusammenhang zwischen einem Angreifer, einer Bedrohung sowie einer Schwachstelle. Durch die Intention des Angreifers eine vorsätzliche Handlung zu begehen entsteht eine Bedrohung, die über eine mögliche Schwachstelle ausgenutzt werden kann.

Definition 2.2.12 Risiko

Das Risiko (engl. risk) entspricht der Annahme einer Wahrscheinlichkeit für die Ausnutzung einer Schwachstelle durch einen Angreifer sowie die daraus resultierende Auswirkung auf die Organisation.

Definition 2.2.13 Schützenswertes Gut

Die Auswirkung bezieht sich auf die Gefährdung von schützenswerten Güter (engl. Asset) einer Organisation wie beispielsweise die Vertraulichkeit von personenbezogen Daten (engl. personally identifying information) [23], [73]. Dadurch kann es Angreifern gelingen, Informationen für weitere Aktivitäten zu sammeln (engl. information gathering) oder aber gezielt unbemerkt Aktivitäten durchzuführen.

Definition 2.2.14 Schutzmaßnahme

Um potentielle Risiken für Angriffe zu minimieren, werden Schutzmaßnahmen (engl. safeguards/countermeasures) eingesetzt, um Angreifern das Ausnutzen einer Schwachstelle zu erschweren oder diese durch die Integration einer Maßnahme bereits bei der Entwicklung zu vermeiden.

Bedrohungsanalysen

In der Informationssicherheit gibt es unterschiedliche Bedrohungen (engl. threats) bzw. Angriffstypen, die bei IT-Systemen auftreten können [74]. Um mögliche Bedrohungen bereits in der Entwicklungsphase zu finden, werden Bedrohungsanalysen durchgeführt [68]. Darüber hinaus werden für gefundene Bedrohungen entsprechende Risikowerte berechnet, die ein Ergebnis aus Eintrittswahrscheinlichkeit × Schadenshöhe repräsentieren. Die Analyseergebnisse dienen weiter für die Auswahl geeigneter Schutzmaßnahmen zur Abwehr von Angriffen, die über eine Bedrohung eintreten können. Um ein strukturiertes Vorgehen zu ermöglichen haben sich dafür verschiedene Frameworks [75], [76], [77] und Methoden[78], [79], [80], [81] etabliert. Davon bietet die Microsoft STRIDE Methode auch eine in der Industrie, Wissenschaft und Security-Community verbreitete und anerkannte Taxonomie zur Klassifizierung von Angriffen und Bedrohungen [82]. Das Akronym STRIDE steht dabei für die Angriffsklassen *Spoofing*, *Tampering*, *Repudiation*, *Information Disclosure*, *Denial of Service* und *Elevation of Privilege*) in die Angriffe auf die Informationssicherheit allgemein eingeteilt werden können (s. Tabelle 2.1). Zusätzlich ist jeder Bedrohung eine Security-Eigenschaft zugeordnet, die dadurch gefährdet wird.

Insider-Bedrohungen

Insider-Bedrohungen stellen derzeit eine der größten Herausforderungen im Bereich der Cyber-Sicherheit dar [83], [84]. Auf der Basis von einer Umfrage [85] von 204 Unternehmen aus 13 unterschiedlichen Branchen mit mehr als 1000 Beschäftigten ergab sich eine Steigerung der registrierten Angriffe auf die Informationssicherheit durch *Insider* von 3200 im Jahr 2018 auf 4700 im Jahr 2020.

Insider können nach Cole et al. [86] in vier unterschiedliche Kategorien eingeordnet werden. *Pure Insiders* enthalten rechtmäßige Arbeitnehmer, die eine notwendige Menge an Berechtigungen für die Ausführung ihrer Arbeit besitzen (Zugang zum Gebäude, Zugangsdaten zur Nutzung von definierten Netzwerkdiensten). Die Klasse *Inside Associates* umfasst Personen von Fremdfirmen (Zulieferer, Sicherheitsdienste, Reinigungspersonal), die eine bestimmte Menge an internen Zugängen zu Gebäuden oder auch IT-Infrastruktur besitzen. Da-

Tabelle 2.1: Übersicht der STRIDE-Kategorien (basierend auf [78]).

Bedrohung	**Security-Eigenschaft**	**Bedrohungs-Definition**	**Beispiele**
Spoofing	Authentifizierung	Verfälschung der eigenen Identität	Ausgabe als eine andere Person oder Netzwerkadresse
Tampering	Integrität	Manipulation von Informationen auf einem Datenträger oder im Netzwerk	Modifizieren, Hinzufügen oder Entfernen von Netzwerkpaketen
Repudiation	Nichtabstreitbarkeit	Abstreiten einer durchgeführten Aktion (gesendete Nachricht, ausgeführte Funktion)	„Habe nicht das Netzwerkpaket xy gesendet“ oder „Habe nicht den Kauf xy getätigt“
Information Disclosure	Geheimhaltung	Erlangung von Informationen für nicht autorisierte Personen	Erlaubt dem Angreifer den Zugriff auf Dateien, E-Mails oder Datenbanken
Denial of Service	Verfügbarkeit	Überlastung von Ressourcen, die zur Bereitstellung von Diensten benötigt werden	Ausführung von einer Vielzahl an parallelen Netzwerkanfragen, die normalerweise nicht auftreten
Elevation of Privilege	Autorisierung	Ausführung von Aktionen, für die jemand nicht autorisiert ist	Ermöglicht einem Angreifer das Ausführen von Funktionen mit Administrator-Rechten

neben stellen *Inside Affiliates* Personen wie Familienangehörige oder Freunde von Arbeitnehmern dar, die offiziell keine Berechtigung besitzen, jedoch die Zugangsdaten des zugehörigen Arbeitnehmers durch Diebstahl erlangt haben. Als vierte Klasse stellen *Outside Affiliates* Personen dar, die in keiner Verbindung zu einer Organisation stehen und dadurch offiziell keine Zugänge besitzen, jedoch über verschiedene Angriffswege z.B. durch *Social Engineering* oder eine schwach gesicherte Wireless Local Area Network (WLAN)-Verbindung, Zugang zur internen Infrastruktur erlangt haben. Eine weiterführende Detaillierung sowie Taxonomie von Ereignissen durch Insider-Bedrohungen ist in [83] gegeben.

Definition 2.2.15 Insider-Bedrohung

Pfleeger et al. bezeichnen Insider-Bedrohungen als Insider-Aktionen, die Daten, Prozesse oder Ressourcen einer Organisation in einer störenden oder unerwünschten Weise gefährdet [87].

Definition 2.2.16 Insider

Pfleeger et al. bezeichnen Insider als Personen, die einen legitimierten Zugang für Computer und Netzwerke einer Organisation besitzen [87]. Brackney and Anderson bezeichnen Insider als eine bereits vertrauenswürdige Person mit Zugang zu sensiblen Informationen und Informationssystemen [88].

Definition 2.2.17 Social Engineering

Als Social-Engineering Angriffe werden Aktivitäten bezeichnet, die andere Person durch ein glaubwürdigen Vorwand dazu bewegen, sensible Informationen heraus zu geben, die anschließend für einen Angriff verwendet werden [23].

Definition 2.2.18 WLAN

Als WLAN wird eine Technologie bezeichnet, die es ermöglicht eine Kommunikation zwischen verschiedenen Netzwerkteilnehmern über eine Luftschnittstelle herzustellen. Mit dem Begriff wird meist der IEEE Standard 802.11[89] assoziiert.

Im Vergleich zu Outsider-Bedrohungen, die nach der STRIDE Taxonomie klassifizierbar sind, werden Bedrohungen von Insidern anhand der Absichten, Motive und Handlungen einer Person unterschieden, da diese nicht zwingend eine direkte Verletzung der Security-Eigenschaften beinhalten. So verletzt beispielsweise ein Insider mit legitimierten Zugangsdaten eines Benutzers keine Security-Eigenschaften im Netzwerk. Jedoch im Fall eines Datendiebstahls von einem Server die Vertraulichkeit. Nach Prabhu et al. [90] lassen sich diese in vier verschiedene Arten klassifizieren:

- **Versehentlicher (engl. accidental) Insider:** Beinhaltet eine Person, die keine böswillige Absicht aufweist, jedoch durch ein Fehlverhalten der zugehörigen Organisation einen Schaden zufügt oder das potentielle Risiko dafür erhöht.
- **Nachlässiger (engl. negligent) Insider:** Umfasst Personen, die keine böswilligen Handlungen ausführen möchten, aber durch ihr passives Risikoverhalten einen Schaden verursachen oder die Eintrittswahrscheinlichkeit dafür erhöhen.
- **Schelmischer (engl. mischievous) Insider:** Diese Art hat weder die Absicht, gegen Sicherheitsrichtlinien zu verstoßen, noch ein Motiv, dies zu tun. Jedoch ist durch sein risikobehaftetes Verhalten die Verursachung eines Schadens innerhalb der zugehörigen Organisation möglich.
- **Böswilliger (engl. malicious) Insider:** Dieser verfolgt durch seine Handlungen eine böswillige Absicht, um einen direkten oder indirekten Schaden bei der zugehörigen Organisation zu verursachen.

Schutzmaßnahmen

Die Maßnahmen zur Minimierung des Risikos in der Informationssicherheit können in drei verschiedene Kategorien eingeteilt werden [23]:

- **Administrative Maßnahmen:** Darin werden Prozessaktivitäten definiert, die sich beispielsweise mit dem Risikomanagement beschäftigen. Darüber hinaus sind in dieser Kategorie auch Security-Schulungen (z.B. Informationen zur Passwortsicherheit) von Mitarbeitern einzuordnen, um eine Sensibilisierung für diese Thematik zu erreichen.
- **Technische Maßnahmen:** Diese umfassen Soft- und Hardwarekomponenten wie beispielsweise IDS (s. Abschnitt 2.2.4) oder Firewalls (s. Abschnitt 2.2.3) aber auch kryptografische Verfahren (s. Abschnitte A.1.2 u. A.1.3), die zur Verschlüsselung oder digitalen Signatur von Informationen verwendet werden.
- **Physikalische Maßnahmen:** Darunter werden Aktivitäten definiert, die das Personal und Organisation sowie die Kommunikation und Infrastruktur betreffen. Als Beispiel kann hier ein Server dienen, der softwaretech-

nisch eine hohe Sicherheit aufweist jedoch an einem leicht zugänglichen Ort betrieben wird. So besteht die Gefahr, dass unbefugte Personen sich physikalischen Zugriff zum System verschaffen können und weitere Bedrohungen wirksam werden.

Für den Schutz von Unternehmenswerten gegen böswillige Angreifer, die von außerhalb versuchen in ein System einzudringen, existiert der *defense-in-depth*-Ansatz. In Abbildung 2.10 sind beispielhaft verschiedene technische Schutzmaßnahmen den Schichten zugeordnet, um ein schützenswertes Gut (Personal Computer) abzusichern.

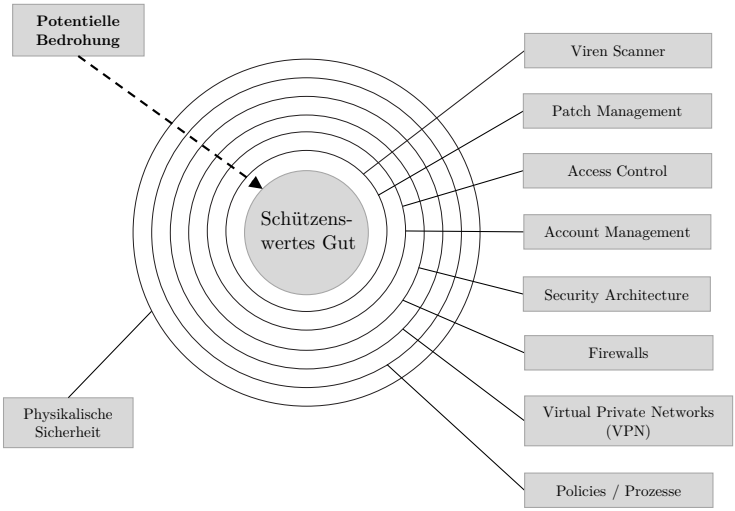


Abbildung 2.10: Anordnung verschiedener technischer Schutzmaßnahmen basierend auf dem Defense-in-Depth Ansatz (basierend auf [23]).

Dadurch wird versucht, die Wahrscheinlichkeit für einen erfolgreichen Angriff durch die verschiedenen Maßnahmen zu minimieren. Folglich müsste ein Angreifer zum Erreichen eines Unternehmenswerts mehrere Maßnahmen überwinden, um sein Ziel zu erreichen [23]. Die genaue Anzahl der Schichten sowie der Schutzmaßnahmen ist dabei von den zuvor ermittelten Bedrohungen (s. Abschnitt 2.2.1) und dem Wert des zu schützenden Guts abhängig. Nach-

folgend werden verschiedene Schutzmaßnahmen näher erläutert, die für den Hauptteil der Arbeit von Relevanz sind.

Definition 2.2.19 Defense-in-Depth

Als Defense-in-Depth wird in der Informationssicherheit ein Konzept bezeichnet, dass durch die Anordnung verschiedener Schutzmaßnahmen ein mehrschichtiges Verteidigungssystem gegen IT-Angriffe beinhaltet.

Sicherheitsrichtlinie (Policy)

Eine Sicherheitsrichtlinie (engl. Policy) dient in Unternehmen und Behörden zur Umsetzung von definierten Richtlinien [54], die geschäftliche oder rechtliche Aspekte in Bezug auf die Informationssicherheit enthalten und durch Schutzmaßnahmen umgesetzt werden müssen. Dabei können Richtlinien auf unterschiedlichen Abstraktionsebenen und in verschiedenen Detaillierungsgraden existieren (s. Abbildung 2.11). Auf der obersten Ebene werden Policies in natürlicher Sprache spezifiziert und enthalten unternehmensspezifische- oder gesetzliche Vorgaben, die einzuhalten sind. Davon abgeleitet werden prozessspezifische Richtlinien, die allgemeine Anforderungen enthalten, ausgearbeitet. Danach erfolgt eine Kategorisierung der Richtlinien in Bezug auf die darin enthaltenen Security-Eigenschaften wie beispielsweise die Integrität von versendeten Nachrichten. Es kann hingegen auch die Anforderung bestehen, dass Ressourcen auf einem System durch die Kontrolle und Durchsetzung von Berechtigungen in Form einer Zugriffskontrolle abgesichert werden müssen. Für die Gewährleistung der Security-Eigenschaften werden danach auf unterster Ebene Informationstechnik (IT)-spezifische Vorgaben für zu verwendende Sicherheitsprotokolle oder Standards in maschinenlesbarer Sprache wie Extensible Markup Language (XML) [91] beschrieben, die IT-Komponenten umsetzen. Eine spezifische IT-Komponente könnte dabei ein Attribute-based Access Control (ABAC)-Modul (s. Abschnitt 2.2.2) darstellen, das eine hinterlegte Policy durchsetzt.

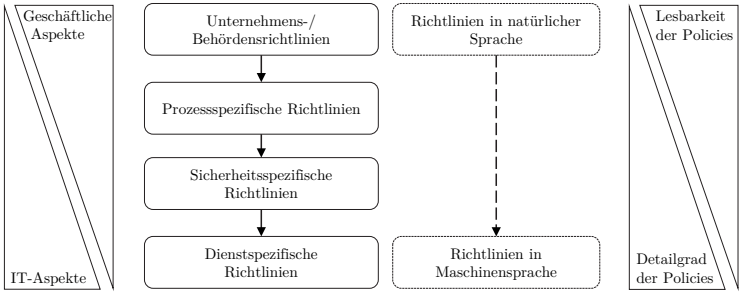


Abbildung 2.11: Übersicht verschiedener Abstraktionsebenen von Sicherheitsrichtlinien (basierend auf [54]).

2.2.2 Modelle & Techniken der Zugriffskontrolle

Definition 2.2.20 Zugriffskontrolle

Die Zugriffskontrolle (engl. access control) verhindert unautorisierte Ressourcenzugriffen und gewährleistet damit die Sicherheitseigenschaft *Autorisierung* (s. Abschnitt 2.2.1). Unter diesem Begriff werden in dieser Arbeit Modelle und Techniken zugeordnet, die eine Zugriffskontrolle auf Anwendungsebene umsetzen.

Die Zugriffskontrolle stellt dadurch eine regelkonforme Nutzung der Ressourcen sicher [92]. Die existenten Zugriffskontrollmodelle kontrollieren allgemein, welches Subjekt (z.B. ein Benutzer, Gerät, Funktion bzw. Service) in einem System auf eine bestimmte Ressource/Objekt (z.B. Datei, Verzeichnis, Funktion bzw. Service) zugreifen darf. Der Ablauf bis zu einem erfolgreichen Zugriff lässt sich in vier Teilschritten beschreiben (s. Abbildung 2.12).

Zunächst authentisiert sich ein Subjekt (Initiator einer Handlung) gegenüber dem IT-System. Danach erfolgt die Überprüfung der Identität durch eine Authentifizierung. Ist diese erfolgreich wird eine Autorisierung durch das eingesetzte Zugriffskontrollmodell durchgeführt, das dem jeweiligen Subjekt bestimmte Zugriffsberechtigungen auf der Basis von spezifizierten Sicherheitsrichtlinien gewährt (lesen, schreiben oder ausführen). Die Kontrollmodelle sind

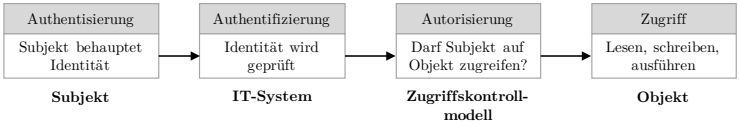


Abbildung 2.12: Schematischer Ablauf einer Zugriffskontrolle (basierend auf [93]).

dabei auf verschiedenen Systemebenen integrierbar (z.B. Netzwerk, Middleware oder Anwendung). Dabei lassen sich die verschiedenen Methoden der Zugriffskontrolle auf vier verschiedene Modelle (DAC, MAC, RBAC, ABAC) abbilden:

Benutzerbestimmbare Zugriffskontrolle (DAC)

Bei der benutzerbestimmbaren Zugriffskontrolle (engl. Discretionary Access Control (DAC)) ist für die Vergabe von möglichen Berechtigungen für eine Ressource ausschließlich der Eigentümer verantwortlich [68]. Für die Zuweisung und Verwaltung der Berechtigungen jedes Subjekts werden der jeweiligen Ressource eine Zugriffskontrollliste (engl. Access Control List (ACL)) zugeordnet (s. Abschnitt 2.2.2), in der beispielsweise Lese- und Schreibrechte festgelegt sind [94].

Regelbasierte Zugriffskontrolle (MAC)

Die regelbasierte Zugriffskontrolle (engl. Mandatory Access Control (MAC)) stellt ein Zugriffsmodell für die strikte Kontrolle und Durchsetzung von Berechtigungen dar. Im Vergleich zu DAC werden Berechtigungen auf Ressourcen nicht auf Subjekte abgebildet, sondern es erfolgt eine Zuweisung von Sicherheits-Kennzeichnungen (engl. security labels) [23]. Die Kennzeichnungen können beispielsweise verschiedene Abstufungen (öffentlich, vertraulich, geheim oder streng geheim) beinhalten. Gleichzeitig werden diese Kennzeichnungen auch den beteiligten Subjekten einer Organisation zugewiesen. Möchte ein Subjekt auf eine Ressource zugreifen, prüft die zentral implementierte Zugriffskontrolle (z.B. in einem Betriebssystem), ob beide die entsprechende Kennzeichnung bzw. Sicherheitsstufe besitzen.

Rollenbasierte Zugriffskontrolle (RBAC)

Bei der rollenbasierten Zugriffskontrolle (engl. Role-based Access Control (RBAC)) werden die Berechtigungen als Menge bestimmten Rollen zugeordnet. Die Rollen können dabei beispielsweise unterschiedlichen Abteilungen (Einkauf, Versand, Vorstand) eines Unternehmens entsprechen. Dadurch entfällt die einzelne Zuweisung an Berechtigungen für jedes Subjekt bzw. Objekt. Die Kontrolle, Durchsetzung und Verwaltung von Berechtigungen der RBAC-Zugriffskontrolle erfolgt dabei zentral (z.B. innerhalb eines Betriebssystems). Wird ein neuer Mitarbeiter eingestellt, muss dieser nur noch einer Rolle zugewiesen werden und erhält damit die hinterlegte Menge an Berechtigungen. Darüber hinaus reduziert dies den Verwaltungsaufwand des verantwortlichen IT-Mitarbeiters, da Änderungen an einer zentralen Stelle möglich sind [23].

Attributbasierte Zugriffskontrolle (ABAC)

Der Grundgedanke des ABAC Ansatzes besteht darin, Zugriffe auf Ressourcen von Dienstnutzern über Attribute zu erlauben oder zu verweigern. Dazu wird zwischen den folgenden Attributen unterschieden:

- Nutzerattribute: beschreiben den Dienstnutzer näher z.B. Alter, Abteilung, Titel.
- Aktionsattribute: beschreiben die Aktion, die auf die Ressource ausgeübt werden soll z.B. lesen, schreiben, löschen.
- Ressourcenattribute: beschreiben das Objekt, auf das zugegriffen wird z.B. Bankkonto, Dokument.
- Umgebungsattribute: Attribute, die sich mit Zeit, Ort oder dynamischen Aspekten des Zugriffskontrollszenarios befassen z.B. Zugriffserlaubnis nur zu bestimmten Zeiten.

Durch die Anwendung von Attributen wird die starre Kopplung zwischen Benutzern und Rollen sowie Rollen und Berechtigungen (vgl. RBAC) flexibler gestaltet [94],[92],[95],[96]. Um den dynamischen Anforderungen der verteilten Systeme gerecht zu werden, wird eine dynamische Zugriffskontrollentscheidung benötigt. Dazu werden die Nutzerattribute zur Laufzeit ausgewertet und

mit einer hinterlegten Policy (s. Abschnitt 2.2.1) abgeglichen. Ein prinzipieller Ablauf dieser Zugriffskontrolle ist die Abbildung 2.13 dargestellt. Möchte ein Subjekt auf ein Objekt zugreifen (1) prüft das ABAC-Modul unter Einbezug der aktuellen Objekt-, Aktions- und Umgebungsattribute, ob der Zugriff auf Basis der hinterlegten Policy erfolgen darf (2) und setzt die Zugriffsentscheidung (3) (erlauben, blockieren) entsprechend durch. Zur Umsetzung und Implementierung wird u.a. der eXtensible Access Control Markup Language (XACML)-Standard [97] empfohlen (s. Abschnitt *XACML*). Dieser unterstützt die Integration von Subjekt- und Objektattributen in Zugriffsrichtlinien, die die Basis des ABAC-Ansatzes darstellen. Die XACML Architektur und die damit verbundenen Module legen die Autorisierungsinfrastruktur fest, die für ein ABAC-Modell notwendig sind.

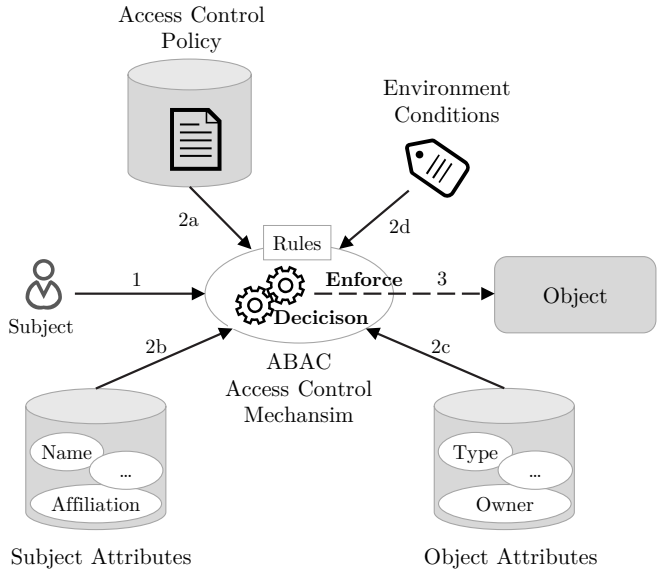


Abbildung 2.13: Prinzipieller Ablauf einer ABAC-basierten Zugriffskontrolle (basierend auf [98]).

eXtensible Access Control Markup Language (XACML)

Die XACML ist eine im Konsortium der Organization for the Advancement of Structured Information Standards (OASIS) definierte Policy-Sprache, die die Zugriffskontrolle auf Ressourcen standardisiert. Sie beinhaltet eine attributsbasierte Zugriffssteuerungsrichtliniensprache, eine Architektur sowie ein Verarbeitungsmodell, das beschreibt, wie Zugriffsanforderungen basierend auf in Policies definierten Regeln für eine attributbasierte Autorisierung erfolgen müssen. Der Vorteil dieser Policy-Sprache ist die Übertragbarkeit von Zugriffsrechten sowie eine feingranulare Zugriffssteuerung [92], [97], [54].

Der XACML Standard definiert die drei grundlegenden Policy-Elemente *Rule*, *Policy* und *Policy-Set*. Ein Policy-Set besteht aus einer oder mehreren Policies, kann aber auch auf weitere Policy-Sets referenzieren. Eine Policy besteht aus einer Vielzahl an Regeln (rules), einer Kennung für den *Rule-Combining Algorithm* und optional aus einer Reihe von Verpflichtungen (Obligations). Jedes der drei Hauptelemente bezieht sich auf genau ein *Target* [97], [99]].

- Target: Das Target dient der Entscheidung, ob ein Policy-Set, eine Policy oder eine Rule für die Anfrage relevant ist und wird dementsprechend ausgewertet. Dazu erhalten die Targets die Attribute aus den vier Kategorien *Subject*, *Resource*, *Action* und *Environment*. In den Policies ist es hingegen nicht verpflichtend in jeder Kategorie Werte für die Attribute anzugeben. Die Werte des Targets werden mit den jeweiligen Werten der Attribute aus der Anfrage verglichen. Gibt es eine Übereinstimmung, so wird die Policy als relevant erachtet und ausgewertet. Beispiel: "Die folgenden Regeln gelten, wenn ein Bankkunde (Subjekt) auf seinen Bausparplan (Ressource) nach fünf Jahren (Umgebung) lesend zugreift (Aktion)". Stimmen die Attributswerte der Policy mit denen des Targets überein, so wird die Policy als relevant angesehen und näher betrachtet.
- Rule: Jede Rule besteht aus einem Ziel (Target), einem Effekt (Effect) und einer Bedingung (Condition). Das *Target* beurteilt, ob eine Regel für die Anfrage relevant ist. Der *Effect* trifft bei Erfüllung der Regel die Entscheidung *erlauben (engl. permit)*, sonst *verweigern (engl. deny)*. Eine *Condition* trifft eine Aussage über Attribute bei der Bewertung durch *wahr (engl. true)*, *unwahr (engl. false)* oder *nicht ermittelbar (engl. indeterminate)*.

- Combining Algorithm: Bei einer Autorisierungsentscheidung ist es möglich, dass mehrere *Rules* einer Policy das gleiche *Target* haben. Damit es zwischen den *Rules* nicht zu Konflikten kommt, kombinieren die *Rule Combinig Algorithms* die unterschiedlichen Resultate der Rules zu einem Gesamtresultat. Selbiges gilt für die Policy-Sets. Hierfür gibt es einen *Policy Combining Algorithm*, der die Ergebnisse der einzelnen Policies zu einem Gesamtergebnis kombiniert.
- Obligations: Die *Obligations* führen zu einer feingranularen Zugriffssteuerung. Sie sind in einer Rule, Policy oder einem Policy-Set festgelegt. Bei der Durchsetzung einer Autorisierungsentscheidung muss der PEP die festgelegten Aktionen durchführen, z.B. einen verschlüsselten Ressourcenzugriff.

Um den Anforderungen verteilter Systeme gerecht zu werden, wird in der XACML Architektur zwischen Policy Administration (Verwaltung), Policy Enforcement (Durchsetzung) und Policy Evaluation (Auswertung oder Abgleich) unterschieden. Dazu wurden in XACML die folgenden Funktionsmodule definiert [92]:

- Policy Enforcement Point (PEP)
- Policy Decision Point (PDP)
- Policy Information Point (PIP)
- Policy Administration Point (PAP)

Die Funktionsmodule und sowie der Datenfluss einer spezifizierten XACML-Architektur sind in Abbildung 2.14 dargestellt.

Die Hauptaufgabe des PAP ist die Speicherung und Verwaltung der Zugriffsrichtlinien. Er erstellt die Policies (1), legt den Policy Typ fest (Integrität, Autorisierung, Authentifizierung) und stellt sie dem PDP zur Verfügung. Der PEP erhält eine Zugriffsanfrage vom Dienstnutzer (2). Anschließend sendet der PEP die Zugriffsanfrage an den Context Handler (3). In der Anfrage können zusätzliche Attribute der Subjekte, der Ressource, der Aktion sowie der Umgebung enthalten sein. Der Context Handler erstellt einen XACML-Anforderungskontext, fügt optional Attribute hinzu und sendet ihn an den PDP (4). Jegliche zusätzliche Subjekt-, Ressourcen-, Aktions- und Umgebungsattribute werden vom

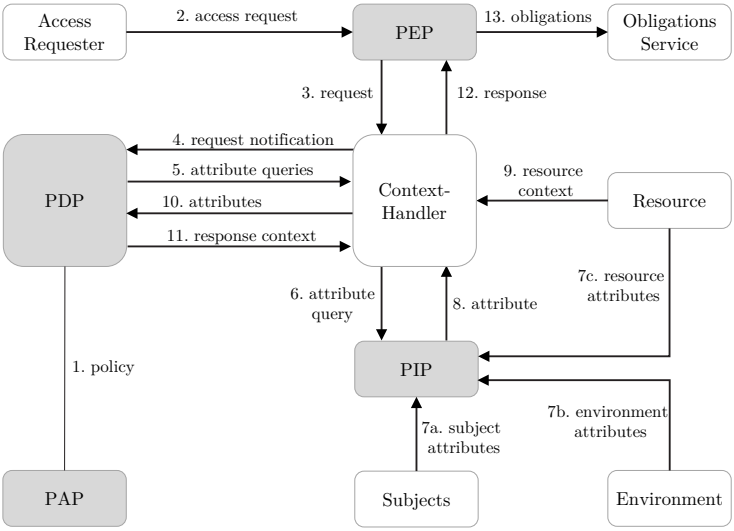


Abbildung 2.14: Darstellung der XACML-Architektur mit verschiedenen Funktionsmodulen für die Zugriffskontrolle (basierend auf [97]).

PDP beim Context Handler angefordert (5). Im darauffolgenden Schritt fordert der Context Handler die benötigten Attribute beim PIP an (6). Die Aufgabe des PIP besteht in der Verwaltung der Attribute. Er erhält die angeforderten Attribute und sendet sie an den Context Handler (7,8). Optional bindet der Context Handler die Ressource in den Anforderungskontext mit ein (9). Anschließend sendet der Context Handler die geforderten Attribute (und optional auch die Ressource) an den PDP, der die Policy direkt auswertet (10). Der PDP sendet den Antwortkontext einschließlich der Autorisierungsentscheidung an den Context Handler zurück (11). Der Conetxt Handler übersetzt den Antwortkontext des PDP in das systemspezifische Antwortformat des PEP und sendet die Antwort an den PEP (12). Falls der Zugriff auf die Ressource gewährt wird, gibt der PEP den Zugriff auf die Ressource frei, andernfalls wird der Zugriff untersagt. Zusätzlich ist der PEP dafür verantwortlich, dass die Obligations erfüllt werden.

Techniken der Zugriffskontrolle

Im Bereich der Zugriffskontrolle haben sich zwei verschiedene Techniken zur Zuweisung von Berechtigungen auf Subjekte bzw. Objekte etabliert. Beide Varianten werden über eine Zugriffskontrollmatrix (engl. access control matrix) definiert (s. Tabelle 2.2) [23]. Die Berechtigungen werden damit entweder über Fähigkeiten (engl. capabilities) auf Subjekte oder über eine Zugriffskontrollliste (ACL) auf Objekte abgebildet.

Fähigkeiten (Capabilities): Diese Art von Tabelle definiert eine Menge von Berechtigungen für ein bestimmtes Subjekt [100]. In Tabelle 2.2 sind die Fähigkeiten jedes Subjekts horizontal für die jeweiligen drei Objekte (Dateien 1 bis 3) spezifiziert. Eine Fähigkeit kann dabei über verschiedene Formate abgebildet werden (Token, Schlüssel, Ticket). Wenn im Betrieb ein Subjekt auf ein Objekt zugreifen möchte, prüft das darunterliegende System, ob das Subjekt die notwendigen Berechtigungen besitzt, welche zuvor übermittelt werden.

Zugriffskontrollliste (ACL): Diese Zugriffslisten werden vorwiegend in Betriebssystemen oder Netzwerkgeräten (z.B. Switches oder Router) verwendet. Dabei wird jedem Objekt eine Liste mit den Berechtigungen der jeweiligen Subjekte zugewiesen [23].

Tabelle 2.2: Darstellung einer Zugriffskontrollmatrix mit Kennzeichnung von Capability und ACL (adaptiert von [23])

Zugriffskontrollmatrix

	Subjekt	**Datei 1**	**Datei 2**	**Datei 3**	**Datei 4**
Capability	**Larry**	Lesen	Lesen, Schreiben	Lesen	Lesen, Schreiben
	Curly	Vollzugriff	Kein Zugriff	Vollzugriff	Lesen
	Mo	Lesen, Schreiben	Kein Zugriff	Lesen	Vollzugriff
	Bob	Vollzugriff	Vollzugriff	Vollzugriff	Kein Zugriff
				ACL	

2.2.3 Firewalls

Firewalls

Definition 2.2.21 Firewall

Als Firewall wird eine Software- oder Hardwarekomponente bezeichnet, die Netzwerkpakete überwacht und auf Basis eines definierten Regelwerks filtert. Eine Firewall ist dadurch eine Schutzmaßnahme aus dem Gebiet der Zugriffskontrolle, die auf der Netzwerkebene agiert. Nach der RFC 4949 [101] ist eine Firewall ein Gateway, das den Datenverkehr zwischen zwei verschiedenen Netzwerken (z.B. dem Internet und einem internen Firmennetzwerk) einschränkt, um Ressourcen vor Bedrohungen aus dem anderen Netzwerk zu schützen (s. Abbildung 2.15)

Firewalls sind in der IT ein fester Bestandteil, um eine zuvor definierte Sicherheitsstrategie in Form von Zugriffsrestriktionen durchzusetzen [68]. Es haben sich dabei über die Jahre verschiedene Arten von Firewall-Systemen entwickelt, die in unterschiedliche Klassen[2] (Paketfilter, Proxy und Applikationsfilter) eingeteilt werden [102]. Diese Schutzmaßnahmen können dabei in unterschiedlichen Architekturen eingesetzt werden und sind darüber hinaus oft in einer Kombination zu finden. Klassische Paketfilter werden dabei direkt in Netzwerkgeräte wie z.B. Router integriert. Hingegen kommt auf Servern eine Kombination aus Paket- und Applikationsfilter zum Einsatz.

2.2.4 Intrusion Detection Systeme

Im Vergleich zu Firewalls oder Verschlüsselungsverfahren, die als proaktive Maßnahmen definiert sind, werden IDS den reaktiven Schutzmaßnahmen zugeordnet [103], [23]. Dies ist damit begründet, dass durch IDS Systeme mögliche Angriffe erst während des Auftretens erkannt werden können. Hingegen sind proaktive Maßnahmen dafür vorgesehen, präventiv die potentielle

[2] Detaillierte Erläuterungen zu den unterschiedlichen Filterarten sind in Abschnitt A.1.4 enthalten.

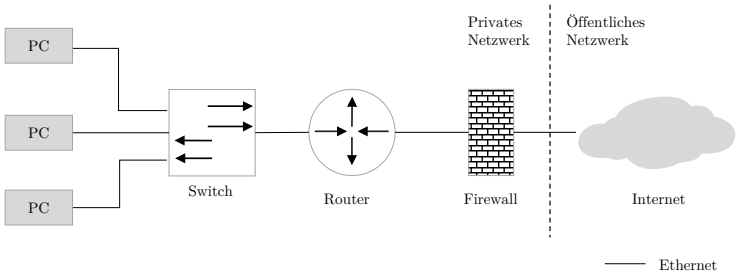


Abbildung 2.15: Integration einer Firewall zur Überwachung des Datenverkehrs zwischen einem öffentlichen- und privaten Netzwerk.

Angriffsfläche des betrachteten Systems zu minimieren. Allgemein analysieren IDS zur Erkennung von Angriffen den Datenverkehr in Netzwerken, werten System Log-Dateien aus oder untersuchen Aktivitätsabläufe eines Benutzers. Grundlegend wird dabei je nach Einsatzort in *Host- bzw. Netzwerk-basiert* unterschieden [23], [102]. Dabei kommen Host-basierte IDS (HIDS) ausschließlich auf Workstations oder Server zum Einsatz, um Anomalien innerhalb dieser Systemgrenze zu erkennen. Eine Anomalie könnte ein Versuch eines Benutzers darstellen, systemkritische Dateien zu verändern oder allgemein das System bzw. den Server in einen risikobehafteten Zustand versetzen zu wollen. Daneben analysieren Netzwerk-basierte IDS (NIDS) die Datenströme innerhalb von Netzwerken, indem von allen Paketen eine Kopie angelegt wird, um diese dann auf vordefinierte Merkmale wie beispielsweise Protokollabweichungen oder bösartige Nutzdaten zu prüfen. Die Erkennungstechniken beider Systeme können in zwei verschiedene Arten gegliedert werden. *Signatur-basierte* Verfahren basieren auf Merkmalen die von bereits bekannten Angriffen extrahiert und regelmäßig aktualisiert werden. Dies bringt die Einschränkung mit sich, dass nur sehr ähnliche Angriffe erkannt werden können. Dagegen bleiben neuartige Angriffe teilweise unerkannt. Hingegen erkennen *Anomalie-basierte* Verfahren Abweichungen in Bezug auf vordefinierte Merkmale, die aus einem Normalverhalten extrahiert werden. Häufig werden dazu statistische, protokoll-spezifische, regelbasierte und heuristische Techniken verwendet.

Auf Basis erkannter IDS-Ereignisse können zentralisiert an eine verantwortliche Person gemeldet werden, die diese detailliert auswerten kann, um mög-

lichst Falschalarme ausschließen zu können. Als konkrete Gegenmaßnahme kann dann eine Anpassung bestehender Firewall-Regeln bzw. Zugriffsrichtlinien erfolgen, um einen laufenden Angriff abzuwehren oder Systeme gegen zukünftige Angriffe weiter zu härten. Diese Gegenmaßnahmen werden übergreifend den Intrusion Prevention Systemen (IPS) zugeordnet [23].

IDS-Kategorien

Die verfügbaren IDS-Systeme lassen sich grundlegend in zwei Kategorien gliedern. Ein Signatur-basiertes Intrusion Detection System (SIDS) verwendet für die Erkennung von Angriffen vordefinierte Angriffsmuster [104]. Die Signatur stellt dabei eine Art Datenbankeintrag dar, der Merkmale von bereits bekannten Angriffen enthält. Diese müssen vorher über entsprechendes Expertenwissen eingetragen werden. Daher wird diese Art auch als wissensbasierte Erkennung (engl. knowledge-based detection) bezeichnet. Das System erstellt im Betrieb ebenfalls Signaturen von eingehenden Daten und vergleicht und sucht regelbasiert nach Übereinstimmungen in der Datenbank (s. Abbildung 2.16).

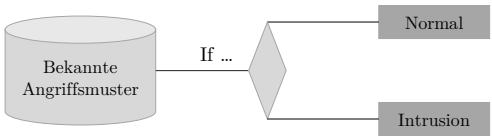


Abbildung 2.16: Schematischer Aufbau zur Funktionsweise eines SIDS (basierend auf [105]).

Die Erkennungsrate von bekannten Angriffen wird allgemein als hoch eingestuft. Zudem gibt es nur eine geringe falsch-postiv Rate. Unbekannte bzw. neue Angriffsmuster kann das System hingegen nicht erkennen. Das gilt insbesondere für *Zero-Day Attacken*, da hier noch keine bekannte Signatur in der Datenbank vorhanden ist. Da in Zukunft eine Zunahme von neuen Schwachstellen sowie Zero-Day Angriffen wahrscheinlich ist, sind SIDS für eine Absicherung nicht effektiv genug [105].

Die zweite Kategorie bildet ein Anomalie-basiertes Intrusion Detection System (AIDS), das im Vergleich zu SIDS das Normalverhalten von Benutzern,

Anwendungen oder Systemen analysiert und in einem Modell speichert. Jede signifikante Abweichung zwischen beobachteten und dem Model bekannten Verhalten wird als Anomalie (Intrusion) eingestuft. Die dafür definierten Schwellwerte können je nach Anwendung und geforderter Empfindlichkeit unterschiedlich spezifiziert sein. Die Entwicklung eines Anomalie-basierten Erkennungsansatzes gliedert sich in zwei Phasen. Die erste Phase bildet die Trainingsphase, in der das Modell mit dem Normalverhalten angelernt wird. In der darauffolgenden Testphase wird ein von der Trainingsphase unabhängiger Datensatz mit enthaltenen Anomalien verwendet, um die Erkennungsfähigkeit bzgl. bisher unbekanntem Verhalten zu prüfen. Da bei AIDS keine spezifischen Signaturen gespeichert werden, sind z.B. auch Zero-Day Angriffe detektierbar [105]. Ein weiterer Vorteil besteht in der Erkennung von Insider-Angriffen. Ein Angreifer müsste beispielsweise für einen erfolgreichen Angriff, das Verhalten eines gestohlenen Benutzerkontos genau kennen, um keine Anomalien zu verursachen. Nachteilig ist hingegen bei dieser Methode, dass die Anzahl von falsch-positiv Fällen im Vergleich zu SIDS höher ist, da eine vollständige Abbildung des Normalverhaltens im Modell schwierig ist.

Definition 2.2.22 Zero-Day Attacke

Als Zero-Day Attacke werden Angriffe bezeichnet, die am selben Tag nach der Entdeckung einer Schwachstelle durchgeführt werden, um diese auszunutzen [105].

AIDS-Erkennungsmethoden

Im Bereich der AIDS gibt es unterschiedliche Erkennungsmethoden, die für das Trainieren bzw. Testen von Daten anwendbar sind [105]. Die Methoden lassen sich dabei in drei Hauptkategorien (maschinelles Lernen, Wissensbasis sowie Statistik-basiert) klassifizieren (s. Abbildung 2.17). Ein weiterer Detaillierungsgrad ist in [106] gegeben.

Alle Techniken beinhalten dabei folgende drei generischen Funktionskomponenten bzw. Stufen [107]:

- Parametrisierung: In dieser Phase werden die zu beobachtenden Instanzen des Zielsystems in einer vorab definierten Struktur dargestellt.

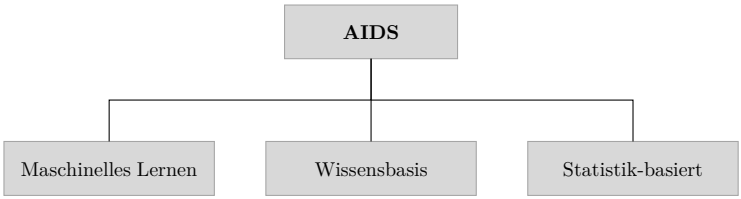


Abbildung 2.17: Klassifizierung von AIDS-Erkennungsmethoden [107].

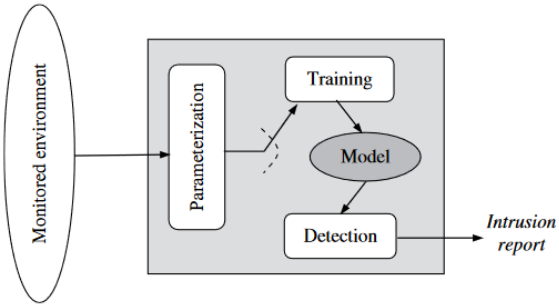


Abbildung 2.18: Generische funktionale Struktur eines AIDS [107].

- Trainingsstufe: Das Normalverhalten wird in einem Modell abgebildet. Dies erfolgt entweder manuell oder automatisiert.
- Detektionsstufe: Die beobachteten Daten werden mit dem zuvor erstellten Modell abgeglichen. Überschreitet die Abweichung einen spezifizierten Schwellwert, wird eine Anomalie-Meldung ausgegeben.

Die Statistik-basierten IDS Techniken erstellen basierend auf Daten (z.B. Netzwerkdaten) bestimmte Profile, die ein statistisches Verhalten abbilden [107]. Diese Profile enthalten beispielsweise die Datenrate, die Anzahl von Paketen eines bestimmten Protokolls oder die Anzahl an gleichzeitigen Verbindungen. Im Betrieb vergleicht dieser IDS-Typ die Profile von zwei Datensets (Trainingsdaten und aktuell zu verarbeitenden Daten). Treten dabei Abweichungen auf,

wird anhand eines definierten Schwellwerts (Abweichungsgrad) klassifiziert, ob bestimmte Daten als Anomalie gekennzeichnet werden.

Eine weitere Technik bilden wissensbasierte AIDS. Im Vergleich zu SIDS wird durch menschliches Wissen ein Normalverhalten des Systems in Form von Regeln beschrieben. Dazu werden zunächst verschiedene Attribute und Klassen aus den Trainingsdaten identifiziert und anschließend eine Menge von Klassifizierungsregeln, Parametern oder Verfahren abgeleitet, um ein legitimiertes Verhalten zu beschreiben. Im Betrieb wird jedes davon abweichende Verhalten als Anomalie eingeordnet.

Die dritte IDS-Technik umfasst Methoden des maschinellen Lernens, die ein Verhaltensmodell auf annotierten Trainingsdaten erstellen, um damit Muster zu erkennen. In vielen Fällen deckt sich die Anwendbarkeit von maschinellen Lernprinzipien mit der von statistischen Techniken, obwohl erstere darauf abzielen, ein Modell zu erstellen, das seine Leistungsfähigkeit auf der Grundlage früherer Ergebnisse verbessert. Ein maschinell lernendes IDS hat dagegen die Fähigkeit, seine Erkennungsstrategie zu ändern, wenn es neue Informationen (annotierte Daten) erhält.

2.3 Künstliche Intelligenz

Die Hauptaufgabe der Informatik besteht grundlegend aus der strukturierten sowie automatisierten Verarbeitung von Daten bzw. Informationen, die im klassischen Sinne durch Menschen erstelle Programme erfolgt [108]. Dabei existieren jedoch Problemstellungen, die nur schwer bzw. durch zeitaufwändige Programmierungen lösbar sind. So ist beispielsweise für Menschen die Identifikation einer Katze auf einem Foto gut lösbar. Diese Aufgabe durch eine vom Menschen programmierte Computeranwendung abzubilden ist dagegen aufwändig. Die Künstliche Intelligenz (KI) adressiert diese Art von Problemen, die heutzutage in unterschiedlichen Bereichen (z.B. automatisiertes Fahren, Informationssicherheit) vertreten sind [109], [110]. Eine Schlüsseltechnologie der KI bildet dabei maschinelles Lernen.

2.3.1 Maschinelles Lernen

Der Begriff *Maschinelles Lernen* ist ein Oberbegriff für eine Sammlung von etablierten Techniken und Algorithmen der Informatik, die es ermöglichen eine KI zu realisieren (eine anschauliche Klassifizierung ist in [111] gegeben). Die Anwendung von KI ist breit gefächert und wird u.a. bei der Spracherkennung, Erkennung von Bildern, automatisierten Fahrfunktionen sowie bei der Erkennung von Angriffen auf die Informationssicherheit eingesetzt. Die KI ist dabei in der Lage ein Wissen basierend auf Erfahrungen aufzubauen. Die Aufgabe besteht im Wesentlichen darin, eine Eingabe auf eine Ausgabe zu transformieren (s. Abbildung 2.19), um damit eine Aufgabe zu lösen, die zuvor nicht durch einen Menschen über eine Logik programmiert wurde [108].

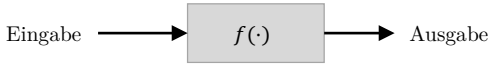


Abbildung 2.19: Prinzip der Informationsverarbeitung (basierend auf [108]).

Konventionell berechnet ein programmierter Algorithmus auf Basis der Eingabe die Ausgabe. Im Bereich des maschinellen Lernens wird dieses Prinzip verändert. Das zugrundeliegende Lernverfahren erstellt aus der Eingabe sowie Ausgabe ein Programm (Modell), welches in der Lage ist, die Eingabe entsprechend der bekannten Ausgabe zu transferieren (s. Abbildung 2.20).

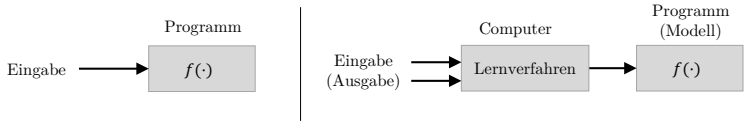


Abbildung 2.20: Vergleich der prinzipiellen Funktionsweise zur Lösung einer Aufgabe auf Basis der konventionellen Programmierung (links) sowie dem maschinellen Lernen (rechts). Basierend auf [108].

Die Erstellung eines Modells erfolgt in einer sogenannten Trainingsphase, in der mittels Trainingsdaten die Erfahrungen angelernt werden. Das maschinelle Lernen ist dabei nicht mit einem Lernen bei Menschen zu vergleichen. Vielmehr erfolgt eine Anpassung (engl. fitting) des Modells an die Trainings-

daten. Es kann als Optimierungsproblem angesehen werden, bei dem auf Basis vorhandener Daten sowie bekannten Ausgabe (Ziel) ein möglichst geeignetes Modell aus einer Menge unterschiedlicher Modelle herausgesucht wird.

Methoden des Lernens

Für die Trainingsphase des maschinellen Lernens existieren verschiedene Methoden, die in Abhängigkeit der zur Verfügung stehenden Daten anwendbar sind. Grundlegend lassen sich diese in vier Arten unterschieden.

- **Überwachtes Lernen:** Das überwachte Lernen (engl. supervised Learning) basiert auf vorhandenen Eingabe-Ausgabe Kombinationen, die in Trainingsdaten enthalten sind und als Beispiele dienen. Darauf basierend wird dann eine Funktion abgeleitet, die gegebene Eingaben auf die bekannten Zielwerte abbilden kann. Da die Ausgabewerte bekannt sind, werden diese Daten auch als annotiert (engl. labeled) bezeichnet. Während der Trainingsphase erfolgt die Anpassung der Modellparameter, die meist durch das verwendete Lernverfahren definiert sind. Da die Zielwerte bereits bekannt sind, kann die in der Trainingsphase laufende Optimierung überwacht werden, wie genau das aktuelle Modell die Ausgabe berechnet.
- **Unüberwachtes Lernen:** Beim unüberwachten Lernen (engl. unsupervised Learning) stehen im Vergleich zum überwachten Lernen wesentlich weniger Informationen in Bezug auf die vorhandenen Daten zur Verfügung. Das bedeutet es existieren keine annotierten Daten (Beispiele) sondern lediglich Eingabedaten. Das System muss dadurch eigenständig bestimmte Zusammenhänge bzw. Muster auf Basis der vorhandenen Daten ableiten. Die Herausforderung liegt dabei in der Bewertung der entdeckten Muster, die aufgrund der fehlenden Zielgröße während der Lernphase hinsichtlich ihrer Güte quantifiziert werden können.
- **Halbüberwachtes Lernen:** Das halbüberwachte Lernen (engl. semi-supervised Learning) nutzt eine Kombination aus den beiden zuvor erläuterten Lernmethoden. Damit wird versucht aus einer Menge aus Eingabedaten, die nur zum Teil annotiert sind, ein entsprechendes Modell zu optimieren.

- **Bestärkendes Lernen:** Als viertes Lernverfahren ist das sogenannte bestärkende Lernen (engl. reinforcement learning) definiert, das sich zu den zuvor erläuterten Verfahren grundlegend unterscheidet. So kann hier von einer Art Belohnungsmethode gesprochen werden. Zu Beginn der Trainingsphase stehen noch keine Trainingsdaten zur Verfügung, sondern werden mittels eines Systems, das in Interaktion mit seiner Umwelt steht, generiert. Die durchgeführten Aktionen führen dabei direkt zu einer Reaktion oder können indirekt durch eine Zustandsänderung der Umwelt zeitlich versetzt ein Feedback bzgl. den vom System getroffenen Entscheidungen übermitteln. Die Rückmeldung an das System umfasst entweder eine Belohnung oder Bestrafung in Bezug auf die Zielerreichung, sodass das System für eine Abfolge von Aktionen versucht die Anzahl der Belohnungen zu maximieren. Letztlich wird nicht jede einzelne Aktion bewertet, sondern nur die gesamte Abfolge. Diese ist dadurch auch mit der Bewertung einer Spielstrategie vergleichbar, die entweder zu einem Sieg oder einer Niederlage führte.

2.3.2 Computerlinguistik

Die Computerlinguistik (CL) (engl. Natural Language Processing (NLP)) beschäftigt sich mit Methoden für die maschinelle Text- und Sprachverarbeitung mittels Computeralgorithmen. Die Sprachverarbeitung bildet ein Teilgebiet der KI und verbindet die Sprachwissenschaft mit der Informatik. Innerhalb der Spracherkennung haben sich grundlegend zwei Ansätze ausgeprägt (Spracherkennung mittels Mustervergleich sowie die statistische Erkennung) [112]. Da die Arbeit in Kapitel 5 eine statistische Erkennungsmethode verwendet, wird diese nachfolgend näher erläutert.

Statistische Spracherkennung

Die stochastische Modellierung wird verwendet, um die Variantenvielfalt von Sprachsignalen in Bezug auf Spracherkennung abzubilden [112]. Betrachtet man die Spracherkennung auf Basis der informationstheoretischen Sicht lässt sich diese auf ein Dekodierungsproblem zurückführen (s. Abbildung 2.21). Ausgehend von einem Sprecher, der eine Mitteilung machen möchte, transformiert eine Abfolge von Wörtern W durch seinen Sprechapparat in ein akusti-

sche Signal s. Die Übertragung in Richtung Erkenner kann dabei durch unterschiedliche Medien erfolgen, bevor dieser aus dem digitalisierten Sprachsignal bestimmte Merkmalssequenzen X extrahiert. Weiter kann X als Ausgang des Übertragungskanals betrachtet werden, der als Eingabe W erhalten hat. Der linguistische Decoder hat dann die Aufgabe, aus der Merkmalssequenz X die Wortfolgen der Mitteilung zu schätzen.

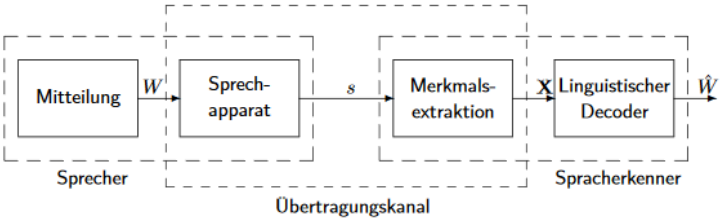


Abbildung 2.21: Informationstheoretische Sicht der Spracherkennung [112].

Sprachmodellierung

Die Sprachmodellierung (engl. language modeling) steht in direkter Beziehung mit der Spracherkennung und wird dabei als Sammlung von A-priori Kenntnissen über die Sprache definiert, die eine Menge von Wortfolgen ohne akustische Aspekte beinhaltet [112]. Dieses Wissen steht dabei als Vorwissen zur Verfügung, bevor eine Mitteilung stattfindet. Das gesammelte Wissen beinhaltet neben der allgemeinen Sprache auch die Informationen über die Kommunikationssituation. Die Situation lässt sich über Erfahrungswerte abbilden, die darin verwendete Wörter sowie deren Häufigkeit speichern. Ein Sprachmodell ist häufig aus mehreren Teilmodellen aufgebaut, von denen jedes einen Teilaspekt der Sprache repräsentiert (z.B. Häufigkeit der Wörter oder Satzgrammatik). Für die Modelle sind zwei Formen ausgeprägt.

- **Statistische Sprachmodelle:** Die statistische Sprachmodellierung nutzt Erfahrungswerte durch Messen oder Zählen von Ereignissen, um beispielsweise die Häufigkeit von Wörtern zu erfassen. Diese werde dann mittels einer Sprachdatensammlung, auch Korpus genannt, gespeichert.

Diese Art von Sprachmodell beschreibt die Erzeugung der Sprache mittels einem Zufallsprozess.

- **Wissensbasierte Sprachmodelle:** Diese Art von Sprachmodellen befasst sich mit der Erfassung von linguistischem Expertenwissen, welches beispielsweise grammatikalisches Wissen über die Konjugation von Verben beinhalten kann. Die Grundlage dieses gespeicherten Wissens ist im Vergleich zu statistischen Sprachmodellen nicht die zahlenmäßige Erfassung von Ereignissen der Sprache, sondern vielmehr die dahinterliegenden Zusammenhänge bzw. Gesetzmäßigkeiten.

Statistische Sprachmodellierung

Die statistische Spracherkennung beschäftigt sich mit der Aufgabe für eine vorhandene Merkmalssequenz X die Wortfolge W mit der größten A-posteriori-Wahrscheinlichkeit $P(W|X)$ zu ermitteln. Da die Schätzung dieser Wahrscheinlichkeit nahezu nicht realisierbar ist, wird stattdessen das Produkt $P(X|W)P(W)$ maximiert. Dabei ist die Wahrscheinlichkeit $P(X|W)$ als *akustisches Modell* definiert. Darüber hinaus wird das *Sprachmodell* durch die Wahrscheinlichkeitsverteilung $P(W)$ definiert, welches unabhängig von der Merkmalssequenz X ist und einzig A-priori-Kenntnisse beinhaltet.

Im Vergleich zu *formalen Sprachen* wird über Sprachmodelle gewöhnlich keine Aussage bzgl. *richtig* oder *falsch* einer Wortfolge getroffen. Anders formuliert erfolgt keine scharfe Abgrenzung, ob diese zu einer Sprache gehören oder nicht. Dagegen ist prinzipiell jede Wortfolge möglich, jedoch mit unterschiedlichen Wahrscheinlichkeiten.

Definition 2.3.1 Formale Sprache

Als formale Sprache wird eine Wortmenge $L \subseteq V_T^*$ über dem Alphabet V_T bezeichnet, sofern ein endliches formales System existiert, das L vollständig beschreibt [112]. Die Wortmenge L kann unbeschränkt sein.
V^* ist die Menge aller Wörter, die sich durch eine beliebige Reihung von Zeichen aus der Menge V bilden lassen (sogenannte Kleenesche Hülle). V_T ist dabei das Alphabet, eine endliche, nichtleere Menge von Zeichen, die auch als Terminalsymbole bezeichnet werden, z.B. $V_T = \{a, b, c, ...\}$.

Linguistische Korpora

Ein Korpus dient im Rahmen der CL als empirische Datengrundlage für das Training von Programmen der Sprachverarbeitung [113]. Ein linguistischer Korpus wird als digital gespeicherte Sammlung von gesprochener sowie schriftlicher Äußerungen definiert, die dadurch maschinenlesbar sind. Eine weitere Eigenschaft dieser Korpora ist, dass diese explizit authentische Sprachdaten enthalten, die auf einer linguistisch unreflektierten Kommunikationssituation basieren.

N-Gramm Modelle

Zur Erkennung einer Wortfolge $W_1^K = w_1 w_2 ... w_k$ mit $w_k \in V$ ist die Verwendung eines Sprachmodells möglich, um die Wahrscheinlichkeit $P(W_1^K)$ zu bestimmen. Durch Anwendung des Multiplikationsgesetzes ist eine Zerlegung in bedingte Wahrscheinlichkeiten der einzelnen w_k möglich, jeweils basierend auf den vorangegangenen Wörtern $w_1, ..., w_{k-1}$:

$$P(W_1^K) = P(w_1) \cdot P(w_2|w_1) \cdot \cdots \; P(w_K|w_1 ... w_{K-1}) = \prod_{k=1}^{K} P(w_k|W_1^{k-1}) \tag{2.1}$$

Da theoretisch beliebig viele Wortfolgen W existieren und nur ein kleiner Teil als Stichprobentexte vorhanden sind, ist eine empirische Schätzung der wahren Verteilung von $P(W)$ nicht möglich. Durch die Aufteilung in bedingte Wahrscheinlichkeiten $P(w_k|w_1...w_{k-1})$ lässt sich das vorhandene Dilemma umgehen. Durch die Annahme, dass die Abhängigkeit zwischen weit auseinander liegenden Wörtern einer Wortfolge nur schwach gegeben ist, kann für die Berechnung der Wahrscheinlichkeiten die Anzahl der Wörter vor w_k auf einen definierten Wert limitiert werden.

Das N-Gramm Modell [114] entspricht dabei einer Markov-Kette $(n-1)$-Ordnung

dieser Annahme und definiert in Abhängigkeit von n verschiedene N-Gramme. Die Approximation basiert dabei nur auf den vorangegangen $N-1$ Wörtern:

$$P(w_k|w_1...w_{k-1}) \approx P(w_k|w_{k-N+1}...w_{k-1}) \tag{2.2}$$

Dabei werden die N-Gramme in Abhängigkeit der Länge N als Unigramm (N=1), Bigramm (N=2), Trigramm (N=3) und Tetragramm (N=4) bezeichnet. Das *Unigramm* bildet die einfachste Approximation, die kein vorangegangenes Wort berücksichtigt:

$$P(w_k|w_1...w_{k-1}) \approx P(w_k) \tag{2.3}$$

Hingegen berücksichtigt das *Bigramm* genau das jeweilige Vorgängerwort einer Wortfolge und wird in der Praxis sehr häufig eingesetzt:

$$P(w_k|w_1...w_{k-1}) \approx P(w_k|w_{k-1}) \tag{2.4}$$

Das *Trigramm* und *Tetragramm* (s. Gleichungen 2.5 u. 2.6) erhöht die Anzahl der vorangegangenen Wörter einer Wortfolge auf zwei bzw. drei:

$$P(w_k|w_1 ... w_{k-1}) \approx P(w_k|w_{k-2}, w_{k-1}) \tag{2.5}$$

$$P(w_k|w_1 ... w_{k-1}) \approx P(w_k|w_{k-3}, w_{k-2}, w_{k-1}) \tag{2.6}$$

Definition 2.3.2 Markov-Kette

Ein Markov-Prozess mit endlichem oder abzählbarem (also diskretem) Zustandsraum heißt Markov-Kette [115].

Definition 2.3.3 Markov-Eigenschaft

Ein stochastischer Prozess $(X_t)_{t \in T}$ heißt Markov-Prozess, genau dann, wenn für alle $t_0 < t_1 < \cdots < t_{n+1}, t_i \in T$ und $x_0, \ldots, x_{n+1} \in S, n \in \mathbb{N}$: $P(X_{t_{n+1}} = x_{n+1} | X_{t_n} = x_n, \ldots, X_{t_0} = x_0) = P(X_{t_{n+1}} = x_{n+1} | X_{t_n} = x_n)$. Diese Eigenschaft heißt Markov-Eigenschaft [115].
Dabei entspricht Menge T der Parametermenge. Indizes $t \in T$ interpretiert als Zeiten. Menge S ist definiert als Zustandsraum. Werte der Zufallsvariablen X_t interpretiert als Zustände eines Systems.

Definition 2.3.4 Stochastischer Prozess

Ein stochastischer Prozess ist eine Familie von Zufallsvariablen $(X_t)_{t \in T}$ auf einem Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, P)$ mit Werten in einer Menge S [115].

Schätzen von N-Gramm Wahrscheinlichkeiten

Für die Schätzung von Wahrscheinlichkeiten der N-Gramme ist die Maximum Likelihood Estimation (MLE) anwendbar [114]. Dazu wird die Anzahl eines bestimmen N-Gramms in einem Korpus ermittelt und normalisiert:

$$P(w_k|w_{k-n+1}^{k-1}) = \frac{C(w_{k-n+1}^{k-1} w_k)}{C(w_{k-n+1}^{k-1})} \tag{2.7}$$

Für ein Bigramm ergibt sich daraus:

$$P(w_k|w_{k-1}) = \frac{C(w_{k-1} w_k)}{C(w_{k-1})} \tag{2.8}$$

Nachfolgend wird das Vorgehen auf Basis eines exemplarischen Minimal-Korpus dargestellt, der drei unterschiedliche Sätze enthält (der Satzanfang sowie das Satzende werden durch <s> bzw. </s> markiert):

- 1. Satz: <s> I am Sam </s>
- 2. Satz: <s> Sam I am </s>
- 3. Satz: <s> I do not like green eggs and ham </s>

Durch die Verwendung eines Bigramms lassen sich beispielsweise folgende Wahrscheinlichkeiten berechnen:

- $P(I| < s >) = \frac{2}{3}$
- $P(Sam| < s >) = \frac{1}{3}$
- $P(do|I) = \frac{1}{3}$

Parameterglättung

Die allgemeine Herausforderung in der CL besteht darin, auf der Basis einer begrenzten Trainingsmenge, eine präzise (linguistische) Ausdruckskraft zu erreichen. Jedoch kann ein Sprachmodell niemals alle Wortkombinationen erhalten. Dies kann dazu führen, dass bestimmte in der Realität auftretende Wortkombinationen, die in den Trainingsdaten nicht enthalten sind, eine Wahrscheinlichkeit von null aufweisen. Um dieses Auftreten zu vermeiden,

wird durch Glättungsmethoden (engl. smoothing) versucht, dies zu kompensieren [116]. Dafür werden die enthaltenen Häufigkeiten w im ermittelten Korpus N durch einen Glättungsparameter k angepasst.

Allgemein ist diese Glättungsmethode wie folgt definiert:

$$p_{Add-k}(w) = \frac{\#w + k}{N + kV} \tag{2.9}$$

Darin repräsentiert k den Glättungsparameter der einen Wert zwischen $0 < k \leq 1$ aufweisen kann. Der Parameter V umfasst dabei die Anzahl der im Korpus vorhandenen Wörter. Die allgemeine Glättungsvorschrift lässt sich beispielsweise wie folgt auf ein Bigramm übertragen:

$$p_{Add-k}(w_i|w_{i-1}) = \frac{\#(w_{i-1}, w_i) + k}{\#(w_{i-1}) + kV} \tag{2.10}$$

3 Angriffsanalyse und Stand der Technik/Wissenschaft

Für die informationstechnische Absicherung von Fahrzeugarchitekturen ist es erforderlich bisherige Angriffe sowie deren ausgenutzte Schwachstellen zu untersuchen, um identifizierte Schwächen bei zukünftigen Entwicklungen zu adressieren. Im Hinblick auf diese Arbeit wird eine statistische Auswertung vergangener Angriffe vorgenommen und darauf basierend ein Abgleich zum aktuellen Stand der Technik und Wissenschaft vorgenommen, welche Sicherheitsschwächen aktuell durch die Forschung aber auch durch Normen und Standards adressiert werden. Gestützt wird diese Analyse durch die Taxonomie für Angriffe der automotive Domäne von Sommer et al. [34], die zur detaillierten Klassifizierung 23 verschiedene Kategorien enthält. Daneben umfasst die Taxonomie eine öffentlich verfügbare Sammlung von 162 publizierten Angriffen [117]. Darüber hinaus bieten die Autoren auf dieser Datengrundlage eine erweiterte Sammlung an, in der die Angriffe in deren einzelne Angriffsschritte aufgeteilt sind und sich dadurch 413 Schritte auf Basis von 162 Angriffen ergeben. Seit Mitte 2022 ist eine aktualisierte Version mit 361 Angriffen sowie 621 Angriffsschritten verfügbar [117]. Klassische Datenbanken für IT-Schwachstellen wie beispielsweise die *Common Vulnerabilities and Exposures (CVE)* [118] umfassen dagegen nur vereinzelte, in Textform beschriebene Schwachstellen von Fahrzeugen. Eine umfangreiche Analyse ist damit nicht möglich. Die Kategorien der Taxonomie enthalten dagegen detaillierte Informationen zu beispielsweise genutzten Angriffswegen (engl. attack paths), Angriffsklassen oder verletzten Security-Eigenschaften. Im Hinblick auf diese Arbeit sind dabei Statistiken bzgl. der Zugriffskontrolle sowie genutzte Schnittstellen von besonderem Interesse, um daraus die Anforderungen für die Entwicklung eines effektiven Ansatzes abzuleiten.

3.1 Bisherige Angriffe

Für eine quantitative Auswertung der Datenbasis [117] wird der Betrachtungszeitraum auf zehn Jahre festgelegt (2010 - 2019), da die enthaltenen Angriffe über das Jahr 2019 hinaus noch nicht vollständig erfasst sind. Die Jahre vor 2010 wurden ebenfalls nicht berücksichtigt, da diese nur eine geringe Anzahl aufweisen. Neben einer jährlichen Auswertung der insgesamt 525[1] Angriffe wird zusätzlich eine Einteilung in safety-relevant und diagnosebasiert vorgenommen (s. Abbildung 3.1). Die safety-relevanten Angriffe umfassen Aktivitäten, die durch Ausnutzung einer Schwachstelle nicht nur informationstechnische Eigenschaften verletzt haben, sondern parallel auch einen Einfluss auf die funktionale Sicherheit des Fahrzeugs hatten. Gegenüber Angriffen mit ausschließlich informationstechnischen Auswirkungen weisen diese eine höhere Relevanz auf (s. auch Abschnitt 1.2). Die diagnosebasierten Angriffe stellen Aktivitäten dar, bei denen die Angreifer mithilfe von Diagnoseprotokollen (s. Abschnitt A.1.1) in die On-Board Kommunikation des Fahrzeugs eingedrungen sind. Dieses Filterkriterium wurde gewählt, um zu analysieren, welchen Anteil diese Sonderfunktionen in Bezug auf die Informationssicherheit in Fahrzeugen aufweisen. Durch die gewählten Kriterien lassen sich auf der Basis von Abbildung 3.1, die jeweiligen prozentualen Anteile der beiden Klassen quantifizieren. Die safety-relevanten Angriffe besitzen einen Anteil von 14 % der Gesamtzahl. Die diagnosebasierten Angriffe stellen einen Anteil von 9 % der Gesamtzahl dar.

Es ist bei dieser Auswertung zudem anzumerken, dass in den Jahren 2013 und 2015 lokale Peaks ausgeprägt sind. Dagegen schwächt sich die Anzahl der Angriffe in den drei nachfolgenden Jahren ab. Es sollte darauf basierend nicht abgeleitet werden, dass dies ein tatsächlicher Rückgang an Angriffen repräsentiert, da die erfassten Zahlen bestimmten Randbedingungen unterliegen. Die ausgewertete Sammlung umfasst lediglich die öffentlich publizierten Angriffe. Dagegen gelangen nicht immer alle durchgeführten Angriffe an die Öffentlichkeit oder werden an die Hersteller gemeldet. Dadurch kann eine nicht quantifizierbare Dunkelziffer existieren. Des Weiteren kam es in den genannten Jahren zu medienwirksamen Aktivitäten einiger Forscher, die zusätzliche

[1] Die Anzahl der Angriffe beinhaltet sowohl Single-Step als auch Multi-Step Angriffe gemäß der Taxnomie von Sommer et al. [34].

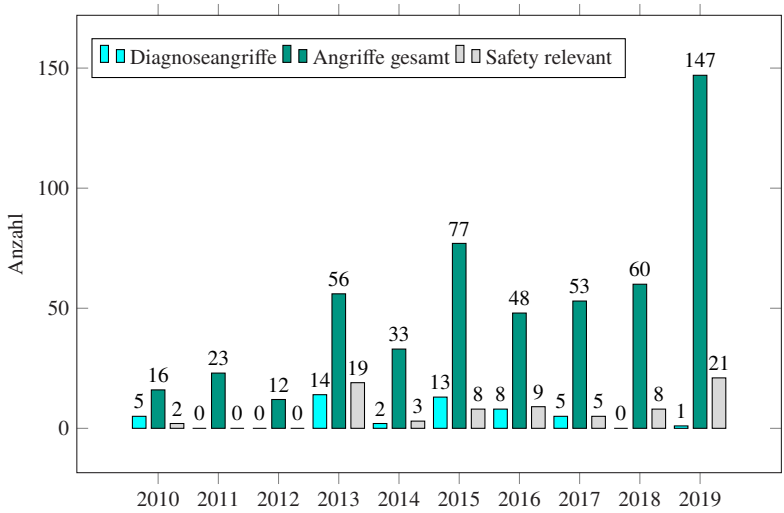


Abbildung 3.1: Übersicht von Angriffen auf Fahrzeuge im Zeitraum 2010 - 2019 basierend auf [117].

Forschungsaktivitäten im Bereich der Security in Fahrzeugen auslösten, wodurch mehr Veröffentlichungen erschienen sind. Das bedeutet, die Anzahl der bekannten Schwachstellen hängt im Wesentlichen von dem durchgeführten Analyseaufwand ab. Des Weiteren ist zu berücksichtigen, dass neue Fahrzeugarchitekturen innerhalb der gesamten Fahrzeugentwicklungsphase entstehen, die eine durchschnittliche Laufzeit von ca. vier Jahren aufweisen [45]. Folglich muss davon ausgegangen werden, dass es zu zeitlichen Verschiebungen von neu entdeckten Schwachstellen bzw. daraus resultierenden Angriffen kommen kann. Allgemein muss mit einem ansteigenden Risiko für potentielle Schwachstellen in den nächsten Jahren gerechnet werden, welches aus dem zunehmenden Vernetzungsgrad und neuen Systemen resultiert [119]. Diese Annahme wird zudem durch einen automotive Security-Bericht bestärkt, der einen deutlichen Aufwärtstrend von Schwachstellen in den Jahren von 2010 - 2019 registriert [120]. Des Weiteren ist ein erhöhtes Angriffsrisiko auch durch zukünftige Trends ableitbar, da Remote-Zugriffe aufgrund der wachsenden Verbreitung sowie Funktionsvielfalt von Diagnosediensten ansteigen werden [121]. Gleichzeitig wird aktuell durch die Entwicklung von Standards, Pro-

zessen sowie Schutzmaßnahmen versucht, das Risiko für Schwachstellen in Zukunft zu minimieren (s. auch Abschnitt 3.2).

3.1.1 Analyse der Angriffsschnittstellen

Für die Minimierung des Risikos durch geeignete Schutzmaßnahmen ist es notwendig, eine Analyse der genutzten Angriffswege bzw. Schnittstellen durchzuführen, um diese zukünftig entsprechend absichern zu können (s. Abbildung 3.2). Zusätzlich ist es zielführend die dabei ausgenutzten Schwächen bzw. Schwachstellen zu analysieren, um Schutzmaßnahmen zu verbessern bzw. gezielt integrieren zu können, damit diese Angriffswege in zukünftigen Fahrzeugen nicht mehr existent sind. Die Analyse ergibt, dass 28 % der Angriffe auf der OBD-Schnittstelle basierten und damit den größten Anteil darstellen. Der zweithöchste Anteil (24 %) bildet die Radio Frequency Identification (RFID)-Schnittstelle. Darauf folgen Mobilfunk-basierte Angriffswege (17 %). Die hohe Anzahl an OBD-basierten Angriffen kann u.a. darauf zurückgeführt werden, dass die OBD-Schnittstelle bei vorhandenem physikalischen Zugang zum Fahrzeug direkt zugänglich ist. Zudem existieren frei erhältliche Software-Tools mit passenden CAN-Bibliotheken, Beschreibungen sowie Hardwareschnittstellen [122, 123]. Der zeithöchste Anteil per RFID repräsentiert Angriffe, die ein Fahrzeug-Diebstahl als Angriffsziel beinhalten und sich auf drahtlose Fahrzeugzugangssysteme konzentrieren.

Für die weitere Analyse lässt sich eine Systemgrenze definieren, um dabei in zwei verschiedene Angriffstypen (*Insider, Outsider*) unterscheiden zu können. In Domänen-basierten E/E-Architekturen (s. auch Abschnitt 2.1.1) liegt diese Grenze im zentralen Gateway. Folglich werden alle Verbindungen der Außenwelt (OEM-Backend, Infrastruktur) über die Schnittstellen wie beispielsweise WLAN oder OBD über diesen Knoten nach statisch definierten Routingtabellen in das interne Fahrzeugnetzwerk geroutet.

Definition 3.1.1 OBD-Schnittstelle

Der OBD-Schnittstelle sind in dieser Arbeit Angriffe zugeordnet, die auf dem Standard CAN-Protokoll oder Diagnoseprotokoll basieren (s. auch Abschnitt A.1.1). Damit repräsentiert der Begriff lediglich die standardisierte physikalische Steckverbindung, mit der eine drahtgebundene Verbindung zu einer E/E-Fahrzeugarchitektur möglich ist.

Definition 3.1.2 RFID-Schnittstelle

Die RFID-Schnittstelle referenziert in dieser Arbeit die drahlosen Fahrzeugzugangssysteme, die auf der RFID-Technologie basieren. RFID definiert eine Sender-Empfänger Technologie, die eine automatisierte und berührunglose Identifikation sowie Lokalisierung von Objekten über Funkwellen umsetzt [124].

Ausnahmen stellen dabei drahtlose Schnittstellen mit kurzer Reichweite wie beispielsweise *Keyless Entry* oder Reifendruckkontrollsysteme dar. Angreifer, die über diese Schnittstellen versuchen in Fahrzeuge einzudringen, stellen *Outsider* dar. Agiert ein Angreifer hingegen innerhalb der Systemgrenze, stellt dieser einen *Insider* dar. Ein derartiges Szenario umfasst beispielsweise die Kompromittierung eines Steuergerätes durch einen Angreifer (Schnittstelle: *ECU*), der dieses als Eintrittspunkt für das interne Fahrzeug nutzt und als legitimer Teilnehmer agiert (s. auch Abschnitt 3.1.3).

Die Angriffe auf Basis dieser Schnittstelle zeigten, dass dadurch gezielte Funktionsausführungen möglich waren, die safety-kritische Auswirkungen hatten [125]. Als weiterer Insider-Angriff kann die Schnittstelle *CAN-Bus* angesehen werden. Bei diesem Angriffsweg hatten die Angreifer physikalischen Zugriff zu einem internen CAN-Bus und waren dadurch wiederum Teilnehmer der Kommunikation. Da das CAN-Protokoll standardmäßig keine Security-Eigenschaften enthält, hatten die angegriffenen Fahrzeuge keinen Schutz gegen unautorisiertes Lesen und Senden von Nachrichten. Die einzige Herausforderung lag dabei bei der Vermeidung von Kollisionen gleicher Nachrichten IDs (s. auch Abschnitt A.1.1). Sendet der Angreifer eine Botschaft, die eigentlich von einer anderen ECU stammt, erhält das jeweilige Empfängersteuergerät kurz hintereinander zwei Botschaften mit gleicher ID und verwirft standardmäßig

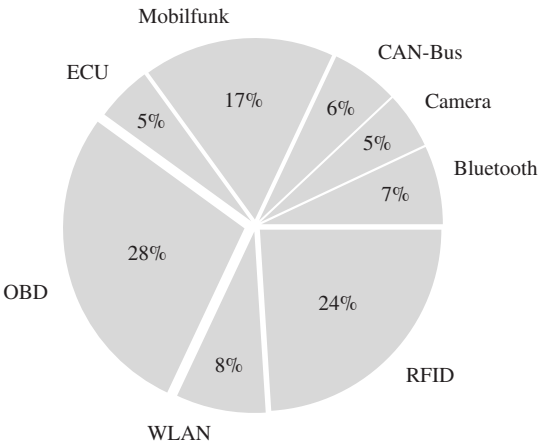


Abbildung 3.2: Prozentuale Verteilung der verwendeten Schnittstellen bei automotive Angriffen im Zeitraum 2010 - 2019 basierend auf [117].

letztere. Bei der weiteren Analyse zeigt sich, dass die Angreifer versuchten, das legitimierte Steuergerät vom Bus abzutrennen indem es in einen inaktiven Sendezustand (z.B. Diagnosemodus) versetzt wurde (s. auch Abschnitt 3.1.3). Die anderen Steuergeräte hatten dagegen keine Möglichkeiten, den unautorisierten Sender zu erkennen.

Auf der Basis der genutzten Angriffsschnittstellen ist eine weitere Klassifizierung in lokal- sowie remote-ausführbare Angriffe möglich (s. Abbildung 3.3). Die betrachteten remote-ausführbaren Schnittstellen bestehen dabei aus Mobilfunk, WLAN, RFID, Bluetooth, Kamera/Light Detection and Ranging (LiDAR). Im betrachteten Zeitraum ist der Anteil der lokal-basierten Angriffe (physikalischer Zugriff notwendig) um 16 % höher, gegenüber drahtlos ausgeführten Remote-Angriffen.

Der Remote-Anteil könnte sich aufgrund des ansteigenden Vernetzungsgrades von Fahrzeugen mit der Außenwelt über vorwiegend drahtlose Schnittstellen deutlich erhöhen. In einer Studie von PwC [7] werden bis zum Jahr 2025 über 470 Millionen vernetzte Fahrzeuge auf den Straßen weltweit unterwegs sein. Zudem kommt als weitere Veränderung hinzu, dass immer mehr klassische

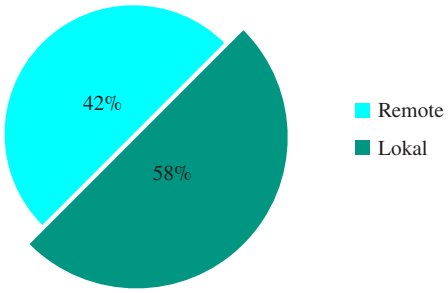


Abbildung 3.3: Verteilung der lokal- und remote-ausgeführten automotive Angriffe in den Jahren 2010 - 2019 basierend auf [117].

IT-Protokolle den Einzug ins Fahrzeug halten [14]. Dies könnte wiederum die Einstiegshürde für automotive Angriffe senken, da für diesen Bereich bereits verschiedene Tools und Anleitungen zum Ausnutzen von Schwachstellen existieren [126, 127, 128].

3.1.2 Betrachtung der verletzten Security-Eigenschaften

Für die Entwicklung und Implementierung effektiver Schutzmaßnahmen ist es entscheidend, welche Security-Eigenschaften der Informationssicherheit (s. auch Abschnitt 2.2.1) bei den bisherigen Angriffen verletzt wurden (s. Abbildung 3.4), da jede Verletzung durch eine fehlende oder nicht sichere Maßnahme oder einer Kombination aus beidem resultiert. Die Verletzung der Authentizität stellt mit 37 % aller Angriffe den höchsten Anteil dar. Dies lässt sich bei der weiteren Betrachtung der angegriffenen E/E-Architekturen erklären, da bis auf die Ausnahme der Diagnose (s. auch Abschnitt A.1.1), keine Maßnahmen zur Gewährleistung der Authentizität implementiert waren. Einen Anteil von 30 % besitzt die Security-Eigenschaft der Vertraulichkeit. Die Analyse der Ursachen führt zum Ergebnis, dass teilweise Schutzmaßnahmen zur Gewährleistung der Security-Eigenschaften implementiert waren, diese allerdings Schwächen aufwiesen, die zu den ausgenutzten Schwachstellen führten. Zur Verdeutlichung dient z.B. eine im Jahr 2013 ausgenutzte Schwachstelle in Form eines schwach implementierten Security Access (s. auch Abschnitt A.1.5) [125].

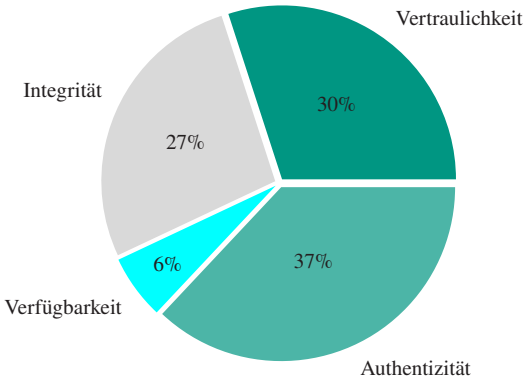


Abbildung 3.4: Prozentuale Verteilung der verletzten Security-Eigenschaften bei automotive Angriffen in den Jahren 2010 - 2019 basierend auf [117].

Diese ermöglichte den Forschern Miller et al. die Firmware eines Steuergerätes zu exportieren und anschließend über *Reverse-Engineering* Techniken zu analysieren, um an weitere Informationen zu gelangen. Des Weiteren stellen 27 % der Angriffe eine Verletzung der Integrität dar, die durch Tampering-Angriffe (s. Abschnitt 2.2.1) erzielt wurden. Als Beispiel dient hier ein weiterer Angriff von Miller et al. [125], indem es den Forschern gelungen ist eine veränderte Firmware auf eine ECU zu flashen, um danach beliebige CAN-Botschaften im internen Fahrzeugnetzwerk zu senden. Die vierte Security-Eigenschaft der Verfügbarkeit hat mit 6 % den geringsten Anteil von den betrachteten Angriffen.

Definition 3.1.3 Reverse-Engineering

Als Reverse-Engineering werden Vorgänge bezeichnet, die versuchen, einzelne Komponenten und deren Zusammenhänge eines betrachteten Gesamtsystems ohne Detailwissen zu identifizieren. Im Fahrzeugbereich könnte dies eine Beobachtung des On-Board Netzwerkverkehrs darstellen, um übertragene Nutzdaten (z.B. Sensorwerte) sowie deren Absender und Empfänger zu ermitteln [129].

Definition 3.1.4 Kontext

Als Kontext werden innerhalb der IT verschiedene Arten von Informationen bezeichnet, die dazu eingesetzt werden können, um eine Situation einer Entität (z.B. Person oder Objekt) zu beschreiben [130]. In einer beispielhaften Interaktion zwischen zwei Entitäten (Benutzer und Server) definiert ein Kontext nun Informationen, welche einen Einfluss auf die Interaktion haben könnten (z.B. Zeitpunkt, Ort, gerade parallellaufende Zugriffe).

3.1.3 Analyse ausgewählter Angriffe

Für die Analyse und zur besseren Nachvollziehbarkeit ausgewählter Angriffe, dient eine exemplarische E/E-Fahrzeugarchitektur (s. Abbildung 3.5), die auf Basis verschiedener Hersteller abgeleitet ist und ein aktuelles Design repräsentiert. Die nachfolgend erläuterten Angriffe basierten teilweise auf davon abweichenden Architekturen, die als Vorgängerversionen gelten (s. auch Abschnitt 2.1.1). Jedoch ist diese Architektur als Referenzobjekt zur Erläuterung der von den Angreifern genutzten Netzwerkpfade geeignet. Darüber hinaus werden die ausgewählten Angriffe durch die zuvor beschrieben Angriffscharakteristiken klassifiziert (s. Tabelle 3.1), sowie entsprechende Angriffsklassen nach STRIDE (s. Tabelle 2.1) zugewiesen.

Adventures in Automotive Networks and Control Units 2013

Bereits im Jahr 2013 untersuchten die Forscher Miller et al. Möglichkeiten [125], die ein Angreifer nutzen könnte, um das Fahrverhalten gezielt zu beeinflussen. Dabei verwendeten sie zwei Fahrzeugtypen von unterschiedlichen Herstellern. Die gefundenen Schwächen bzw. ausgenutzten Schwachstellen beziehen sich dabei auf on-Board und Diagnose-spezifische CAN-Botschaften. Die E/E-Architektur der beiden Fahrzeuge basierte auf einem zentralisierten Konzept. Verglichen mit der in Abbildung 3.5 gezeigten Struktur sind keine Domänen-Controller vorhanden. Es werden lediglich zwei CAN-Busse verwendet, die über ein Gateway miteinander verbunden sind. Darüber hinaus gibt es keine Netzwerksegmentierung zwischen On-Board Netzwerk und

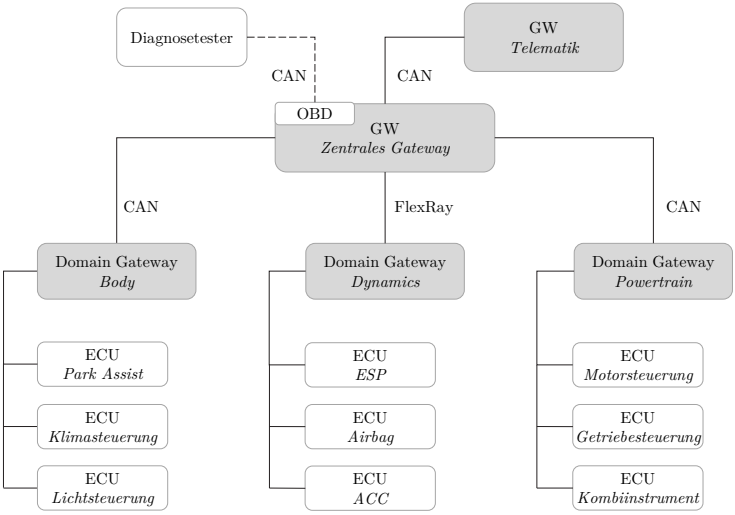


Abbildung 3.5: Beispielhafte E/E-Fahrzeugarchitektur basierend auf drei Domänen-Controllern (adaptiert von [RDBK18]).

OBD-Schnittstelle, sodass beide CAN-Netzwerke über diesen Zugriffspunkt direkt erreichbar sind. Da das CAN-Protokoll nach der ISO 11898 [131] keine Security-Funktionalitäten beinhaltet und die Hersteller keine zusätzlichen Security-Maßnahmen implementiert hatten, gelang es den Forschern verschiedene Fahrzeugsysteme wie beispielsweise die Lenkung oder Bremse über On-Board CAN-Botschaften (Angriffstyp: *Spoofing* und *Tampering*, s. Abschnitt 2.2.1) anzusteuern. Darüber hinaus wurde eine erhebliche Schwäche bzgl. des implementierten Security-Access (s. Abschnitt A.1.5) Algorithmus für die erweiterten Diagnosefunktionen (s. auch Abschnitt A.1.1) gefunden. Wurde ein Steuergerät mit einer Anfrage zum Wechsel in eine erweiterte Diagnosesitzung aufgefordert, sendete das beteiligte Steuergerät permanent den selben Seed zur Berechnung des passenden Schlüssels. Dies hatte zur Folge, dass ein Aufzeichnen der Kommunikation zwischen einem Diagnosegerät und dem Fahrzeug ausreichte, um den vom Diagnosegerät erzeugten Schlüssel mitzulesen. Durch dieses Paar - bestehend aus Seed und Schlüssel - ließ sich der verwendete Algorithmus rekonstruieren. Die dadurch ausnutzbare Schwach-

stelle wurde u.a. dazu verwendet, eine veränderte Firmware auf das Motorsteuergerät zu flashen. Allgemein ist die Überwindung des Security-Access als kritisch zu betrachten, da auf Steuergeräten kein Rechtemanagement existiert. Der Angreifer erlangt dadurch einen Vollzugriff (Lese- und Schreibzugriff) ohne weitere Beschränkungen.

Jeep Hack 2016 (OBD)

Im Jahr 2016 veröffentlichten die Forscher Miller und Valasek acht erfolgreiche Angriffe auf ein Fahrzeug [39]. Als Basis für den Angriff nutzten die Autoren verschiedene Diagnosefunktionen über einen physikalischen Zugriff der OBD-Schnittstelle. Als ersten Angriffsschritt wurde die Park-Assistenz ECU über eine Diagnosenachricht in den Zustand *Bootrom Mode* versetzt, der zur Beendigung einer aktiven Teilnahme am Netzwerkverkehr auf dem CAN-Bus führte. Dieser Schritt ermöglichte es ihnen, ihre eigenen Nachrichten (Angriffstyp: *Spoofing*, s. Abschnitt 2.2.1) anstelle der Park-Assistenz ECU zu senden, wodurch Funktionen wie beispielsweise das Auslösen der Parkbremse oder die Ansteuerung der Lenkung bei jeder Geschwindigkeit ausführbar waren. Normalweise würden die ausgenutzten Assistenzfunktionen nur bei Geschwindigkeiten $< 5km/h$ funktionieren. Durch die bewusste Störung der Assistenz-ECU wurde dieses Sicherheitsfeature umgangen. Zudem war die Security-Eigenschaft der Nachrichten-Authentifizierung (s. Abschnitt 2.2.1) nicht gewährleistest sowie keine darauf aufbauende Rechteverwaltung, die vom Fahrzeugzustand abhängige Berechtigungen kontrolliert und durchsetzt.

Definition 3.1.5 Bootrom Mode

Als Bootrom Mode wird eine erweiterte Diagnosesitzung bezeichnet, die es erlaubt eine neue Firmware auf das Steuergerät zu flashen [41].

Airbag Angriff 2017

Auf Basis der Diagnosefunktionalität von Fahrzeugen wurde im Jahr 2017 im Rahmen eines Security-*Penetrationstests* eine Schwachstelle in Bezug auf sicherheitskritische Airbag-Systeme gefunden und publiziert [DBRK17]. Als

Eintrittspunkt diente ebenfalls die OBD-Schnittstelle. Daneben gibt es bei Fahrzeugen, ab dem Jahr 2014 einen Standard, der die Implementierung einer *End of Life (EOL)* Funktionalität fordert, welche durch die ISO 26021 [132] spezifiziert ist. Der in der Norm exemplarisch dargestellte Security-Access Algorithmus (s. Abschnitt A.1.5) zur Absicherung der Diagnose EOL-Funktion wurde herstellerübergreifend ohne Veränderungen implementiert, wodurch die gefährliche Schwachstelle entstand. Demnach wurde für das dabei zugrundeliegende Challenge-Response Verfahren (Security-Access) entgegen dem Standard nur ein Byte statt zwei Bytes verwendet. Dadurch reduzierte sich die Anzahl der möglichen Schlüsselkombinationen gravierend von 2^{16} auf 2^8 Varianten.

Dazu kam eine zweite Implementierungsschwäche bzgl. dem verwendeten Algorithmus. Für eine Authentifizierung des Senders (z.B. Diagnosegerät) gegenüber dem Fahrzeug musste die empfangene Zufallszahl (Seed, Länge = 1 Byte) zur Erstellung des passenden Schlüssels lediglich durch die Anwendung des Einerkomplements verändert werden, um in eine erweiterte Diagnosesitzung (s. Abschnitt A.1.1) zu wechseln. Dieser Sitzungstyp ermöglichte die Aktivierung aller Zündladungen von Airbags sowie Gurtstraffer des Fahrzeugs. Zudem war eine derartige Funktionsausführung bei manchen Fahrzeugmodellen auch während der Fahrt möglich, da keine aktuelle Geschwindigkeit oder weitere Sensorwerte wie z.B. die Sitzbelegung oder der Zustand des Gurtschlosses als zusätzliche Absicherungsmethoden miteinbezogen wurden.

Definition 3.1.6 Penetrationstest

Als Penetrationstest wird eine Security-Testmethode bezeichnet, wobei der Tester nur ein beschränktes Wissen über die zu testende Anwendung oder System hat. Der Tester agiert dadurch aus Sicht eines Angreifers und versucht über nach außen sichtbare Schnittstellen in das System einzudringen [133].

Tabelle 3.1: Übersicht ausgewählter Angriffe auf Fahrzeuge.

Referenz	[125]	[39]	[DBRK17]
Schnittstelle	OBD	OBD	OBD
Protokoll	CAN, Diagnose	CAN, Diagnose	CAN, Diagnose
Safety-kritisch	ja	ja	ja
Verletzte Security-Eigenschaft	Authentifizierung, Integrität	Authentifizierung	Authentifizierung
Angriffstyp	Spoofing, Tampering	Spoofing	Spoofing, Information Disclosure
Ausführung	lokal	lokal	lokal

Definition 3.1.7 EOL (Diagnose)

Die EOL Funktionalität beinhaltet die Aktivierung von pyrotechnischen Einrichtungen (z.B. Airbags, Gurtstraffer) bei Fahrzeugen am Ende des Lebenszyklus über eine standardisierte Diagnosefunktion [132].

Automotive Insider Angriffe

Ein aus informationstechnischer Sicht gefährlicher Systemzustand tritt ein, wenn Angreifer auf Basis eines Insiders agieren (s. Abschnitt 2.2.1). Diese Art von Angriffen lassen sich im automotive Bereich durch verschiedene, bereits aufgetretene Vorfälle erläutern. Aus Sicht des betrachteten Systems (Fahrzeugarchitektur) stellt diese Art einen Angriff dar, bei dem vorhandene Schutzmaßnahmen durch erlangtes Insider-Wissen (z.B. Zertifikate, Passwörter, Berechtigungen) umgangen werden.

In 2013 gelang es beispielsweise Miller et al. [125] ein Steuergerät mit manipulierter Konfiguration zu flashen. Die Voraussetzung für diesen erfolgreichen Angriff bildete eine Schwachstelle innerhalb des Security-Access (s. Abschnitt A.1.5), die den Angreifern eine erweiterte Diagnosesitzung (s. Abschnitt A.1.1) ermöglichte. Durch diese Berechtigungen waren sie in der Lage, einen manipulierten CAN-Kommunikationsablauf zu flashen bzw. auszuführen. Da sich

diese ECU im internen Fahrzeugnetzwerk befand und von den anderen Netzwerkteilnehmern als legitimierter Netzwerknoten angesehen wurde, erlangten die eigentlichen Nicht-Insider, einen Insider-Status. Andere Knoten konnten nicht unterscheiden, ob die versendeten CAN-Botschaften des kompromittierten ECUs von einem Angreifer oder der eigentlichen Funktion stammen.

Ein weiterer Vorfall ereignete sich im Jahr 2018 [134] als ein Mitarbeiter eines Automobilherstellers interne Dokumente aus der Entwicklung an unbefugte Dritte weitergab. Dieser Vorfall hatte zunächst einen rein wirtschaftlichen Schaden zur Folge, da geistiges Eigentum unautorisiert weitergeben wurde. Jedoch wäre daraus auch ein konkreter Angriff auf Fahrzeuge denkbar, falls in diesen Dokumenten z.B. vertrauliches Schlüsselmaterial für Remote-Zugänge enthalten wäre.

Ein ähnlicher Vorfall ereignete sich im Jahr 2020 erneut als ein Automobilhersteller ein online erreichbares Software-Repository nicht umfassend absicherte [135]. Dadurch war es Dritten möglich, sich für einen Zugang ohne weitere Prüfung zu registrieren und dadurch einen Insider-Status zu erlangen. Der dort hinterlegte Quellcode stammte von einem Konnektivitäts-Steuergerät, dass Fahrzeuge mit der Cloud des Fahrzeugherstellers verbindet und dadurch einen Fernzugriff ermöglicht. Darin enthalten waren u.a. Schnittstellenbeschreibungen für interne Herstellersysteme, Passwörter sowie weitere Anmeldeinformationen.

3.1.4 Zusammenfassung und Diskussion - Automotive Angriffe

Die analysierten Angriffe zeigen, dass die jeweiligen E/E-Architekturen keine ausreichenden Security-Schutzmaßnahmen beinhalteten. Die dabei nicht erfüllten Schutzziele wurden in Abschnitt 3.1 erläutert. Des Weiteren zeigte sich, dass 39 % der Angriffe im betrachteten Zeitraum über die OBD-Schnittstelle ausgeführt wurden, die dadurch lokal-basierte Angriffe darstellen. Zudem zeigte sich durch eine Untersuchung von Wen et al. [136], dass drahtlose *OBD-Dongles* die teilweise zum *Management von Flottenfahrzeugen* zum Einsatz kommen, erhebliche Schwachstellen aufweisen. Dafür analysierten die Forscher insgesamt 77 Dongles verschiedener Hersteller und kamen zu dem Ergebnis, dass 85 % der Dongles keine Verbindungs- oder Anwen-

dungsauthentifizierung enthielten. Des Weiteren erlauben 37 % der Dongles einen direkten Zugriff auf den CAN-Bus der OBD-Schnittstelle. Auch das BSI listet diese Untersuchung im Lagebericht zur Cyber-Sicherheit in der Automobilbranche [31]. Bezogen auf die Airbag-Schwachstelle (s. Abschnitt 3.1.3) kann dadurch angenommen werden, dass aus einem lokal-basierten Angriff auch ein Remote-Angriff erfolgen könnte. Des Weiteren wäre ein skalierbarer Angriff auf komplette Fahrzeugflotten möglich, die für die Übertragung von Fahrzeugdaten mit OBD-Dongles ausgestattet sind [137].

Die durchgeführte Analyse hat folgende Untersuchungsergebnisse ergeben:

- Die OBD-Schnittstelle wurde von den untersuchten Angriffen am häufigsten als Eintrittspunkt für das Ausnutzen von Schwachstellen in der On-Board Kommunikation von Fahrzeugen verwendet.
- In bisherigen Fahrzeugsystemen gab es keine zuverlässige Zugriffskontrolle, die Berechtigungen von Fahrzeugfunktionen kontrolliert und durchsetzt.
- Die Security-Eigenschaft Authentizität wurde am häufigsten bei den untersuchten Angriffen verletzt (37 %-Anteil). Diese steht wiederum im engen Zusammenhang mit der Autorisierung.
- Bisherige Systeme boten keine Möglichkeiten, Angriffe während ihres Auftretens zu erkennen (IDS-Systeme).

Definition 3.1.8 OBD-Dongle

Als OBD-Dongle wird ein Adapter bezeichnet, der mit der OBD-Schnittstelle des Fahrzeugs verbunden werden kann, um Diagnosedaten von Fahrzeugen auszulesen. Der Adapter kann über drahtlose Schnittstellen wie Bluetooth oder WLAN mit Smartphones und zugehörigen Apps gekoppelt werden [136].

Definition 3.1.9 Management von Flottenfahrzeugen

Das Management einer Fahrzeugflotte von Unternehmen umfasst jeden Aspekt des Lebenszyklus eines Fahrzeugs von der Beschaffung bis zur Entsorgung [138]. Im laufenden Betrieb übermitteln OBD-Dongles den Unternehmen verschiedene Fahrzeugdaten wie z.B. Fahrstil, zurückgelegte Strecken oder Kraftstoffverbrauch. Diese Daten werden von Unternehmen analysiert, um beispielsweise die Umweltbilanz oder Betriebskosten optimieren zu können.

3.2 Stand der Technik und Wissenschaft

Die analysierten Angriffe (s. Abschnitt 3.1) zeigen bisherige Sicherheitsprobleme in Fahrzeugsystemen auf. Da der Vernetzungsgrad bei zukünftigen Fahrzeugen durch neue E/E-Architekturen weiter zunehmen wird [18] (s. auch Abschnitt 2.1.1) steigt parallel auch das Risiko für neue Schwachstellen. Aus diesem Grund müssen bekanntgewordene Angriffe kontinuierlich analysiert werden, um geeignete Schutzmaßnahmen ableiten zu können. Die Sicherheitssysteme müssen einerseits eine präventive Wirkung erzielen (z.B. durch Firewalls oder Zugriffskontrollen, s. Abschnitte 3.2.1 und 3.2.2), um das Risiko für einen erfolgreichen Angriff zu reduzieren. Auf der anderen Seite müssen reaktive Systeme integriert werden (z.B. IDS, s. Abschnitt 2.2.4), um Angriffe während ihres Auftretens erkennen zu können, um darauf basierend entsprechende Abwehrmaßnahmen wie beispielsweise die Änderung einer Firewall-Konfiguration oder die Anpassung von Berechtigungen einer Zugriffskontrolle vorzunehmen (s. Abschnitt 3.2.3). Neben veröffentlichten Ansätzen aus der Wissenschaft gibt es auch Weiterentwicklungen im Bereich von Guidelines, Regularien und Standards in Bezug auf die Informationssicherheit im Automobil, die bei Forschung und Entwicklung von neuen Konzepten eine Rolle spielen (s. Abbildung 3.6)[2].

3.2.1 Automotive Firewalls

Durch die Angriffe auf Fahrzeuge der letzten Jahre (s. Abschnitt 3.1), wurden als Gegenmaßnahmen verschiedene Ansätze zu Firewalls (s. Abschnitt 2.2.3) für automotive Netzwerke publiziert, um Netzwerknachrichten durch verschiedene Filtertechniken zu kontrollieren. Im Jahr 2014, präsentierten Seifert et al. [139] eine Firewall für Fahrzeuggateways, die Kommunikationsdaten auf Basis von spezifizierten Zeitgrenzen filtert. Die Autoren verwenden dafür einen zeit-beschrifteten Zustandsautomat (engl. timed automaton), um ein spezifi-

[2] Teile der nachfolgenden Abschnitte wurden in [RGKS20] veröffentlicht. Die gelisteten und beschriebenen Ansätze der Technik u. Wissenschaft basieren auf einer Recherche in den Datenbanken *IEEE Xplore* sowie *Google Scholar*. Dabei wurden explizit Indexbegriffe wie CPS Security, Automotive Firewalls, RBAC u. ABAC, IDS verwendet, um verwandte Arbeiten zu Themenschwerpunkten dieser Arbeit zu identifizieren.

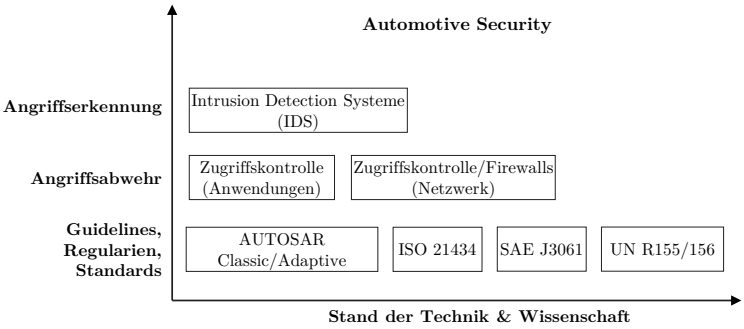


Abbildung 3.6: Übersicht von Automotive Security Aspekten in Bezug auf den aktuellen Stand der Technik und Wissenschaft.

ziertes Zeitverhalten der Fahrzeugkommunikation abzubilden. Dadurch ist dieser Ansatz ausschließlich für statisch definierte Netzwerke (s. Abschnitt 2.1.2) einsetzbar.

Ein weiterer Firewall-Ansatz wurde von Pesé et al. [140] basierend auf dem Ethernet-Protokoll beschrieben, die sowohl Software- als auch Hardwareanforderungen für den Einsatz im Fahrzeug untersuchten. Für die Filterung von Angriffen ist der Ansatz dreistufig aufgebaut. Auf unterster Ebene wird ein klassischer Paketfilter verwendet, der auf einem Field Programmable Gate Array (FPGA) implementiert ist. Über Softwarealgorithmen werden zwei weitere Stufen erweitert, die Möglichkeiten zur Durchsatzratenbegrenzung (engl. ratelimiting) und zustandsbasierten Filterung ermöglichen. Die von ihnen durchgeführte Evaluierung enthält u.a. verschiedene Untersuchungen zu Ende-zu-Ende Laufzeiten, Datendurchsatzraten sowie Speicherverbrauch.

Im Bereich von Automotive Ethernet publizierten Holle et al. [141] eine Firewall zur Integration in Gateways, die sowohl einen Paket- als auch einen Applikationsfilter enthält. Die Autoren gehen dagegen nicht im Detail auf den implementierten Algorithmus ein. Darüber hinaus war der Ausgangspunkt für die Entwicklung einer Gateway-Firewall von Luo et al. [142] eine durchgeführte Bedrohungs- und Risikoanalyse einer exemplarischen Fahrzeugarchitektur. Die Forscher analysierten danach unterschiedliche Angriffspfade über Eintrittspunkte wie beispielsweise Sensoren, Schnittstellen oder Steuergeräte.

Tabelle 3.2: Vergleich von publizierten automotive Firewall-Ansätzen.

Referenz	[139]	[140]	[141]	[142]
Domäne	Automotive	Automotive	Automotive	Automotive
Filterart	Paket	Paket	Paket, Applikation	Paket
Protokoll	CAN, Ethernet, FlexRay	Ethernet	Ethernet	CAN, Ethernet
Ort	Gateway	Gateway	Gateway	Gateway
Kontext	-	-	-	-
Filtertechnik	statisch	dynamisch	dynamisch	dynamisch

Darauf basierend leiteten sie ihr Firewall-Konzept ab, das neben zustandsbasierter Paketfilterung auch IDS-Funktionen bietet, die Anomalien auf Basis von Informationsentropie erkennt.

Die beschriebenen Ansätze lassen sich über verschiedene Eigenschaften klassifizieren (s. Tabelle 3.2). Neben der Filterart, dem gefilterten Protokoll sowie dem Ort wo die Firewall in der E/E-Architektur integriert wird, gibt es zwei weitere relevante Unterscheidungsmerkmale. Hier handelt es sich um die Fragen, ob bzw. welche Kontextinformationen bei der Filterung miteinbezogen werden. Allerdings beschreibt keiner der untersuchten Ansätze eine Kontext-basierte Filterung, wie beispielsweise auf Basis des aktuellen Fahrzeugzustands.

3.2.2 Zugriffskontrolle

Die Auswertung der verletzten Security-Eigenschaften (s. Abschnitte 3.1.2 u. 3.1.3) bei Angriffen der letzten Jahre zeigt die bisher unzureichende Gewährleistung der Authentizität sowie Autorisierung in Fahrzeugsystemen. Dadurch konnten Angreifer durch eine Kompromittierung eines Steuergerätes bzw. einer darauf integrierten Applikation ihre Rechte uneingeschränkt ausweiten und die vollständige Kontrolle erlangen. Durch ein implementiertes Rechtemanagement wäre der Zugriff nur auf die jeweilige übernommene Applikation beschränkt gewesen (s. auch Abschnitt 3.1.3) [143]. Aufgrund dieser fehlenden Schutzmaßnahme wurden im Bereich der Wissenschaft verschiedene Ansätze für eine Zugriffskontrolle publiziert. Die Autoren Kim et al. [144] stellten einen Ansatz für eine dezentrale Autorisierung von Funktionen basierend auf ABAC

(s. auch Abschnitt 2.2.2) vor, der als zusätzliches Modul in die AUTOSAR Classic Plattform integriert wurde. Die für die Zugriffskontrolle verwendeten Attribute enthalten Informationen zu CAN-Nachrichten, ECU Funktionen sowie Umgebungseigenschaften (Empfänger ECUs, Nachrichtenfrequenz), die auf den unterschiedlichen Netzwerkknoten über ACLs (s. Abschnitt 2.2.2) abgebildet sind.

Ein weiterer Ansatz wurde von Hamad et al. [145] publiziert, der sich mit einer policy-basierten Kommunikation befasst. Das Konzept beschreibt einen vertrauenswürdigen Prozess zur Entwicklung einer Sicherheitsrichtlinie mit unterschiedlichen Stakeholdern (OEMs und Zulieferer). Die Richtlinie beinhaltet dabei die Spezifizierung von Security-Eigenschaften (z.B. Vertraulichkeit oder Integrität) sowie Protokollvoraussetzungen für Schnittstellen von Services. Darüber hinaus wurde ein Security-Modul entworfen, das auf jeder ECU im Fahrzeug als verteilte Proxy-Firewall (s. Abschnitt A.1.4) für TCP/IP Kommunikation integriert werden kann. Unter Verwendung dieses Moduls ist es möglich eine fein-granulare Zugriffskontrolle auf Basis von IP-Adressen, Ports sowie spezifizierten Security-Eigenschaften für jede Applikation in Abhängigkeit zur hinterlegten Sicherheitsrichtlinie zu kontrollieren.

Die Autoren Gupta et al. [146] fokussieren ihre Forschung auf den stetig wachsenden Bereich der Konnektivität von Fahrzeugen mit der Umwelt und daraus entstehenden Risiken für neue Schwachstellen in Bezug auf die Informationssicherheit. Dabei erläutern die Forscher zuerst verschiedene Architekturvarianten und beschreiben unterschiedliche Modelle der Zugriffskontrolle (ACLs, Capability-based Access Control (CapBAC), ABAC) für statische und dynamische Kommunikation. Danach erfolgt eine ausführliche Beschreibung wie diese Ansätze in die beschriebenen Architekturen integriert werden könnten.

Ein Ansatz innerhalb der CPS-Domäne im Bereich der intelligenten Stromnetze (engl. smart grids) wurde von Ruland et al. [147] vorgestellt. Die Autoren beschreiben in ihrer Arbeit eine Firewall basierend auf einem ABAC-Zugriffsmodell die Berechtigungen über XACML Sicherheitsrichtlinien (s. Abschnitt 2.2.2) kontrolliert und durchsetzt. Zusätzlich enthält der Beitrag eine domänenspezifische Architektur, die zeigt, wie die vorgestellten ABAC-Module integriert werden können. Ein verwandter Ansatz im Bereich von automatisierten Industrieanlagen, die ebenfalls der CPS-Domäne zugeordnet sind, wurde von Huh et al. [148] publiziert. Die Forscher zeigen darin aktuelle Herausforderungen bzgl. der Informationssicherheit auf und adressieren den Be-

reich der Zugriffskontrolle durch die Integration eines RBAC-Zugriffsmodells in eine verteilte Industrieanlagensteuerung. Zusätzlich werden für Zugriffsentscheidungen der aktuelle Ort sowie der Gerätetyp (z.B. fest installierte Anlage oder mobiles Gerät) ausgewertet.

Einen Ansatz für eine Zugriffskontrolle innerhalb einer automotive Middleware wurde von Hugot et al. [149] publiziert. Die Autoren benennen in ihrer Arbeit die Schwächen von bisherigen automotive Steuergeräten in Bezug auf ein fehlendes Rechtemanagement. Darüber hinaus führen die Forscher an, dass eine effektive Kontrolle der Berechtigungen ab ISO/OSI Schicht 5 (s. Abschnitt A.1.1) erfolgen sollte, um einen anwendungsbezogenen Kontext miteinbeziehen zu können.

Definition 3.2.1 Kontextbezogene automotive Security

Die Nutzung von kontextbezogenen Informationen für Security-Entscheidungen führt zu genauen, effizienten und zeitnahen Sicherheitsentscheidungen, die dynamische Fahrzeug- und Flottenumgebungen unterstützen [150].

Definition 3.2.2 Fahrzeugzustand

Der Fahrzeugzustand stellt eine kontextbezogene Information im Fahrzeug dar. Die Informationen können dabei bestimmte Funktionszustände einer Software (z.B. Steuergerät befindet sich im Diagnosemodus) umfassen oder ein oder mehrere Sensorinformationen (z.B. Sitzbelegung, Raddrehzahl) darstellen, die den Zustand (z.B. Fahrzeug in Bewegung, aktuelle Sitzbelegung) des Fahrzeugs genauer beschreiben.

Außerdem sollte die Zugriffskontrolle bei Zugriffsentscheidungen immer den aktuellen Fahrzeugzustand berücksichtigen. Als Beispiel nennen die Autoren eine Kommunikation zwischen einer Adaptive Cruise Control (ACC) Fahrerassistenzfunktion sowie einer Motorsteuerungseinheit, die nur dann erlaubt sein sollte, wenn sich die ACC Funktion im eingeschalten Zustand befindet. Die Zugriffsentscheidungen werden dadurch dynamisch getroffen. Das präsentierte Konzept basiert dabei auf einem MAC Zugriffsmodell (s. Abschnitt 2.2.2), dass die Datenströme zwischen Sendern und Empfängern unter Verwendung eines

Tabelle 3.3: Vergleich publizierter Ansätze für die Zugriffskontrolle innerhalb von CPS.

Referenz	**[144]**	**[145]**	**[146]**	**[147]**	**[148]**	**[149]**
Domäne	Automotive	Automotive	Automotive	Smart Grids	Automatisierung	Automotive
Modell	ABAC	eigen	CapBAC, ABAC	ABAC	RBAC	eigen
Technik	Policies	Policies	ACLs, Policies	Policies	ACL	Policies
Kontext	-	-	-	Zeit	Ort, Gerätetyp	Fahrzeugzustände

Zustandsautomaten kontrolliert. In diesem Automaten werden die jeweiligen Zustände des Fahrzeugs abgebildet, um dadurch unerlaubte Zustandsübergänge zu verhindern. Des Weiteren wurde die Zugriffskontrollmethode auf eine service-orientierte Kommunikation auf Basis des SOME/IP Protokolls (s. Abschnitt A.1.1) abgebildet und drei unterschiedliche Implementierungsvorschläge erläutert.

Die erläuterten Ansätze der Zugriffskontrolle können durch verschiedene Charakteristika miteinander verglichen werden, um Unterscheidungsmerkmale zu identifizieren (s. Tabelle 3.3). Dabei werden die in Abschnitt 2.2.2 beschriebenen Zugriffsmodelle und Techniken verwendet. Zudem wird dargestellt, ob bzw. welche Art von Kontextinformationen für die Entscheidung von Zugriffsanfragen miteinbezogen werden.

3.2.3 Intrusion Detection Systeme

Im Bereich der On-Board Kommunikation von Fahrzeugen existieren verschiedene IDS Ansätze (s. Tabelle 3.4), um Angriffe zu erkennen. Die Forscher Al-Jarrah et al. präsentierten in [103] eine wissenschaftliche Analyse der bisherigen Ansätze in diesem Forschungsbereich. Dazu klassifizierten sie 42 Methoden in zwei unterschiedliche Erkennungstechniken (Cyber und Physical). Die Begründung dieser Einteilung lässt sich darauf zurückführen, dass aktuelle Fahrzeuge ein CPS darstellen und dadurch beide Features beinhalten. *Cyber-Features* repräsentieren Kommunikations- und Protokolleigenschaften

wie beispielsweise die Datenrate, die zeitliche Abfolge von Nachrichten oder Signalcharakteristiken. Daneben werden durch *Physical-Features* bestimmte Sensorsignale oder eine Kombination davon abgebildet die beispielsweise die aktuelle Geschwindigkeit darstellen oder Rückschlüsse auf den aktuellen Fahrzeugzustand ermöglichen. Die Autoren kommen zu dem Ergebnis, dass 34 Ansätze auf Cyber-Features basieren. Ein IDS-Ansatz, der diese Features verwendet und eine hybride Struktur aufweist, wurde beispielsweise von Weber et al. [151] vorgestellt. Die erste Erkennungstechnik nutzt das bekannte Wissen aus der Kommunikationsspezifikation, die von OEMs während der Entwicklung definiert wird und prüft diese auf Basis von statischen Analysen. Die zweite verwendete Technik ist dagegen als lernende Methode aufgebaut, die Signalverläufe auf deren Plausibilität analysiert. Physikalisch bedingt kann beispielsweise ein Geschwindigkeitssignal auf Grund der limitierten Beschleunigung eines Fahrzeugs nicht sprunghaft ansteigen. Für das Training des genutzten Machine-Learning Algorithmus wurde ein reales CAN-Datenset verwendet.

IDS-Ansätze, die Physical-Features verwenden, lassen sich weiter in physikalische Charakteristiken (ISO/OSI Layer 1, s. Abschnitt A.1.1) oder genutzte Sensorwerte zum Erfassen von Kontext und Semantik in Bezug auf den aktuellen Fahrzeugzustand unterteilen. Die erste Variante wird u.a. in [152], [RDB+19], [153] verwendet, die physikalische Signaleigenschaften für die Erkennung von Anomalien verwenden. Als Beispiel kann an dieser Stelle *Scission* [152] genannt werden, das mittels einer Überabtastung die Signalpegel jeder versendeten CAN-Nachricht als Normalverhalten einlernen kann und damit einen Fingerabdruck für jede legitimierte ECU erstellt. Im Fall eines Angriffs durch ein zusätzlich eingefügtes Steuergerät am CAN-Bus, kommt es bei der Versendung von CAN-Nachrichten zu Abweichungen bzgl. der Signalpegel, wodurch die gefälschte Nachricht erkannt wird.

Des Weiteren ergab die Untersuchung von Al-Jarrah et al., dass nur zwei der Ansätze die Erkennung von Anomalien auf Basis von Kontext und Semantik der übertragenen Informationen durchführen. Die Autoren Wasicek et al. [154] verwenden in ihrem Ansatz einen kontext-abhängigen Erkennungsalgorithmus, der auf Sensorreferenzmodellen basiert. Dafür werden verschiedene statistische Kenngrößen (z.B. Varianz oder Standardabweichung) von unterschiedlichen Sensorquellen ermittelt und als Datenset einem Künstlich Neuronalen Netz (KNN) 2.3.1 als Trainingsdaten übergeben. Im realen Fahr-

zeugbetrieb kann dann analysiert werden, ob im jeweiligen Betriebszustand die Sensorwerte plausibel sind. Die Evaluierung des Ansatzes wurde über eine Anomalie des Chip-Tunings durchgeführt, die eine erhöhte Menge Kraftstoff bei der Einspritzung verursachte und erfolgreich erkannt wurde.

Darüber hinaus adressieren Kalutarage et al. [155] ebenfalls das Forschungsfeld einer kontext-abhängigen Anomalieerkennung. Der Ansatz basiert auf der Berechnung von Auftrittswahrscheinlichkeiten durch die Verwendung von N-Grammen (s. Abschnitt 2.3.2), die auf übertragene CAN-Botschaften angewendet werden. Die Nachrichten werden dafür in definierten Zeitfenstern ausgewertet, um für die beteiligten Identifier die entsprechenden Wahrscheinlichkeiten zu bestimmen. Diese Methode ermöglicht eine Auswertung, welche CAN-Botschaften in einem definierten Zeitraum zueinander in Bezug stehen. Eine vom Angreifer nicht zum aktuellen Kontext passende eingeschleuste CAN-Botschaft würde eine sprunghafte Veränderung der Auftrittswahrscheinlichkeit verursachen und wird dadurch als Anomalie erkannt. Der Ansatz betrachtet dabei ausschließlich die CAN-Identifier. Würde der Angreifer hingegen die Nutzdaten manipulieren, bliebe der Angriff unerkannt.

Die Anwendung der N-Gramm Methode im Bereich der Anomalieerkennung wurde auch bereits in anderen Domänen wie beispielsweise der klassischen IT gezeigt. Die Forscher Khreich et al. [156] verwendeten N-Gramme zur Überwachung von Systemlaufzeiten. Dafür werden mehrere N-Gramme mit unterschiedlicher Länge für die Bestimmung der Auftrittswahrscheinlichkeiten der Systemaufrufe eingesetzt. Eine One-Class Support Vector Machine überwacht darüber hinaus die Wahrscheinlichkeiten und prüft diese auf Anomalien. Eine weitere Forschungsarbeit im Bereich von N-Grammen wurde von Zolotukhin et al. [157] vorgestellt, die diese mit Growing Hierarchical Self-Organizing Maps (GHSOMs) [158] kombinieren, um Anomalien in Hypertext Transfer Protocol (HTTP) Netzwerkanfragen zu erkennen.

Die publizierten Ansätze können dabei grundlegend in deren Anwendungsdomäne sowie genutzten Features klassifiziert werden (s. Tabelle 3.4). Darüber hinaus ist in der Klassifizierung auch enthalten, ob der jeweilige Ansatz auch Kontextinformationen (z.B. aktuelle Sensorinformationen eines Fahrzeugs) für die Anomalieerkennung mit einbezieht und auf welcher Basis die Evaluierung durchgeführt wurde (beispielsweise reale oder synthetische Daten).

Tabelle 3.4: Vergleich publizierter IDS-Ansätze

Referenz	[151]	[152]	[154]	[155]	[156]	[157]
Domäne	Automotive	Automotive	Automotive	Automotive	IT	IT
Feature	Cyber	Physical	Physical	Cyber	Cyber	Cyber
Protokoll/ Anwendung	CAN	CAN	CAN	CAN	System-laufzeiten	HTTP
Kontext	nein	nein	ja	ja	ja	ja
Evaluierung	Synthetische Daten	Reale Hardware	Reale Daten	Reale Daten	Reale Daten	Reale Daten

3.2.4 Automotive Security - Guidelines/Regularien/Standards

Durch die publizierten Angriffe in den letzten Jahren haben die Automobilbranche, Standardisierungsorganisationen als auch staatliche Kommissionen verschiedene Standards und Regularien spezifiziert bzw. beschlossen (s. Tabelle 3.5).

Industriestandards

Im Rahmen des AUTOSAR Standards wurde begonnen, Protokolle sowie Security-Funktionen zu spezifizieren. Dadurch wurde beispielsweise das SecOC-Modul (s. Abschnitt A.1.5) zur Absicherung der CAN-Kommunikation eingeführt, um die Security-Eigenschaften der Authentizität und Integrität (s. Abschnitt 2.2.1) durch Nachrichtenauthentifizierungscodes (s. Abschnitt A.1.3) abzusichern. Darüber hinaus wurde für die Automotive Ethernet Kommunikation (s. Abschnitt A.1.1), der IT-Standard Internet Protocol Security (IPSec) [159] zur Absicherung von IP-Paketen adaptiert. Dabei gibt es zwei verschiedene Varianten. Einerseits wird auf Basis des Authentication Header (AH) das gesamte IP-Paket signiert und dadurch eine Authentifizierung des Senders sowie die Authentizität und Integrität der Pakete gewährleistet. Auf der anderen Seite werden mittels des Encapsulating Security Payload (ESP) die Nutzdaten verschlüsselt, um zusätzlich die Vertraulichkeit zu gewährleisten. Des Weiteren wurde aus der IT ebenfalls der Transport Layer Security (TLS)

Standard zur Absicherung der Anwendungsdaten (ISO/OSI Schicht 7, s. Abschnitt A.1.1) übernommen und spezifiziert. Dabei bietet dieses Protokoll eine Sender- und Empfängerauthentifizierung auf der Basis von digitalen Signaturen sowie asymmetrische und symmetrische Verschlüsselungsverfahren (s. Abschnitt A.1.2) für die Sicherstellung einer vertraulichen Verbindung.

Daneben erfolgte mit der Einführung der Version 18-03 [160] des AUTOSAR Adaptive Standards die Spezifikation des Identity und Access Management (IAM) Moduls, das für die service-orientierte Kommunikation ein Rechtemanagement ermöglicht. Bis zu diesem Zeitpunkt existierte lediglich für Diagnoseanwendungen eine eingeschränkte Rechteverwaltung (Security-Access, s. Abschnitt A.1.5). Für jeden Service (Subjekt) kann eine Liste an Berechtigungen für den Zugriff auf andere Services (Objekte) in einer Manifestdatei hinterlegt werden. Dadurch folgt die Spezifikation der Zugriffsmethode auf Basis von Zugriffsausweisen (engl. capabilties, CapBAC, s. auch Abschnitt 2.2.2). Möchte ein Service während der Laufzeit auf einen anderen Service zugreifen, erfolgt die Prüfung der notwendigen Berechtigungen sowohl auf der Sender-ECU als auch auf der Empfänger-ECU mittels einem PDP (s. auch Abschnitt 2.2.2). Die vom PDP getroffene Zugriffsentscheidung wird danach über einen spezifizierten PEP durchgesetzt. Daneben wurde durch die Aufnahme des Protokolls Data Distribution Service (DDS) [161] in AUTOSAR die Möglichkeit geschaffen, die bereits in DDS verankerte Zugriffskontrolle zu verwenden [162].

Regularien & Standards

Im Jahr 2016 wurde das *Cybersecurity Guidebook for Cyber-Physical Systems* J3061 [36] von der SAE publiziert. Dieser Leitfaden adressiert dabei die Thematik der Einführung eines Security-Entwicklungsprozesses und vermittelt Vorschläge über konkrete Prozessschritte. Zudem befasst sich das Werk mit Methoden und Werkzeugen (z.B. Bedrohungs- und Risikoanalysen), die für die Absicherung von Fahrzeugen verwendet werden können. Daneben wird über verschiedene Erläuterungen zu beispielsweise Bedrohungen, Schwachstellen oder Schutzmaßnahmen ein Basiswissen im Bereich der Informationssicherheit vermittelt.

Tabelle 3.5: Übersicht von existierenden bzw. (geplanten) Guidelines, Regularien und Standards in der Automotive-Domäne.

Referenz	**AUTOSAR Classic**	**AUTOSAR Adaptive**	**ISO 21434**	**SAE J3061**	**UN R155/156**
Art	Industrie-Standard	Industrie-Standard	Standard	Guideline	Regulierung
Einführung	2005	2017	2021	2016	2022
Bereich	SW-Architektur (signal-orientiert)	SW-Architektur (service-orientiert)	Informations-sicherheit	Informations-sicherheit	Informations-sicherheit
Fokus	SW-Entwicklung	SW-Entwicklung	Entwicklungs-prozess	Entwicklungs-prozess	gesamter SW-Lebenszyklus

Mit der ISO 21434 [37] ist seit dem Jahr 2021 ein erster Cybersecurity-Standard für Fahrzeuge verfügbar. Der Standard definiert hauptsächlich die Security-Aktivitäten und Nachverfolgbarkeit im Entwicklungsprozess, die dabei in Abhängigkeit des ermittelten Sicherheitsrisikos der jeweiligen Funktion gesteuert sind. Darüber hinaus werden auch bestimmte Anforderungen definiert, die beispielsweise die Durchführung eines kontinuierlichen Security Monitorings umfassen. Hingegen sind keine konkreten Implementierungsvorschläge zu bestimmten Security-Maßnahmen enthalten.

Im Jahr 2022 wird die erste gesetzliche Verordnung für Cybersecurity und Software-Updates in der Automobilbranche in Kraft treten. Dabei handelt es sich um die von der *World Forum for Harmonization of Vehicle Regulations (WP.29)* Gruppe verabschiedeten Regelwerke UN R155 [163] und UN R156 [164] der UNECE. Darin werden Fahrzeughersteller in Mitgliedsländern der UNECE verpflichtet, bestimmte Vorgehensweisen in Bezug auf die Informationssicherheit während der Entwicklung nachweisbar umzusetzen, um eine Typgenehmigung [165] für neue Fahrzeuge zu erhalten. Darüber hinaus gilt diese Verordnung während des gesamten Lebenszyklus, wodurch Hersteller auch verpflichtet werden, einen zuverlässigen Updateprozess für gefundene Schwachstellen zu etablieren.

Definition 3.2.3 Typgenehmigung

Im Typgenehmigungsverfahren wird ein bestimmter Typ eines Fahrzeugs durch einen technischen Dienst einer länderspezifischen Behörde überprüft, ob alle geltenden Anforderungen erfüllt sind. Wird dieser Prozess erfolgreich durchlaufen erhält der Hersteller eine Typgenehmigung. Damit wird dieser ermächtigt alle produzierten Fahrzeuge dieses Typs in den Markt einführen zu dürfen [166].

3.2.5 Zusammenfassung und Diskussion - Stand der Technik und Wissenschaft

Durch die zunehmende Sensibilisierung der Industrie und Wissenschaft für das Thema Informationssicherheit in Fahrzeugen, u.a. durch publizierte Angriffe (s. Abschnitt 3.1), wurde in den letzten Jahren sowohl an Konzepten als auch an Standards bzw. Regularien für die Entwicklung und den Betrieb zukünftiger Systeme gearbeitet.

Firewalls: Die analysierten Firewall-Ansätze verfolgen grundlegend die Security-Eigenschaft der Autorisierung auf Netzwerkebene. Dadurch kann beispielsweise verhindert werden, dass nicht spezifizierte CAN-Nachrichten von einer Fahrzeugdomäne in eine andere geroutet wird (z.B. von Powertrain- zu Chassis-Domäne, s. Abbildung 3.5). Des Weiteren ist eine Überprüfung und Filterung von Nachrichten zwischen verschiedenen Schnittstellen (s. auch 3.1.1) möglich. So kann beispielsweise das Risiko minimiert werden, dass ein Angreifer CAN-Botschaften über die OBD-Schnittstelle in die On-Board Kommunikation einschleusen und safety-kritische Fahrfunktionen ansteuern (s. auch 3.1.3). Bezogen auf die Angriffsklassifikation nach *STRIDE* (s. auch Abschnitt 2.2.1) können Firewalls, einen Schutz gegen *Tampering* sowie *Denial of Service (DoS)* Angriffe bieten. Jedoch kann diese Maßnahme keine vollständige Sicherheit gegen diese Art von Angriffen bieten. So kann eine im zentralen Gateway bzw. Domain-Controller integrierte Firewall lediglich die Pakete zwischen zwei verschiedenen Netzwerken kontrollieren. Innerhalb eines CAN-Netzwerks ist diese Maßnahme, gegen die erwähnten Angriffe aufgrund der Broadcast-Eigenschaft des CAN-Protokolls jedoch wirkungslos. Zudem greift keiner der Ansätze die Möglichkeit auf, die Filterung auf Basis des

aktuellen Fahrzeugzustands durchzuführen, um eintreffende Botschaften präziser auf deren Sicherheitsrisiko bewerteten zu können. Beispielweise wäre das Auslösen der Zündladungen während der Fahrt bzw. bei einer Sitzbelegung im publizierten Airbag-Angriff (s. Abschnitt 3.1.3) nicht möglich gewesen, wenn das Gateway in diesem Anwendungsfall entsprechende Botschaften gefiltert hätte.

Zugriffskontrolle: Die Ansätze der Zugriffskontrolle dienen zur Gewährleistung der Integrität, Vertraulichkeit und Verfügbarkeit von Informationen und bieten einen Schutz gegen *Tampering*, *DoS* oder *Elevation of Privileges* Angriffe (s. auch Abschnitt 2.1). Die vorgestellten Ansätze aus dem Bereich der Wissenschaft adressieren dabei unterschiedliche Modelle der Zugriffskontrolle, Protokolle sowie automotive Plattformen. Die Kontrolle und Durchsetzung von Berechtigungen für Fahrzeugfunktionen hätten die Möglichkeiten der bisher publizierten Angriffe reduziert. So hätten die Forscher im erläuterten Jeep Hack 2013 (s. Abschnitt 3.1.3) nach der Überwindung des Security-Access keinen Vollzugriff auf alle Funktionen erhalten, sondern nur auf eine zuvor definierte Menge an Berechtigungen. Des Weiteren beziehen die Ansätze keine Kontextinformationen auf Basis des aktuellen Fahrzeugzustands in die Zugriffsentscheidungen mit ein. Die einzige Ausnahme bildet dabei der Ansatz von Hugot et al. [149], der auf diese wichtige Information hinweist, aber keine weiteren Angaben für eine zuverlässige Erfassung dieses Zustandes macht. Der publizierte Ansatz fokussiert dabei eine Ethernet-basierte SOME/IP Kommunikation innerhalb von kommenden SOA-Architekturen (s. auch Abschnitt 2.1.1). Ein Ansatz für signal-orientierte Architekturen ist dagegen nicht enthalten.

Intrusion Detection Systeme: Im Bereich der Detektion von Angriffen bzw. Anomalien in Fahrzeugsystemen wurde durch die Arbeit von Al-Jarrah et al. [103] eine Taxonomie zur Klassifizierung von bisher publizierten Ansätzen eingeführt. Die Forscher schlussfolgern aus ihrer Analyse, dass bisherige Ansätze mit der Ausnahme von [154] den Kontext und die Semantik der Daten nicht in die Anomalieerkennung mit einbeziehen. Ein bisher nicht adressiertes Forschungsgebiet ist die Anomalieerkennung im Bereich der Diagnose in Fahrzeugen, die aufgrund bisher gezeigter Angriffe (s. Abschnitt 3.1) und zunehmend remote-ausführbarer Diagnoseanwendungen Risiken für neue Angriffsszenarien bietet [121].

Guidelines/Regularien/Standards: Im Bereich der Standardisierung sowie gesetzlichen Regelungen lässt sich über die letzten Jahre erkennen, dass sowohl der Gesetzgeber also auch die Industrie das Gebiet der Informationssicherheit in Fahrzeugen verstärkt in den Fokus nehmen. Im Vergleich zu Spezifikationen der Industrie (AUTOSAR Standard), die konkrete Implementierungsvorgaben zu Security-Maßnahmen aufnehmen, adressieren Normungsorganisationen wie die ISO oder SAE vorwiegend die Security-Aktivitäten sowie Prozesse und deren Integration innerhalb der Entwicklungsphase. Das in AUTOSAR enthaltene Modul SecOC sowie weitere Security-Protokolle (s. auch Abschnitt 3.2.4) adressieren die bisher fehlende Kanalsicherheit in Bezug auf die Eigenschaften Authentizität und Integrität bei der Übertragung von Nachrichten (z.B. in CAN-Netzwerken), die u.a. *Spoofing* und *Tampering* Angriffe ermöglichten (s. auch Abschnitt 3.1.3). Des Weiteren spezifiziert der AUTOSAR Adaptive Standard ein Modul für die Zugriffskontrolle innerhalb von Steuergeräten, die Berechtigungen von Services kontrolliert. Die Berechtigungen werden dabei über *Capabilities* (s. auch Abschnitt 2.2.2) definiert, die keine aktuellen Fahrzeugzustände berücksichtigen. Eine Herausforderung ist dabei der Widerruf (engl. revocation) von bereits zugewiesenen Berechtigungen, die den Services innerhalb der implementierten Manifestdatei individuell zugewiesen wurden.

Daneben wird es im Bereich der gesetzlichen Regularien durch die Verordnungen der UNECE einen Paradigmenwechsel in Bezug auf die Cybersicherheit für OEMs geben. Zuvor waren die Hersteller beispielsweise in Deutschland lediglich indirekt über das Produkthaftungsgesetz (ProdHaftG) verpflichtet, die Informationssicherheit durch geeignete Prozesse und Maßnahmen in Fahrzeugen zu berücksichtigen. Durch die UNECE-Verordnungen (R155/156) werden Hersteller dazu verpflichtet, dass Fahrzeugsysteme Angriffe erkennen und abwehren können (s. Abschnitt 7.3.7 in [163]).

Definition 3.2.4 Produkthaftungsgesetz

Das Produkthaftungsgesetz regelt in Deutschland nach §1 die Haftung eines Herstellers, wenn durch ein fehlerhaftes Produkt ein Mensch verletzt oder getötet wird. Die Haftung wird dabei nur ausgeschlossen, wenn der Fehler nach dem zur Entwicklungszeit verfügbaren Stand der Wissenschaft und Technik nicht erkannt werden konnte [167].

3.3 Diskussion und Einordnung

Bisherige Fahrzeugsysteme konnten die Security-Eigenschaften Authentizität und Autorisierung nicht- oder nur teilweise gewährleisten. Aktuell gibt es in der Wissenschaft nur vereinzelte Ansätze, die speziell die Thematik der Zugriffskontrolle aufgreifen und damit die Autorisierung in Fahrzeugsystemen adressieren. Parallel existieren auch in anderen Domänen, wie beispielsweise der intelligenten Stromnetze oder in großen Industrieanlagen, die ebenfalls der Gruppe der CPS angehören, ähnliche Herausforderungen, die durch Ansätze im Bereich der Zugriffskontrolle und Firewalls adressiert wurden. Jedoch enthalten diese Konzepte keine Berücksichtigung von aktuellen Zuständen der beteiligten Systeme. Diese würden jedoch einen Mehrwert bzgl. einer Minderung von Fähigkeiten eines Angreifers bieten, da Berechtigungen feingranularer definierbar sind. Dadurch wäre es beispielsweise einem Angreifer mit erlangtem Zugriff auf ein Fahrzeugnetzwerk nur möglich, bestimmte Funktionen ausführen, solange sich das Fahrzeug im Stillstand befindet. Dass diese Informationen für die Kontrolle und Durchsetzung von Berechtigungen entscheidend sind, stellten die Autoren Hugot et al. [149] in ihrer Arbeit an einem automotive System dar. Es fehlt dagegen ein Ansatz, wie diese Systemzustände sicher erfasst werden können bzw. welche Sensoren sich dafür eigenen, um Manipulationen zu verhindern.

Die Notwendigkeit der Zugriffskontrolle wurde auch innerhalb der Industrie erkannt und im Rahmen des AUTOSAR Adaptive Standards spezifiziert. Jedoch ist die Zugriffskontrolle ausschließlich für eine service-orientierte Kommunikation ausgelegt. Des Weiteren werden ebenfalls keine aktuellen Systemzustände in die Entscheidungsfindung bei der Prüfung von Berechtigungen miteinbezogen. Zudem besteht eine Herausforderung bzgl. des Widerrufs von bereits implementierten Berechtigungen von Services.

Im Bereich der Angriffserkennung (IDS) innerhalb der Fahrzeugkommunikation lassen sich folgende Erkenntnisse aufgreifen. Die publizierten Ansätze fokussieren die Anomalieerkennung auf Basis der On-Board Kommunikation. Hingegen wurde die Diagnosekommunikation bisher nicht adressiert. Deren wichtige Rolle in Bezug auf die Informationssicherheit belegen die in Abschnitt 3.1 ausgewerteten Angriffe sowie zukünftige Remote-Diagnoseanwendungen. Darüber hinaus fehlen geeignete Testdaten (Auf-

zeichnungen mit Diagnosekommunikation), um IDS-Ansätze auf diesem Forschungsgebiet evaluieren zu können.

Außerdem beziehen die untersuchten Ansätze, bis auf zwei Ausnahmen (s. auch Abschnitt 3.2.3), keine Kontext- bzw. Semantik-basierende Informationen in die Erkennung mit ein. Des Weiteren fiel auf, dass bisher keiner der Ansätze eine Kombination aus beiden Schutzmaßnahmen, d.h. die dynamische Anpassung von Berechtigungen in einer Zugriffskontrolle aufgrund von erkannten Anomalien umsetzte.

4 Angriffsprävention durch Zugriffskontrolle bei Diagnosefunktionen

Die bisher fehlende Vergabe und Kontrolle von Zugriffsrechten auf automotive Steuergeräten in signal-orientierten Netzwerken ermöglichte es Angreifern über Diagnosenachrichten, teilweise safety-kritische Funktionen anzusteuern (s. Abschnitt 3.1). Das in dieser Arbeit entwickelte Konzept der Automotive Attribute-based Access Control (A-ABAC) für signal-orientierte Netzwerke soll das Risiko für zukünftige Angriffe minimieren, indem zugewiesene Berechtigungen für Anwendungen kontrolliert und durchgesetzt werden. Hierfür wird ein Zugriffsmodell der klassischen IT adaptiert sowie kontextbezogene Fahrzeuginformationen (s. Definition 3.2.1) für eingehende Zugriffsentscheidungen verwendet.

4.1 Entwicklung eines Zugriffskontrollansatzes

Das entwickelte A-ABAC Framework (s. Anhang A.11) adressiert verschiedene Schwächen der Autorisierung, die bei bisher bekannten Angriffen (s. Abschnitt 3.1) ausgenutzt wurden. Die Zugriffsentscheidungen werden dabei u.a. auf Fahrzeugzuständen getroffen, die über verschiedene Sensorquellen bestimmbar sind. Da diese Quellen selbst einen Angriffspunkt für Manipulationen darstellen wird eine Methodik zur Bewertung des Risikos bereitgestellt. Die Zugriffskontrolle kann sowohl auf ECU- als auch auf Gateway-Ebene erfolgen, um entweder Funktions- (Diagnose) oder Nachrichten-bezogen (CAN-Botschaften) hinterlegte Berechtigungen einer Zugriffsrichtlinie (s. Abschnitt 2.2.1) zu kontrollieren bzw. durchzusetzen.

4.1.1 Funktionale- und nicht-funktionale Anforderungen

Für die Entwicklung der automotive Zugriffskontrolle werden nachfolgend Anforderungen (engl. requirements) spezifiziert, die auf Basis der identifizierten derzeit offenen Forschungslücken (s. auch Abschnitt 3.2.5) abgeleitet sind.

- **RQ1:** Die Zugriffskontrolle soll spezifizierte Berechtigungen für Diagnoseanwendungen kontrollieren und durchsetzen.
- **RQ2:** Die Zugriffsentscheidung soll in Abhängigkeit des aktuellen Fahrzeugzustands erfolgen.
- **RQ3:** Die Berechtigungen für Funktionen sollen in einem geeigneten Format für automotive ECUs beschreibbar sein.
- **RQ4:** Die Zugriffskontrolle soll verteilt in die Fahrzeugarchitektur integriert werden, um auf unterschiedlichen Architektur-Komponenten (z.B. ECUs und Gateways) die Zugriffsentscheidungen kontrollieren und durchsetzen zu können.
- **RQ5:** Die Zugriffskontrolle soll in E/E-Architekturen mit signal-orientierter Kommunikation integrierbar sein.
- **RQ6:** Eine zentrale Möglichkeit zur Anpassung und Widerruf von bereits zugewiesenen Berechtigungen für Funktionen soll gegeben sein.

4.1.2 Architektur

Das Konzept der A-ABAC Zugriffskontrolle basiert auf einer Adaption des in der klassischen IT etablierten ABAC-Zugriffsmodells (s. auch Abschnitt 2.2.2). Dieses Modell bietet gegenüber dem Zugriffsmodell RBAC den Vorteil, dass Subjekte für den Zugriff auf Ressourcen nicht einzeln erstellt und verwaltet werden müssen. Stattdessen erfolgt die Zugriffsentscheidung auf Basis einer zugewiesenen Zugriffsrichtlinie (s. auch Abschnitt 4.1.8) in die bisheriges Wissen über bekannte Angriffe und Schwachstellen einfließt (s. Abbildung 4.1). Darüber hinaus wird in die Zugriffsentscheidung der aktuelle Fahrzeugzustand (s. Definition 3.2.2) miteinbezogen (s. auch Abschnitte 4.1.3 u. 4.1.6), um kontextabhängige Sicherheitsentscheidungen umzusetzen. Die einzelnen Module

der Zugriffskontrolle sind nachfolgend beschrieben. Die Durchsetzung von Berechtigungen (PEP) kann dabei auf jedem Netzwerkknoten erfolgen. Das Erlauben oder Blockieren einer Zugriffsanfrage erfolgt hingegen in leistungsstarken Domänen-Gateways, die zusätzliche Zustandsinformationen (PIP) des Fahrzeugs bereitstellen.

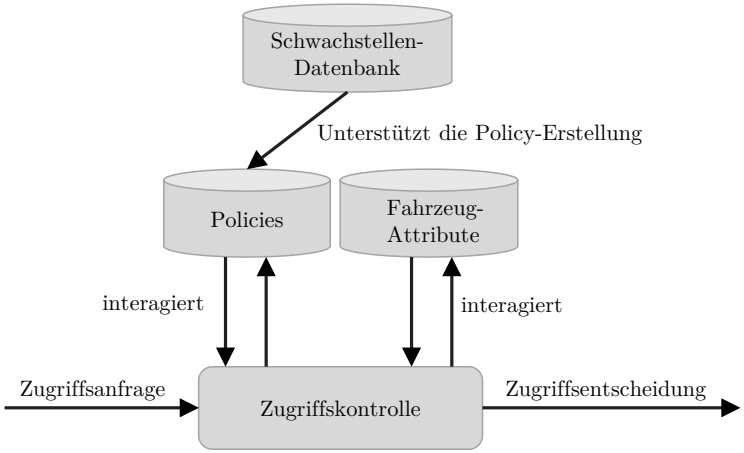


Abbildung 4.1: Prinzipieller Aufbau der A-ABAC Zugriffskontrolle unter Einbezug von Zugriffsrichtlinien und Fahrzeugattributen [BRS+20].

Die A-ABAC enthält für die Kontrolle und Durchsetzung von Zugriffsberechtigungen verschiedene Module (PEP, PDP, PIP, PAP, s. auch Abschnitt 2.2.2). Eine mögliche Integration ist anhand einer exemplarische Domänen-E/E-Architektur illustriert (s. Abbildung 4.2). Dieser Architekturtyp wird derzeit in aktuellen Fahrzeugbaureihen eingesetzt und wird voraussichtlich auch in Zukunft in hybriden Architekturen (service- und signal-orientiert, s. auch Abschnitt 2.1.2) für AUTOSAR Classic Domänen verwendet. Die Architektur enthält ein Backbone Ethernet-Netzwerk bestehend aus einem zentralen Gateway sowie mehreren Domänen-Gateways, die untergeordnete Netzwerke (CAN, LIN, FlexRay) mit dazugehörigen ECUs miteinander verbinden. Das zentrale Gateway übernimmt zusätzlich die Routingfunktion zu externen Netzwerkteilnehmern wie z.B. OBD-Hardware. Die Erweiterung des Telematik-Gateways ermöglicht eine drahtlose Kommunikation über eine Luftschnittstelle (Mobil-

funk, WLAN, Bluetooth) zu weiteren externen Netzwerkknoten wie beispielsweise einem OEM-Backend.

Module der Zugriffskontrolle

Durch den verteilten Aufbau von Funktionen innerhalb der E/E-Architektur werden die A-ABAC-Module auf verschiedene Netzwerkkomponenten integriert. Die jeweiligen Funktionen der unterschiedlichen Module werden nachfolgend erläutert:

- **Policy Enforcement Point (PEP):** Das Modul ist für eingehende Zugriffsanfragen und die Durchsetzung der Zugriffsentscheidungen (erlauben, verweigern) des PDP zuständig. Es ist auf jeder ECU bzw. jedem Gateway integriert, um diese Funktionalität umsetzen zu können.
- **Policy Decision Point (PDP):** Die Aufgabe des PDPs besteht darin, übermittelte Zugriffsanfragen des PEP auf Basis der geltenden Zugriffsrichtlinien (s. auch Abschnitt 4.1.8) zu prüfen und anschließend eine Zugriffsentscheidung an den PEP zu übertragen. Als Beispiel kann hier eine Diagnoseanfrage (Aktion) von einem Mechaniker (Subjekt) genannt werden, der auf das Motorsteuergerät (Ressource) des Fahrzeugs zugreifen möchte. Der PDP hat darüber hinaus die Aufgabe die aktuell gültigen Umgebungsbedingungen (z.B. Sitzbelegung, Fahrzeugzustand, Standort) in die Zugriffsentscheidung mit einzubeziehen. Diese werden auf Anfrage vom PIP bereitgestellt. Die getroffene Entscheidung wird abschließend an den PEP für die Durchsetzung übermittelt. Im Vergleich zu Capability-basierten Zugriffstechniken (s. Abschnitte 2.2.2 bzw. AUTOSAR Adaptive Standard in 3.2.5) bieten die PDPs die Möglichkeit, zugewiesene Zugriffsrichtlinien bei Änderungen zentralisiert zu aktualisieren, falls Berechtigungen angepasst bzw. widerrufen werden müssen.
- **Policy Information Point (PIP):** Dieses Modul erfasst und verarbeitet benötigte Fahrzeuginformationen basierend auf vertrauenswürdigen Sensorinformationen (s. auch Abschnitt 4.1.7), die als Umgebungsbedingungen für die Entscheidungen notwendig sind. Die PIPs sind dabei auf den Gateways integriert, um direkt mit den PDPs zu interagieren. Für eine eingehende Diagnoseanfrage, ist beispielsweise die Information über

den aktuellen Fahrzeugzustand (z.B. im Stillstand oder in Bewegung) entscheidend für die Entscheidungsfindung des PDP.

- **Policy Administration Point (PAP):** Der PAP ist der zentrale Punkt für die Administration und Verteilung von Zugriffsrichtlinien durch den Fahrzeughersteller. Im Backend kann der Hersteller unterschiedliche Richtlinien bzw. Versionen verwalten, Änderungen vornehmen und diese auf Fahrzeuge über die Luftschnittstelle ausrollen. Darüber hinaus können vorhandene Richtlinien bei Bekanntwerden von neuen Security-Schwachstellen entsprechend angepasst und getestet werden bevor ein Update für die betroffenen Fahrzeuge ausgerollt wird.

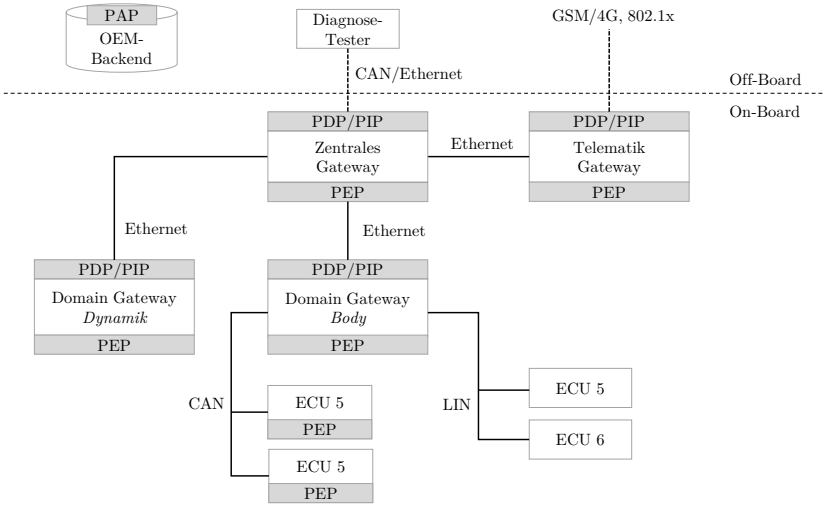


Abbildung 4.2: Integration der verschiedenen A-ABAC Module in eine exemplarische Domänenbasierte E/E-Architektur (basierend auf [RDG+19]).

4.1.3 Metamodell

Das zentrale Element der A-ABAC bildet der Fahrzeugzustand, welcher über Fahrzeugattribute beschrieben wird (s. Abbildung 4.3). Die Attribute beste-

hen aus Sensorinformationen und Funktionszustandsinformationen. Für die Sammlung bzw. Bereitstellung der Informationen ist das PIP-Modul verantwortlich. Durch Auswertung der Fahrzeugattribute (s. Abschnitt 4.1.6) können unterschiedliche Fahrzeugzustände bestimmt werden. So ist beispielsweise der Zustand *in Bewegung* durch Auswertung der Raddrehzahlsensoren bestimmbar. Ein Funktionszustand beschreibt hingegen Zustände von Steuergeräten oder einzelnen Funktionen (z.B. ECU in Bootrom-Mode oder ACC aktiv).

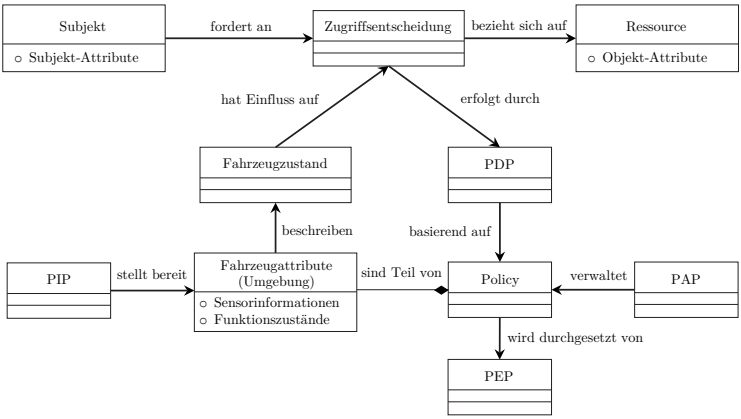


Abbildung 4.3: Metamodell der A-ABAC mit Fokus auf die Zugriffsentscheidung. (Notation nach UML 2.)

4.1.4 Kommunikationsprotokoll

Für den Austausch der benötigten Informationen zwischen den verschiedenen A-ABAC-Modulen wird ein Protokoll spezifiziert, das in die Nutzdaten eines Netzwerkprotokolls eingebettet werden kann. In dieser Arbeit wird das Protokoll in ein Standard CAN-Frame integriert (s. Abbildung 4.4). Das A-ABAC Frame besteht aus Header- und Nutzdateninformationen.

Der Header *Command (CMD)* umfasst Steuerbefehle wie beispielsweise die Anfrage eines Zugriffs auf eine Ressource oder die Übermittlung einer getroffenen Zugriffsentscheidung durch einen PDP. Im Nutzdatenteil des Fra-

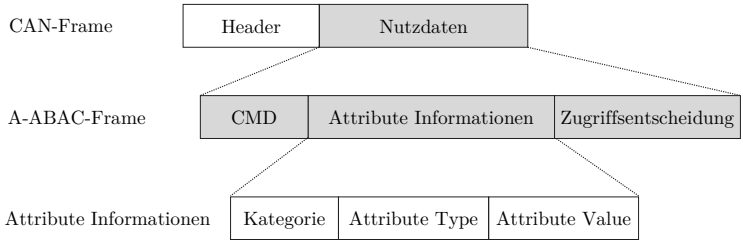


Abbildung 4.4: Einbettung einer A-ABAC Nachricht in eine CAN-Botschaft zur Übertragung von Informationen für die Zugriffskontrolle (basierend auf [RDG+19]).

mes werden *Attribute Informationen* (Kategorie, Attributsname, Attributswert) übertragen (s. auch Abschnitt 4.1.6). Zusätzlich ist ein Feld für die *Zugriffsentscheidung* enthalten, die ein PDP auf eine Zugriffsanfrage vom PEP übermittelt. Um Spoofing oder Tampering-Angriffe (s. Abschnitt 2.2.1) auf die A-ABAC-Frames zu verhindern ist die Absicherung der übertragenen on-Board CAN-Botschaften durch AUTOSAR SecOC (s. Abschnitt A.1.5) notwendig. Da auf Basis des Standard CAN-Protokolls nur acht Nutzdatenbytes zur Verfügung stehen ist die Absicherung durch SecOC allgemein problematisch [168]. Nach der NIST sollte die Media Access Control (MAC)-Länge mindestens 64 Bit betragen, um eine ausreichende Sicherheit zu gewährleisten [169]. Zukünftige CAN-Erweiterungen wie CAN-FD oder CAN-XL (s. Abschnitt A.1.1) sollten diese Längenprobleme jedoch lösen. Für die Absicherung externer Netzwerkteilnehmer wäre eine Zertifikats-basierte Authentifizierung mit einer Public Key Infrastructure (PKI) (s. Abschnitt A.1.3) möglich.

4.1.5 Autorisierung

Für den Zugriff eines Subjekts auf eine Ressource wird auf Basis des A-ABAC Zugriffsmodells eine Autorisierungssequenz durchlaufen, bei der die Module PEP, PDP und PIP die Zugriffsentscheidung kontrollieren und durchsetzen (s. Abbildung 4.5). Die drei dargestellten Zugriffsanfragen basieren dabei auf einer Diagnoseanwendung. In der ersten Sequenz erfolgt eine direkte Zugriffsanfrage auf eine Ressource (Diagnosefunktion) ohne vorherige erfolgreiche Autorisierung. Die Anfrage (*ReqDiag*) wird dabei von den Modulen PDP/PIP abgelehnt

und die Entscheidung (*Access denied*) vom PEP durchgesetzt. Bezogen auf die dargestellte E/E-Architektur (s. Abbildung 4.2) würde die Blockierung der Anfrage im zentralen Gateway erfolgen und keine Nachricht in das interne Fahrzeugnetzwerk geroutet werden.

Die zweite Sequenz repräsentiert den Autorisierungsprozess, der für einen Ressourcenzugriff erfolgreich durchlaufen werden muss. Entspricht die Anfrage der geltenden Zugriffsrichtlinie, die der PDP prüft, sendet dieser eine Genehmigung (*AccessGranted*) an den PEP. Daraufhin erfolgt die Freigabe der Ressource durch den PEP (*UnlockRessource*), die über den PDP an die angefragte Ressource übertragen wird. Die erfolgreiche Autorisierung wird durch die Ressource gegenüber dem PDP/PIP bzw. PEP bestätigt (*RessourceUnlocked*). Der beteiligte PEP übermittelt abschließend eine Genehmigung zum anfragenden Subjekt (*AccessGranted*). Danach erfolgt der Start einer neuen Diagnosesession durch das beteiligte Subjekt. Da die Autorisierung erfolgreich abgeschlossen ist, werden die Anfragen vom PEP bzw. PDP/PIP direkt an die Zielressource weitergeleitet.

Die beschriebenen Szenarien basieren auf dem Anwendungsfall, dass das Subjekt einen externen Netzwerkteilnehmer darstellt. Hingegen kann eine Autorisierung auch innerhalb des internen Fahrzeugnetzwerks zwischen zwei ECUs erfolgen. Dadurch übernimmt die anfragende ECU die Rolle des Subjekts.

4.1.6 Fahrzeugspezifische Attribute

Für die Integration einer ABAC-basierten Zugriffskontrolle in Fahrzeugarchitekturen können verschiedene automotive spezifische Attribute genutzt werden (s. Tabelle 4.1). Als Basis für die Adaption dienen die Attribute aus der traditionellen IT des ABAC-Zugriffsmodells (s. auch Abschnitt 2.2.2).

Werden diese auf die Fahrzeugdomäne abgebildet, können auch hier vier verschiedene Kategorien (*Subjekt, Aktion, Umgebung und Ressource*) definiert werden. Dabei repräsentieren *Subjekte* beispielsweise außerhalb des Fahrzeugs Personen, Einrichtungen wie Werkstätten oder externe Netzwerkteilnehmer. Hingegen repräsentieren Subjekte im Fahrzeug z.B. Steuergeräte oder darauf basierende Funktionen.

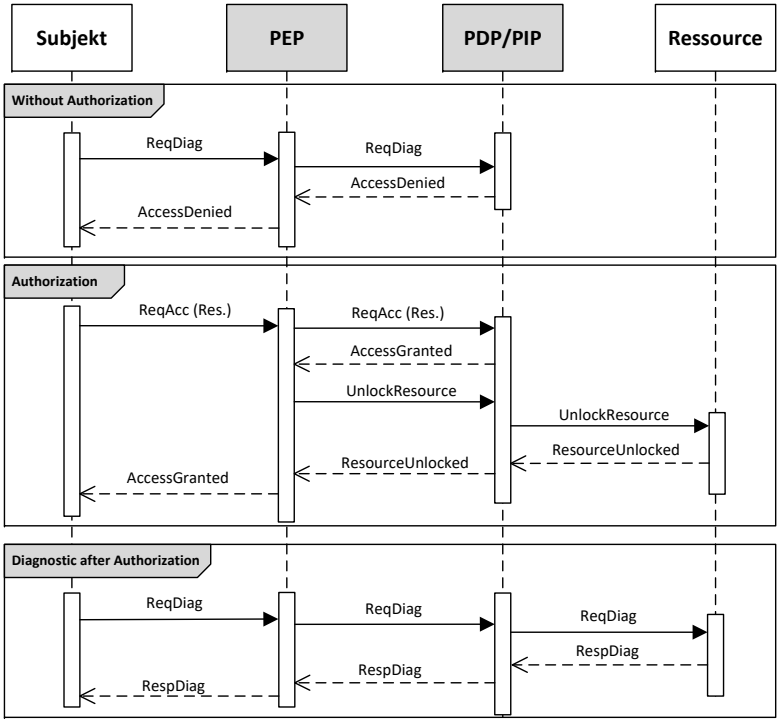


Abbildung 4.5: Ablauf einer A-ABAC Autorisierungssequenz am Beispiel einer Diagnoseanwendung (basierend auf [RDG+19]).

Die Kategorie *Aktion* umfasst verschiedene Zugriffs- bzw. Ausführungsarten wie beispielsweise das Lesen oder Schreiben von Daten auf Steuergeräten. Des Weiteren können Aktionen aber auch bestimmte Ansteuerbefehle umfassen, wie z.B. das Öffnen des Kofferraums oder das Entsperren der Lenkradverriegelung.

Der dritte Attributstyp bildet die *Umgebung* die Signale bzw. deren Sensorquellen beinhalten. Die Umgebung ist dadurch eine Information, die einen Zustand des Fahrzeugs beschreibt. Diese können auf Fahrzeugebene beispielsweise die Zustände *in Bewegung*, *im Stillstand* oder die *aktuelle GPS-Position*

Tabelle 4.1: Auszug verschiedener Attributsinformationen die in Fahrzeugen verfügbar sind und relevant für die Zugriffsentscheidungen der A-ABAC sind.

Kategorie	**Attributsname**	**Attributswert**
Subjekt	Rolle	Werkstatt
Subjekt	Händler_ID	123456
Subjekt	ECU_Typ	Motorsteuerung
Aktion	Aktion_ID	lesen
Aktion	Aktion_ID	öffnen
Umgebung	Motorstatus	Betrieb
Umgebung	Gurtschloss_VL	1
Umgebung	Sitzbelegung_VL	1
Ressource	Ressource_ID	Diagnosedaten
Ressource	Ressource_ID	Türschloss

darstellen. Auf Sensor- bzw. Signalebene werden Zustandsinformationen beispielsweise vom Gurtschloss oder der aktuellen Sitzbelegung übermittelt. Ein weiterer Zustand kann dagegen auch beinhalten, ob sich ein Steuergerät aktuell in einem Diagnosemodus befindet und dadurch nicht aktiv an der On-Board Kommunikation teilnimmt. Diese Information wäre hilfreich gewesen, um beispielsweise den Angriff (Jeep Hack 2016, s. auch Abschnitt 3.1.3) abzumildern. Für die Erfassung der Zustandsinformationen auf Basis von Sensoren muss jedoch berücksichtigt werden, dass Angreifer diese manipulieren können. Aus diesem Grund ist eine Bestimmung des Risikos sinnvoll, um Sensorquellen zu klassifizieren (s. Abschnitt 4.1.7).

Die vierte Kategorie *Ressource* definiert Objekte, auf die Subjekte durch Aktionen zugreifen. Ein Anwendungsfall könnte dabei beispielsweise ein Zugriff durch eine externe Diagnosehardware (Subjekt) darstellen, die auf Diagnosedaten (Ressource) eines Steuergerätes lesend zugreift (Aktion), während sich das Fahrzeug im Stillstand (Umgebung) befindet. Darüber hinaus können Ressourcen auch Funktionen aus der Fahrzeugsteuerung darstellen (z.B. Lenkung ansteuern, Ansteuerung der Bremsen), die von Subjekten ausgeführt werden.

4.1.7 Auswahl physikalischer Attribute

Die integrierte Sensorik in Fahrzeugen bietet die Möglichkeit verschiedene Fahrzeugzustände zu bestimmen, um diese für die Attributs-basierte Zugriffskontrolle zu verwenden. Für die Durchführung einer Zugriffsentscheidung ist es wichtig, den genauen Fahrzeugzustand (s. Definition 3.2.2) als Kontext (s. Definition 3.2.1) mit zu berücksichtigen, da Security-Angriffe in verschiedenen Fahrzeugzuständen bzw. Fahrmanöver unterschiedliche Auswirkungen auf die funktionale Sicherheit haben können. Eine Manipulation des Lenkwinkels durch einen Angreifer kann während der Fahrt beispielsweise zu safety-kritischen Situationen führen, die hingegen im Stillstand keine weiteren Auswirkungen zur Folge haben. Ob das Fahrzeug beispielsweise gerade in Bewegung ist, kann durch die Auswertung der Raddrehzahlsensoren oder Global Positioning System (GPS)-Informationen erfolgen. Des Weiteren liefern Sitzbelegungs- oder Gurtschlosssensoren eine präzise Aussage über die aktuelle Zahl der Insassen.

Die Vertrauenswürdigkeit bzw. Manipulationssicherheit von Sensoren in einem Fahrzeug kann variieren, das bei der Auswahl berücksichtigt werden muss. Daher wird zur Klassifizierung der Sensorquellen bzgl. deren Manipulierbarkeit durch Angreifer eine adaptierte *Common Vulnerability Scoring System (CVSS)* [170] Metrik verwendet. Dieser offene Industriestand wird neben der Industrie auch in der Wissenschaft zur Bewertung der Schweregrade von Schwachstellen im Bereich der Informationssicherheit eingesetzt [34], [171]. Außerdem wird diese Metrik im automotive Cybersecurity-Standard (ISO 21434, s. Abschnitt 3.2.4) empfohlen. In der nachfolgenden Sensorbewertung wird der ermittelte Schweregrad (engl. severity) als Risiko bzgl. der Manipulierbarkeit durch einen Angreifer definiert.

Die CVSS-Metrik besteht insgesamt aus drei Teilen (Basis-, Zeitkontext- und Umfeld-Metrik). In dieser Arbeit wird die Basis-Metrik verwendet, da die darin enthaltenen Charakteristika sich in Bezug auf die Zeit nicht verändern und damit auf die Sensoreigenschaften in Fahrzeugen abbildbar sind. Grundlegend ist diese in die Ausnutzbarkeit-Metrik (engl. exploitability metrics) sowie Einfluss-Metrik (engl. impact metrics) unterteilt. Die erste Untergruppe enthält die nachfolgenden fünf Kriterien.

- Der *Angriffsvektor* (engl. attack vector (AV)) beinhaltet Einträge, wie ein Angreifer das Angriffsziel erreichen kann. Ist z.B. das Zielsystem aus der Ferne oder nur mit physikalischem Zugriff erreichbar.
- Die *Angriffskomplexität* (engl. attack complexity (AC)) umfasst die Auswahl von bestimmten Vorbedingungen. Eine mögliche Vorbedingung wäre beispielsweise, dass sich der Angreifer innerhalb einer Verbindung zwischen zwei Kommunikationsteilnehmern befinden muss.
- Die *erforderlichen Berechtigungen* für das Zielsystem (engl. privileges required (PR)) definieren, welche Berechtigungen der Angreifer besitzen muss, um einen erfolgreichen Angriff durchzuführen.
- Die Benutzerinteraktion (engl. user interaction (UI)) beinhaltet, ob für einen erfolgreichen Angriff eine direkte Benutzereingabe am Zielsystem notwendig ist. Wenn dies der Fall ist, hätte ein Angreifer nicht die Möglichkeit diese Aktion per Remote-Zugriff auszuführen.
- Als letzter Eintrag in dieser Untergruppe wird der Bereich (engl. Scope (S)) einer Schwachstelle definiert. Damit wird angegeben, ob bei einer erfolgreichen Ausnutzung einer Schwachstelle noch weitere Ressourcen betroffen sind, die nicht direkt angegriffen werden.

Die *Einfluss-Metrik* bildet daneben den Einfluss einer ausgenutzten Schwachstelle auf die Eigenschaften (Vertraulichkeit (C), Integrität (I) und Verfügbarkeit (A)) der Informationssicherheit ab. Die adaptierte CVSS-Metrik zur Bewertung von Fahrzeugsensoren wird in Tabelle 4.2 auf Basis von vier verschiedenen Sensoren beispielhaft dargestellt. Jeder Sensor ist dabei unter Anwendung der Basis-Metrik bewertet. Da die Metrik ihren Ursprung in der klassischen IT hat, wird das Kriterium *erforderliche Berechtigungen* (PR) bzgl. dessen Definition auf das Fahrzeug angepasst. In Bezug auf die Sensorbewertung wird damit festlegt, welche Art von Zugriff ein Angreifer auf den Sensor haben muss. Falls kein Zugriff notwendig ist, wird dieser mit *None* deklariert. Der Wert *Low* wird zugewiesen, wenn ein Zugriff auf bestimmte Schnittstellen im Fahrzeug notwendig ist (z.B. CAN-Bus oder OBD-Schnittstelle). Der höchste Wert *High* wird vergeben, wenn der Angreifer für eine Manipulation einen physikalischen Zugriff auf die Sensorleitungen benötigt. Die Schwierigkeit besteht dabei zusätzlich in der Interpretation der teilweise proprietären Protokolle. Ein Beispiel wie Angreifer Sensoren ohne direkten Zugriff mani-

pulieren konnten ist in [172] gegeben. Darin beschreiben die Forscher wie ein Kamerasensor (montiert an der Frontscheibe) mit verschiedenen Grafikmustern vom Straßenrand aus manipuliert wird. Dagegen wäre eine Manipulation eines Raddrehzahlsensors (s. auch Tabelle 4.2) wesentlich aufwändiger, da ein physikalischer Zugriff auf den Sensor sowie Protokollkenntnisse notwendig wären.

Tabelle 4.2: Bewertung verschiedener Fahrzeugsensoren bzgl. deren Manipulierbarkeit durch einen Angreifer auf Basis der adaptierten CVSS Basis-Metrik [RDBK18].

Sensor	AV	AC	PR	UI	S	C	I	A	Score	Severity
Raddreh.	Phys.	Low	High	None	Chang.	None	High	High	6.5	medium
Beschl.	Adja.	Low	High	None	Chang.	None	High	High	8.1	high
Sitzbel.	Phys.	Low	High	None	Chang.	None	High	High	6.8	medium
ACC-Radar	Netw.	Low	None	None	Chang.	None	High	High	10	critical

Anhand eines ACC-Radarsensors wird die Einteilung mittels der adaptierten CVSS-Metrik in die unterschiedlichen Kategorien nachfolgend exemplarisch erläutert. Die Zuweisung *Network* wird dem Angriffsvektor zugewiesen, da dieser Sensor von außen ohne physikalischen Zugriff manipuliert werden kann. Die Angriffskomplexität ist *low*, da Radarsensoren ohne weiteren Absicherungsmaßnahmen von außen durch Spoofing-Angriffe (s. Abschnitt 2.2.1) beeinflusst werden können [173]. Des Weiteren sind für diese Aktivitäten keine bestimmten Berechtigungen *(PR = None)* bzw. Benutzerinteraktionen *(UI = None)* am beteiligten Zielsystem (Fahrzeug) notwendig. Es muss weiter davon ausgegangen werden, dass eine erfolgreiche Manipulation einen Einfluss *(S = Changed)* auf verschiedene Systeme im Fahrzeug hat, da mehrere Funktionen dieses Sensorsignal verarbeiten.

Die Analyse der durch diese Aktivitäten beeinträchtigten Security-Eigenschaften ergibt das Ergebnis, dass sowohl die Integrität *(I = High)* als auch die Verfügbarkeit *(A = High)* dieses Sensorsignals verletzt werden. Die Eigenschaft der Vertraulichkeit wird als nicht beeinträchtigt gewertet *(C = None)*, da die On-Board Kommunikation basierend auf dem CAN-Protokoll bisher unverschlüsselt übertragen wird [174].

Die Endbewertung *Score* dient für einen Vergleich bzgl. der Vertrauenswürdigkeit bezogen auf die bewerteten Sensoren. Die CVSS-Metrik definiert neben

der Berechnung einer numerischen Punktzahl unterschiedliche Risikointervalle (s. Tabelle 4.3). Für die Auswahl geeigneter Sensoren wird die Grenze auf *8,9 (Hoch)* festgelegt. Alle Sensoren, die ein höheres Bewertungslevel aufweisen, sind für die Bestimmung des Fahrzeugzustands nur durch die Anwendung zusätzlicher Maßnahmen geeignet, da ein hohes Risiko bzgl. deren Manipulierbarkeit existiert. Die Bewertung des ACC-Radarsensors ergibt eine Punktzahl von *10,0*. Darauf basierend lässt sich ableiten, dass der Sensor durch weitere Maßnahmen geschützt werden muss. Eine mögliche Zusatzabsicherung wäre eine Kombination aus mehreren Sensoren (Sensorfusion), um die empfangenen Werte zu plausibilisieren [175].

4.1.8 Zugriffsrichtlinien (Policies)

Die Grundlage für die Definition von Zugriffsberechtigungen auf Basis des ABAC-Zugriffsmodells stellen die zu definierenden Zugriffsrichtlinien dar. In der klassischen IT werden diese über den XACML Standard beschrieben (s. auch Abschnitt 2.2.2), der ohne zusätzliche Tools schwer lesbar bzw. verständlich sind. Die OASIS hat diese Komplexität erkannt und die vereinfachte Beschreibungssprache Abbreviated Language for Authorization (ALFA) [176] spezifiziert. Diese domänenspezifische Sprache (DSL) definiert keine vollständig neue Beschreibungssprache, sondern reduziert die Detailtiefe des XACML Standards. Dadurch besteht weiter die Möglichkeit eine erstellte ALFA-Policy in eine XACML-Policy umzuwandeln. Die ALFA-Sprache kann dazu verwen-

Tabelle 4.3: Qualitative Ratingskala gemäß der CVSS-Basis-Metrik [170] zur Bestimmung der Severity bzgl. einer Sensormanipulation durch einen Angreifer basierend auf dem ermittelten Score-Wert.

Rating	**CVSS Score**
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

det werden eine fahrzeugspezifische Zugriffsrichtlinie zu beschreiben, die in Fahrzeugsteuergeräte integriert werden kann (s. Auflistung 4.1).

Auflistung 4.1: Exemplarische Zugriffsrichtlinie (Policy) zur Definition von Zugriffsberechtigungen auf Basis einer Diagnoseanwendung (basierend auf [RDG+19]).

```
1  policy AccessDiagnosisData {
2  target clause
3  attribute.role      == "engineer"
4  or  attribute.role  == "technician"
5  or  attribute.role  == "mechanic"
6  apply firstApplicable
7  ReadDiagnosis
8  }
9
10 rule ReadDiagnosis {
11 permit target clause
12 attribute.action          == "read"
13 and attribute.resourceID  == "diagnosticsData"
14 condition
15 attribute.status          == "connected"
16 and attribute.status      == "vehicle at standstill"
17 }
```

Die erstellte Policy repräsentiert exemplarisch eine durch ALFA beschriebene Zugriffsrichtlinie für die Definition von Berechtigungen verschiedener Subjekte auf Basis einer Diagnoseanwendung. Die Richtlinie bildet allgemein Konditionalsätze ab. In diesem Beispiel können *Ingenieure, Techniker oder Mechaniker nur Diagnosedaten auslesen, solange das Testmodul mit dem Fahrzezuug verbunden ist und sich das Fahrzeug im Stillstand befindet*. Die Attributsnamen sind dabei Basierend auf dieser exemplarischen Anforderung erfolgt die Definition der passenden Attribute für die Subjekte (engineers, technicans and mechanics), Aktionen (lesen), Umgebungsattribute (Verbindungsstatus des Diagnosetesters, Fahrzeug im Stillstand) sowie Ressourcen (Diagnosedaten) der Zugriffsrichtlinie.

In Zeile 1 wird die ableitete Aktion beschrieben. Die Zeilen 2 bis 5 beschreiben das *Target* (s. Abschnitt 2.2.2), für welche Subjekte die Policy gültig ist. In Zeile 6 wird der *Rule-combining Algorithm* (s. Abschnitt 2.2.2) definiert, der bei Auswertung der Richtlinie angewendet wird. In diesem Beispiel wird die erste anwendbare Regel übernommen. Des Weiteren wird eine Regel definiert,

welche den Lesezugriff auf Diagnosedaten spezifiziert (ab Zeile 10). Dazu wird in den Zeilen 11 - 13 die eigentliche Regel beschrieben (*permit*), die es den definierten Subjekten erlaubt, auf Diagnosedaten lesend zuzugreifen. Die Berechtigung ist zusätzlich an Bedingungen (*condition*) gebunden (Zeilen 14 - 16), die für das Zugriffsrecht erfüllt sein müssen. Falls eine oder mehrere Bedingungen dagegen nicht erfüllt sind, wird der Zugriff verweigert.

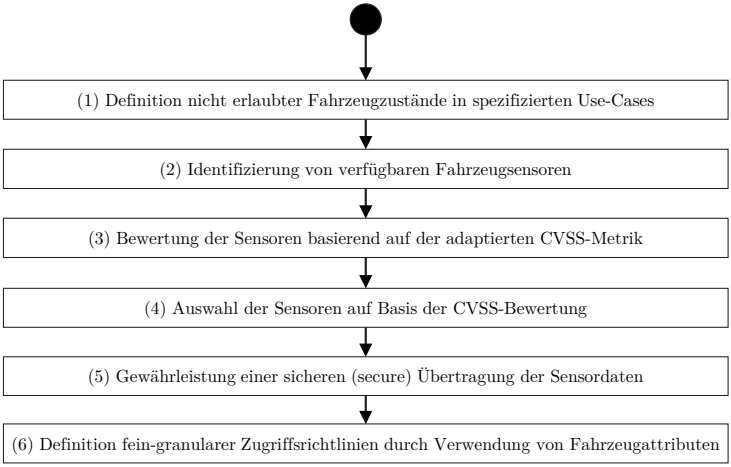


Abbildung 4.6: Ablauf zur Auswahl und Bewertung von geeigneten Sensoren zur Nutzung als Fahrzeugattribute in Zugriffsrichtlinien (basierend auf [RDBK18]).

Für die Erstellung der Zugriffsrichtlinien wird eine Methodik definiert, um die Berechtigungen zusätzlich an Fahrzeugattribute zu koppeln (s. Abbildung 4.6). Dabei werden zunächst *unsichere bzw. nicht erlaubte Fahrzeugzustände (1)* in unterschiedlichen Anwendungsfällen definiert. Darauf folgt die *Identifizierung von verfügbaren Fahrzeugsensoren (2))* innerhalb der betrachteten Fahrzeugbaureihe, die für die Bestimmung der Fahrzeugzustände geeignet sind. Die vorhandenen Sensoren werden daraufhin auf Basis der adaptierten CVSS-Metrik bewertet (s. auch Abschnitt 4.1.7). Anschließend erfolgt auf Basis der durchgeführten Risikobewertung eine *Auswahl der benötigten Sensoren (4)*, um die im ersten Schritt definierten Fahrzeugattribute erfassen zu können. Danach muss die *Sensorübertragung entsprechend abgesichert werden (5)*, um eine Manipulation der Sensordaten auf dem Übertragungsweg zu verhindern.

Als letzter Schritt folgt die *Definition von fein-granularen Zugriffsrichtlinien (6)* unter Einzug der nicht erlaubten Fahrzeugzustände.

4.1.9 Attributs-basierte Netzwerk-Firewall

Neben der Integration der Zugriffskontrolle auf einzelne Steuergeräte für die Kontrolle von Berechtigungen auf Basis von Funktionen (Anwendungsebene), können die fahrzeugspezifischen Attribute auch für die Verwendung auf der Netzwerkebene in Firewalls (s. auch Abschnitt 2.2.3) genutzt werden. Dieser Einsatz erlaubt eine Paketfilterung z.B. in den Gateways [RDBK18]. Dafür werden zu Beginn allgemeine Zugriffsrichtlinien definiert und diese danach in konkrete Filterregeln für Firewalls übersetzt (s. Tabelle 4.4).

Tabelle 4.4: Auszug einer Zugriffsrichtlinie für die Durchsetzung von Filterregeln auf Basis von fahrzeugspezifischen Attributen für die Integration in Fahrzeug-Gateways (basierend auf [RDBK18]).

Nr.	Zugriffsrichtlinie	Filterregel
1	Keine Ansteuerung von Fahrfunktionen solange min. eine ECU in Bootrom-Mode	Blockierung aller CAN-Botschaften (CAN IDs xy), die relevant für Fahrbetrieb & Status Bootrom-Mode ECUs >= 1
2	Keine erweiterten Diagnosesitzungen für spezifische Diagnosefunktionen, solange Fzg. in Bewegung	Blockierung aller Diagnoseanfragen mit SID xy, wenn v_Fzg. >= 6 km/h
3	Keine Diagnoseanfragen in Bezug auf EOL-Funktionen (Airbag, Gurtstraffer), solange Fzg. in Bewegung oder eine Sitzbelegung	Blockierung aller Diagnoseanfragen mit SID 10 LEV 04, wenn Status_Sitzbelegung ! = 0 or v_Fzg. >= 6 km/h oder Status_Gurtschloss ! = 0

Die dargestellte Zugriffsrichtlinie ist basierend auf dem *Jeep Hack* im Jahr 2016 sowie dem Airbag Hack 2017 abgeleitet (s. auch Abschnitt 3.1.3), um genutzte Angriffstechniken bereits im Gateway abzumildern bzw. zu verhindern.

Als allgemeine Richtlinie gilt in diesem Fall, dass keine Fahrfunktionen angesteuert werden dürfen, solange sich eine ECU im Bootrom-Mode befindet. Die dazugehörige Filterregel umfasst die Blockierung aller CAN-Nachrichten IDs, die relevant für den Fahrbetrieb sind (s. *Nr. 1*, Tabelle 4.4).

Der nächste Eintrag (*Nr. 2*) erlaubt keinen Wechsel in erweiterte Diagnosesitzungen (s. Abschnitt A.1.1), solange das Fahrzeug in Bewegung ist. Als Filterregel kann diese Anforderung auf spezifische Diagnosefunktionen in Abhängigkeit der aktuellen Fahrzeuggeschwindigkeit abgebildet werden.

Eine weitere Regel (*Nr. 3*) könnte beispielsweise die EOL-Funktionalität des Fahrzeugs adressieren. So wird allgemein die Richtlinie definiert, dass derartige Diagnoseanfragen blockiert werden, solange bestimmte Fahrzeugattribute (z.B. Sitzbelegung) von erlaubten Werten abweichen. Dafür werden als Filteregel betreffende Diagnosefunktionen mit Fahrzeugzustandsinformationen gekoppelt. Diese Maßnahme hätte beispielsweise das Risiko für die erfolgreiche Ausnutzung der Airbag-Schwachstelle aus dem Jahr 2017 (s. auch Abschnitt 3.1.3) minimiert.

4.2 Experimentelle Evaluierung auf Basis von Diagnoseanwendungen

In der experimentellen Untersuchung werden verschiedene Angriffsszenarien in einem prototypischen Netzwerk mit integrierter A-ABAC Zugriffskontrolle durchgeführt. Die Szenarien sind teilweise aus den in Abschnitt 3.1 analysierten Angriffen abgeleitet, um einen realen Bezug herzustellen. Da die Diagnosekommunikation im Vergleich zur On-Board Kommunikation keinen Echtzeitanforderungen unterliegt, werden diese Randbedingungen in dieser Arbeit nicht betrachtet.

4.2.1 Evaluierungsnetzwerk

Für die Untersuchung des A-ABAC Ansatzes wird eine vereinfachte Domänenbasierte E/E-Architektur (s. Abbildung 4.7) verwendet. Die darin enthaltenen Netzwerknoten sind über das CAN-Protokoll vernetzt (s. Abbildung 4.2). Die Architektur enthält ein zentrales Gateway sowie je ein Domänen-Gateway, mit dem je zwei Steuergeräte verbunden sind. Als externen Zugang für Diagnoseanwendungen wird am zentralen Gateway ein zusätzlicher CAN-Bus bereitgestellt, der bei realen Fahrzeugen über die OBD-Schnittstelle zugänglich ist. Zusätzlich sind auf den Gateways jeweils PIP/PDP sowie ein PEP

integriert (s. auch Abschnitt 4.1.2), um eingehende Zugriffsanfragen auf Basis der hinterlegten Zugriffsrichtlinie sowie aktuellen Umgebungsbedingungen eine Zugriffsentscheidung zu treffen. Bei den nachfolgenden Zugriffsszenarien wird der Datenpfad zwischen einem externen Subjekt (Simulation) sowie einer Ressource im internen Fahrzeugnetzwerk (ECU) betrachtet.

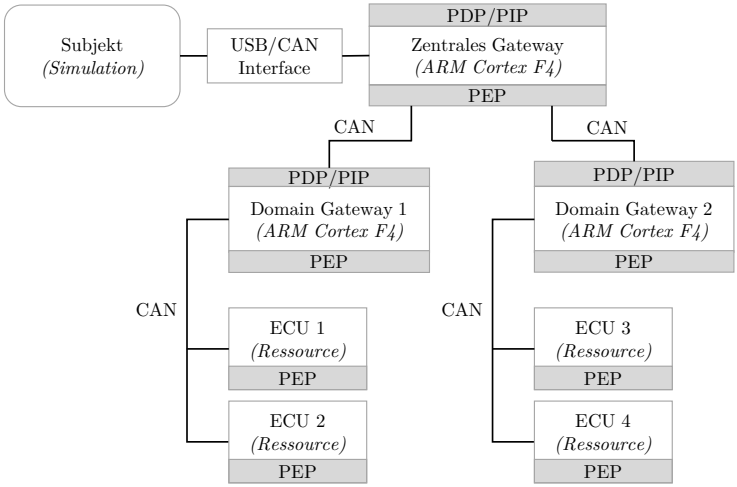


Abbildung 4.7: Übersicht der verwendeten E/E-Architektur zur Evaluierung des A-ABAC Ansatzes (basierend auf [RDG+19]).

Die Integration eines PAPs erfolgt in diesem Testaufbau nicht, da dieser eine Instanz innerhalb eines OEM-Backends darstellen würde und daher nicht Teil dieser Arbeit ist. Zur Übertragung der A-ABAC spezifischen Informationen wird das entwickelte Protokoll (s. auch Abschnitt 4.1.4) verwendet und in die Nutzdaten der CAN-Frames eingebettet. Als Plattform für jeden Netzwerkknoten wird ein ARM Cortex F4 basiertes Entwicklungsboard [177] mit FreeRTOS [178] verwendet. Das Subjekt, welches Diagnoseanfragen sendet, wird durch einen Simulationsrechner realisiert, der über einen USB/CAN Adapter mit dem Testnetzwerk verbunden ist. Da der Fokus bei dieser Evaluierung auf der Autorisierung von Zugriffsanfragen liegt, wird im Testaufbau keine Absicherung des Kommunikationskanals umgesetzt. Bei realen Fahrzeugen könnte dies beispielsweise über die AUTOSAR SecOC Spezifikation (s. Abschnitt A.1.5) für

die On-Board Kommunikation erfolgen oder bei Diagnoseanwendungen durch den Einsatz von Zertifikaten basierend auf einer PKI (s. Abschnitt A.1.3).

4.2.2 Untersuchung verschiedener Zugriffsszenarien

Für die Verifizierung des A-ABAC Konzepts werden drei verschiedene Anwendungsfälle mit zugehörigen Zugriffs-Policies (s. Abschnitt A.2.1) definiert. Darüber hinaus wird der gesamte Datenverkehr auf den CAN-Bussen mit zusätzlicher Messhardware parallel aufgezeichnet. Über den Messaufbau ist es ebenfalls möglich, das Verhalten des Netzwerkverkehrs aus der Rolle eines Angreifers zu beeinflussen.

Anwendungsfall 1: Der erste Fall repräsentiert jeweils eine Anfrage für einen Lese-/Schreibzugriff eines externen Diagnosetesters sowie OBD-Dongles auf die internen Komponenten (ECU 1 und ECU 2). Die hinterlegte Policy erlaubt nur für einen Diagnosetester beide Zugriffsarten. Für alle anderen Geräte wird der Zugriff blockiert. Die Anfrage wird daraufhin vom zentralen Gateway im PDP ausgewertet. Die Untersuchung zeigt, dass der Schreibzugriff des OBD-Dongles blockiert wird, da die Subjektattribute nicht der spezifizierten Policy entsprechen.

Anwendungsfall 2: In diesem Szenario sendet ein externer Diagnosetester in sehr kurzen Zeitintervallen verschiedene Diagnoseanfragen, um dadurch eine DoS Angriff (s. auch Abschnitt 2.2.1) zu simulieren. Der beteiligte PEP im zentralen Gateway, der die Anfragen empfängt, kann diese durch eine integrierte Durchsatzratenbegrenzung (engl. rate limiting) auf Basis der kurzen Zeitintervalle erkennen und wird dadurch nicht überlastet.

Anwendungsfall 3: Das dritte Szenario basiert auf der aufgedeckten Airbag-Schwachstelle im Jahr 2017 (s. auch Abschnitt 3.1.3). Die EOL-Funktionalität wird daher im Testaufbau auf *ECU 3* implementiert. Zusätzlich wird eine passende Zugriffsrichtlinie auf dem *Gateway 1* hinterlegt, die auf Basis des Angriffs abgeleitet wird. Das betrifft hauptsächlich die Einbeziehung von Attributen (z.B. Sitzbelegung, Fahrzeuggeschwindigkeit, Gurtschloss), die den aktuellen Fahrzeugzustand beschreiben. Danach werden über den Simulationsrechner mehrere Diagnoseanfragen zur Ausführung der EOL-Funktion getestet, während im Fahrzeugnetzwerk verschiedene Fahrzeugzustände simuliert werden. Es zeigt sich, dass der PDP im zentralen Gateway die Anfragen so-

fort blockiert, sobald eine Verletzung der Zugriffsrichtlinie eintritt (z.B. durch nicht erlaubte Umgebungsbedingungen).

4.2.3 Speichergrößen

Die benötigten Speichergrößen der unterschiedlichen A-ABAC Klassen sind in Tabelle 4.5 dargestellt. Insgesamt weisen die Module eine Größe von 3,7 kB auf. Dabei kann die Speichergröße der Klasse *Policy* durch die Anzahl der verwendeten Attribute und definierten Regeln variieren. Das vollständige A-ABAC Framework ist nur für die Implementierung eines PDP notwendig, der beispielsweise auf den Gateways integriert wird. Für eine bessere Einordnung der Speichergrößen, enthält eine aktuelles Domänen-Gateway einen Flash-Speicher von bis zu 6 MB [179]. Ein PEP benötigt dagegen nur ein reduziertes Framework, da Zugriffsentscheidungen seitens des PDP entweder erlaubt oder verweigert werden.

Tabelle 4.5: Übersicht der verschiedenen Speichergrößen von A-ABAC-Modulen.

Software-Komponente	**Größe [kB]**
A-ABAC Base	2.4
Policies	0.9
Attribute	0.4
Summe	**3.7**

Durch die ermittelten Speichergrößen der verschiedenen Komponenten kann belegt werden, dass die A-ABAC keine wesentliche Speicherbelegung auf den ECUs bzw. Gateways verursacht. Damit ist eine Integration auf reale automotive Hardwarekomponenten möglich.

4.3 Zusammenfassung

Die erfolgreichen Angriffe auf Fahrzeuge lassen sich zu einem Großteil auf die fehlende Gewährleistung der Security-Eigenschaften Authentizität sowie der Autorisierung zurückführen (s. auch Kapitel 3). Dadurch war es möglich,

dass Angreifer beispielsweise durch eine Kompromittierung einer ECU einen Vollzugriff erlangten und dadurch beliebige CAN-Botschaften zur Ansteuerung von Fahrfunktionen senden konnten. Der präsentierte A-ABAC Ansatz für die Integration einer Zugriffskontrolle in signal-orientierte Netzwerke bietet die Möglichkeit Berechtigungen von Subjekten für den Zugriff auf Objekte über Zugriffs-Policies abzubilden. Die Richtlinien sind auf Netzwerkknoten wie beispielsweise Gateways integrierbar damit die zugehörigen PDPs entsprechende Zugriffsentscheidungen von eingehenden Anfragen durchführen können. Ein wichtiges Merkmal stellt dabei die Berücksichtigung von Fahrzeugzustandsinformationen dar, die Kontext-basierte Zugriffsentscheidungen erlauben. Die Bestimmung aktueller Fahrzeugzustände erfolgt dabei auf Sensorinformationen, die PIPs bereitstellen. Da die Vertrauenswürdigkeit von integrierten Fahrzeugsensoren aufgrund möglicher Manipulationen durch einen Angreifer je nach Einbauort und Sensortyp variieren kann, wird eine Methodik zur Auswahl und Bewertung von geeigneten Sensoren bereitgestellt. Getroffene Zugriffsentscheidungen der PDPs werden auf den Steuergeräten über die integrierten PEPs durchgesetzt.

5 Angriffserkennung durch Anomalieerkennung bei Diagnosefunktionen

Neben der Integration von präventiven Schutzmaßnahmen in E/E-Architekturen wie anhand der Zugriffskontrolle in Kapitel 4 beschrieben, sind für die Steigerung der Informationssicherheit proaktive Maßnahmen wie IDS-Systeme notwendig. Das Ziel ist dabei mögliche Angriffe während ihres Auftretens zu erkennen indem Abweichungen (Anomalien) von einem bekannten Normalverhalten bestimmt werden. Erkannte Anomalien können wiederum verwendet werden, um in einem weiteren Schritt eine Anpassung bisheriger Schutzkonzepte vorzunehmen. Die ermittelten Erkenntnisse können dazu verwendet werden, bereits definierte und zugewiesene Berechtigungen einer Zugriffskontrolle anzupassen, um einen laufenden Angriff zu blockieren (z.B. zeitlich beschränkte Blockierung des Diagnosezugangs) oder präventiv Berechtigungen anzupassen (Menge an Zugriffsrechten eines Benutzers anpassen). Der nachfolgend entwickelte Erkennungsansatz fokussiert die Anomalieerkennung innerhalb der Diagnosekommunikation, die bisher auf Basis des analysierten Stands der Technik & Wissenschaft (s. Abschnitt 3.2) noch nicht näher adressiert wurde. Für den entwickelten Ansatz der Anomalieerkennung wird als Ausgangssituation angenommen, dass ein Angreifer (unautorisierte Person) in Besitz von Zugangsdaten eines legitimierten Besitzers kommt. Dadurch agiert dieser als Insider (auch *Outside Affiliates* genannt, s. Abschnitt 2.2.1), wodurch beispielsweise eine Zugriffskontrolle nicht mehr zwischen einem Angreifer bzw. legitimierten Benutzer unterscheiden kann. Der Angreifer ist damit in der Lage alle dem jeweiligen Benutzer zugewiesenen Berechtigungen für den Zugriff auf Diagnoseanwendungen im Fahrzeug zu verwenden.

Das angenommene Angriffsszenario (s. Abbildung 5.1) beinhaltet dabei nicht nur einen lokalen Zugriff, sondern auch Remote-Zugriffe, die zunehmend über Cloud-Infrastrukturen und drahtlose Übertragungstechnologien (z.B. Mobil-

funk) ausführbar sind. Die darin definierten Kommunikationskanäle werden in Bezug auf die Manipulation von Netzwerkpaketen als sicher angesehen (Gewährleistung der Sicherheitseigenschaften Authentizität/Integrität). Im Vergleich zu lokal basierten Angriffen ist bei Remote-Zugriffen ein höheres Schadensrisiko anzunehmen, da sich das Fahrzeug zeitweise in Bewegung (aktiver Fahrbetrieb) befinden kann. Ein böswilliges Auslösen von sicherheitskritischen Diagnosefunktionen (z.B. ECU-Reset) könnte in diesem Systemzustand einen erheblichen Einfluss auf das Fahrzeugverhalten haben. Um derartige Insider-Angreifer zu erkennen muss das (Kommunikations-) Verhalten analysiert werden, um Abweichungen zum Normalverhalten (legitimer Benutzer) zu detektieren. Die Analyse von eingehenden Diagnosebotschaften aus der Umwelt (als Systemgrenze wird hier das Fahrzeug angenommen) wird dabei im zentralen Fahrzeuggateway durchgeführt.

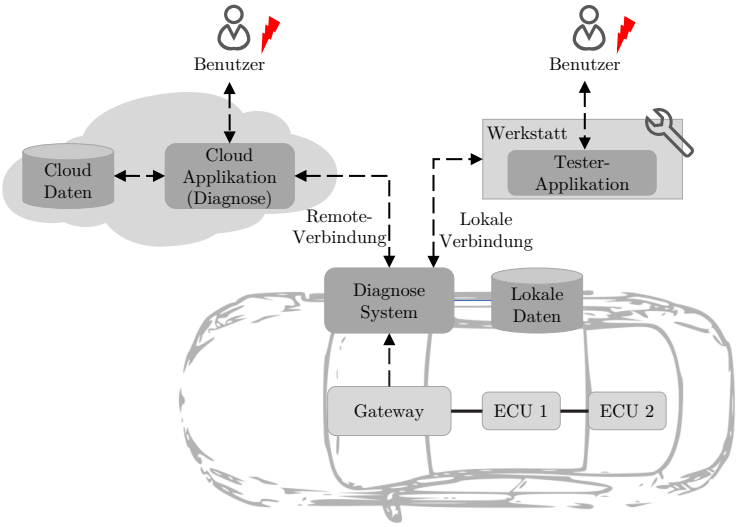


Abbildung 5.1: Schematische Darstellung von möglichen Diagnoseverbindungen zwischen Diagnosetester in Werkstätten (lokaler Zugriff) und Cloud-Applikationen (Remote-Zugriff). Angenommene Angriffspunkte (Kompromittierung von Benutzer-Accounts) sind mit einem roten Blitz dargestellt. (Darstellung angelehnt an [121].)

5.1 Entwicklung eines Erkennungsansatzes für Anomalien

Der in dieser Arbeit entwickelte AIDS-Ansatz (s. Abschnitt 2.2.4) für Diagnoseanwendungen in Fahrzeugen adressiert eine bisher noch offene Forschungsthematik im Bereich des untersuchten Stands der Technik und Wissenschaft (s. Abschnitt 3.2) in Bezug auf eine kontext-basierte Anomalieerkennung. Diese soll im Fahrzeug eingehende Diagnosesequenzen analysieren, um auf der Grundlage eines bekannten Normalverhaltens, davon abweichende Diagnosesequenzen als Anomalien zu klassifizieren. Dadurch soll die Erkennungstechnik in der Lage sein, mögliche Insider-Angreifer zu erkennen, die von einem externen Netzwerkknoten Diagnosebotschaften an das Fahrzeug senden. Dafür wird eine Methode aus dem Bereich der CL adaptiert (s. Abschnitt 2.3.2), um mittels verschiedener Sprachmodelle bestimmte Verhaltensmuster aus der Kommunikation abzubilden. Innerhalb des Fahrzeugs wird davon ausgegangen, dass die eingehende externe Diagnose-Kommunikation im zentralen Gateway auf eine signal-orientierte CAN-Kommunikation geroutet wird.

5.1.1 Merkmale der Diagnosekommunikation

Die Fahrzeugdiagnose gemäß UDS-Protokoll erfolgt nach dem *Request/Response* Verfahren, die ausgehend von einer externen Diagnoseanwendung gestartet wird und in einer spezifischen Reihenfolge abläuft. Je nach Diagnoseanwendung bzw. Auswahl eines bestimmten Diagnosevorgangs (z.B. Löschen des Fehlerspeichers einer ECU) werden vordefinierte Diagnosesequenzen an das Fahrzeug gesendet. Dies kann beispielsweise nach dem Start einer Sitzung zunächst die Ausführung einer vollständigen Steuergerätesuche beinhalten (s. Abbildung 5.2). Danach erfolgt durch den Diagnose-Benutzer die Auswahl einer bestimmten ECU, auf der weitere Diagnosefunktionen (Sub-Funktionen) ausgeführt werden können. Die Anzahl der Steuergeräte und verfügbaren Funktionen sind allgemein abhängig von der jeweiligen Fahrzeugbaureihe sowie der zugehörigen Ausstattungsvariante. Das Kommunikationsmuster wird dabei über zwei Faktoren definiert. Einerseits sind verfügbare Funktionen über die Herstellerspezifikation bekannt und dadurch in Form von zusammenhängenden Diagnoseabfolgen in Diagnose-Anwendungen implementierbar. Andererseits beeinflusst der aktive Benutzer die Muster, da diese je nach Erfahrungslevel un-

terschiedliche Ausführungsreihenfolgen im Rahmen eines Diagnosevorgangs wählt.

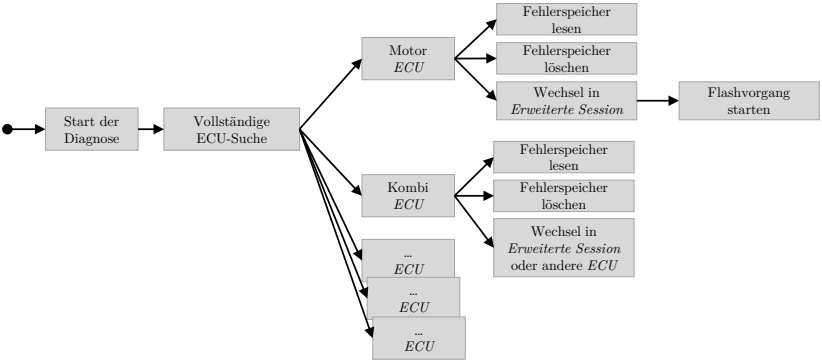


Abbildung 5.2: Exemplarische Übersicht von möglichen Diagnosepfaden, die ein Benutzer durchlaufen könnte.

5.1.2 Klassifizierung von Anomalien in der Diagnosekommunikation

Im Bereich CPS lassen sich Anomalien allgemein auf Basis der jeweils verwendeten IDS-Features klassifizieren (s. Abschnitt 3.2.3). Auf dieser Basis ist eine weitere Klassifikationsebene existent, die eine Fluss-basierte (engl. Flow-based) von einer Nutzdaten-basierte (engl. Payload-based) Erkennung unterscheidet [103]. Die Flussdaten können dabei Informationen wie beispielsweise die Zykluszeiten von Nachrichten oder die Auswertung von Header-Daten sowie die Analyse der *Entropie* von Nachrichten umfassen. Im Gegensatz dazu, analysieren bzw. bewerten Nutzdaten-basierte Ansätze den Inhalt von Nachrichten. Mit dem Fokus auf die On-Board CAN-Kommunikation, verfeinert Müter et al. [180] die Informationen, die für die Erkennung von Anomalien betrachtet werden sollten. Dafür definieren die Forscher acht verschiedene Anomalie-Sensoren (s. Tabelle 5.1), die jeweils einen bestimmten Anomalietyp adressieren. Eine Klassifizierung von Anomalien für die Diagnosekommunikation ist derzeit nicht existent. Eine Einordnung ist jedoch auf der Grundlage der verschiedenen Sensoren von Müter et al. möglich (s. Tabelle 5.1).

Tabelle 5.1: Übersicht von Anomalieerkennungssensoren basierend auf Müter et al. [180] sowie mögliche Anwendbarkeit auf die Diagnosekommunikation.

Nr.	Sensor	Beschreibung	Diagnose
1	Formality	Header-Analyse (z.B. Nachrichtenlänge)	ja
2	Location	Nachricht auf CAN-Bus x erlaubt?	ja
3	Range	Abgleich Signal-Wertebereich mit Spezifikation	partiell
4	Frequency	Zykluszeiten	nein
5	Correlation	Abgleich mit Nachrichten auf anderen Bussen	nein
6	Protocol	Korrekte Reihenfolge, Challenge-Response Protokolle	ja
7	Plausibility	Nutzdatenbewertung auf Plausibilität	partiell
8	Consistency	Bewertung von Daten auf Basis mehrerer Quellen	partiell

Definition 5.1.1 Entropie

Die Entropie definiert innerhalb der Informationstheorie das Maß des mittleren Informationsgehalts einer Nachricht [112].

Definition 5.1.2 Anomalie-Sensor

Ein Anomalie-Sensor überwacht nach Müter et al. [180] eine spezifische Eigenschaft innerhalb der On-Board Fahrzeugkommunikation (z.B. Zykluszeit von Nachrichten). Die Überwachung basiert dabei auf einem spezifizierten Verhalten wie beispielsweise einer Kommunikationsmatrix.

Die Diagnosekommunikation unterscheidet sich teilweise durch andere Kommunikationseigenschaften von der On-Board CAN-Kommunikation, wodurch nicht alle Sensoren für ein potentielles Diagnose-IDS adaptierbar sind. Nachfolgend werden die Unterschiede der einzelnen Sensoren erläutert:

- Formality: Die Prüfung formaler Eigenschaften auf Header-Ebene, ist bei Diagnosenachrichten prinzipiell gegeben, da beispielsweise Adressinformationen sowie zugehörige Nachrichtenlängen in der Kommunikationsmatrix spezifiziert sind.

- Location: Für jede Domäne der On-Board CAN-Kommunikation (z.B. Antriebsstrang, Komfort) ist eine definierte Menge an bestimmten Nachrichten spezifiziert. Die Erkennung davon abweichender Nachrichten ist auch für Diagnosenachrichten möglich, da diese ebenfalls definiert sind.
- Range: Die Analyse von Signal-Wertebereichen auf Basis der Spezifikation ist nur teilweise auf die Diagnose übertragbar, da Diagnose-Nachrichten verschiedene Arten von Informationen übertragen. Als Beispiel sind hier z.B. Daten bei einem Flashvorgang im Rahmen eines Updates zu nennen. Ein derartiger Sensor ist dadurch nur partiell für Diagnosenachrichten einsetzbar.
- Frequency: Die Nachrichtenfrequenz bildet ein wesentliches Unterscheidungsmerkmal zwischen der On-Board und Diagnose-Kommunikation. Da Diagnose-Nachrichten ausschließlich sporadisch übertragenen werden und keine definierten Zykluszeiten zugewiesen sind, ist eine Anomalieerkennung auf dieser Ebene nicht umsetzbar. Eine Ausnahme bildet die Überwachung einer Mindest-Zeitdifferenz von zwei aufeinanderfolgenden Nachrichten, um beispielsweise einen DoS-Angriff (s. Abschnitt 2.2.1) zu verhindern.
- Correlation: Diese Art von Sensor prüft ein paralleles Auftreten von Nachrichten auf anderen im Fahrzeug existenten Bussen, um Abweichungen von der Spezifikation zu erkennen. Ein zeitlich paralleles Auftreten ist für bestimmte Nachrichten beabsichtigt, um eine Informationen von einem Quellen-Steuergerät über ein Gateway in verschiedene Architektur-Domänen zu verteilen. Für die Diagnosekommunikation ist dieses Verhalten jedoch untypisch, da die Kommunikation zielgerichteter erfolgt.
- Protocol: Der Sensor prüft die Einhaltung eines bestimmten Protokollablaufs. Müter et al. führen auf dieser Basis die Möglichkeit zur Analyse der Protokollreihenfolge an, um durch einen Angreifer verursachte Abweichungen zu erkennen. Als Beispiel nennen die Autoren das Challenge-Response Verfahren (Security-Access, s. Abschnitt A.1.5) für erweiterte Diagnosedienste.
- Plausibility: Die semantische Analyse ermöglicht die Überprüfung der Plausibilität von Daten. Ein Beispiel ist die Identifikation von Sprüngen in einem Geschwindigkeitssignal innerhalb eines definierten Zeitinter-

valls. Diese Art von Prüfung ist nur partiell auf die Diagnose abbildbar, da die Nutzdaten nicht immer plausibilisierbar sind (z.B. Flash-Daten oder Ansteuerbefehle).

- Consistency: Diese Art der Analyse erweitert die Plausibilitätsprüfung, indem weitere Quellen für die Prüfung miteinbezogen werden. Ein Beispiel wäre der Abgleich des Geschwindigkeitssignals (basierend auf den Raddrehzahlen) mit der Geschwindigkeit des GPS-Sensors. Eine Adaptierung dieses Sensors wäre für die Diagnose partiell möglich. Jedoch ist nicht jede Art von Diagnosenachricht mit anderen Quellen plausibilisierbar.

5.1.3 Anomalie-Sensoren für die Insider-Erkennung

Basierend auf dem Stand der Technik und Wissenschaft (s. Abschnitt 3.2) sind aktuell keine IDS-Ansätze existent, die den Fokus auf die Erkennung von Anomalien innerhalb der Diagnosekommunikation legen. Jedoch führt die Arbeit von Weber [111] eine Bewertung durch, welche Art von Erkennungsmethode (statische oder lernende Checks) basierend auf den Sensoren von Müter et al. (s. Tabelle 5.1) realisierbar bzw. notwendig sind. Statische Checks umfassen dabei Überwachungsregeln, die aus der Spezifikation heraus in ausführbaren Code übersetzt sind. Dagegen beruhen lernende Checks auf Trainingsdaten beispielsweise unter der Anwendung von Methoden des maschinellen Lernens (s. Abschnitt 2.3.1). Dabei ordnet Weber den Sensoren Nr. 1 - 5 eine Realisierung mittels statischer Checks zu. Die Sensoren Nr. 7 und 8 enthalten dagegen lernende Checks. Der Protocol-Sensor (Nr. 6) ist gesondert zu betrachten, da hierbei beide Arten von Checks realisierbar sind. Für bekannte Protokollabläufe (z.B. dem Security-Access in Diagnoseprotokollen, s. Abschnitt A.1.5) sind die Abfolgen exakt spezifiziert und dadurch für statische Checks geeignet. Hingegen ist bei proprietären Protokollen ein lernender Check notwendig, da hier ein Verhalten trainiert bzw. abgeleitet werden muss.

Transferiert auf die automotive Diagnosekommunikation lassen sich daraus folgende Erkenntnisse ableiten:

- Auf der Basis einer vorhandenen Spezifikation ist je nach Detaillierungsgrad die Ableitung von Regeln für statische Checks möglich (Sensoren 1 - 3).
- Für die Sensoren 7 und 8 ist nur eine eingeschränkte Möglichkeit für lernende Checks gegeben, da die Diagnosenachrichten nicht grundsätzlich die passenden Merkmale im Vergleich zu On-Board CAN-Nachrichten aufweisen.
- Der Protocol-Sensor (Nr. 6) bietet im Vergleich zu den anderen Sensoren das größte Potential bzgl. einer Extraktion bzw. Analyse von Verhaltensmustern und bildet damit die Grundlage zur Erkennung von Insider-Angriffen.

Spezifizierung von kontextabhängigen Diagnose-Anomalien

Da die Diagnose bisher noch nicht mit Fokus auf die Erkennung von Insider-Angriffen (s. Abschnitt 2.2.1) in der Technik und Wissenschaft adressiert wurde, sind bisher keine zugehörigen Anomalietypen spezifiziert. Aus diesem Grund werden nachfolgend Verhaltensmuster abgeleitet, die ein derartiger Angriffstyp auslösen könnte. Den Mustern liegt die Annahme zugrunde, dass der Insider keine Kenntnis über das exakte Normalverhalten eines legitimierten Benutzers (z.B. Werkstattmitarbeiter) hat. Es können lediglich über standardisierte Kommunikationseigenschaften des UDS-Diagnoseprotokolls (s. Abschnitt A.1.1), Teile bestimmter Zusammenhänge (Sequenzabfolgen) rekonstruiert werden. Dazu ist davon auszugehen, dass ein Angreifer ähnlich wie ein Penetrationstester (s. Abschnitt 3.1.3) nur ein eingeschränktes Wissen über sein Zielobjekt (Fahrzeugsystem) hat und versucht offene Schwachstellen durch beispielsweise die Anwendung des *Trial&Error*-Prinzips zu finden. Dadurch resultiert ein anderes Kommunikationsverhalten im Vergleich zu legitimierten Benutzern. Auf Basis eines bekannten Normalverhaltens lassen sich daraus kontextabhängige Anomalietypen definieren:

1. Veränderung der Ausführungsreihenfolge: Da der Insider keine exakte Kenntnis über die Protokollabfolge des Normalverhaltens hat, verursacht dieser durch sein Vorgehen eine Abweichung.

2. Einfügen zusätzlicher Nachrichten: Ein Insider könnte über ein potentielles Vorwissen (Analyse von standardisierten Abläufen), Teile der Kommunikation rekonstruieren und zusätzliche Diagnosesequenzen einfügen, um ein definiertes Ziel (z.B. Ansteuerung eines Aktors) zu erreichen.

3. Ersetzen von Nachrichten: Ein Insider könnte ebenfalls bestimmte Teile eines von ihm bekannten Kommunikationsablaufs verändern, indem er gezielt einzelne Sequenzen ersetzt, um wiederum ein definiertes Ziel zu erreichen.

4. Veränderung des kontextuellen Zusammenhangs innerhalb einer Diagnosenachricht: Der Insider versucht Hersteller-spezifische Funktionen bzw. Unterfunktionen anzusteuern, die nicht dem legitimierten Nutzungsverhalten entsprechen.

Definition 5.1.3 Trial&Error-Prinzip

Als Trial&Error-Prinzip wird eine heuristische Methode bezeichnet, die durch die Variation potentieller Lösungsmöglichkeiten, eine gewünschte Lösung finden soll.

5.1.4 Ableitung von Features für die Anomalieerkennung

Die Abfolge von Diagnosesequenzen ist einerseits abhängig von der implementierten bzw. verwendeten Diagnoseanwendung sowie zum anderen vom jeweiligen Anwendungsbenutzer. Auf dieser Grundlage lassen sich Verhaltensmuster ableiten, die für die Erkennung von Anomalien nutzbar sind. In der klassischen IT wird die Analyse von Benutzerverhalten als *User Entity Behavior Analytics (UEBA)* bezeichnet [181]. Dabei werden beispielsweise bei einem Computersystem, das durch einen Benutzer mit einem Server interagiert, bestimmte Features wie die Zeitstempel des ersten oder letzten Zugriffs am Tag, Zeitdauer zwischen dem ersten und letzten Zugriff oder bestimmte Ausführungsreihenfolgen von Anwendungen analysiert. Als Erkennungstechniken für Anomalien sind dafür statistische Sprachmodelle (engl. Statistical Language Model (SLM)) verbreitet, welche N-Gramme verwenden [182], [183].

Definition 5.1.4 User Entity Behavior Analytics (UEBA)

Unter User and Entity Behavior Analytics werden Verfahren definiert, die ein Verhalten von IT-Entitäten überwachen sowie analysieren. Das Ziel ist dabei die Erkennung von Abweichungen (Anomalien) von einem bekannten Normalverhalten [184].

Für Diagnoseanwendungen kann eine Adaption dieser Ansätze erfolgen, indem ebenfalls verschiedene Erkennungsfeatures definiert werden. So ist aus der Reihenfolge der Sequenzen, die von der Anwendung zum Fahrzeug versendet werden, ein Normalverhalten ableitbar. Das Verhalten kann dabei auf unterschiedlichen Entitäten entsprechen. Zum einen ist beispielsweise ein Normalverhalten auf Basis eines Benutzers definierbar. Zum anderen kann das Normalverhalten auch einer Menge von Benutzern entsprechen, die eine Anwendung ausführen. Sendet ein Angreifer nun eigene Diagnosesequenzen zum Fahrzeug, entstehen Abweichungen zum Normalverhalten, da dieser in einer anderen Reihenfolge vorgeht (s. Abbildung 5.3).

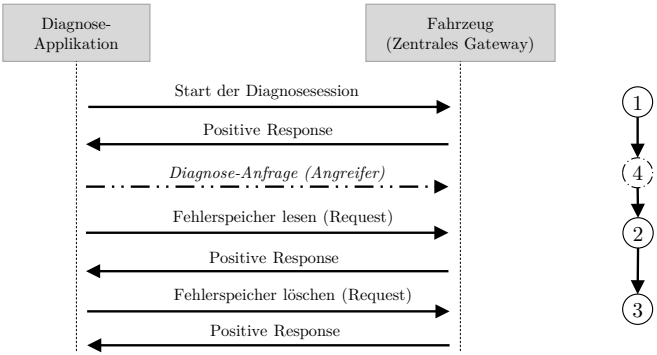


Abbildung 5.3: Exemplarische Abfolge an Diagnoseschritten zwischen einer Diagnoseapplikation und einem Fahrzeug.

Das Normalverhalten eines Benutzers würde in dieser exemplarischen Darstellung einer Abfolge der Sequenznummern (1, 2, 3) entsprechen. Ein Angreifer könnte dagegen die Sequenzabfolge (1, 4, 2, 3) senden. Dabei entspricht die

jeweilige Sequenznummer einer bestimmten Diagnosefunktion (z.B. Fehlerspeicher lesen).

5.1.5 Erkennung und Klassifikation

Für eine Erkennung von Verhaltens-spezifischen Anomalien muss ein Protocol-Sensor sowohl auf Sequenz- als auch auf Byte-Ebene die Kontext-basierten Informationen der Diagnose-Frames analysieren sowie klassifizieren. Durch die Anwendung einer hybriden Erkennungstechnik (Sequenz- und Byte-Ebene) lassen sich grundlegend drei Fälle definieren (s. Abbildung 5.4). Zuerst ist eine Überprüfung auf einer Sequenz-basierten Ebene möglich. Wird keine Anomalie erkannt, werden die Daten als *normal* (*Fall 1*) klassifiziert. Wird hingegen eine Anomalie erkannt, erfolgt eine zusätzliche Überprüfung auf der Byte-Ebene des jeweiligen Diagnose-Frames. Diese Prüfung führt entweder zum Ergebnis *keine Anomalie erkannt* (*Fall 2*) oder *Anomalie erkannt* (*Fall 3*).

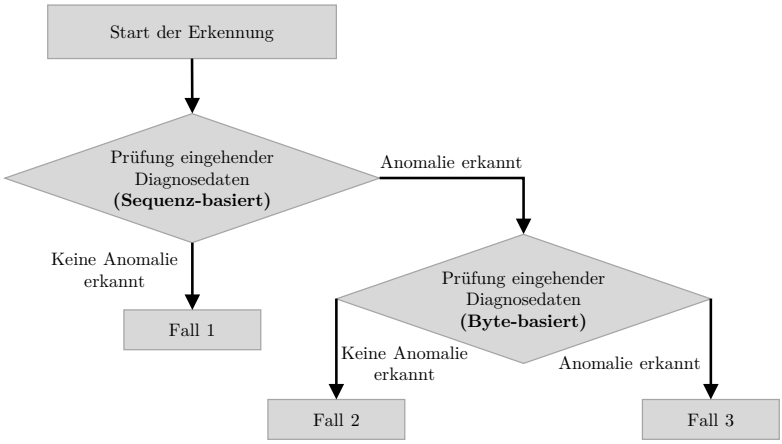


Abbildung 5.4: Ablaufplan einer hybriden Anomalieerkennung auf Basis eingehender Diagnosenachrichten.

Die Fallunterscheidung ermöglicht weitere Analysen, um Erkenntnisse über mögliche Anomalien abzuleiten:

Im **Fall 1** ist die Sequenzreihenfolge der untersuchten Daten (Verhalten) in den Trainingsdaten enthalten und wird daher als Normalverhalten eingestuft.

Der **Fall 2** beinhaltet vier verschiedene Verhaltensaktivitäten:

- Eine Sequenz wurde möglicherweise durch eine im Trainingsdatenset vorhandene Sequenz ersetzt, sodass damit die normale Sequenzreihenfolge nicht korrekt ist. Zur Verifizierung dieses Anomalietyps kann der folgende Schritt durchgeführt werden: Neuberechnung der Wahrscheinlichkeiten auf Basis der empfangenen Diagnosenachrichten unter Ausschluss der als Anomalie erkannten Sequenz. Danach wird weiterhin eine Anomalie detektiert.
- Normale Flow Control Frame (FC) Diagnose-Nachrichten: Die FC-Nachrichten (s. Abschnitt 2.1.3) weisen nicht immer einen eindeutigen kontext-bezogenen Zusammenhang auf und treten in Diagnoseabläufen unregelmäßig auf. Dadurch kann es durch den Sequenz-basierten Ansatz bei diesen Botschaftstypen zu falsch-positiv Anomalien führen. Durch eine nachgelagerte Auswertung des PCI-Bytes (s. Abbildung A.3) ist eine Verifizierung möglich.
- Abweichende Sequenzreihenfolge bzgl. des Normalverhaltens: Die Sequenznummern, die als Anomalie detektiert werden, können in der Reihenfolge vertauscht werden, um eine erneute Berechnung der Wahrscheinlichkeiten durchzuführen. Wird diese ohne erkannte Anomalien abgeschlossen, bestätigt diese die ursprünglich veränderte Sequenzreihenfolge.
- Wiedereinspielen einer zusätzlichen Nachricht (s. Definition A.1.8) die aus dem Normalverhalten aufgezeichnet wurde: Die als Anomalie erkannte Sequenz kann zunächst ausgeschlossen werden und mit den verbleibenden Sequenzen eine erneute Berechnung durchgeführt werden. Ist daraufhin keine Anomalie mehr vorhanden, bestätigt dies das Einfügen einer zusätzlichen Diagnosenachricht, die eine Abweichung vom Normalverhalten darstellt.

Der **Fall 3** umfasst folgende Aktivitäten:

- Eine Diagnosesequenz wurde durch eine vom Normalverhalten abweichende Nachricht ersetzt (ähnlich 1. Variante von Fall 2). Die erkannte

Anomalie muss manuell weiter analysiert werden, um den Einfluss bewerten zu können.

- Einfügen einer zusätzlichen Nachricht, die nicht zum Normalverhalten passt. Die mittels dem Sequenz-basierten Ansatz erkannte Anomalie kann zunächst isoliert und eine erneute Berechnung durchgeführt werden. Ist die Sequenz-Anomalie dadurch aufgelöst jedoch im Byte-basierten Ansatz weiter detektierbar, muss eine manuelle Analyse bzgl. den enthaltenen Nutzdaten erfolgen.

5.1.6 Existente Kontext-basierte Erkennungsmethoden

Im Bereich der Wissenschaft und Technik (s. Abschnitt 3.2.3) existieren derzeit lediglich zwei Ansätze, die eine Kontext-basierte Anomalieerkennung für die On-Board Fahrzeugkommunikation adressieren. Im ersten Ansatz nutzen die Autoren Wasicek et al. [154] verschiedene Sensorquellen, um mittels Korrelation statistische Abhängigkeiten der unterschiedlichen Sensorgrößen während des aktiven Fahrzeugbetriebs abzuleiten. Transferiert auf die Diagnosekommunikation sowie die Ableitung eines Benutzerverhaltens ist diese Methode nur eingeschränkt geeignet. Im Rahmen von Diagnosesitzungen besteht vorwiegend kein aktiver Fahrbetrieb, sodass die Sensoren für beispielsweise die Gaspedalstellung oder Motordrehzahl bzgl. einer Insider-Erkennung, keine verwertbaren Informationen liefern. Der zweite Ansatz von Kalutarage et al. [155] nutzt die N-Gramm Methode aus dem Bereich der CL als Basis zur Erkennung von Anomalien. Die Methode ermöglicht es, eine Auftretenswahrscheinlichkeit von einzelnen CAN-Nachrichten zu berechnen. Als Erkennungsfeature nutzen die Autoren ausschließlich die jeweilige Nachrichten-ID. Die durchgeführte Evaluierung zeigte, dass der Ansatz zwei verschiedene Spoofing-Angriffe 2.2.1 zuverlässig erkannte. Dabei handelte es sich um böswillig eingefügte Nachrichten zur Manipulation der Motordrehzahl sowie Getriebestellung. Hingegen wären Veränderungen der Nutzdaten eines CAN-Frames nicht detektierbar. Der Ansatz bietet allerdings ein Potential für die Adaptierung auf die Diagnosekommunikation, da ein kontextuelles Verhalten extrahierbar ist.

5.1.7 Herausforderungen bei der Erkennung von Diagnose-Anomalien

Da der aktuelle Stand der Wissenschaft und Technik (s. Abschnitt 3.2.3) bisher keine Arbeit zu IDS-Ansätzen im Bereich von automotive Diagnoseanwendungen enthält, existieren folglich auch keine öffentlich verfügbaren Kommunikationsdatensätze auf Basis von Diagnoseprotokollen. Für die Entwicklung eines IDS-Ansatzes muss in dieser Arbeit eine eigene Aufzeichnung von Diagnosekommunikation erfolgen. Dabei ist anzumerken, dass auf dieser Grundlage lediglich ein Proof-of-Concept erfolgen kann, da sich die Aufzeichnung auf Diagnosevorgänge einer Fahrzeugbaureihe und eines legitimierten Benutzers beschränkt. Zusätzlich ist die Aufzeichnung von Diagnosenachrichten im Vergleich zur On-Board CAN-Kommunikation schwieriger, da nicht zyklisch und automatisiert eine hohe Buslast vorhanden ist. Die nachfolgende Untersuchung basiert daher auf einem selbst aufgezeichneten Realdatenset, das ungefähr 5000 Diagnosenachrichten enthält.

5.2 Konzept für die Erkennung spezifischer Diagnoseanomalien

Ausgehend von den grundsätzlichen Überlegungen des vorangegangen Abschnitts in Bezug auf ein potentielles Angriffsszenario, dessen Erkennung sowie die damit verbundenen Herausforderungen wird nachfolgend ein IDS-Ansatz entwickelt, der eine erste Grundlage für die Erkennung von Anomalien in der Diagnosekommunikation liefert. Die dafür verwendete Methode basiert auf der Adaptierung des IDS-Ansatzes von Kalutarage et al. (s. Abschnitt 3.2.3), der unter Verwendung von N-Grammen (s. Abschnitt 2.3.2) Abweichungen von einem Normalverhalten in der On-Board CAN-Kommunikation zuverlässig detektiert[1].

[1] Teile des entwickelten Erkennungsansatzes für Anomalien in Diagnoseanwendungen wurden in [RLF+20] publiziert.

5.2.1 Funktionale- und nicht-funktionale Anforderungen

Für die Entwicklung des Erkennungsansatzes werden bzgl. Erkennungstechnik sowie des zu analysierenden Kommunikationsprotokolls nachfolgende Anforderungen definiert.

- **RQ1:** Die Anomalieerkennung soll Protocol Data Units (PDUs) (s. Definition A.1.7) des UDS-Diagnoseprotokolls auf Abweichungen bzgl. des Normalverhaltens erkennen.
- **RQ2:** Die eingehenden Diagnosesequenzen sollen im zentralen Gateway des Fahrzeugs analysiert werden.
- **RQ3:** Das Normalverhalten soll über ein geeignetes Verfahren aus vorhandenen Diagnosesequenzen extrahierbar sein.
- **RQ4:** Das IDS soll auf Basis einer signal-orientierten CAN-Kommunikation arbeiten.
- **RQ5:** Das System soll sowohl Anomalien in der Sequenzreihenfolge als auch Veränderungen innerhalb einer Diagnosebotschaft (Byte-Ebene) erkennen können.

5.2.2 Systemübersicht und Funktionsweise

Auf Basis der Anforderungen wird das System so ausgelegt, dass das IDS im Fahrzeuggateway integriert wird (s. Abbildung 5.5). Dadurch ist es in der Lage eingehende Diagnosedatenströme von externen Netzwerkkomponenten (Cloud-Applikationen, lokale Diagnosehardware) zu analysieren. Unter der Verwendung der N-Gramm Methode werden Diagnosesequenzen analysiert und deren Auftrittswahrscheinlichkeiten (s. Abschnitt 5.4.2) berechnet. Das dafür genutzte Erkennungsmodell (s. Abschnitt 5.2.3) gliedert sich in zwei verschiedene Ebenen. Zunächst wird auf Basis der eingehenden Sequenzabfolgen analysiert, ob Abweichungen zum Normalverhalten vorliegen. Daran anknüpfend erfolgt eine Prüfung auf Byte-Ebene eines einzelnen Diagnose-Frames. Der zweistufige Ansatz ermöglicht eine Klassifikation der erkannten Anomalien (s. Abschnitt 5.1.5).

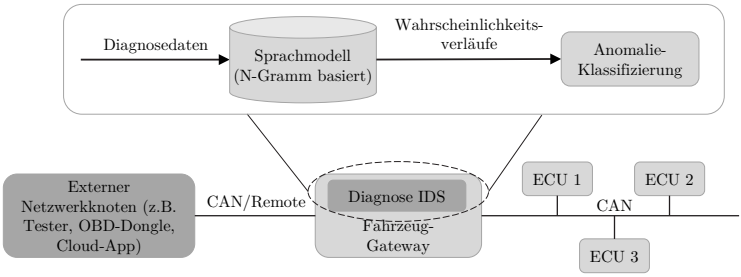


Abbildung 5.5: Schematischer Aufbau des IDS zur Erkennung von Anomalien in eingehenden Diagnosebotschaften.

5.2.3 Erkennungsmodelle

Natürliche Sprachen wie beispielsweise Deutsch, Englisch oder Spanisch können im Vergleich zu formalen Sprachen (z.B. Programmiersprachen wie ANSI C) Mehrdeutigkeiten enthalten und kontext-bezogene Informationen vermitteln. Damit Computer die natürlichen Sprachen und deren Kontext verarbeiten und einordnen können sind mathematische Modelle notwendig. Ein etabliertes Modell in der CL bilden statistische Sprachmodelle (s. Abschnitt 2.3.2). Dabei werden von einer Sequenz (bestehend aus m Wörtern oder Sätzen), die bedingten Wahrscheinlichkeiten $p(w_1, w_2, ..., w_m)$ für das Auftreten des nächsten Wortes in einer Sequenz berechnet.

Die bestimmte Anordnung von Diagnosebotschaften (definiert durch Protokolle sowie dem zugrundeliegenden Benutzerverhalten, s. Abschnitt A.1.1) kann als Sprache verstanden werden. Dabei entspricht die Abfolge der Botschaften einem Satz sowie die Botschaften selbst einem Wort. Auf der Ebene einer Botschaft kann diese als Satz sowie die enthaltenen Bytes als Wörter definiert werden. Auf Basis dieser Definition ist die N-Gramm Technik auf die Diagnose anwendbar, um Verhaltensmuster aus der Kommunikation abzubilden. Der folgende Ansatz greift diese Idee auf, um eine kontext-basierte Anomalieerkennung zu entwickeln. Die dabei zugrundeliegenden Sprachmodelle ermöglichen die Berechnung einer Auftretenswahrscheinlichkeit von eingehenden Nachrichten bzw. bestimmter Sequenzen. Sendet ein Angreifer seine eigene Sequenzreihenfolge ergibt sich ein zum Normalverhalten veränderter Kontext,

der als Anomalie definiert wird. Die Berechnung der Wahrscheinlichkeiten erfolgt dabei über zwei Sprachmodelle, die mit unterschiedlichen Features (Sequenz und Byte-basiert) arbeiten.

Sequenz-basiertes Modell

Aufgrund der sequentiellen Übertragungseigenschaft von Diagnosebotschaften existiert im Vergleich zur natürlichen Sprache kein Satzanfang sowie Satzende. Für die Definition eines Satzes wird daher ein Sliding Window (s. Abbildung 5.6) verwendet, dass zu jedem Abtastzeitpunkt eine definierte Anzahl an Diagnose-Frames umfasst. Die Größe des Fensters m definiert dabei die Satzlänge S, um daraus die Wahrscheinlichkeiten von verschiedenen N-Grammen zu berechnen (z.B. $n = 2$ für ein Bigramm).

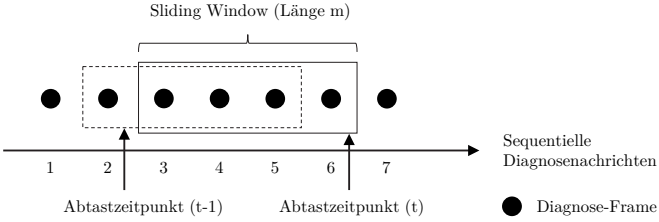


Abbildung 5.6: Aufbau des Sequenz-basierten Sprachmodells unter Verwendung eines Sliding Windows für die Berechnung der N-Gramme von eingehenden Diagnosenachrichten.

Byte-basiertes Modell

Nachdem ein Sprachmodell auf Basis der einzelnen Sequenzen innerhalb einer Diagnosesitzung definiert wurde, kann auf einer weiteren Ebene der Kommunikationsdaten ein zusätzliches Sprachmodell spezifiziert werden. Dieses stützt sich auf einzelne Bytes einer Diagnose-PDU des Unified Diagnostic Services (UDS)-Protokolls (s. Abbildung 5.7), die in die Nutzdaten eines CAN-Frames (s. Abbildung A.2) eingebettet sind. Die Bytes werden dabei als einzelne Wörter interpretiert, um einen kontextuellen Zusammenhang abzubilden. Eine PDU repräsentiert damit einen vollständigen Satz. Im Gegensatz

zum Sequenz-basierten Sprachmodell ist für die Berechnung der N-Gramme kein Sliding-Window notwendig, da immer eine vollständige PDU ausgewertet wird.

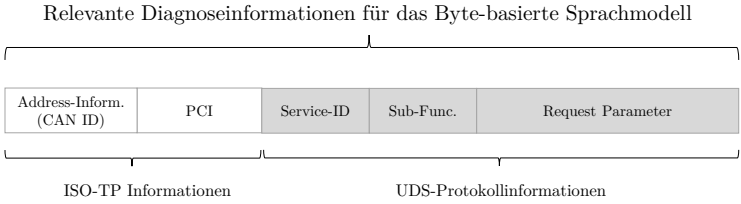


Abbildung 5.7: Aufbau einer Diagnosenachricht mit Transport- und UDS-Protokollinformationen eingebettet in ein CAN-Frame, die im Byte-basierten Modell verarbeitet werden.

Für eine Diagnosekommunikation nach dem UDS-Standard (s. Abschnitt A.1.1) ist der kontextuelle Zusammenhang über die einzelnen Nutzdatenbytes gegeben. Beispielweise steht die *Service ID* in Zusammenhang mit den verfügbaren *Sub-Funktionen*, die wiederum abhängig von der Adresse (*CAN-ID*) des jeweiligen Steuergeräts ist. Auf dieser Grundlage lassen sich statistische Zusammenhänge berechnen. Wird beispielsweise von einem Angreifer versucht eine Sub-Funktion anzusteuern, die statistisch gesehen bisher nie im Zusammenhang mit einer entsprechenden CAN-ID stand, wird eine niedrige Auftretenswahrscheinlichkeit über das Sprachmodell berechnet.

Einflüsse durch die Verwendung unterschiedlicher N-Gramme

Für das Sequenz-basierte Sprachmodell ist es entscheidend, welche Parameter für die N-Gramme n bzw. Satzlänge (Sliding Window) m gewählt werden, da diese Werte die Wahrscheinlichkeitsverläufe beeinflussen. Aus dem Stand der Wissenschaft ist im Bereich der natürlichen Sprachen bekannt, welche Auswirkungen sich durch die Anwendung von unterschiedlichen N-Grammen auf Basis verschiedener Mengen an Trainingsdaten ergeben. Daraus folgt, dass N-Gramme höherer Ordnung nicht immer einen positiven Einfluss auf die Berechnung der Auftrittswahrscheinlichkeit haben. Die Arbeit von Nguyen [185] zeigt, dass sich die berechneten Wahrscheinlichkeitswerte durch die Verwendung von N-Grammen ($n > 3$) deutlich verringern im Vergleich zu einem

Bigramm. Dieses Phänomen lässt sich auf die steigende Anzahl der unbekannten N-Gramme zurückführen, die in Testdaten auftreten, jedoch nicht in den Trainingsdaten enthalten sind. Die Arbeit von Allison et al. [186] enthält eine Untersuchung welchen Einfluss die Korpusgröße (s. Abschnitt 2.3.2) unter der Verwendung von verschiedenen N-Grammen auf den Überdeckungsgrad von Testdaten hat. Demnach ergibt sich bei einem untersuchten Sprachkorpus von 160k Wörtern auf Basis von Bigrammen ein Überdeckungsgrad von 33,2 %. Unter Verwendung desselben Korpus ergibt sich bei Nutzung von Trigrammen ein Überdeckungsgrad von 5,8 %. Wird die Anzahl der Wörter im Korpus auf 100 Millionen erhöht, ergeben sich Überdeckungsgrade von 88,8 % (Bigramm) sowie 57,6 % (Trigramm). Der Überdeckungsgrad korreliert im Bereich der Spracherkennung mit der *Wortfehlerrate* (engl. Word Error Rate (WER)).

Unter der Annahme eines ausreichend großen Korpus ist die Verwendung eines möglichst großen n empfehlenswert, da der darin enthaltene Informationsgehalt ansteigt. In der Praxis ist dies meist nicht möglich und die Datensätze enthalten nicht alle theoretisch vorkommende Wortkombinationen. Das kann dazu führen, dass fehlende Kombinationen mit einer Wahrscheinlichkeit von 0 im Korpus gespeichert werden. Durch die Verwendung von Glättungsverfahren (s. Abschnitt 2.3.2) wird dieser Problematik entgegengewirkt.

Definition 5.2.1 Wortfehlerrate

Die Wortfehlerrate gibt den prozentualen Anteil von Wörtern an, die zur Gesamtmenge falsch detektiert wurden [112].

Varianten und Modelle

Betrachtet man im Rahmen der Diagnosekommunikation mögliche Varianten und Modelle von Fahrzeugen, die in Bezug auf die Anomalieerkennung auf Basis der N-Gramm Methode im Realbetrieb von Interesse sind, ist eine Klassifizierung durch die verschiedenen Baureihen eines Herstellers möglich (s. Abbildung 5.8). Ausgehend von einer Diagnose-Anwendung, die entweder lokal auf einem Werkstatt-Diagnosegerät oder remote in einem Hersteller-Backend läuft, ist die Menge aller darin auftretenden Diagnosevorgänge für Fahrzeuge einer bestimmten Baureihe als Normalverhalten (Korpus) definierbar.

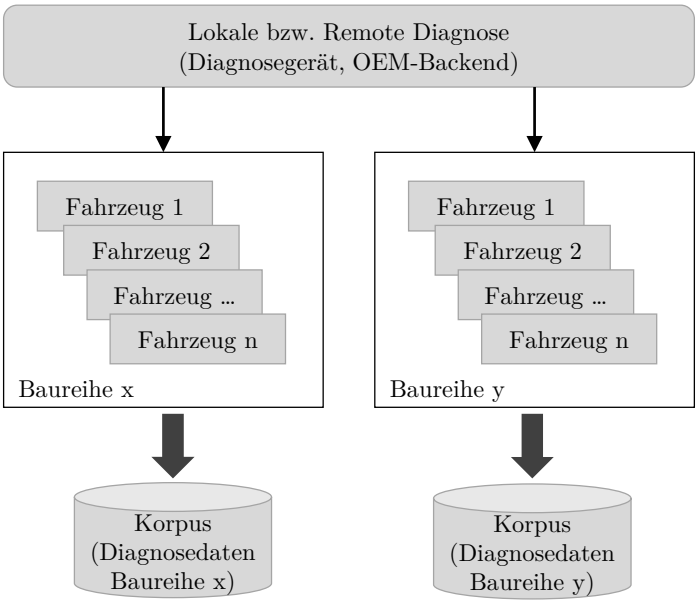


Abbildung 5.8: Schematische Übersicht an unterschiedlichen Varianten (Baureihen) eines Herstellers und sich daraus ergebende Diagnosedaten (Korpus), die durch die lokale und Remote-Diagnose entstehen und für das Training von Sprachmodellen nutzbar sind.

Darüber hinaus würde die Berücksichtigung der individuellen Ausstattungsvariante eines Fahrzeugs einen weiteren Detaillierungsgrad ermöglichen. Bezogen auf die Anomalieerkennung wird an dieser Stelle jedoch angenommen, dass diese Detailtiefe keinen erheblichen Mehrwert in Bezug auf die Insider-Erkennung liefern würde. Als Folge ergibt sich, dass nicht plausible Diagnoseabfolgen (z.B. Reset einer nicht verbauten ECU) in Bezug auf eine spezifische Ausstattungsvariante nicht erkannt werden. Jedoch stellt dieses Szenario keine potentielle Bedrohung dar.

Eine Möglichkeit zur Sammlung dieser Daten wäre die Übermittlung von Diagnosevorgängen eines Fahrzeugs an eine zentrale Hersteller-Datenbank, um auf dieser Grundlage das Training der Sprachmodelle (Sequenz- und Byte-basiert) in einem OEM-Backend durchzuführen. Ein individuelles Fahrzeug

erhält somit lediglich trainierte IDS-Sprachmodelle, die über Software over-the-Air (SOTA) den gesamten Lebenszyklus hinweg kontinuierlich aktuell gehalten werden. Die Nutzung von Flottendaten, Übertragung an ein OEM-Backend und deren zentralisierte Auswertung zur Erkennung von Anomalien der On-Board Fahrzeugkommunikation wurde bereits von Hofmockel in [106] vorgestellt.

5.2.4 Verfahren zur Erkennung von Anomalien

Nach der Adaption der zwei Sprachmodelle auf die Diagnosekommunikation wird ein Verfahren zur Detektion von Anomalien benötigt, das nach Beendigung der Trainingsphase eingehende Testdaten analysiert und in normal bzw. anormal klassifiziert. Ein dafür etabliertes Verfahren ist die Verwendung von Anomalie-Scores [187], [155], [106]. Dabei bilden die Trainingsdaten die Referenz für die Ermittlung der Schwellenwerte (Grenzwerte). In [106] sind unterschiedliche Möglichkeiten für die Bestimmung des Schwellenwerts in Bezug auf verschiedene Ausgangssituationen (Art der vorliegenden Trainingsdaten) erörtert. Eine geeignete Methode stellt dabei die Verwendung eines x-Quantils Q_x als Schwellenwert dar, das auf Basis der ermittelten Scores S der Trainingsdaten definiert wird:

$$\epsilon_{q_x} = Q_x(S) \tag{5.1}$$

Die Methode besitzt den Vorteil, dass Ausreißer (seltene Ereignisse die einen hohen Score aufweisen) in Trainingsdaten ausgeschlossen werden können. Falls der Trainingsdatensatz hingegen ausschließlich ein Normalverhalten abbildet, ist ein 100 %-Quantil wählbar. Für die Anwendung dieser Methode im Bereich der N-Gramme ist die Umwandlung der Scores in Wahrscheinlichkeiten vorteilhaft, da die verwendeten Sprachmodelle ebenfalls Wahrscheinlichkeiten berechnen. Dafür ist über die Häufigkeitsverteilung (Histogramme) der verschiedenen N-Gramme eine Schätzung der Parameter für die Verteilungsfunktionen möglich, um beispielsweise eine Gamma- oder Normalverteilung basierend auf den Trainingsdaten anzunähern (s. Abbildung 5.9).

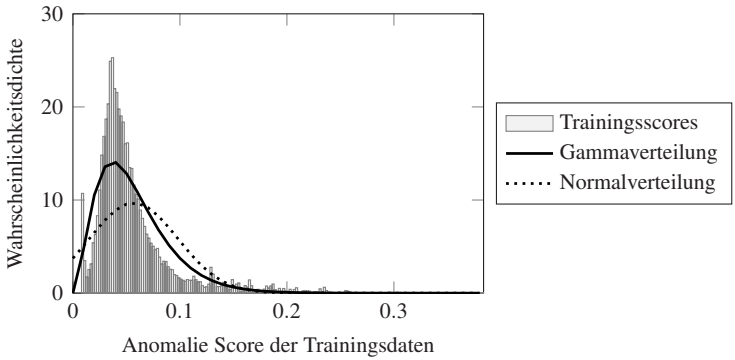


Abbildung 5.9: Exemplarische Darstellung von Anomaliescores und angenäherten Verteilungsfunktionen [106].

Unter Verwendung des Schwellenwertes ist eine Klassifikation von eingehenden Diagnosedaten automatisiert möglich. Dazu wird kontinuierlich die Wahrscheinlichkeit $p(x)$ berechnet und mit dem Schwellenwert verglichen (s. Gleichung 5.2).

$$p(x) \begin{cases} \geq \epsilon & \text{Anomalie} \\ < \epsilon & \text{Normalbereich} \end{cases} \tag{5.2}$$

5.3 Untersuchung verschiedener Erkennungsmodelle

Für die Untersuchung der entwickelten Sprachmodelle werden verschiedene Parameter wie die Größe des Sliding Windows m sowie N-Gramm-Länge untersucht, um den Einfluss bzgl. der Anomalieerkennung zu untersuchen. Diese Ergebnisse werden anschließend mit existenten Auswertungen aus dem Bereich der Sprachverarbeitung (s. Abschnitt 5.2.3) verglichen, um Erkenntnisse ableiten zu können. Dabei werden zur Validierung der N-Gramm Modelle im ersten Schritt Normaldaten (ohne Anomalien) mit Daten, die synthetisch angereicherte Anomalien enthalten, verglichen (s. Abbildung 5.10). Die synthetisch

erzeugten Anomalien basieren dabei auf einer Manipulation des kontextuellen Zusammenhangs gemäß den zuvor generisch beschriebenen Anomalien (s. Abschnitt 5.1.3) innerhalb der Diagnosekommunikation auf der Sequenz- und Byte-Ebene.

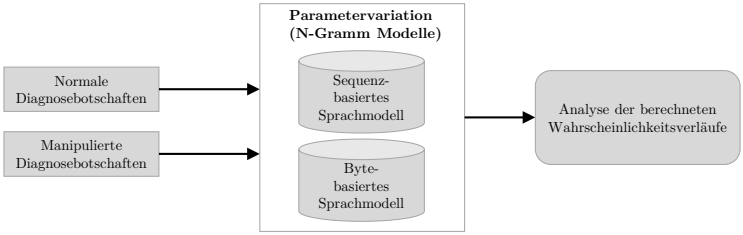


Abbildung 5.10: Schematischer Aufbau zur Untersuchung der verschiedenen Sprachmodelle.

Danach erfolgt in einem zweiten Schritt (s. Abschnitt 5.4) die Untersuchung des entwickelten Anomalie-Detektion Moduls, das auf Basis der Normalverteilung und Definition eines Schwellwerts ϵ (s. Abschnitt 5.2.4) Abweichungen erkennt.

5.3.1 Aufbau & Training der Modelle

Für den Aufbau bzw. das Training der Sprachmodelle sind zunächst verschiedene Datensets notwendig. Für die Gewinnung der Daten (Trainingsdaten) werden reale Diagnosedaten zwischen einem Werkstattdiagnosegerät sowie einem Fahrzeug aufgezeichnet. Dabei führt ein Werkstattbenutzer mehrere Diagnosevorgänge an einem Fahrzeug durch, die als Referenz für ein Normalverhalten dienen. Im Anschluss erfolgt eine Aufbereitung der Rohdaten sowie das eigentliche Training unterschiedlicher N-Gramm Sprachmodelle.

Aufzeichnen von Realdaten

Zur Generierung und Aufzeichnung von realen Diagnosedaten wird ein Werkstatt-Diagnosegerät über die OBD-Schnittstelle mit einem Testfahrzeug verbunden (s. Abbildung 5.11). Das Verbindungskabel enthält dabei einen

integrierten Bypass, um einen Logging-Computer als zusätzlichen CAN-Netzwerkknoten für die Aufzeichnung aller übermittelten CAN-Botschaften zu verwenden. Während der Diagnosesitzung werden verschiedene Diagnoseabläufe und Abläufe auf ECUs (z.B. Airbag, Kombiinstrument oder Motorsteuerung) ausgeführt, um diese als *Trainingsdaten* für die nachfolgende Untersuchung zu nutzen. Neben den Trainingsdaten wird ein zweiter unabhängiger *Testdatensatz* mit verschiedenen Diagnoseabläufen aufgezeichnet. In Summe enthalten beide Datensätze ungefähr 5000 Diagnosebotschaften.

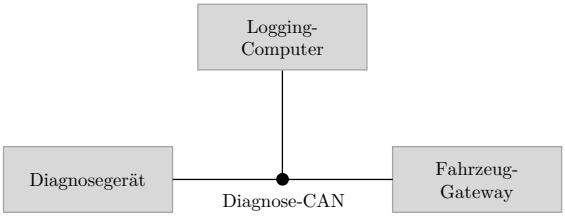


Abbildung 5.11: Schematischer Aufbau zur Aufzeichnung von realen Diagnosenachrichten zwischen Diagnosegerät und Fahrzeug über einen Logging-PC mittels OBD-Bypass.

Datenaufbereitung

Da in dieser Arbeit das Ziel auf der Anomalieerkennung von eingehenden Diagnosesequenzen liegt, muss eine Datenvorverarbeitung erfolgen, bevor diese als Trainings- bzw. Testdaten verwendbar sind. Dazu werden gemäß den definierten Anforderungen für die Weiterverarbeitung nur Diagnosebotschaften verwendet, die das Diagnosegerät zum Fahrzeug versendet. Darüber hinaus werden aus den Rohdaten keine Zeitstempel verwendet, da auf Grundlage dieser Informationen eine Extraktion des Verhaltens als nur sehr eingeschränkt möglich erachtet wird. Diese Annahme lässt sich damit begründen, dass Diagnoseabläufe zeitlich nicht konstanten Zeitintervallen unterliegen, die vom jeweiligen Benutzer und dessen Eingabe abhängen. Diese kann wiederum bei jeder Diagnosesitzung zeitlich variieren. An dieser Stelle wird darauf verwiesen, dass innerhalb der On-Board Kommunikation sehr wohl Ansätze existieren, die das zeitliche Verhalten (Nachrichtenfrequenz) als Feature für die Anomalieerkennung verwenden [103]. Als Voraussetzung muss hierbei ein

statisch spezifiziertes Kommunikationsverhalten existieren (K-Matrix, s. Abschnitt 2.1.2).

Modellbildung

Die Modellbildung der Sprachmodelle erfolgt mittels der aufbereiteten Trainingsdaten. Dafür werden die Sprachmodelle unter Verwendung unterschiedlicher N-Gramme (z.B. Bi- oder Trigramme) trainiert (s. Abbildung 5.12). Als Datengrundlage dient für alle Modelle der Korpus (5000 Nachrichten) aus zuvor aufgezeichneten Diagnoseabläufen (Trainingsdatensatz).

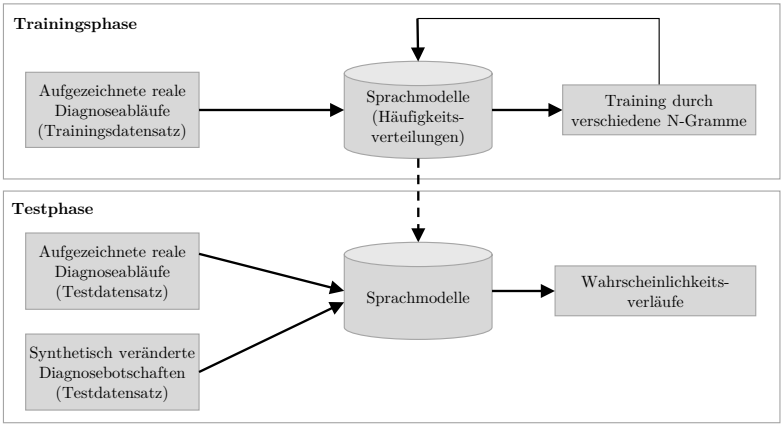


Abbildung 5.12: Übersicht des Modellbildung (Trainingsphase) sowie nachfolgender Testphase mit Normal- und Anomaliedaten zur Berechnung der Wahrscheinlichkeitsverläufe.

Die Modelle enthalten nach Abschluss jeweils eine Häufigkeitsverteilung (Korpus) der analysierten Diagnosedaten basierend auf dem jeweiligen N-Gramm Typ und bilden danach die Grundlage für die Berechnung der Wahrscheinlichkeitsverläufe während der Testphase (s. Abschnitte 5.3.2 u. 5.3.3). Dafür werden zum einen real aufgezeichnete Diagnoseabläufe (Testdatensatz) verwendet. Zum anderen werden auf diesem Datensatz basierend, synthetisch veränderte Diagnoseabläufe erzeugt, um verschiedene Anomalietypen (s. Abschnitt 5.1.3) abzubilden.

Parameterglättung

Um die Qualität der trainierten Häufigkeitsverteilungen (s. vorheriger Abschnitt) zu verbessern wird die *Add-k* Glättungsmethode (s. Abschnitt 2.3.2) von Lidstone angewendet, die sich in der Praxis als anwendbar zeigte [188].

Erzeugung von Anomalien

Da für die Untersuchung des entwickelten Erkennungsansatzes keine realen Anomaliedaten vorliegen müssen diese synthetisch aus den Testdaten generiert werden. Dafür wird angenommen, dass ein Angreifer mit einem legitimierten Zugang über einen authentifizierten externen Zugangspunkt (z.B. Diagnosegerät oder OEM-Backend), eigene Diagnosebotschaften in das Fahrzeugnetzwerk einschleust und dadurch eine Abweichung vom Normalverhalten erzeugt. Durch diesen Angriffsweg verletzt dieser keine spezifizierten Sicherheitseigenschaften (z.B. Vertraulichkeit oder Authentizität), die potentiell auf dem Verbindungskanal durch verschiedene Sicherheitsmechanismen (z.B. Security-Protokolle, Zertifikate) sichergestellt werden. Die generierten Anomalien umfassen dabei drei verschiedene Varianten (s. Abschnitt 5.3.2).

5.3.2 Sequenz-basierter Erkennungsansatz

Um die Fähigkeit der Anomalieerkennung mittels der adaptierten N-Gramm Methode auf die Diagnosekommunikation zu untersuchen, werden nachfolgend verschiedene Parametervariationen auf Basis der trainierten Sprachmodelle durch drei unterschiedliche Test-Szenarien analysiert (s. Abbildung 5.13), die konkrete Beispiele der spezifizierten Diagnose-Anomalien repräsentieren (s. Abschnitt 5.1.3). Dabei wird neben der Verwendung unterschiedlicher N-Gramme auch untersucht, welchen Einfluss die Variation der Größe des Sliding Windows auf die Erkennung hat.

Test-Szenario 1

Dieses Szenario repräsentiert den ersten Anomalietyp (s. Abschnitt 5.1.3), indem ein Angreifer versucht eine im Normalverhalten enthaltene Sequenz bzw.

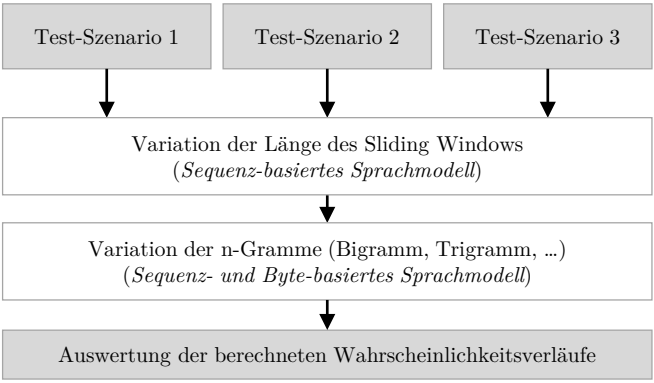


Abbildung 5.13: Ablauf der Untersuchung von Sequenz- und Byte-basierten Sprachmodellen durch Variation der Länge des Sliding Windows sowie Verwendung unterschiedlicher N-Gramme.

mehrere Sequenzen durch veränderte zu ersetzen. Die Originaldaten (Normalverhalten) entsprechen dabei real aufgezeichneten Daten (s. Tabelle 5.2). Zusätzlich werden zwei Sequenzabfolgen mit enthaltenen Anomalien erstellt (Anomalie a und b). Dabei ist in der ersten Anomalieabfolge die 9. und in der zweiten die 7., 8. und 9. Sequenz verändert.

Die nachfolgend dargestellten Diagramme (s. Abbildung 5.14) zeigen die Verläufe der berechneten Wahrscheinlichkeiten auf der Datengrundlage des Testszenarios 1 unter Verwendung eines Bigramms. Dabei wird die Sliding Window Größe m (s. Abschnitt 5.2.3) zwischen zwei und fünf variiert. Der Smoothing Parameter k beträgt dabei 0.001.

Die Berechnung der Wahrscheinlichkeiten erfolgt dabei zu unterschiedlichen Zeitpunkten bzw. nach dem Empfang verschiedener Diagnosenachrichten. Ist beispielsweise die Sliding Window Größe $m = 2$ (Satzlänge S), startet die Berechnung der ersten Wahrscheinlichkeit P nach dem Eintreffen der zweiten Sequenz wie folgt:

$$P(S_{1,2}) = P(w_1, w_2) \tag{5.3}$$

Tabelle 5.2: Auszug von verschiedenen Sequenzabfolgen bestehend aus realen Diagnosebotschaften (Normaldaten o) sowie synthetisch eingefügten Anomalien. (In Spalte Anomaliedaten (a) ist die 9. Sequenz verändert. In Spalte Anomaliedaten (b) sind die Sequenzen 7 - 9 verändert.)

Nr.	Normaldaten (o)	Anomaliedaten (a)	Anomaliedaten (b)
1	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00
2	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00
...	...	...	...
6	60 03 19 02 0C 00 00 00	60 03 19 02 0C 00 00 00	60 03 19 02 0C 00 00 00
7	60 30 00 00 00 00 00 00	60 30 00 00 00 00 00 00	**60 30 10 00 00 00 00 00**
8	60 02 10 01 00 00 00 00	60 02 10 01 00 00 00 00	**60 02 10 05 00 00 00 00**
9	60 02 10 03 00 00 00 00	**60 02 40 03 00 00 00 00**	**60 02 40 03 00 00 00 00**
10	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00
...	...	...	...

Ab der 9. Sequenz (s. Abbildung 5.14) sinkt die Wahrscheinlichkeit von Anomalie *a* sprunghaft ab, sodass im Vergleich zu den Normaldaten eine große Differenz an der 9. und 10. Sequenznummer entsteht:

$$P(S_o) = P(o_8, o_9) \leftrightarrow P(S_a) = P(o_8, a_9) \tag{5.4}$$

$$P(S_o) = P(o_9, o_{10}) \leftrightarrow P(S_a) = P(a_9, o_{10}) \tag{5.5}$$

Dabei entsteht die größte Differenz an der 9. Sequenz, da sich an dieser Stelle die Anomalie befindet. Ab der 10. Sequenz erhöht sich die Wahrscheinlichkeit, da diese mit den Normaldaten übereinstimmt. Mit Berechnung der 11. Sequenz weisen die Wahrscheinlichkeiten der Normal- bzw. Anomaliedaten *a* wieder denselben Wert auf. Dadurch ist die Anomalie im Vergleich zu den Normaldaten mit der geringsten Wahrscheinlichkeit eindeutig identifizierbar.

Bei der Analyse des Wahrscheinlichkeitsverlaufs auf Basis der Anomaliedaten *b* fällt auf, dass die Wahrscheinlichkeit ab der eigentlichen Anomalie (7. Sequenz) deutlich geringer wird und bei den darauffolgenden Sequenzen wieder etwas ansteigt. Dieses Verhalten lässt sich auf die zugrundeliegenden Trainingsdaten zurückführen, die bei dieser Kombinatorik eine höhere Auftritts-

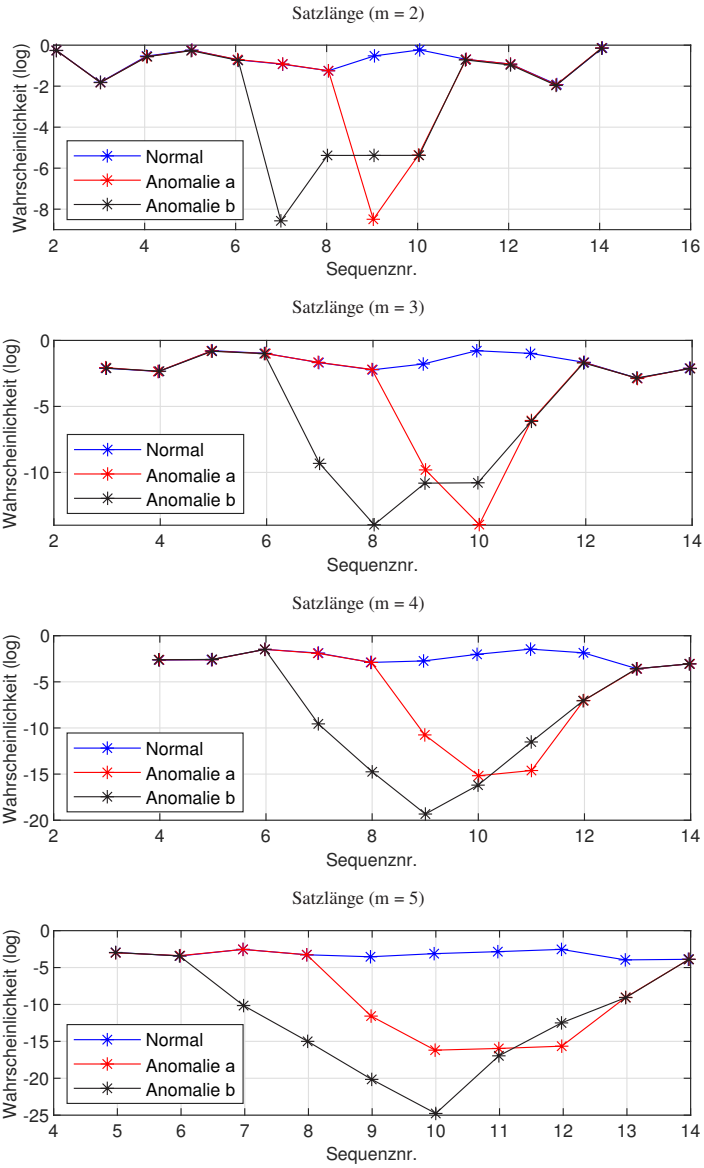


Abbildung 5.14: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der Sliding Window Größe m für das 1. Testszenario.

wahrscheinlichkeit der Sequenzen (8,9,10) im Sprachmodell erzeugen. Jedoch liegt P zwischen der 8. und 10. Sequenz immer noch deutlich unter dem Wert der Normaldaten.

Wird die Satzlänge nun auf $m = 3$ gesetzt, ergibt sich eine erste Differenz zwischen dem Anomaliedatenset a und den Normaldaten an der 10. Sequenz. Diese Verschiebung lässt sich über die zugrundeliegende Berechnung erklären:

$$P(S_8) = P(w_6, w_7, w_8) \tag{5.6}$$
$$P(S_9) = P(w_7, w_8, w_9) \tag{5.7}$$
$$P(S_{10}) = P(w_8, w_9, w_{10}) \tag{5.8}$$
$$P(S_{11}) = P(w_9, w_{10}, w_{11}) \tag{5.9}$$
$$P(S_{12}) = P(w_{10}, w_{11}, w_{12}) \tag{5.10}$$

Dabei lässt sich ableiten, dass die Anomalie in der 9. Sequenz enthalten ist (s. Abbildung 5.14). Wird die Satzlänge m weiter erhöht steigt die Anzahl der Sequenzen die einen abweichenden Wahrscheinlichkeitswert zu den Normaldaten aufweisen und eine Identifikation der genauen Anomaliesequenz wird schwieriger.

Nach der Variation der Satzlänge wird außerdem untersucht, welchen Einfluss auf die Erkennung von Anomalien unterschiedliche N-Gramme haben. Dafür werden dieselben Daten (s. Tabelle 5.2) sowie der gleiche Smoothing Parameter ($k = 0.001$) verwendet. Die Satzlänge wird dabei konstant auf $m = 5$ gesetzt. Unter Verwendung von vier unterschiedlichen N-Grammen (s. Abbildung 5.15) zeigt sich, dass die Anzahl der Ausreißer (Anomaliepunkte) bei jedem N-Gramm Typ gleich ist. Hingegen variieren im Vergleich die berechneten Wahrscheinlichkeiten relativ betrachtet.

Test-Szenario 2

In diesem Szenario wird angenommen, dass ein Angreifer durch das Einfügen einer zusätzlichen Nachricht eine Abweichung zum Normalverhalten erzeugt (Anomalietyp 2, s. Abschnitt 5.1.3). Dabei werden neben normalen Testdaten zwei Sequenzabfolgen mit enthaltenen Anomalien verwendet (s. Tabelle 5.3).

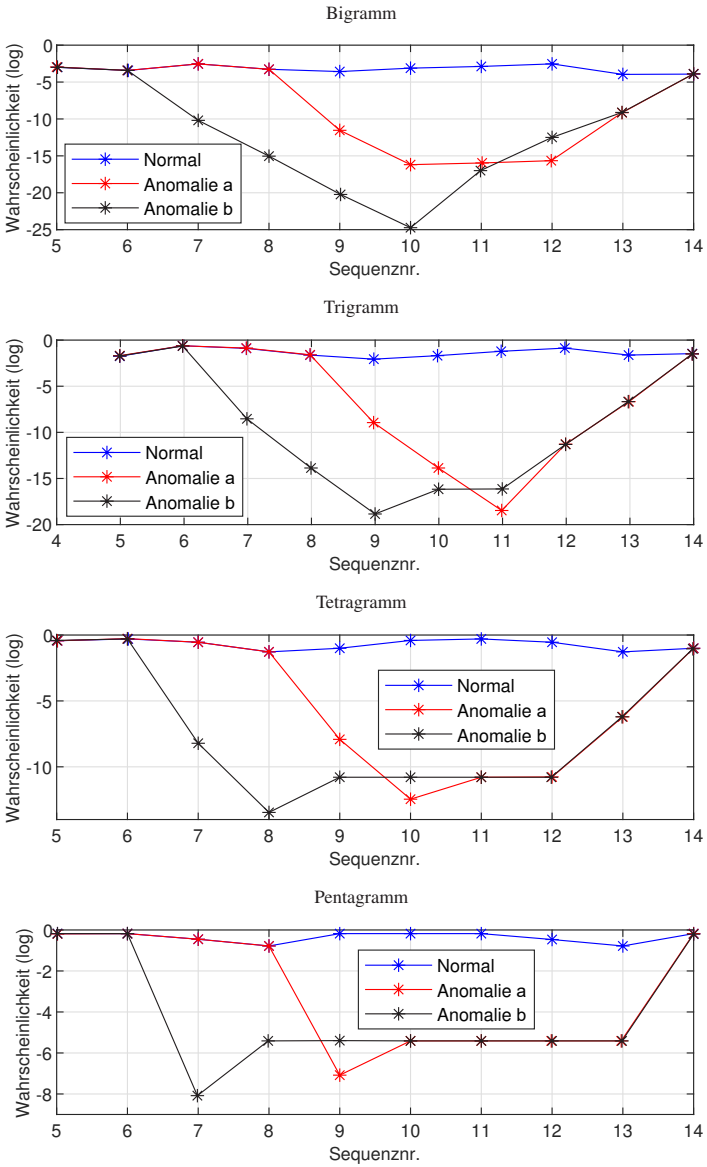


Abbildung 5.15: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der N-Gramme für das 1. Testszenario.

In den Anomaliedaten (a) besteht die eingefügte 10. Sequenz aus einer Diagnosebotschaft, die in den Normaldaten enthalten ist, jedoch nicht zu dieser Sequenzabfolge passt. Hingegen ist die 10. Sequenz in den Anomaliedaten (b) eine vom Normalverhalten vollständig abweichende Nachricht.

Tabelle 5.3: Auszug einer Sequenz bestehend aus realen Diagnosebotschaften (Normaldaten o) sowie synthetisch eingefügten Anomalien. (In den Spalten Anomaliedaten (a) & (b) ist eine 10. Sequenz eingefügt, die im Vergleich zum Normalverhalten nicht enthalten ist.)

Nr.	Normaldaten (o)	Anomaliedaten (a)	Anomaliedaten (b)
1	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00
...	...	...	...
9	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00
10	None in Original	**60 02 10 01 00 00 00 00**	**60 02 10 01 00 20 40 12**
11	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00
...	...	...	...

Auch in diesem Szenario wird nochmals eine Parametervariation des Sliding Windows ($m = 2 - 5$) basierend auf einem Bigramm vorgenommen. Darüber hinaus werden mit einer Satzlänge von $m = 5$ verschiedene N-Gramme verwendet. Dabei ergeben sich basierend auf den Anomaliedaten (a und b) dieselben Wahrscheinlichkeitsverläufe (s. Abbildungen A.12 u. A.13), da beide Varianten eine Abweichung vom Normalverhalten verursachen. Das Sequenzbasierte Modell kann hier keine weitere Unterscheidung bzgl. des genauen Nachrichteninhalts treffen.

Test-Szenario 3

Das dritte Szenario der Untersuchung bildet den Fall ab, dass ein Angreifer durch eine Veränderung der Sequenzreihenfolge vom Normalverhalten abweicht (Anomalietyp 3, s. Abschnitt 5.1.3). Dafür werden real aufgezeichnete Diagnosesequenzen als Testdaten verwendet (s. Tabelle 5.4). Die durch einen Angreifer ausgelöste Anomalie wird durch eine veränderte Sequenzabfolge abgebildet, die eine Vertauschung der 8. und 9. Sequenz beinhaltet.

Tabelle 5.4: Auszug einer Sequenz bestehend aus realen Diagnosebotschaften (Normaldaten o) sowie synthetisch eingefügten Anomalien. (In Spalte Anomaliedaten (a) sind die 8. und 9. Sequenz im Vergleich zu den Normaldaten vertauscht.)

Nr.	Normaldaten (o)	Anomaliedaten (a)
1	60 02 10 03 00 00 00 00	60 02 10 03 00 00 00 00
...	...	...
8	60 02 10 01 00 00 00 00	**60 02 10 03 00 00 00 00**
9	60 02 10 03 00 00 00 00	**60 02 10 01 00 00 00 00**
10	60 03 22 F1 50 00 00 00	60 03 22 F1 50 00 00 00
...	...	...

Das Szenario weist eine Ähnlichkeit zum 1. Test-Szenario auf, da eine Vertauschung auch einer Ersetzung von Nachrichten entsprechen kann. Jedoch wird hier nur die Reihenfolge der Nachrichten geändert und nicht der eigentliche Nachrichteninhalt, der weiter mit dem Normalverhalten übereinstimmt. Die Wahrscheinlichkeitsverläufe werden erneut mit verschiedenen Satzlängen m sowie vier unterschiedlichen N-Grammen berechnet (s. Abbildungen A.14 u. A.15). Auf der Basis des Sequenz-basierten Ansatzes ist eine Erkennung der 8. und 9. Sequenz als Anomalie möglich. Hingegen ist keine Bewertung bzgl. des Nachrichteninhalts auf Basis der Nutzdatenbytes möglich. Das bedeutet der Ansatz kann nicht unterscheiden, ob die als Anomalie detektierten Nachrichten gutartige oder bösartige Nachrichteninhalte auf Basis des Normalverhaltens sind.

Diskussion - Sequenz-basierter Ansatz

Die Wahl des N-Gramm Typs zeigt verschiedene Auswirkungen auf die zugehörigen bedingten Wahrscheinlichkeiten der Testdaten. Nach Abschnitt 5.2.3 gilt in der CL, dass eine N-Gramm Länge $n > 3$ im Vergleich zu einem Bi- bzw. Trigramm keine zuverlässigeren Auftretenswahrscheinlichkeiten liefern. Theoretisch wäre bei größeren N-Grammen der Informationsgehalt höher. Jedoch zeigt sich in der Sprachverarbeitung, dass reale Korpora (s. Abschnitt 2.3.2) mögliche Wortkombinationen nie vollständig abbilden und dadurch Kombinationen mit einer Wahrscheinlichkeit von 0 auftreten können. Mit Blick auf die untersuchten Test-Szenarien dieser Arbeit zeigt sich, dass bei der Verwendung

von Tetra- bzw. Pentagrammen teilweise über mehrere Sequenzen hinweg die gleiche Wahrscheinlichkeit berechnet wird (s. Abbildung 5.15) und dadurch auf fehlende Kombinationen in den Trainingsdaten zurückzuführen ist. Darüber hinaus gilt nach Wu [189], dass die Effizienz der Modelle durch eine Erhöhung von $n > 3$ nicht mit dem Ressourcenverbrauch skalieren.

5.3.3 Byte-basierter Erkennungsansatz

Das zweite Erkennungsmodell nutzt die N-Gramm Methode auf Basis einzelner Diagnosenachrichten (PDU-Ebene). Wie beim Sequenz-basierten Modell werden konkrete Testdaten (Ausschnitt, s. Tabelle 5.5) verwendet, um den Einfluss unterschiedlicher N-Gramme auf die Berechnung der Wahrscheinlichkeiten zu untersuchen. Die enthaltenen Anomalien repräsentieren dabei den vierten Anomalietyp (s. Abschnitt 5.1.3). Die Originalnachricht bildet in diesem Untersuchungsszenario eine Diagnose-Botschaft zum Reset eines Steuergerätes (s. Abschnitt A.3). Die Sequenzen Nr. 2 und 3 sind dagegen auf ein Angriffsszenario transferierbar, in dem ein Angreifer versucht nach dem *Trial&Error*-Prinzip bestimmte Funktionen auszuführen. Die veränderten Sub-Function Bytes (2.1.3) liegen in einem OEM-spezifischen Bereich (0x40 - 0x5F) und damit außerhalb von standardisierten Funktionen des UDS-Protokolls (s. Abschnitt A.1.1). Es wird in diesem Szenario davon ausgegangen, dass diese Sub-Function nicht definiert ist und sich dadurch eine Abweichung vom Normalverhalten ergibt.

Tabelle 5.5: Darstellung einer Diagnosenachricht zum Reset einer spezifischen ECU sowie zwei veränderten Anomalienachrichten (Nr. 2 und 3).

Nr.	Diagnosenachricht	Beschreibung
1	60 02 11 01 00 00 00 00	Original CAN Message
2	60 02 11 **40** 00 00 00 00	Änderung des Sub-Function Bytes
3	60 02 11 **5F** 00 00 00 00	Änderung des Sub-Function Bytes

Jede Diagnosenachricht entspricht dabei einem vollständigen Satz. Aus diesem Grund erfolgt keine Variation der Satzlänge m, da nach UDS-Standard A.1.1 eine Länge von acht Datenbytes definiert werden sollte (s. auch Tabelle 5.5).

Bei davon abweichenden Längen muss die Satzlänge entsprechend angepasst werden.

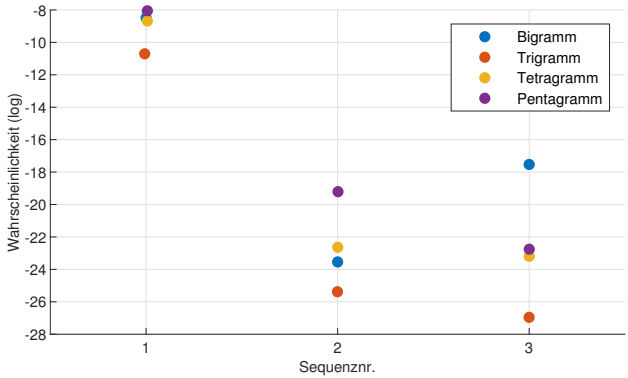


Abbildung 5.16: Berechnete Wahrscheinlichkeiten auf Basis des Byte-basierten Modells für unterschiedliche N-Gramme. (Die 1. Sequenz beinhaltet Normaldaten. Die 2. und 3. Sequenz enthalten Anomalien.)

Die berechneten Wahrscheinlichkeiten (s. Abbildung 5.16) auf Basis der Testdaten (s. Tabelle 5.5) weisen eine deutliche Differenz zwischen Original- und Anomaliedaten auf. Daneben ist zu erkennen, dass der verwendete N-Gramm Typ einen geringen Einfluss auf die berechneten Wahrscheinlichkeiten hat. Alle N-Gramme berechnen eine deutlich geringere Wahrscheinlichkeit der Anomaliesequenzen (Nr. 2 und 3) im Vergleich zur Originalnachricht (Nr. 1).

5.3.4 Hybrides Framework zur Erkennung von Anomalien

Auf Basis der zuvor durchgeführten Untersuchung wird ein hybrides Framework (s. Abbildung 5.17) entwickelt, um beide Erkennungsfeatures (Sequenz- und Byte-basiert) zu vereinen. Das Framework besteht aus vier verschiedenen Modulen, die zum einen für die Trainingsphase sowie zum anderen für die Analyse der Kommunikationsdaten während des Betriebs notwendig sind. Für beide Phasen dient eingangsseitig das Aufzeichnungsmodul für die Speicherung der zu analysierenden Kommunikationsdaten.

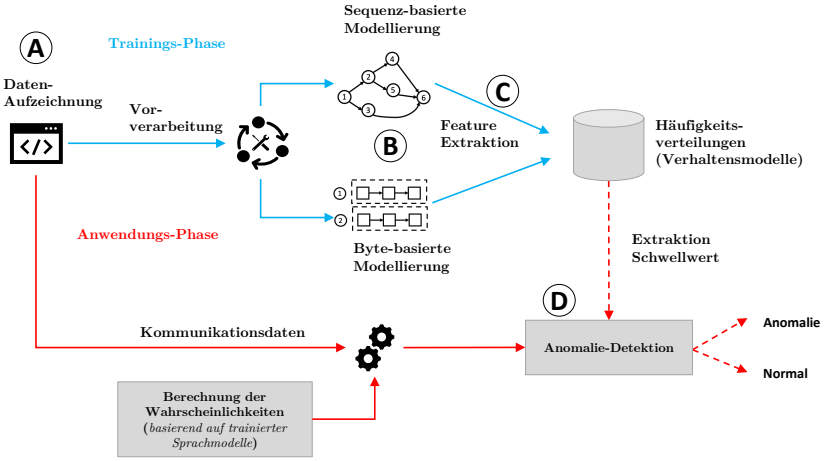


Abbildung 5.17: Schematischer Aufbau des Detektion-Frameworks für Anomalien in Diagnosekommunikationsdaten.

A) Aufzeichnung & Vorverarbeitung von Diagnosedaten

Das erste Modul (Datenaufzeichnung) erfasst die Kommunikationsdaten eingangsseitig und speichert diese ab. Für das Training sowie den Test eines Modells mittels überwachten Lernens (s. Abschnitt 2.3.1) werden voneinander unabhängige Datensätze benötigt. Nach Abschluss der Trainingsphase dient dieses Modul in der Betriebsphase als Zwischenspeicher.

B) Modellierung (Sequenz- und Byte-basiert)

In diesem Schritt erfolgt die Erstellung sowie das Training der Sequenz- und Byte-basierten Sprachmodelle, um anschließend entsprechende Verhaltensmuster (Features) der Diagnose-Kommunikationsdaten zu extrahieren.

C) Feature Extraktion

In diesem Schritt wird die Extraktion von Verhaltensmustern auf Basis der Sequenz- bzw. Byte-basierten Sprachmodelle unter Verwendung der N-Gramm Technik ausgeführt. Die Extraktion der Features erfolgt durch ein Sliding Window mit der Länge $m = 5$ (Sequenz-basiertes Modell) sowie der Verwendung von Trigrammen, da diese nach dem aktuellen Stand der Wissenschaft (s. Abschnitt 5.2.3) bei kleinen Trainingsdatensätzen im Vergleich zu $n > 3$ bessere Ergebnisse zeigen. Für das Byte-basierte Sprachmodell wird ebenfalls ein Trigramm verwendet. Daraus ergibt sich jeweils für beide Features eine Häufigkeitsverteilung, die das Normalverhalten der analysierten Diagnosekommunikation repräsentieren. Weiter bilden diese Verteilungen die Grundlage zur Berechnung der Auftretenswahrscheinlichkeiten im Anwendungsbetrieb.

D) Anomalie-Detektion

Für die automatisierte Anomalie-Detektion von eingehenden Diagnosesequenzen im Anwendungsbetrieb wird ein passende Erkennungsmethode benötigt, die eingehende Diagnosedaten im Realbetrieb in normal bzw. anormal klassifiziert. Nach dem Trainieren der N-Gramm Sprachmodelle muss eine Annäherung einer Verteilungsfunktion erfolgen (s. auch Abschnitt 5.2.3), um die Schwellenwerte ϵ der zwei Modelle zu bestimmen. Die dafür benötigten Trainingsdaten müssen auf einer spezifischen Fahrzeugbaureihe basieren (s. auch Abschnitt 5.2.3). Der prinzipielle Ablauf ist in Abbildung 5.18 dargestellt.

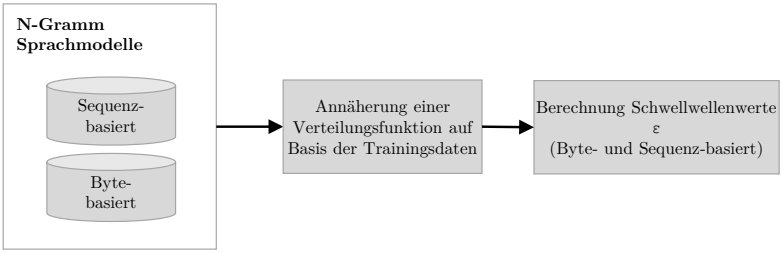


Abbildung 5.18: Ablauf zur Bestimmung der Anomalie-Schwellenwerte ϵ für den Byte- bzw. Sequenz-basierten Erkennungsansatz.

5.4 Prototypische Umsetzung

Im Rahmen der prototypischen Umsetzung wird das hybride Framework untersucht, ob eine Erkennung von Anomalien innerhalb der Diagnosekommunikation unter Verwendung der trainierten Sprachmodelle über einen definierten Schwellwert ϵ im Rahmen der Anwendungs-Phase (s. Abbildung 5.17) prinzipiell möglich ist. Die Kombination der zwei Modelle bietet den Vorteil die erkannte Anomalie zu klassifizieren (s. Abschnitt 5.1.3).

5.4.1 Aufbau & Durchführung

Für die Untersuchung werden verschiedene Anomaliedaten aus den Testszenarien 1 - 3 (s. Abschnitt 5.3.2) verwendet und ausgewertet. Für die Berechnung der Schwellenwerte beider Sprachmodelle wird eine Annäherung durch eine Normalverteilung durchgeführt (s. auch Abschnitt 5.2.3). Die unbekannten Parameter der Normalverteilung werden (s. auch Abschnitt 5.2.4) dabei geschätzt. Dafür wird die MLE-Methode [190] für den Erwartungswert μ bzw. die Varianz σ^2 eingesetzt:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} a_i \tag{5.11}$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (a_i - \mu)^2 \tag{5.12}$$

Die ermittelten Parameter sind in Tabelle 5.6 aufgeführt. Die enthaltenen Schwellenwerte entsprechen jeweils einem 100 %-Quantil, da die verwendeten Trainingsdaten ausschließlich ein Normalverhalten ohne Anomalien beinhalten.

Tabelle 5.6: Übersicht der berechneten Parameter für die Stichprobe des Sequenz- bzw. Byte-basierten Modells.

Parameter	Sequenz-basiert	Byte-basiert
Erwartungswert μ	-1.071	-1,793
Varianz σ	1,214	1,411
Schwellenwert ϵ	-6,06	-8.75

5.4.2 Berechnete Wahrscheinlichkeitsverläufe

Unter Verwendung der Sprachmodelle werden auf Basis der Anomaliedaten verschiedene Wahrscheinlichkeitsverläufe berechnet (s. Abbildung 5.19). Durch die Anwendung der Test-Szenarien 1 und 2 mittels des Sequenz-basierten Ansatzes wird eine Anomalie an der 10. Stelle erkannt. Dieses Ergebnis enthält jedoch keine weiteren Informationen. Es sind an dieser Stelle zwei verschiedene Varianten möglich. Die Sequenz könnte eine an sich korrekte Nachricht darstellen, die im Normalverhalten existiert, aber nicht im korrekten kontextuellen Zusammenhang steht (vom Normalverhalten abweichende Reihenfolge). Die zweite Variante könnte einen Angreifer darstellen, der eine vom Normalverhalten abweichende Diagnosenachricht an das Fahrzeug sendet.

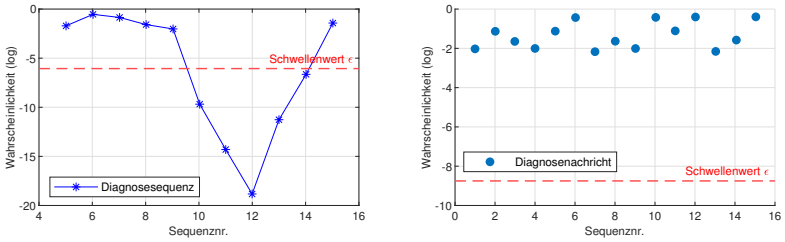
Durch die Analyse der Eingangsdaten mittels des Byte-basierten Sprachmodells lässt sich die Anomalie genauer klassifizieren, sodass hier eine vom Angreifer eingeschleuste Nachricht eindeutig den Erkennungsschwellwert unterschreitet.

Im dritten Testszenario erfolgt eine vom Normalverhalten abweichende Sequenzreihenfolge der gesendeten Diagnosenachrichten. Das Sequenz-basierte Sprachmodell kann die Anomalie korrekt erkennen. Das Byte-basierte Modell erkennt hingegen keine Auffälligkeiten, da auf Byte-Ebene keine Manipulation stattfindet. Die Einordnung ist daher ebenfalls korrekt.

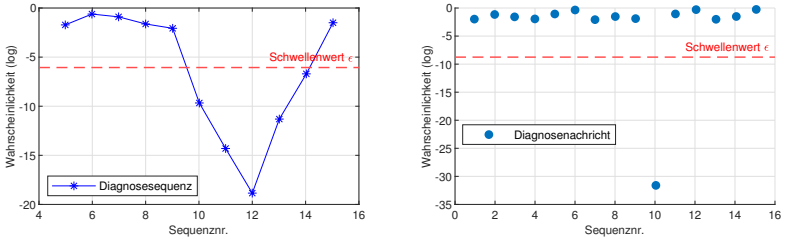
5.4.3 Auffälligkeiten

Bei der Berechnung der Wahrscheinlichkeiten auf Basis der aufgezeichneten Testdaten fällt auf, dass speziell beim Auslesen der Fehlerspeicher von

Testszenario 1: Einfügen einer Nachricht aus Normalverhalten (falsche Reihenfolge).



Testszenario 2: Einfügen einer Angreifer-Nachricht.



Testszenario 3: Vertauschen der Sequenzreihenfolge (Nachrichten im Normalverhalten enthalten).

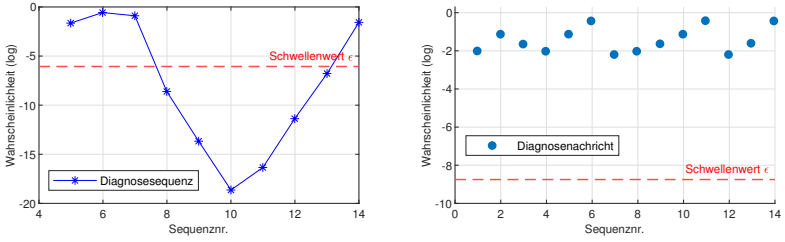


Abbildung 5.19: Berechnete Wahrscheinlichkeiten durch das hybride N-Gramm Framework auf Basis unterschiedlicher Testdaten (links: Sequenz-basiert, rechts: Byte-basiert).

Steuergeräten einige Diagnosenachrichten mit dem Sequenz-basierten Modell als Anomalie klassifiziert werden. Hingegen detektiert der Byte-basierte Ansatz keine Abweichung zum Normalverhalten. Ein exemplarischer Wahrscheinlichkeitsverlauf für ein derartiges Verhalten auf Basis einer Motor-ECU (s. Abschnitt A.4.1) ist in Abbildung 5.20 dargestellt.

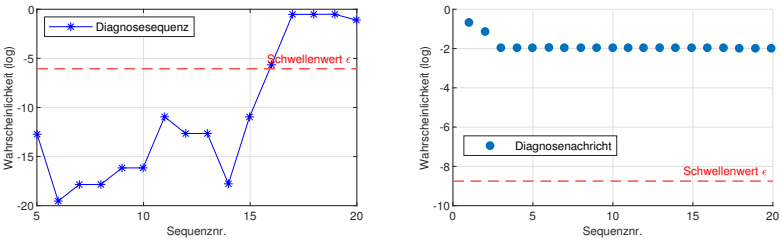


Abbildung 5.20: Auswertung von aufgezeichneten Motor-Diagnosebotschaften mit enthaltenen FC-Frames (links: Sequenz-basiert, rechts: Byte-basiert).

Der Grund für dieses Verhalten lässt sich über das verwendete CAN-Transportprotokoll ISO-TP (s. Abschnitt A.1.1) erklären, wodurch die Empfänger zum Sender zur Steuerung der Übertragungsgrößen der Datenpakete bestimmte FC-Frames (s. Abschnitt A.1.1) versenden. In verwendeten Test-Datenset sendet das Diagnosegerät zum Auslesen des Fehlerspeichers eine Anfrage (SID = 0x22 und SID = 0x19, s. Abschnitt A.3). Daraufhin sendet jede ECU dem Sender ein First Frame (FF)-Frame (s. Abschnitt A.1.1), da die Daten aufgrund der Größe segmentiert werden müssen. Danach muss der Empfänger (Diagnosegerät) jeder ECU mit einem FC-Frame antworten, um zu übermitteln, wie viele aufeinanderfolgende Consecutive Frame (CF)-Frames (s. Abschnitt A.1.1) eingehen dürfen (abhängig von der Größe des Empfangspuffers). Die Reihenfolge der gesendeten FC-Frames ist dabei zufällig, sodass hier kein Kontext bezogenes Training der Sprachmodelle erfolgen kann.

5.5 Zusammenfassung

Die N-Gramm Methode ist ein etabliertes Verfahren in der Computerlinguistik und wird darüber hinaus auch im Bereich von Intrusion-Detection-Systemen eingesetzt. Der in dieser Arbeit entwickelte IDS-Ansatz adaptiert die N-Gramm

Methode auf die automotive Domäne zur Erkennung von Anomalien in der Diagnosekommunikation. Dafür wird die Eigenschaft genutzt, dass die übertragenen Diagnosenachrichten in einem kontextuellen Zusammenhang stehen und darüber ein Verhalten ableitbar ist. Dieses wird durch ein hybrides Sprachmodell (Sequenz- und Byte-basiert) abgebildet. Für die prototypische Umsetzung des Ansatzes wurden real aufgezeichnete Diagnosesequenzen mit synthetischen Anomalien angereichert und als Testdaten verwendet. Es wurde gezeigt, dass das System in der Lage ist, die vier verschiedenen Anomalietypen im Rahmen der verwendeten Testdaten zu erkennen. In Bezug auf die eingangs beschriebene Problemstellung von Insider-Angreifern im Bereich der Diagnoseanwendungen zeigt der entwickelte Ansatz die prinzipielle Möglichkeit zur Erkennung dieses Angriffstyps, indem Abweichungen vom Normalverhalten detektierbar sind. Zur Steigerung der Informationssicherheit könnte der IDS-Ansatz als zusätzliche Maßnahme in das zentrale Gateway des Fahrzeugs integriert werden, um eine zusätzliche Absicherungsschicht zu bilden, falls eine präventive Maßnahme (z.B. das Rechtemanagement) durch Erlangung von legitimierten Zugangsdaten von einem Angreifer überwunden wurde. Zudem sollte das integrierte IDS-System durch eine zusätzliche Schutzmaßnahme (z.B. Firewall) abgesichert werden, um die Sicherheitseigenschaft *Verfügbarkeit* bei einem möglichen DoS-Angriff sicherzustellen. Mittels einer Filterregel könnte in der Firewall eine maximal erlaubte Datenrate (Anzahl an Diagnosebotschaften in einem bestimmten Zeitintervall) definiert werden, um eine Überlastung dieser Kontrollkomponente zu verhindern.

6 Zusammenfassung & Ausblick

Die aktuellen Trends wie das automatisierte Fahren oder die Elektromobilität implizieren gleichzeitig eine Steigerung der Konnektivität von Fahrzeugen mit Blick auf die Anzahl der Sensoren, Netzwerktechnologien sowie die Kommunikation mit der Außenwelt (Backends, andere Fahrzeuge oder Infrastruktur). Damit steigt gleichzeitig das Risiko von Cyber-Angriffen auf die Informationssicherheit der gesamten automotive IT. Mit Blick auf das einzelne Fahrzeug können derartige Angriffe auch die funktionale Sicherheit gefährden. Dadurch ergibt sich die Motivation, neue Fahrzeugarchitekturen durch geeignete Schutzmaßnahmen abzusichern, um das Risiko für erfolgreiche Angriffe zu minimieren. Dazu sind sowohl präventive als auch pro-aktive Maßnahmen in E/E-Architekturen notwendig.

6.1 Beiträge der Arbeit

Im Rahmen dieser Arbeit wurden zunächst bisher bekannte Angriffe bzw. Schwachstellen in den Jahren 2010 - 2019 analysiert. Diese Analyse bildete die Grundlage zur Identifikation von Schwächen der informationstechnischen Absicherung, die zur jeweiligen Verletzung der Security-Eigenschaften und letztlich zu erfolgreichen Angriffen führte. Dabei stellte die Verletzung der Authentizität bzw. die damit implizierte Autorisierung mit 37 % den höchsten Anteil der untersuchten Angriffe dar. Für eine detaillierte Erläuterung dieser Security-Eigenschaft wurden exemplarisch drei Angriffe näher erläutert. Anknüpfend dazu erfolgte eine Untersuchung und Einordnung zum Stand der Technik und Wissenschaft auch im Hinblick auf aktuelle Guidelines, Regularien und Standards. Auf Basis dieser Ergebnisse wurden offene Forschungslücken im Bereich der Zugriffskontrolle auf Netzwerk- und Anwendungsebene sowie im Bereich IDS mit Fokus auf die Diagnosekommunikation identifiziert. So adressieren im Bereich der Wissenschaft nur wenige Ansätze die Thematik

zur Integration eines Rechtemanagements. Zudem war in bisherigen automotive Architekturen nur ein eingeschränktes und unsicheres Rechtemanagement (Security-Access) ausschließlich für Diagnoseanwendungen vorhanden. Die in dieser Arbeit entwickelte attributbasierte Zugriffskontrolle ermöglicht eine feingranulare Kontrolle und Durchsetzung von definierten Sicherheitsrichtlinien (Policies), die auf der Beschreibungssprache ALFA aufsetzen. Der Fokus liegt dabei auf der Verknüpfung von Berechtigungen (z.B. Lese- und Schreibzugriff) mit Fahrzeugzuständen. Der entwickelte Ansatz wurde abschließend prototypisch umgesetzt und mittels verschiedener Use-Cases getestet.

Neben dieser präventiven Maßnahme wurde im Bereich der pro-aktiven Maßnahmen eine bisherige Lücke mit Fokus auf die Diagnosekommunikation identifiziert, die u.a. bei der Erkennung von Insider-Angriffen relevant ist. Die Arbeit adressiert diese und erarbeitet einen IDS-Ansatz auf Basis der N-Gramm Methode. Zunächst wurden dafür spezifische Merkmale der Diagnose herausgearbeitet, die dann zum Trainieren der zugehörigen entwickelten Sprachmodelle dienen. Im Anschluss erfolgte die Untersuchung verschiedener Modelle auf Basis exemplarischer Testdaten, die mit synthetischen Anomalien angereichert wurden. Im Rahmen einer prototypischen Umsetzung wurde die prinzipielle Funktionsweise nachgewiesen.

Da zukünftig neben der bisher etablierten signal-basierten Kommunikation das SOA-Paradigma in Fahrzeugarchitekturen Anwendung findet, ergeben sich durch dieses hybride Architekturdesign Auswirkungen auf die Anwendung bisheriger Security-Maßnahmen. Die Arbeit greift dazu die Absicherungsmaßnahmen der Zugriffskontrolle und IDS aus signal-basierten Netzwerken auf und analysiert Potentiale sowie Einschränkungen bzgl. deren Adaptierbarkeit auf SOAs.

6.2 Reaktion auf Sicherheitsvorfälle

Die Informationssicherheit innerhalb der Automobilindustrie wird in Zukunft weiter an Bedeutung gewinnen. Neben individuellen Interessen der Hersteller, ihre Produkte bestmöglich abzusichern nehmen aufkommende Regularien eine zentrale Rolle ein. So werden bis im Jahr 2023 insgesamt 18 Standards Inkrafttreten, die innerhalb der automotive Domäne einen Einfluss auf die Fahrzeugtypgenehmigung haben (die am relevantesten, s. Abbildung 6.1) [191].

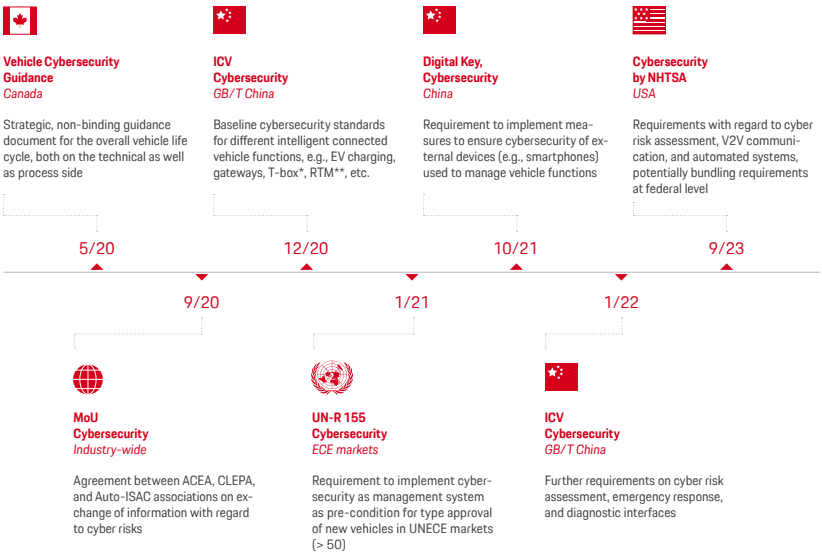


Abbildung 6.1: Wichtige automotive Cybersecurity Regularien and Standards die bis 2023 Inkrafttreten [191].

Für die Einhaltung der Regularien ist wiederum die Anwendung von verfügbaren Standards notwendig. So wird innerhalb der *UN-R 155 Cybersecurity* Verordnung gefordert, dass Fahrzeughersteller ein Cyber Security Management System (CSMS) etablieren müssen. Infolgedessen würde der ISO 21434 Standard (s. Abschnitt 3.2.4) Anwendung finden, der auf Prozessebene entsprechende Aktivitäten spezifiziert [192]. Darin wird u.a. explizit ein Incident-Response-Prozess gefordert, der einen definierten Umgang mit auftretenden Security-Schachstellen umfasst, die beispielsweise durch ein integriertes IDS in Fahrzeugen erkannt werden können [193]. Ein derartiger Prozess kann grundlegend in fünf Schritte unterteilt werden (s. Abbildung 6.2). Zunächst müssen Intrusions im Fahrzeug erkannt und protokolliert werden (1). Der AUTOSAR Standard [194] spezifiziert dafür bereits verschiedene Security-Events, die ein herstellerspezifisches IDS verwenden kann. Nach der Übertragung (2) an ein Security Operations Center (SOC) müssen die Events analysiert (3) werden, um zugehörige Schwachstellen zu identifizieren. Dies kann entweder

auf Basis einzelner Fahrzeuge oder auf Basis von gesammelten Flotten-Daten erfolgen. Danach muss der Hersteller geeignete Gegenmaßnahmen entwickeln und testen (4). Im letzten Schritt erfolgt die Behebung der Schwachstellen durch ein Software-Update, dass je nach Verfügbarkeit über eine Luftschnittstelle (z.B. Mobilfunk) ausgerollt wird.

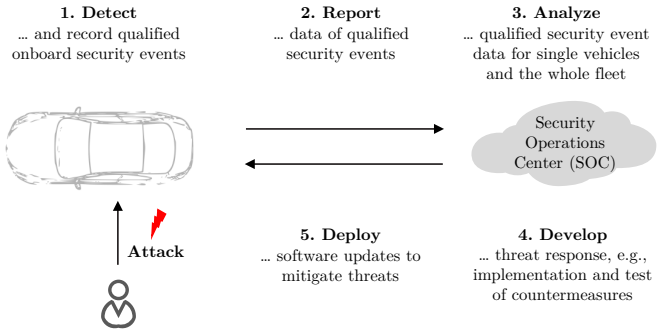


Abbildung 6.2: Schematische Darstellung verschiedener Schritte im Rahmen eines Security-Event-Prozesses (basierend auf [193]).

6.2.1 Security-Events - Analyse, Bewertung, Reaktion

Eine Herausforderung besteht aktuell in der Weiterverarbeitung der Security-Events durch ein *Security Information and Event Management (SIEM)*, die durch Fahrzeuge in ein SOC übermittelt werden. Die Events müssen dazu in einem ersten Schritt detailliert analysiert werden, um beispielsweise Falschalarme (false-positives) herauszufiltern. Aufgrund der hohen Datenmengen sind hierzu vorwiegend Ansätze des maschinellen Lernens (s. Abschnitt 2.3.1) notwendig, um diese automatisiert auszuwerten. Unterstützend kann zusätzlich eine Analyse durch Security-Spezialisten erfolgen. Der dafür erforderliche Zeitraum für derartige Analysen kann stark variieren. Demnach sind Experten zufolge zwischen einigen Sekunden und Minuten auch Stunden bis Wochen denkbar [195]. Letzteres betrifft dann speziell Untersuchungen aus dem Bereich der *IT-Forensik*, die beispielsweise eine Sicherstellung sowie Analyse des betreffenden Fehlerspeichers eines Fahrzeugs umfassen können. Damit ist eine Rekonstruktion der genauen Vorgehensweise des Täters möglich. Auch

hierzu müssen für die automotive Domäne passende Prozesse aufgesetzt und Techniken aus anderen Domänen adaptiert oder neue entwickelt werden. Daneben sind für die detaillierte Analyse der Fahrzeug Security-Events weitere Informationen von Relevanz, die von anderen IDS-Systemen außerhalb des Fahrzeugs erhoben werden. So könnten beispielsweise erkannte Anomalien aus einem Diagnose-IDS (s. Kapitel 5) mit Log-Daten aus einem Backend- oder Mobilfunksystem mittels Korrelationstechniken verglichen werden, um einen möglichen Cyber-Angriff zu verifizieren [195].

Definition 6.2.1 Security Information and Event Management

Das Security Information and Event Management beinhaltet die Erfassung und Verwaltung von Logging-Informationen sowie die Analyse und Berichterstellung mit Fokus auf die Informationssicherheit [196].

Definition 6.2.2 IT-Forensik

Als IT-Forensik wird eine streng methodisch durchgeführte Datenanalyse von Datenträgern sowie in Computernetzwerken bezeichnet [197].

Nach der Analyse müssen die Security Events auf ihre Kritikalität bewertet werden. Handelt es sich evtl. um einen derzeit noch laufenden Angriff? Kann durch die zugehörige Schwachstelle weiterer Schaden angerichtet werden (z.B. skalierender Angriff über eine komplette Baureihe)? Ein erster Ansatz dafür ist in [198] gegeben. Basierend auf der Bewertung muss anschließend eine entsprechende Reaktion (engl. response) erfolgen. Hierbei muss an unterschiedlichen Strategien gearbeitet werden [199]. Welche schnellen Reaktionen sind im Falle eines laufenden Angriffs möglich? Muss der Fahrer unter Umständen mittels visueller, akustischer oder haptischer Warnung gewarnt werden? Sind beispielsweise bestimmte Teilbereiche isolierbar, um die mögliche Angriffsfläche zu begrenzen bzw. den erlangten Zugang des Angreifers zu unterbinden. Eine potentielle Möglichkeit wäre die Sicherheitsrichtlinien einer Zugriffskontrolle (s. Kapitel 4) anzupassen, um spezifizierte Berechtigungen stärker einzuschränken. Dazu müssen Security-Updates schnellstmöglich in die Fahrzeuge übertragen werden (z.B. über die Mobilfunkschnittelle). Jedoch existieren dabei Herausforderungen bzgl. Variantenreichtum sowie spezifizierten Safety-Anforderungen die nicht verletzt werden dürfen [200].

Neben dem Aspekt, dass neue Schwachstellen durch IDS-Systeme erkannt werden, ist zusätzlich ein Prozess notwendig, wie mit von extern gemeldeten Schwachstellen umgegangen wird (z.B. durch Security-Forscher). Auch bei diesen Meldungen muss eine Analyse, Bewertung sowie Reaktion erfolgen. Dabei spielt die Thematik der verantwortungsvollen Offenlegung (engl. responsible disclosure) von Schwachstellen eine zentrale Rolle [201]. Dazu müssen Hersteller und Zulieferer an gemeinsamen Prozessen und Vereinbarungen arbeiten, die den Umgang sowie die Zeit bis zur Offenlegung einer Schwachstelle regeln.

6.2.2 Rückkopplung in den Entwicklungsprozess

Weiterer Forschungsbedarf besteht aktuell auf der Weiterverarbeitung bzw. Rückkopplung der Security-Events in den Entwicklungsprozess, die aus dem aktiven Feldbetrieb durch die SOCs gesammelt werden. Ein erstes wissenschaftliches Forschungsprojekt adressiert seit Juni 2021 diese Thematik im Rahmen des Projekts *UNCOVER* [202]. Die Ziele liegen dabei auf der Entwicklung von Methoden und Werkzeugen zur systematischen Erfassung und Analyse von Security-Events, um damit Wissen über mögliche Schwachstellen sowie Auswirkungen in Fahrzeugarchitekturen zu generieren. Diese Erkenntnisse sollen den Entwicklungsprozess beim Re-Design bestehender Maßnahmen in Form eines Updates oder zur Steigerung der Informationssicherheit bei zukünftigen Entwicklungen unterstützen.

6.3 Informationssicherheit in service-orientierten Fahrzeugarchitekturen

Durch die Weiterentwicklung im Bereich der E/E-Architekturen aufgrund von veränderten Anforderungen und Trends (s. Abschnitt 2.1.2) wird es zunächst eine hybride Variante, bestehend aus signal- und service-orientierter Kommunikation geben. Im Hinblick auf die Gewährleistung der Informationssicherheit durch Zugriffskontrollen auf Anwendungs- und Netzwerkebene müssen verschiedene Aspekte betrachtet werden. Bei bisherigen signal-orientierten Architekturen wird die Kommunikation bereits in der Entwicklung spezifi-

ziert (s. Abschnitt 2.1.2). Das beinhaltet auch mögliche Nachrüstungen im Betrieb (z.B. neue Fahrerassistenz- oder Komfortfunktionen), die Werkstätten durch Software-Updates bzw. Hardwareänderungen umsetzen können. Das dadurch möglicherweise geänderte Kommunikationsverhalten entspricht dabei einer Spezifikation aus dem vorangegangen Entwicklungsprozess und ist daher bereits bekannt. Transferiert man die in dieser Arbeit adressierten Sicherheitsmechanismen (Zugriffskontrolle, Firewalls sowie IDS) auf hybride E/E-Architekturen, ergeben sich Unterschiede zu signal-orientierten Architekturen. Die Gründe für diese Abweichung basieren dabei hauptsächlich auf dem dynamischen Laufzeitverhalten in SOAs[1].

6.3.1 Firewalls

Traditionelle IT-Firewall Techniken existieren bereits seit über 25 Jahren und wurden eingeführt, um Security-Mängel in Protokollen zu kompensieren. Aus diesem Grund können Firewalls als Maßnahmen eingeordnet werden, die bestehende Schwächen von heutigen Kommunikationsprotokollen nicht vollständig auflösen, sondern die möglichen Auswirkungen durch Filtertechniken minimieren. Es existiert weiter das Risiko, dass ein Angreifer diese zusätzlichen Kontrollen umgeht und bestehende Protokollschwächen ausnutzt [68]. Des Weiteren sind Firewalls nicht in der Lage semantische Prüfungen der Nutzdaten durchzuführen. Lediglich Applikations-Level Firewalls (s. Abschnitt 2.2.3) sind fähig bestimmte Angriffsmuster nach spezifizierten Angriffstechniken zu erkennen. Mit Blick auf die Fahrzeugkommunikation sind Firewalls nicht geeignet, um sicherheitskritische Botschaften in Echtzeit bzgl. einem Kontext (z.B. aktueller Fahrzeugzustand, s. Definition 3.2.2) zu inspizieren [149]. Zudem sollte berücksichtigt werden, dass durch die Anwendung von Verschlüsselungstechniken (s. auch Abschnitt A.1.2) bei Automotive Ethernet beispielsweise auf der Anwendungsschicht (s. Abschnitt A.1.1), die übertragenen Daten für Firewalls auf dieser Schicht nicht analysierbar sind.

Transferiert auf zukünftige automotive Netzwerke sollten Protokollschwächen der klassischen IT nicht übernommen werden. Eine Sicherheitsuntersuchung

[1] Teile der nachfolgenden Abschnitte wurden in [RGKS20] veröffentlicht.

von Kreissel et al. [143] zeigte jedoch bereits diverse Protokollschwächen innerhalb des SOME/IP Protokolls (s. Abschnitt A.1.1) auf.

Vielmehr ist die Gewährleistung der grundlegenden Security-Eigenschaften (s. Abschnitt 2.2.1) wie Vertraulichkeit, Authentizität/Integrität bzw. Verfügbarkeit auf Protokollbasis notwendig. Auf Basis der verfügbaren/eingesetzten Protokolle sollte eine Bewertung erfolgen, ob Firewalls noch einen zuverlässigen Schutz bieten oder ob eine Kombination von sicheren Protokollen beispielsweise mit einer Zugriffskontrolle auf Anwendungsebene besser geeignet ist. Anhand einer hybriden Fahrzeugarchitektur (s. Abbildung 2.8) werden für drei enthaltene Netzwerkknoten die Fähigkeiten einer Firewall erläutert:

Gateway (signal/service-orientierte Kommunikation): Diese Gateways führen eine Übersetzung von signal-orientierter Kommunikation (z.B. CAN-Botschaften) auf eine service-orientierte Kommunikation (z.B. SOME-IP Pakete) sowie umgekehrt durch. Ausgehend von der SOA-Domäne wird jeder Service statisch auf eine Botschaft zugewiesen. Das ermöglicht eine Filterung dieser bekannten Botschaften. Darüber hinaus sind auch deren Nutzdaten, wie beispielsweise die Wertebereiche von Sensorsignalen für eine Filterung limitierbar. In umgekehrter Richtung agiert das Gateway als Serviceprovider und stellt Signale (z.B. aktuelle Geschwindigkeit oder Motordrehzahl) aus der signal-orientierten Kommunikation für Service-Clients zur Verfügung. Aus der Sicht einer Firewall ist dieses Szenario für eine Filterung herausfordernder, da prinzipiell jeder Client (entsprechende Berechtigungen vorausgesetzt) einen Service abonnieren kann. Dies beinhaltet, dass die jeweiligen IP-Adressen der Clients über die Laufzeit nicht statisch bestimmbar sind. Folglich sind klassische Paketfilter (ISO/OSI Schicht 3/4, s. Abschnitt 2.2.3) für diese Anwendung nicht geeignet. Stattdessen verleibt einer Filterung auf höheren Schichten (z.B. Anwendungslevel). Derartige Filterungen sind rechenintensiv und dadurch nicht in der Lage automotive Echtzeitanforderungen zu erfüllen. Daneben fehlen in Gateways umfassende Kontextinformationen, um Nachrichten entsprechend zu filtern [149]. Dies ist dagegen nur an den Kommunikations-Endpunkten auf den ISO/OSI Schichten 5 und höher möglich.

Connectivity ECU: Innerhalb dieses Netzwerkknotens erfolgt das Routing zwischen On-/und Off-Board Kommunikation. Dabei ergibt sich die Möglichkeit eine Firewall zu integrieren, die beispielsweise eine Kommunikation zwischen Fahrzeug und einem OEM-Backend kontrolliert. Als Filter wären Paketfilter einsetzbar, die nur Pakete mit der statischen IP-Adresse des Ba-

ckends weiterleiten. Zusätzlich wäre auch eine Kontrolle bzw. Einschränkung bestimmter Ports (s. Abschnitt 2.2.3) möglich.

OBD-Gateway: An diesem Gateway erfolgt ebenfalls ein Übergang zwischen On-/und Off-Board Kommunikation des Fahrzeugs. Mögliche externe Netzwerkknoten sind beispielsweise Diagnose-Tester aber auch OBD-Dongles (s. Abschnitt 3.1.4), die vorwiegend über Diagnoseprotokolle (s. Abschnitt A.1.1) mit dem Fahrzeug kommunizieren. Erfolgt die Kommunikation über das CAN-Protokoll ist eine Paketfilterung auf Basis von CAN-Identifier möglich. Darüber hinaus können auch Eigenschaften des Transportprotokolls ISO TP (s. Abbildung 2.1.3) genutzt werden. Wird hingegen eine Verbindung über das Diagnostics over Internet Protocol (DoIP)-Protokoll[2] aufgebaut, erhält das externe Gerät eine dynamische IP-Adresse. Dadurch ist eine Paketfilterung nur bedingt möglich bzw. bietet keinen umfassenden Schutz, da über die IP-Adresse nicht zwischen einem legitimierten Gerät oder einem Angreifer unterschieden werden kann. Daneben wäre zusätzlich eine Port-Filterung umsetzbar. Dabei ist zu beachten, dass verschiedene Anwendungen oftmals denselben Port verwenden.

6.3.2 Zugriffskontrolle

Die Analyse von Angriffen im automotive Bereich (s. Abschnitt 3.1) zeigte, dass bisher kein oder nur ein eingeschränktes Rechtemanagement (Diagnose) vorhanden war. Aus diesem Grund ist es notwendig auch in SOAs eine konsistente Zugriffskontrolle zu implementieren, um zu verhindern, dass Angreifer durch einen kompromittierten Netzwerkknoten uneingeschränkt Services böswillig ausnutzen. In signal-orientierten Netzwerken ist der Angreifer zumindest dahingehend eingeschränkt, dass Botschaften bestimmter Domänen (z.B. Powertrain) nicht grundsätzlich über alle in der Architektur enthaltenen Gateways geroutet werden (s. auch Abschnitt 2.1.2) und dadurch nicht grundsätzlich eine Kommunikation innerhalb des gesamten Netzwerks möglich ist. Das bedeutet, dass ein Angreifer von einer komprimierten ECU aus nicht einen beliebigen anderen Knoten über das Netzwerk erreichen kann. Hingegen ist dieses Sze-

[2] Das DoIP-Protokoll spezifiziert ein IP-basiertes Transportprotokoll, das vorwiegend bei Ethernet-Verbindungen eingesetzt wird, um Diagnosepakete nach dem UDS-Standard zu übertragen.

nario in einer SOA möglich, da prinzipbedingt jeder Service im Netzwerk erreichbar ist.

Unter der Annahme, dass in zukünftigen SOAs sichere Verbindungen (Kommunikationsprotokoll gewährleistet Sicherheits-Eigenschaften wie beispielsweise Authentizität/Integrität) vorhanden sind, ist zu beachten, dass ein Angreifer durch Insider-Wissen (s. auch Abschnitte 2.2.1 und 3.1.3) weiter eine Bedrohung darstellt. Falls dieser von einer legitimierten ECU aus agiert sind andere Teilnehmer nicht in der Lage den Angreifer zu erkennen, da keine Sicherheits-Eigenschaften des Kanals verletzt werden. Daher sollte die Menge an Berechtigungen für Services nur für die Ausführung der spezifizierten Funktionalität zwingend notwendigen Rechte beinhalten, um die mögliche Angriffsfläche zu minimieren (Prinzip der geringsten Privilegien [203]). Der AUTOSAR Adaptive Standard adressiert dieses Prinzip mit der Spezifikation des IAM-Moduls (s. Abschnitt 3.2.4). Da Fahrzeuge ein CPS repräsentieren (s. Abschnitt 1.2) ist zudem zu beachten (wie auch in signal-orientierten Architekturen), dass diese Systeme verschiedene physikalische Zustände aufweisen (s. Abschnitt 6.3.2) und für eine Zugriffskontrolle von Bedeutung sind [149].

Zugriffsrelevante Zustandsinformationen

Die Einführung von SOAs bietet im Hinblick auf die Bestimmung zugriffsrelevanter Fahrzeugzustände durch die Auswertung von unterschiedlichen Informationsquellen (s. Tabelle 6.1) einfachere sowie erweiterte Möglichkeiten im Vergleich zu signal-orientierten Architekturen. Die Zentralsteuergeräte (s. Abbildung 2.8) verfügen zu jedem Zeitpunkt über aktuelle Informationen der Sensor/Aktor-Ebene. Dadurch kann zentralisiert eine Bestimmung des aktuellen Zustands[3] erfolgen (z.B. Fahrzeug in Bewegung). Darüber hinaus sind weitere Zustandsinformationen ableitbar, die Rückschlüsse auf aktuelle Fahrmanöver/Fahrsituationen ermöglichen.

So sind bei automatisierten Fahrfunktionen Kamera- oder Radarinformationen auswertbar (z.B. vorausfahrendes Fahrzeug erkannt). Zudem sind auch Kartendaten verwertbar, die ortsbezogene Daten liefern (Fahrzeug befindet sich in

[3] Die Zusammenhänge bzgl. Zuständen, Fahrzeugattributen sowie Zugriffsentscheidungen sind im Metamodell in Abschnitt 4.1.3 dargestellt.

Tabelle 6.1: Übersicht verfügbarer Informationsquellen zur Bestimmung von zugriffsrelevanten Zustandsinformationen in automotive SOAs.

Quelle	phys. On-Board Sensorsignale	Umweltinformationen	Funktionszustände
Beispiele	Drehmoment, Leistung, Beschleunigungsrate	Radardaten, Kameradaten, Kartendaten	Fahrerassistenz
Filterart	Paket	Paket	Paket, Applikation
Protokoll	CAN, Ethernet, FlexRay	Ethernet	Ethernet

einer Stadt oder auf einer Autobahn). Zusätzlich stellen aktuelle Zustände von Funktionen wie der Fahrerassistenz oder Regelsystemen ebenfalls verwertbare Informationen bzgl. der Zugriffskontrolle dar. Damit sind sehr präzise kontextbezogene Zugriffsentscheidungen möglich.

6.3.3 Adaptierungspotentiale des A-ABAC Ansatzes

Zur Absicherung von automotive SOAs ist ein Rechtemanagement essenziell, damit Services nur von autorisierten Subjekten abonnierbar bzw. ausführbar sind. In hybriden Architekturen (s. Abbildung 2.8) kann die zunehmende Zentralisierung der Steuergeräte sowie die Abstraktion zwischen Rechen- und Sensor/Aktor-Ebene genutzt werden, um eine feingranulare Zugriffskontrolle auf Serviceebene zu realisieren.

Integration in eine hybride E/E-Architektur

Analog zu signal-orientierten Architekturen lassen sich die Module PEP, PDP sowie PIP (s. Abschnitt 4.1.2) des entwickelten A-ABAC Ansatzes in die service-orientierte Ebene integrieren, die in Fahrzeugen durch die AUTOSAR Adaptive Plattform realisiert wird. In der zugehörigen Spezifikation [204] sind bereits erste Anforderungen für die Implementierung einer Zugriffskontrolle enthalten. So wird beispielsweise explizit die Verwendung von PEPs sowie PDPs gefordert, um die Berechtigungen von Services auf Basis von vordefinierten Policies zu kontrollieren bzw. durchzusetzen. Dabei ist die Standardisierung von Policies nicht Teil von AUTOSAR. Zusätzlich wird innerhalb der

Spezifikation definiert, dass Zugriffsentscheidungen in Abhängigkeit des aktuellen Fahrzeugzustands erfolgen sollen. Ergänzend dazu bietet der entwickelte A-ABAC Ansatz im Vergleich zu AUTOSAR Adaptive folgende Eigenschaften:

- Zugriffskontrollmodell zur Umsetzung der Module PEP und PDP
- Verwendung einer etablierten Beschreibungssprache zur Definition von Sicherheitsrichtlinien (Policies)
- Einbezug aktueller Fahrzeugzustände sowie Bereitstellung einer Methodik zur Ermittlung und Auswahl von Sensoren für die Zustandsermittlung

Anwendungsbeispiel

Für eine exemplarische Demonstration der Anwendbarkeit des A-ABAC Ansatzes in service-orientierte Architekturen, dient als Untersuchungsgegenstand eine reduzierte hybride SOA-Architektur (s. Abbildung 6.3). Auf der Grundlage der *Super Tux Anwendung*[4] werden Berechtigungen für verschiedene Services definiert und deren Kontrolle erläutert. Zudem werden den Berechtigungen unterschiedliche Kontextinformationen zugeordnet, die dynamische Zugriffsentscheidungen ermöglichen.

Die Architektur umfasst dabei zwei verschiedene Kommunikationsparadigmen, die über zwei Domänen-Controller (Gateway) verbunden sind. Innerhalb der signal-orientierten Domäne sind drei Legacy-ECUs (Motor, Steering, LIN) enthalten. Innerhalb der service-orientierten Domäne sind dagegen die leistungsstarken Steuergeräte (Central Computing Cluster, Domänen-Controller) integriert, die über ein Ethernet-Netzwerk vernetzt sind. Für die Kontrolle und Durchsetzung von Zugriffsberechtigungen lassen sich darin verschiedene Module des entwickelten A-ABAC Ansatzes integrieren. Da das Zentralsteuergerät, die zentrale Steuer- und Regelinstanz ist und gleichzeitig die höchste Rechenpower bietet, werden darauf PEP, PDP und PIP integriert. Des Weiteren wird jedem Domain-Controller ein PEP sowie PDP zugeordnet. Für die

[4] Als SuperTux Anwendung wird eine Spieleanwendung eines OEMs bezeichnet, die es Kunden ermöglicht, das Fahrzeug im Stillstand als Spielekonsole zu nutzen [205].

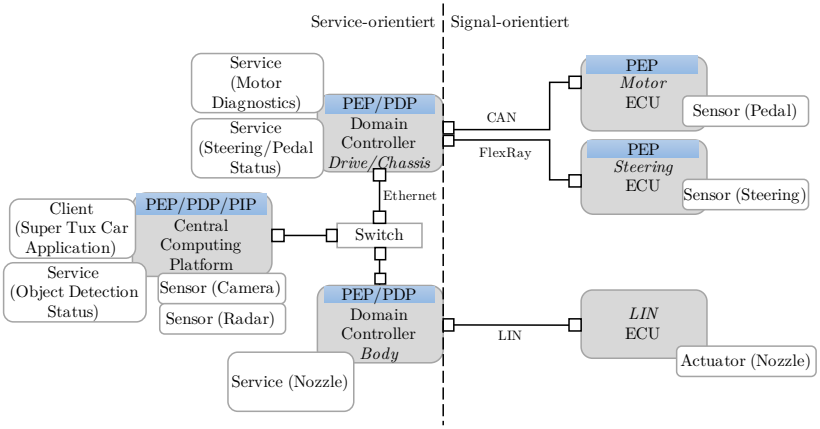


Abbildung 6.3: Exemplarische hybride SOA-Architektur mit Darstellung benötigter Services und Signale auf Basis der *Super Tux Anwendung* (adaptiert von [VOG+20]).

Legacy-ECUs wird aus Performance-Gründen nur ein PEP verwendet, da die Zugriffsentscheidungen vom Domänen-Controller getroffen werden (analog zum Ansatz in signal-orientierten Architekturen, s. Abschnitt 4.1.2).

Applikationsbeschreibung

Die zu betrachtende Applikation (Super Tux Anwendung) ist in der Lage den Luftmassenstrom der Fahrzeugklimatisierung auf Basis der Geschwindigkeit zu regeln und Fahrwerksaktoren (Stoßdämpfer) aktiv anzusteuern, wodurch das Fahrzeug in Kombination mit dem Infotainment-System als eine Art 4-D Fahrsimulator dienen kann. Die Super Tux Anwendung wird dabei als Service-Client auf dem Zentralsteuergerät (s. Abbildung 6.3) ausgeführt. Diese kann wiederum verschiedene Services auf den Domänen-Controller ausführen bzw. abonnieren. Die Funktionen der Services bilden dabei Steuer-und Statusinformationen aus der signal-orientierten Sensor/Aktor Ebene ab. Dadurch ist die Anwendung vom Zentralrechner aus in der Lage einen Aktor anzusteuern bzw. aktuelle Statusinformationen der Sensoren zu erhalten. Die verschiedenen Kommunikationsbeziehungen, die über die Services zwischen beiden

Paradigmen auf Basis der *Super Tux* Anwendung bestehen, lassen sich in einer logischen Architektur darstellen (s. Abbildung 6.4).

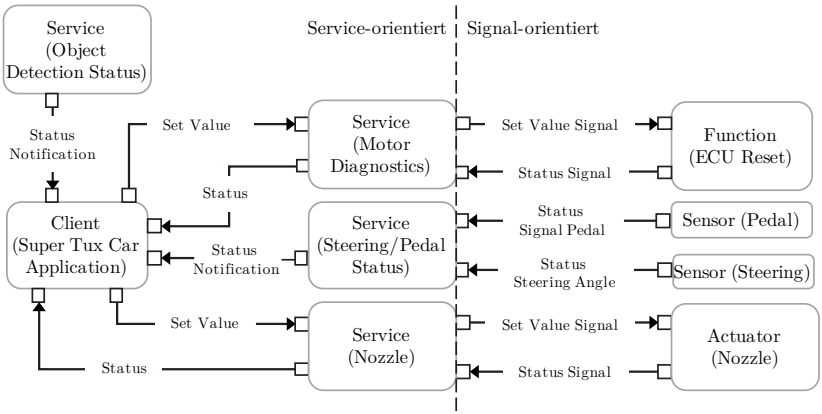


Abbildung 6.4: Logische Kommunikationsbeziehungen zwischen service- und signal-orientierter Kommunikation auf Basis der Super Tux Anwendung (adaptiert von [VOG+20]).

Neben der konventionellen Sensor-/Aktor Ebene ist der Zentralrechner mit Sensoren (Kamera, Radar) in der Lage, Objekte wie z.B. Fahrzeuge oder Personen außerhalb des Fahrzeugs zu erkennen. Die entsprechenden Statusinformationen der Objekterkennung werden über den Service *Object Detection* für Anwendungen wie beispielsweise der *Super Tux Car Application* zur Verfügung gestellt.

Zugriffsberechtigungen

Auf Basis der in Abbildung 6.3 gezeigten Architektur lassen sich Berechtigungen für Anwendungen bzw. Clients über eine Zugriffs-Policy spezifizieren (s. Tabelle 6.2). Die darin enthaltenen Berechtigungen (Aktionen) für den jeweiligen *Client* bezogen auf einen *Provider* sind mit Zustandsinformationen kombinierbar, um feingranulare Zugriffsrechte zu definieren. Die exemplarischen Einträge beziehen sich dabei auf die Super Tux Anwendung, die auf dem Zentralrechner ausgeführt wird und einen Client repräsentiert, der Services von

anderen Provider (s. Abbildung 6.4) nutzt. Die berücksichtigten Zustandsinformationen (s. Abschnitt 6.3.2) sind dabei Fahrzeugzustände (z.B. aktuelle Geschwindigkeit) sowie Umfeldinformationen (z.B. andere Fahrzeuge, Personen). Der Vorteil gegenüber signal-orientierten Architekturen liegt darin, das Sensor-Aktor Informationen nicht auf einer bestimmten ECU oder Domäne verleiben, sondern immer der zentralen Verarbeitungsschicht (Central Computing Cluster) über Services zur Verfügung gestellt werden. Dadurch kann die Kontrolle und Durchsetzung der Berechtigungen zentralisiert auf diesem leistungsstarken Cluster auf Anwendungsebene erfolgen. Da die exemplarische Zugriffsrichtlinie dieselbe Struktur im Vergleich zu der im A-ABAC Ansatz (s. Abschnitt 4.1) verwendeten aufweist, ist eine Beschreibung über die ALFA-Sprache (s. Abschnitt 4.1.8) möglich, um diese über einen PDP auszuwerten. Insgesamt ist eine Adaptierung der ABAC Module (PEP, PDP und PIP) möglich, um diese auf die Zentralsteuergeräte in einer SOA zu integrieren.

Tabelle 6.2: Übersicht einer exemplarischen Zugriffs-Policy, die Service-Berechtigungen für Clients (Subjekte) in Kombination mit dazugehörigen Service-Provider (Objekte) in einer automotive SOA spezifiziert.

Client	**Aktion**	**Provider**	**Erlaubte Zustände**
Super Tux	abonnieren	Domain-Controller Body	Sitzerkennung muss Fahrer melden, Fahrzeug im Stillstand
Super Tux	ausführen	Nozzle	Nur wenn Umfeld-Erkennung kein Objekt meldet, Fahrzeug im Stillstand
Super Tux	abonnieren	Umfeld-Erkennung	Fahrzeug im Stillstand
Super Tux	ausführen	Motor-Diagnose	Fahrzeug im Stillstand, Lesezugriff

Die Limitierung der Zugriffsrechte ermöglichen das Risiko für einen erfolgreichen Angriff, speziell in einem unerlaubten Systemzustand minimieren. Unter der Annahme, dass es einem Angreifer gelingt sich erfolgreich als *Super Tux* Client gegenüber den Service-Anbietern auf den Domänen-Controllern zu authentifizieren, wäre u.a. eine plötzliche Ausführung während der Fahrt oder bei erkannten Objekten außerhalb des Fahrzeugs nicht möglich.

6.3.4 Intrusion Detection Systeme

Die existenten Ansätze im Bereich von automotive IDS basieren vorwiegend auf dem CAN-Protokoll und adressieren dadurch die signal-orientierte Kommunikation (s. Abschnitt 3.2.3). Dagegen gibt es im Vergleich für automotive SOAs bisher nur vier publizierte Arbeiten [206], [207], [208], [209], die erste Möglichkeiten und Ansätze zur Erkennung von Angriffen mit Fokus auf das SOME/IP Protokoll (s. Abschnitt A.1.1) enthalten. Dies ist zum einen auf das in der automotive Domäne relativ neue Kommunikationsparadigma zurückzuführen, zum anderen lassen sich bisherige Ansätze auf Basis signal-orientierter Kommunikation nur eingeschränkt übertragen. Für die Entwicklung zukünftiger IDS-Ansätze für SOAs muss zunächst definiert werden, welche Features für eine Detektion geeignet sind. Darüber hinaus muss analysiert werden, an welchen Stellen eine sinnvolle IDS-Integration erfolgen kann, da sich die Netzwerke durch eine dynamische Rekonfiguration verändern können. Daran knüpft eine weitere Designfrage an, welche Vor- und Nachteile ein Host- bzw. Netzwerk-basiertes IDS (s. Abschnitt 2.2.4) beinhalten. Zusätzlich ist auch eine Analyse der Anwendbarkeit von Signatur- und Anomalie-basierten Techniken notwendig.

Merkmale für die Anomalieerkennung in SOAs

Nach Al-Jarrah et al. (s. Abschnitt 3.2.3) sind bisherige IDS-Ansätze basierend auf dem CAN-Protokoll durch drei verschiedene Methoden klassifizierbar (Fluss-basiert, Nutzdaten/Inhalts-basiert sowie Hybrid). Dabei nutzen Fluss-basierte Erkennungsansätze Netzwerk-spezifische Eigenschaften, wie beispielsweise die jeweiligen Zykluszeiten der Botschaften. Dagegen analysieren Nutzdaten/Inhalts-basierte Ansätze die jeweilige Botschaft auf Basis der in der K-Matrix spezifizierten Eigenschaften (s. Abschnitt 2.1.2). Dadurch sind Auswertungen z.B. auf Signalbasis möglich, ob Abweichungen zu definierten Wertebereichen existieren. Darüber hinaus sind auch Plausibilitätschecks von einzelnen Signalen durchführbar. Transferiert auf automotive SOAs ist diese Methode nur eingeschränkt adaptierbar. Die Gründe sind zum einen die höheren Datenraten (Faktor 200 - 2000 bei 100 MBit/s bzw. 1 GBit/s) von automotive Ethernet (s. Tabelle A.1) und die damit verbundene Erhöhung der enthaltenen Nutzdaten pro Paket. Ein Vergleich zur klassischen IT, die grund-

Tabelle 6.3: Anwendbarkeit von On-Board Anomalie-Detektion-Sensoren [180] - adaptiert und analysiert für die Integration in SOA-Fahrzeugarchitekturen (basierend auf [RGKS20]).

	Kriterien (Anwendbarkeit)						
Detection Sensor	Specif. Based	Num. of Messages	Num. of Bus Systems	Different Message Types	Payload Inspection	Semantic Based	Anwendbar in SOAs
Formality	✓	1	1	n/a	✗	✗	(✓)
Location	✓	1	1	n/a	✗	✗	✓
Range	✗	1	1	n/a	✓	✗	✗
Frequency	✗	n	1	✗	✗	✗	(✓)
Correlation	✓	n	n	✓	✗	✗	✓
Protocol	✗	n	n	✓	✗	✗	(✓)
Plausibility	✗	n	1	✗	✓	✓	✗
Consistency	✗	n	n	✓	✓	✓	✗

legend auf dem Ethernet-Protokoll aufbaut, zeigt sich im Bereich IDS ein ausgeprägter Trend zu Fluss-basierten Erkennungsmerkmalen [210] auf den ISO/OSI Schichten 2 und 3 (s. Abschnitt A.1.1). Dadurch wäre es zielführend diese Art auf die automotive Domäne zu übertragen. Welche exakten Fluss-Merkmale eine derartige SOA aufweist, ist aktuell noch schwer abzuschätzen, da diese Technologie noch am Anfang der Entwicklung steht. Jedoch ist eine Analyse bzgl. der Übertragbarkeit auf Basis der von Müter et al. [180] definierten IDS-Sensoren (s. auch Abschnitt 5.1.2) möglich, um die Anwendbarkeit in Bezug auf SOAs zu identifizieren (s. Tabelle 6.3).

Durch die zuvor erwähnten Einschränkungen bzgl. einer Nutzdaten-Analyse ist eine Übertragung der Sensoren *Range*, *Plausibility*, *Consistency* auf Netzwerkebene nicht generell möglich. Darüber hinaus sind die Kommunikationsbeziehungen zwischen Netzwerkknoten nicht mehr statisch definiert, sondern sind abhängig von den benötigten Informationen während der Laufzeit. Dadurch ist eine Spezifizierung von *Range*, *Frequency* und *Protocol* nicht mehr gegeben (in orange gekennzeichnet). Auf diesen Features basierend sind bisherige Erkennungsmethoden, die beispielsweise auf Methoden des maschinellen Lernens (s. Abschnitt 2.2.4) aufbauen, schwieriger anzuwenden, da sich ein zu trainierendes Normalverhalten dynamisch verändern kann. Betrachtet man den Sensor *Formality* ist dieser mit Einschränkungen übertragbar, da im Vergleich

zu signal-orientierter Kommunikation auf Protokollebene weniger statisch spezifiziert ist und sich damit die Detektionsmöglichkeiten verringern. Daneben ist das Kriterium *Anzahl der Bussysteme* in SOAs nicht definierbar, da das gesamte Netzwerk eine hierarchische Ethernet-Topologie darstellt. Vielmehr ist hier eine Analyse bzgl. definierter *Virtual Local Area Networks (VLANs)* oder die Anzahl von *Topics* auf Basis des DDS Protokolls zielführend, um mit Hilfe des *Location* Sensors, Abweichungen zu erkennen. Allerdings muss gleichzeitig berücksichtigt werden, dass durch neu eingeführte Services im Rahmen eines Updates oder bei einer Rekonfiguration des Netzwerks durch den Ausfall eines Providers, sich Änderungen bzgl. der Kommunikationspfade ergeben. Durch die geänderten Netzwerkeigenschaften wäre dann auch der *Location* Sensor betroffen.

Definition 6.3.1 VLAN

Als VLAN wird ein logisches Teilnetzwerk definiert, das sich innerhalb eines physikalischen Netzes befindet und dieses aufteilt [23]. Eine etablierte Methode ist die Verwendung von Markierungen (Tags) in Netzwerkpaketen auf Basis des IEEE 802.1Q Standards.

Zusammenfassend ergibt sich auf der Grundlage der unterschiedlichen Eigenschaften von signal-basierten bzw. service-orientierten Netzwerken, dass die Übertragung der IDS-Sensoren nur teilweise möglich ist. Dagegen jeden einzelnen Service mit speziellen Monitoring-Funktionalitäten auszustatten, würde wiederum einer Nutzdatenanalyse entsprechen und wäre aus den zuvor genannten Gründen nicht anwendbar. Daher wäre ein mögliches Ziel eines Netzwerk-basierten IDS, die Erkennung auf Middleware Ebene zu integrieren. Ein erster konkreter Ansatz auf Basis des SOME/IP Protokolls ist in [208] gegeben.

Signatur- und Host-basierte Ansätze

Durch die zunehmende Standardisierung von Betriebssystemen sowie Adaptierung von etablierten Protokollen der klassischen IT wie beispielsweise Ethernet sind zukünftig Schwachstellen realistisch, die nicht mehr ausschließlich Hersteller- oder Baureihen-spezifisch sind. Um dieses Risiko zu adressieren sind Bestrebungen für das Teilen von Wissen über Bedrohungen und Schwach-

stellen zwischen verschiedenen Herstellern und Zulieferer von besonderer Relevanz. Durch bekannte Schwachstellen ist typischerweise die Ableitung von Signatur-basierten IDS (s. Abschnitt 2.2.4) möglich, die komplementär zu Anomalie-basierten Systemen einsetzbar sind. Auf Industrie-Ebene gibt es bereits Bestrebungen dieses Wissen im Rahmen des Automotive Information Sharing & Analysis Center (Auto-ISAC) [211] zu teilen. Ergänzend dazu existieren in der Wissenschaft weitere Ansätze zu Meldeplattformen und Prozessen [201] sowie zur verantwortungsvollen Offenlegung (engl. responsible disclosure) von automotive Schwachstellen.

Zusätzlich ist durch die Verwendung von *Portable Operating System Interface (POSIX)*-basierten Betriebssystemen wie beispielsweise AUTOSAR Adaptive eine Fokussierung auf Host-basierte IDS (s. Abschnitt 2.2.4) sinnvoll, da zukünftig auch Drittanbieter verschiedene Dienste in das Fahrzeug integrieren [212]. Daraus ergibt sich neben der Netzwerk-Ebene eine weitere Dynamik bzw. ein nicht statisch definiertes Verhalten auf der Anwendungsebene des jeweiligen Hosts (ECU). Jedoch ist auf dieser Ebene wiederum eine Nutzdatenanalyse anwendbar. Im Rahmen des AUTOSAR-Gremiums gibt es dazu bereits erste Standardisierungsaktivitäten bzgl. eines Host-basierten IDS in SOAs [213]. Diese umfassen ein generisches Architektur-Design für ein verteiltes IDS mit zugehörigen Schnittstellenbeschreibungen sowie standardisierten Anomalietypen (Security-Events).

Definition 6.3.2 POSIX

Als POSIX wird eine standardisierte Programmierschnittstelle bezeichnet, die zwischen Betriebssystem und Anwendungssoftware aufsetzt.

Integration in E/E-Architekturen

Betracht man des Weiteren mögliche Integrationspunkte für ein IDS in SOAs gibt es Unterschiede zu signal-orientierten Architekturen. Bei der Verwendung von CAN-Netzwerken ist durch die Broadcast-Eigenschaft (s. Abschnitt A.1.1), eine Integration an jeder beliebigen Stelle am Bus möglich, um den Netzwerkverkehr zu analysieren. Hingegen muss bei Ethernet bzw. SOA-basierten Netzwerken die Integration von IDS an Stellen der E/E-Architektur erfolgen,

an denen zum einen die erforderlichen Daten zur Verfügung stehen und zum anderen ausreichende Ressourcen zur Verarbeitung vorhanden sind. Eine Möglichkeit stellt die Integration auf Central Computing Clusters (CCCs) [214] dar. Unter Verwendung einer hybriden E/E-Architektur (s. Abbildung 2.8) müssen jedoch Randbedingungen beachtet werden. Besteht beispielsweise die Anforderung, die Datenströme zwischen ESP- und Kamera-ECU zu überwachen, so ist eine Fluss-basierte Analyse mittels einem Netzwerk-basierten IDS auf dem CCC nicht direkt möglich. Der Grund dafür ist die zugrundeliegende Unicast-Verbindung, welche die zwei Netzwerknoten basierend auf Ethernet über die Switches hinweg aufbauen. Dadurch ist entweder eine Analyse direkt auf den Switches möglich oder indirekt auf dem CCC mit der Vorbedingung, dass benötigte Informationen getrennt bereitgestellt werden müssen. Da Services in einer SOA auch mit externen Netzwerkteilnehmern kommunizieren, ist für eine umfassende Fluss-Daten Analyse auch die Überwachung der Datenströme über das Connectivity-Gateway notwendig. Um sowohl interne als externe Datenströme gesamtheitlich zu betrachten, besteht die Möglichkeit, die Informationen an das CCC zur Analyse weiterzugeben. Entsprechende Standardisierungsaktivitäten für ein zentrales oder dezentrales IDS (zur Steigerung der Ausfallsicherheit) für hybride E/E-Architekturen laufen derzeit innerhalb des AUTOSAR-Gremiums [213].

6.3.5 Zusammenfassung

Da in der nächsten Generation von E/E-Architekturen verstärkt eine serviceorientierte Kommunikation zum Einsatz kommt (s. Abschnitt 2.1.2), wurden Unterschiede im Vergleich zur Absicherung von signal-basierten Architekturen in Bezug auf Maßnahmen wie Firewalls, Zugriffskontrolle und IDS erarbeitet. Wesentlich Unterschiede bilden dabei die zunehmende Konzentration von bisher verteilten Steuergeräten auf leistungsstarke Zentralsteuergeräte sowie die nicht mehr ausschließlich statisch definierte automotive Ethernet Kommunikation. Gestützt wurden die Analysen durch die Betrachtung verschiedener Netzwerkknoten innerhalb einer Fahrzeugarchitektur. Darüber hinaus wurde anhand einer beispielhaften Anwendung auf Basis einer hybriden E/E-Architektur erläutert, wie kontextabhängige Zugriffsentscheidungen für Services bzw. Clients durch die Adaptierung des A-ABAC Ansatzes in SOAs möglich sind. Darüber hinaus wurden Unterschiede für die Adaptierung von

signal-basierten IDS-Ansätzen mit Fokus auf Erkennungsfeatures sowie deren Integration in SOAs analysiert.

A Anhang

A.1 Netzwerktechnik und Sicherheit

A.1.1 Kommunikationstechnologien

Im Laufe der Jahre haben sich in der Fahrzeugdomäne beginnend mit CAN verschiedene Kommunikationstechnologien entwickelt und etabliert (s. Tabelle A.1).

Tabelle A.1: Übersicht an verschiedenen Netzwerktechnologien [51, 45].

Technologie	**CAN-(FD)/XL**	**LIN**	**FlexRay**	**Automotive Ethernet (100Base-T1)**
Steuerung	Ereignis	Zeit	Zeit	Ereignis
Zeitverhalten	!deterministisch	deterministisch	deterministisch	!deterministisch
Architektur	Multi-Master	Master-Slave	Multi-Master	Punkt-zu-Punkt
Buszugriff	CSMA/CA	TDMA	TDMA	Voll-Duplex
Adressierung	Nachricht	Nachricht	Nachricht	Quelle/Ziel
Max. Datenrate	1-(10) Mbit/s	20 kbit/s	20 Mbit/s	100 Mbit/s
Nutzdaten	8-(64)/2048 Byte	8 Byte	254 Byte	1500 Byte
Medium	1 Twisted-Pair	1 Twisted-Pair	1-2 Twisted-Pair	1 Twisted-Pair
Topologie	Bus	Bus	Bus, Stern	Stern

ISO/OSI Referenzmodell

Das ISO/Open Systems Interconnection model (OSI) Modell definiert sieben unterschiedliche Kommunikationsschichten (s. Abbildung A.1) als herstellerunabhängige Grundlage für das Design von Netzwerkprotokollen und wurde 1984 standardisiert [215]. Für jede dieser Schichten sind bestimmte Aufgaben bzw. Funktionen zugeordnet, die für eine Kommunikation zwischen Systemen notwendig sind. Eine kurze Beschreibung dieser Aufgaben ist nachfolgend erläutert.

1. Die Bitübertragungsschicht bildet die unterste Ebene und übernimmt Aufgaben für die elektrische, mechanische sowie funktionale Verbindung zum Übertragungsmedium (z.B eine Zweitdrahtleitung).

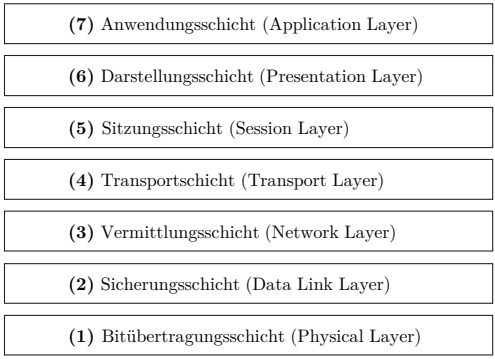


Abbildung A.1: Übersicht der Kommunikationsschichten des ISO/OSI Referenzmodells (basierend auf [215]).

2. Die darüberliegende Sicherungsschicht ermöglicht eine Bereitstellung von Datenpaketen (Frames) sowie Mechanismen zur Fehlererkennung die bei der Übertragung auftreten können. Zusätzlich bietet diese die Funktionalität der physikalischen Adressierung.

3. In der Vermittlungsschicht auf der dritten Ebene werden Routing-Aufgaben (Wegbestimmung zwischen Sender- und Empfänger) definiert sowie eine logische Adressierung bereitgestellt.

4. Die vierte Ebene (Transportschicht) agiert als Bindeglied zwischen höherliegenden Anwendungs- sowie darunterliegenden Transportschichten. Es erfolgt dabei eine Zuordnung auf eine bestimmte Anwendung.

5. Die Sitzungsschicht übernimmt steuerungstechnische Aufgaben in Verbindungen. Beispielweise werden bei entfernten Funktionsaufrufen (engl. Remote Procedure Calls (RPCs)) die Anworten den entsprechenden Aufrufern zugeordnet.

6. Auf der sechsten Ebene (Darstellungsschicht) erfolgt eine Umwandlung der Daten in entsprechende Formate die von oder zu der Anwendungsschicht stammen.

7. Die Anwendungsschicht repräsentiert die oberste Ebene, die eine Verbindung mit den darunterliegenden Schichten sowie der eigentlichen Systemanwendung ermöglicht.

Controller Area Network (CAN)

Der CAN-Bus ist eine Bustechnologie, die Nachrichten über eine ungeschirmte, verdrillte Zweidrahtleitung (engl. twisted pair) durch ein Differenzpegelsignal störsicher überträgt. Darüber hinaus ist diese Technologie üblicherweise als Linientopologie ausgeführt, in der die Teilnehmer über einzelne Stichleitungen mit dem CAN-Bus verbunden sind. Die maximal erreichbare Datenrate ist auf 1 Mbit/s spezifiziert [51]. Im Wesentlichen besteht eine Botschaft (s. Abbildung A.2) aus einem Header, der zur Adressierung einen 11- bzw. 29-bit Identifier (ID) enthält, sowie einen Data Length Code (DLC), der definiert, wie viele Nutzdatenbytes die zugehörige Botschaft enthält. Neben dem eigentlichen Nutzdatenfeld existieren noch einige weitere Felder (z.B. Cyclic Redundancy Check (CRC) oder Acknowledge (ACK)), die zur Kontrolle und Bestätigung einer fehlerfreien Übertragung dienen.

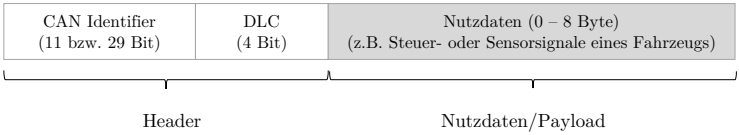


Abbildung A.2: Grundlegender Aufbau einer CAN-Botschaft (basierend auf [51].)

Da das klassische CAN-Protokoll ursprünglich nur für Nutzdaten bis 8 Datenbytes spezifiziert war, wurde im Jahr 2012 die Erweiterung CAN Flexible Data-Rate (CAN FD) [131] eingeführt, um u.a. einen quantitativen Aspekt zu adressieren. Bei der Einführung des CAN-Busses mussten lediglich einige hundert Signale übertragen werden. Dagegen lässt sich die aktuelle Zahl an Signalen in Fahrzeugen in einem fünfstelligen Bereich beziffern [216]. Der CAN-FD Standard erweitert daher die Nutzdatenlänge auf bis zu 64 Bytes und erhöht gleichzeitig die Datenrate auf bis zu 10 Mbit/s. Eine zweite Erweiterung die in zukünftigen Fahrzeugarchitekturen zum Einsatz kommen soll bildet die CAN-XL Technologie [217], die eine Nutzdatenlänge von bis zu 2048

Bytes bei einer Datenrate von 10 Mbit/s gewährleistet. Dazu ist eine Abwärtskompatibilität zu CAN-FD gegeben und der parallele Einsatz auf dem gleichen physikalischen Medium umsetzbar. Dazu besteht die Möglichkeit auch Ethernet-Frames mit dem Protokoll zu übertragen, sodass auch eine serviceorientierte Kommunikation abbildbar ist. Prinzipbedingt wird der CAN als Broadcast-Medium bezeichnet, da jeder verbundene Teilnehmer alle versendeten Nachrichten empfangen kann. Die ID der Botschaften dient hierbei nicht als explizite Quell- oder Zieladresse, sondern kennzeichnet die Botschaft bzw. den Nachrichteninhalt. Empfängerknoten enthalten darüber hinaus einen ID-Akzeptanzfilter, in dem hinterlegt ist, welche Botschaften für diesen speziellen Knoten bestimmt sind. Dadurch verwerfen die anderen ECUs Botschaften, die sie nicht benötigen. Das CAN-Protokoll ist nicht-deterministisch aufgebaut, wodurch keine definierte Übertragungszeiten garantiert werden können [51]. Vielmehr orientiert sich die Übertragung an der ID der Botschaften, die für eine Prioritätssteuerung verwendet wird. Im Standard ist definiert, je niederwertiger die ID, desto höherprior ist die Botschaft. Möchte ein Bus-Teilnehmer eine Nachricht versenden erfolgt die beschriebene Prioritätssteuerung innerhalb der Arbitrierungsphase. In dieser Phase beginnt jeder Teilnehmer, der eine Nachricht versenden möchte, seine ID auf den Bus zu legen. Gleichzeitig liest jeder Teilnehmer auf dem Übertragmedium parallel mit. Bemerkt dieser, dass ein anderer Sender eine höherpriore Botschafts-ID besitzt, wird der Sendevorgang unterbrochen. Dieses Prinzip des Buszugriffs wird in der Netzwerktechnik allgemein als Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) Verfahren [218] bezeichnet, wodurch ein Mehrfachzugriff auf ein Medium möglich ist und Teilnehmer eigenständig mögliche Kollisionen auf dem Übertragungskanal erkennen und vermeiden.

Local Interconnected Network (LIN)

Der LIN wurde zum Ende der 90er Jahre entwickelt und in ISO 17987 standardisiert, um eine kostengünstige Lösung zur Vernetzung von einfachen Sensor-Aktor-Wirkketten zu ermöglichen. Die Übertragung erfolgt dabei auf einer Eindrahtleitung mit einer maximalen Datenrate von bis zu 20 kbit/s und 8 Nutzdatenbytes. Im Gegensatz zum CAN basiert die Kommunikation auf einem Master-Slave Prinzip. Dadurch gibt es in einem LIN-Netzwerk einen Masterknoten und sowie bis zu 16 Slaveknoten. Ausschließlich der Master ist für die Kommunikationssteuerung verantwortlich, da dieser über eine oder

mehrere Tabellen zur Steuerung der Botschaften (engl. scheduler) verfügt. Die Tabellen enthalten dabei detaillierte Botschaftsinformationen wie Header- und Nutzdateninhalte sowie einen definierten Zeitrahmen - auch Zeitschlitz genannt - indem die Botschaft übertragen wird. Die Kommunikation sowie der Buszugriff werden dabei deterministisch nach dem Time Division Multiple Access (TDMA) Verfahren auf Basis der statisch hinterlegten Botschaftsreihenfolge umgesetzt. Die Slave-Teilnehmer besitzen hingegen keinerlei Informationen über die gesamte Kommunikation, sondern können ihre Nutzdatenbytes erst dann auf den Bus legen, wenn der Master ein passendes Frame mit einem Header vorgegeben hat. Folglich sind die Slave-Knoten nicht in der Lage eine Botschaft, abweichend der definierten Reihenfolge, auf dem Bus zu versenden. Das Netzwerk wird wie der CAN als Linientopologie aufgebaut, wobei der LIN-Master überwiegend auf einem CAN-Knoten mit integriert wird und dieser zusätzlich als Gateway fungiert.

FlexRay

Im Jahr 2001 wurde innerhalb der Automobilbranche eine weitere Netzwerktechnologie mit der Bezeichnung *FlexRay* entwickelt, die nach ISO 17458 definiert ist. Der Treiber für diese Aktivitäten war der Bedarf nach einer Technologie, die für sicherheitskritische Datenübertragungen von *X-by-Wire* Systemen geeignet ist [45]. Die FlexRay Technologie bietet eine Datenrate von bis zu 20 Mbit/s und kann dabei bis zu zwei Kanäle gleichzeitig nutzen. Dies ermöglicht ausfallsichere (redundante) Verbindungen. Aufgrund der Kosten ist es wahrscheinlich, dass Automotive Ethernet den FlexRay in naher Zukunft ablösen wird [219].

Definition A.1.1 FlexRay

Unter X-by-Wire werden Systeme definiert bei denen der Steuerimpuls nicht über eine mechanische Verbindung zu einem zugehörigen Aktor übertragen wird, sondern ausschließlich elektronisch. In aktuellen Fahrzeugen stellt dies beispielsweise ein Gaspedal (Drive-by-Wire) dar. Der integrierte Sensor erfasst die Steuerbefehle des Fahrers und übermittelt diese elektronisch an ein zugehöriges ECU, das wiederum entsprechend den Motor (Aktor) regelt [220].

Automotive Ethernet

Der steigende Grad der Vernetzung des Fahrzeuges sowohl intern als auch extern mit der *Außenwelt* (vorwiegend über drahtlose Schnittstellen) sowie steigende Anforderungen an die Bandbreite, erforderten die Entwicklung und Einführung einer neuen Vernetzungstechnologie [221]. Darüber hinaus hat sich in anderen Domänen wie der klassischen IT seit Jahrzehnten in der Vernetzung die Ethernet-Technologie etabliert. Dieser Aspekt wurde innerhalb der Automobilbranche erkannt, sodass vorwiegend aus Kosten bzw. Kompatibilitätsgründen eine Adaptierung sinnvoll ist [51]. Durch strenge fahrzeugspezifische Anforderungen bzgl. des Temperaturbereichs (−40 bis 85°C) oder der Zuverlässigkeit bei Beschleunigungen bis zu $4g$ ist zudem das Gewicht der Verkabelung von besonderer Relevanz [222]. Im Jahr 2011 begannen die ersten Standardisierungsaktivitäten innerhalb der gegründeten *OPEN (One Pair EtherNet) Alliance* zur Definition eines Standards für die fahrzeuginterne Kommunikation. Der Ausgangspunkt bildete dabei die einige Jahre zuvor entwickelte Ethernet-Spezifikation (BroadR-Reach) des Unternehmens Broadcom, die auf nur einer ungeschirmten, verdrillten Zweidrahtleitung basierte [223]. Ein vergleichbarer IT-Standard ist dagegen mit der doppelten Anzahl definiert [222]. Im Jahr 2015 wurde die BroadR-Reach Spezifikation in den öffentlichen Standard (Institute of Electrical and Electronics Engineers (IEEE) 802.3bw [224]) überführt, der die physikalische Schicht von Ethernet für die Anwendung in Fahrzeugnetzwerken für eine Datenrate von 100 Mbit/s spezifiziert (der Standard wird auch als *100Base-T1* bezeichnet). Es laufen daneben bereits weitere Aktivitäten für noch höhere Datenraten, ebenfalls basierend auf nur einer verdrillten Zweidrahtleitung.

Definition A.1.2 Außenwelt

Der Begriff *Außenwelt* wird in diesem Kontext als Überbegriff für Infrastrukturkomponenten wie Ampeln, andere Fahrzeuge oder Backend-Server definiert, die außerhalb der Systemgrenze des Fahrzeugs liegen und dadurch ein externes Netzwerk darstellen.

Auf den höheren Schichten zwei, drei und vier (Data Link, Network, Transport) setzt der Automotive Ethernet Standard wie die traditionelle IT auf eine Kombination aus verschiedenen Protokollen. Aktuell wird derzeit die Kombination

Ethernet mit Transmission Control Protocol (TCP)/Internet Protocol (IP) bzw. User Datagram Protocol (UDP)/IP verwendet. Der Buszugriff spielt in diesem Kontext eine untergeordnete Rolle, da Automotive Ethernet als Voll-Duplex Übertragung definiert ist. Dadurch wird ermöglicht, dass zwei Kommunikationspartner gleichzeitig Nachrichten senden und empfangen können, wodurch keine Kollisionen auftreten. Ein weiterer Grund ist der physikalische Aufbau dieser Netzwerke, die Punkt-zu-Punkt Verbindungen (Stern-Topologie) darstellen und dadurch maximal zwei Kommunikationsteilnehmer auf einer Leitung beinhalten.

Scalable service-Oriented MiddlewarE over IP (SOME/IP)

Ein aktuelles Protokoll, dass in SOA basierten Fahrzeugnetzwerken zum Einsatz kommt ist SOME/IP. Dabei umfasst die zugehörige Spezifikation nicht nur ein Kommunikationsprotokoll, sondern auch eine Middleware, die innerhalb von AUTOSAR standardisiert ist. Daneben bietet der Standard drei unterschiedliche Kommunikationsmöglichkeiten, die zwischen Clients und Server nutzbar sind.

Events: Eine Event-basierte Kommunikation bietet Clients die Möglichkeit, benötigte Services (z.B. Statusinformationen von Fahrzeugsensoren) bei einem Provider zu abonnieren. Der Provider sendet dem Abonnenten entweder die Informationen innerhalb eines definierten Zeitintervalls oder sobald eine Wertänderung auftritt.

Felder: Innerhalb von Services kann eine Definition von Feldern erfolgen, auf die Clients lesend oder schreiben über die Verwendung Getter- und Setter-Methoden zugreifen können.

Methoden: Im Rahmen von Methoden sind ausgehend von den Clients RPCs auf verfügbaren Provider ausführbar. Dabei wird grundlegend in zwei verschiedene Arten unterschieden. Die erste Variante umfasst einen entfernten Methodenaufruf bei dem der beteiligte Provider einen Rückgabewert übermittelt. Die zweite Art basiert der Aufruf auf einer fire&forget Methode, bei der nur die entfernte Funktion aufgerufen wird, aber kein Rückgabewert folgt. Im Fahrzeug kann die zweite Variante beispielsweise für Ansteuerbefehle einer Aktorik eingesetzt werden.

Transportprotokolle

Die zuvor erläuterten Netzwerktechnologien (s. Abschnitt A.1.1) in Fahrzeugen sind im ISO/OSI Modell [215] über die Schichten eins und zwei definiert. Diese enthalten hingegen keine Verfahren zur Adressierung von Nachrichten über Gateways in andere Netzwerke bzw. Versendung von zusammenhängenden Botschaften, die über die maximale Nutzdatenlänge hinaus gehen. Für Diagnoseanwendungen sind diese Funktionalitäten von besonderer Bedeutung, da einzelne Steuergeräte über *Gateways* hinweg gezielt erreichbar sein müssen. Aus diesem Grund wurden für die Protokolle CAN und FlexRay spezielle Transportprotokolle entwickelt, um die Anforderungen der Diagnose zu erfüllen. Das ISO TP für CAN basierend auf der ISO 15765-2[225] definiert insgesamt vier verschiedene Botschaftstypen, die als Transportprotokoll in CAN Botschaften eingebettet sind. In Abbildung A.3 sind die vier Botschaftstypen *Single Frame (SF)*, *FF*, *CF* und *FC* dargestellt.

Definition A.1.3 Gateway

Ein Gateway stellt in der Informatik einen aktiven Netzknoten dar, der zwei unterschiedliche Netzwerke verbindet. Dabei können die Netzwerke beispielsweise auf unterschiedlichen Technologien basieren und/oder unterschiedliche Adressierungsarten verwenden [226].

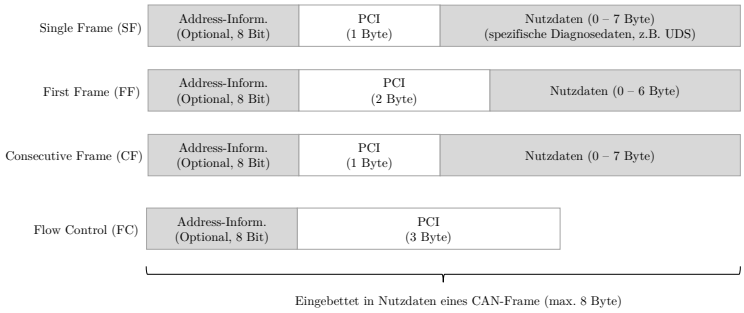


Abbildung A.3: Übersicht verschiedener ISO TP Botschaften, eingebettet in das Nutzdatenfeld einer CAN Botschaft (basierend auf [51]).

Die Adressierung erfolgt dabei über vordefinierte CAN-Identifier. Falls die Adressierung über Gateways notwendig ist, enthält das erste Byte eine eindeutige physikalische Adresse des entsprechenden Steuergerätes. Danach folgt das *Protocol Control Information (PCI)* Feld zur Kennzeichnung, wie die nachfolgenden Diagnosenutzdatenbytes *Service Data Unit (SDU)* vom Empfänger ausgewertet werden sollen. Die Verwendung der entsprechenden Botschaften ist von der zu übertragenden Nutzdatengröße abhängig. Ist diese nicht größer als sieben Bytes wird nur ein SF-Frame versendet. Überschreiten die Daten diese Größe, wird zuerst ein FF-Frame und darauffolgend mehrere CF-Frames versendet, um die gesamten Daten über segmentierte Botschaften zu übertragen. Daneben wird bei der Segmentierung im Gegensatz zur SF-Kommunikation eine Flusskontrolle mittels der FC-Frames benötigt, um zwischen Sender und Empfänger eine Blockgröße zu vereinbaren. Diese gibt an, wie viele CF-Frames hintereinander an den Empfänger gesendet werden können, bevor eine Sendepause vorgenommen wird. Neben dem ISO TP gibt es mit dem ISO 10681 [227] Standard ebenfalls ein Transportprotokoll, welches auf dem FlexRay aufsetzt. Aufgrund des ähnlichen Funktionsprinzips zu ISO TP für CAN wird an dieser Stelle für detaillierte Protokolleigenschaften auf den Standard verwiesen.

Diagnoseprotokolle

Alle Diagnoseprotokolle sind im ISO-OSI Modell auf der Anwendungsschicht (Schicht 7) definiert. Das derzeit aktuellste Diagnoseprotokoll stellt UDS dar, das auf verschiedenen Vernetzungstechnologien wie CAN, FlexRay sowie Automotive Ethernet aufsetzen kann (s. Tabelle A.2). UDS ist durch ISO 14229 [41] spezifiziert. Daneben gibt es weitere Standards die UDS für bestimmte Protokolle wie z.B. CAN (ISO 15765 [228]) und Automotive Ethernet (ISO 13400 [229]) weiter spezifizieren.

Unified Diagnostic Services

Das UDS-Diagnoseprotokoll umfasst verschiedene Felder, die in die Nutzdaten des darunterliegenden Kommunikationsprotokolls wie beispielsweise CAN oder Ethernet eingebettet werden. In Abbildung A.4 ist der Aufbau der drei möglichen UDS Frame-Arten dargestellt. Grundlegend basiert die Kommu-

Tabelle A.2: Einordnung von aktuellen automotive Netzwerkprotokollen (*Diagnoseprotokolle kursiv, Security in bold*).

OSI Layer	**Automotive Netzwerkprotokolle**					
7	*UDS, OBD*	**SecOC**	*UDS*	*UDS, OBD*	*UDS, OBD*	SOME/IP, DDS
6						
5					**TLS**	
4	*ISO TP*		*ISO TP*	*FlexRay TP*	TCP/UDP	
3					IP, **IPSec**	
2	CAN		LIN	FlexRay	Automotive Ethernet	
1						

nikation auf dem *Request/Response* Verfahren, bei dem ein Diagnosegerät entsprechende Anfragen (Service Requests) an das fahrzeuginterne Netzwerk sendet. Ein adressiertes Steuergerät beantwortet daraufhin eine empfangene Anfrage, die über die enthaltene Service Identifier (SID) sowie Subfunction Level (LEV) abgebildet werden, mit einem positiven oder negativen Antwort (Positive Response, Negative Response). Die positiven Antworten enthalten die angefragten Informationen im Feld *Response Parameters*.

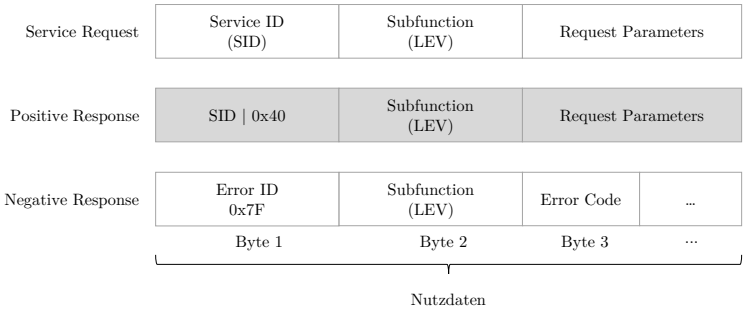


Abbildung A.4: Grundlegender Aufbau unterschiedlicher UDS-Frames (basierend auf [51]).

Der negative Fall tritt beispielsweise ein, wenn angefragte Services nicht zur Verfügung stehen oder ein Fehlerfall vorliegt. Die Art des Fehlers wird dem Diagnosegerät in der Antwort mithilfe des *Error Code* übermittelt. Darüber

hinaus sind im UDS-Standard einige Diagnosefunktionen mit eindeutigen SIDs und LEVs definiert (s. Abschnitt A.3), die eine herstellerübergreifende Gültigkeit besitzen. Des Weiteren gibt es Festlegungen für einheitliche *Error Codes*.

UDS-Diagnosesitzungen

Wird zu einem Steuergerät eine Diagnoseverbindung nach ISO 14229 [41] aufgebaut, kann dieses verschiedene Betriebszustände einnehmen, die als Diagnosesitzungen (engl. Diagnostic Sessions) bezeichnet werden. Zu Beginn einer neuen Diagnoseverbindung befindet sich jedes Steuergerät in einer *Default-Session*. Ausgehend von diesem Zustand kann darauffolgend in erweiterte *Non-Default Sessions* gewechselt werden (s. Abbildung A.5).

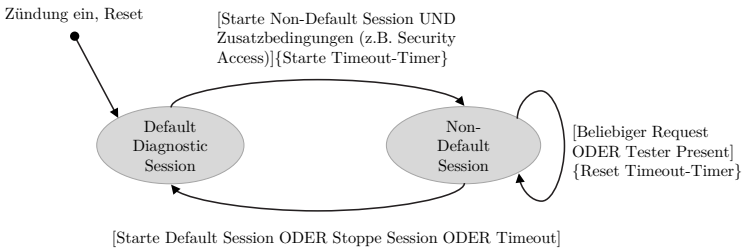


Abbildung A.5: Zustandsautomat für Diagnosesitzungen (basierend auf [51]).

Darüber hinaus definiert der Standard verschiedene Non-Default Sessions (z.B. *programmingSession*, *extendedDiagnosticSession*) mit Vorschlägen an Diagnosefunktionen, die darin integrierbar sind. Daneben können Hersteller auch individuelle Sessions implementieren. Der Zustandsübergang von einer *Default*-Session in eine *Non-Default*-Session kann aus Sicherheitsgründen an eine zusätzliche Abfrage (*Security Access*) geknüpft werden, über den sich ein Diagnosegerät gegenüber dem Fahrzeug authentifizieren muss, bevor eine Autorisierung erfolgt. In dieser Session muss zusätzlich vom Tester eine zyklische Präsenz-Botschaft (engl. Tester present) gesendet werden, damit kein automatischer Zustandswechsel in die Default Session erfolgt.

On-Board Diagnose

Der zur Überprüfung von abgasrelevanten Systemen in Fahrzeugen definierte OBD-Standard (ISO 15031) [62] legt neben der eigentlichen Diagnose auch die Spezifikationen bzgl. Schnittstelle fest, die in jedem Fahrzeug als Steckverbindung im Innenraum verbaut sein muss. Über diese Schnittstelle kann mit dem internen Netzwerk des Fahrzeugs kommuniziert werden. Die Besonderheit von OBD ist die Standardisierung von einheitlichen Servicefunktionen, die für die Diagnose der Abgassysteme erforderlich sind. Damit ist es möglich herstellerunabhängig eine Diagnose durchzuführen, ohne über spezielles Herstellerwissen zu verfügen. OBD setzt ähnlich wie UDS auf bestehende Netzwerkprotokolle wie CAN oder FlexRay inklusive benötigter Transportprotokolle auf und ist daher auf der Schicht 7 des ISO/OSI Modells definiert. Des Weiteren ist es ähnlich wie das UDS Protokoll aufgebaut. Auch OBD basiert auf einem Request/Response Verfahren und nutzt SIDs zur Auswahl von bestimmten Diagnosefunktionen. Darüber hinaus können mit Parameter Identifiers (PIDs) spezielle Messwerte angefragt werden. Eine aktuelle Weiterentwicklung des ursprünglichen OBD-Standards stellt der World Wide Harmonized On-Board-Diagnostic (WWH-OBD) [230] Standard dar, der eine weltweite Harmonisierung bzgl. der Diagnose von abgasrelevanten Systemen bei Personenkraftfahrzeugen sowie Nutzfahrzeugen sicherstellen soll.

A.1.2 Verschlüsselungsverfahren

Im Bereich der Kryptografie existieren verschiedene Verschlüsselungsverfahren, um Informationen gegenüber Dritten geheim halten zu können. Dadurch bildet die Verschlüsselung eine Schutzmaßnahme zur Gewährleistung des Sicherheitsziels *Vertraulichkeit*. Ein wichtiger Grundsatz stellt das *Kerckhoffs'sche Prinzip* von 1883 dar [231]:

Definition A.1.4 Kerckhoffs'sche Prinzip

Kerckhoffs definiert die Anforderung an ein Kryptosystem, dass dieses auch dann noch sicher sein muss, wenn der Angreifer alle Details über das zugehörige System besitzt, mit Ausnahme des zugehörigen Schlüssels [69].

Daraus folgt, dass die Sicherheit eines jeden Verschlüsselungsverfahrens alleinig durch die Geheimhaltung des Schlüssels gewährleistet sein muss, nicht dagegen auf der Geheimhaltung des genutzten Algorithmus. Darüber hinaus existieren Veröffentlichungen zu derzeit als sicher angesehenen Verfahren sowie zugehörige Schlüssellängen durch das NIST [232] oder das BSI [233].

Definition A.1.5 Kryptografie

Der Begriff Kryptografie beschreibt die Wissenschaft, die sich mit dem Verfassen von Geheimschriften befasst, um die Inhalte einer Nachricht zu verbergen [69].

Symmetrische Verfahren

Symmetrische Verschlüsselungsverfahren basieren darauf, dass jedes Subjekt denselben symmetrischen Schlüssel (engl. secret key) ($K = K_E = K_D$) für die Ver- bzw. Entschlüsselung verwendet [68]. Exemplarisch wird nachfolgend eine Sequenz erläutert, in der zwei Kommunikationsteilnehmer (Alice und Bob) eine vertrauliche Nachricht austauschen:

1. Die beiden Kommunikationsteilnehmer vereinbaren einen gemeinsamen sowie geheimen Schlüssel $K_{A,B}$.
2. Um nun einen Klartext M von Alice verschlüsselt an Bob zu senden, erstellt Alice ein Chiffre C mit der Verschlüsselungsfunktion E, die mit $K_{A,B}$ den Klartext M verschlüsselt. Dadurch gilt $C = E(M, K_{A,B})$.
3. Danach kann Bob diesen Text durch die erneute Anwendung des Schlüssels auf dem Chiffre mit der Entschlüsselungsfunktion D den Klartext erzeugen. Es gilt $M = D(C, K_{A,B})$.

Symmetrische Verfahren bieten damit eine Gewährleistung der Vertraulichkeit auf einem Informationskanal wie beispielsweise dem Internet. Dagegen unterstützen diese keinen Schutz der Authentizität bzw. Integrität der versendeten Daten.

Asymmetrische Verfahren

Im Gegensatz zu symmetrischen Verfahren nutzen asymmetrische Verschlüsselungsverfahren sogenannte Schlüsselpaare, die aus einem öffentlichen (engl. public key) und einem privaten Schlüssel (engl. private key) bestehen [68]. Dabei wird der öffentliche Schlüssel über eine Datenbank für jeden einsehbar zur Verfügung gestellt, da dieser keine Geheiminformation enthält. Hingegen muss der private Schlüssel gegenüber Dritten stets geheim gehalten werden. Ein vertraulicher Nachrichtenaustausch zwischen den Kommunikationsteilnehmern Alice und Bob gliedert sich durch die Anwendung eines asymmetrischen Verfahrens wie folgt:

1. Alice und Bob erzeugen sich jeweils die benötigten Schlüsselpaare (K_E^A, K_D^A) und (K_E^B, K_D^B). Dabei stellen K_D^A sowie K_D^B die privaten Schlüssel der Teilnehmer dar.

2. Die öffentlichen Schlüsselpaare K_E^A und K_E^B von Alice und Bob werden öffentlich bekanntgegeben (z.B. durch einen öffentlich erreichbaren Server), damit beide Partner Kenntnis darüber besitzen.

3. Alice ist nun in der Lage mithilfe des öffentlich bekannten Schlüssels von Bob (K_E^B) und dem Klartext M eine Chiffre $C = E(M, K_E^B)$ zu erstellen und an Bob zu versenden.

4. Im Anschluss kann Bob die Chiffre unter Anwendung seines privaten Schlüssels K_D^B entschlüsseln, sodass $M = D(C, K_D^B)$ gilt.

Asymmetrische Verfahren gewährleisten im Gegensatz zu symmetrischen Verfahren neben der Vertraulichkeit auch die Security-Eigenschaften Authentizität sowie Nichtabstreitbarkeit.

A.1.3 Hashfunktionen und digitale Signaturen

Zur Sicherstellung der Sicherheitseigenschaften Authentizität sowie Integrität von Nachrichten werden kryptografische Hashfunktionen eingesetzt. Darüber hinaus existieren digitale Signaturen, die es ermöglichen neben der Nachrichtenauthentizität auch die Authentizität des jeweiligen Senders sicherzustellen.

Ein Empfänger ist dadurch in der Lage, die Echtheit des angegebenen Absenders einer Nachricht zu prüfen.

Hashfunktionen

Kryptografisch sichere *Hashfunktionen* bieten die Möglichkeit, digitale Fingerabdrücke von Objekten (Nachrichten, Daten) zu erstellen [68]. Wird ein solches Objekt beispielsweise zusammen mit dessen Hashwert über ein Netzwerk versendet, kann der Empfänger die Integrität der Nachricht überprüfen, indem er ebenfalls den Hashwert über die empfangenen Informationen errechnet. Bei identischen Hashwerten kann eine Integritätsverletzung ausgeschlossen werden. Damit Hashverfahren als sicher angesehen werden können müssen diese einige mathematische Anforderungen aufweisen [68]:

- Einwegfunktion: Eine Hashfunktion H soll mit gegebenen Eingabedaten M effizient zu berechnen sein. Darüber hinaus darf es mithilfe von vertretbaren Rechenaufwand nicht möglich sein aus einem gegeben Hashwert $h = H(M)$ die Eingabedaten M über eine Umkehrfunktion zu bestimmen. Es nicht effizient möglich sein $M = H^{-1}(y)$ zu berechnen.
- Es darf außerdem nicht möglich sein einen identischen Hashwert h mit einer anderen Eingabedaten M' zu erzeugen. Dadurch darf - auf Basis von vertretbarem Rechenaufwand - $H(M') = h$ nicht gelten.

Erfüllt eine Hashfunktion diese Eigenschaften, so wird diese auch als schwach kollisionsresistente nicht injektive[1] Funktion bezeichnet [68].

Definition A.1.6 Hashfunktion

Als Hashfunktionen werden mathematische Funktionen oder Algorithmen bezeichnet, die eine beliebig lange Eingabemenge auf eine kleinere Zielmenge mit fester Breite abbilden. Das Ergebnis wird daher als Hashwert bezeichnet [234].

[1] $f : A \rightarrow B$ heißt injektiv, wenn $\forall x, y \in A$ gilt: $x \neq y \Rightarrow f(x) \neq f(y)$ [68].

Nachrichtenauthentifizierungscode

Neben der Sicherstellung der Nachrichtenintegrität ist auch die Gewährleistung der Nachrichtenauthentizität notwendig, um sicherzustellen, dass die übertragenen Informationen von einer authentischen Quelle stammen. Diese Anforderung kann eine reine Hashfunktion nicht erfüllen, da diese Funktionen öffentlich bekannt sind und ein Angreifer ebenfalls in der Lage wäre einen entsprechenden Hashwert einer Nachricht zu berechnen. Um auch die Nachrichtenauthentizität sicherzustellen, werden sogenannte Nachrichtenauthentifizierungscodes (engl. Message Authentication Code (MAC)) verwendet, die neben dem eigentlichen Hashalgorithmus zusätzlich für die Berechnung ein gemeinsames Geheimnis (Schlüssel) verwenden [68]. Dadurch sind nur Kommunikationsteilnehmer mit gültigem Schlüssel in der Lage, authentifizierte Nachrichten zu senden bzw. empfangene Nachrichten auf deren Authentizität zu prüfen. Da bei diesem Verfahren alle legitimierten Teilnehmer denselben Schlüssel verwenden, ist eine Senderauthentifizierung nicht möglich [69].

Digitale Signatur

Die digitale Signatur wird dazu verwendet, um Subjekten einen digitalen Identitätsnachweis zu ermöglichen, der gleichzeitig von anderen Subjekten auf Gültigkeit überprüft werden kann [68], [23]. Damit erweitern digitale Signaturen die Sicherheitseigenschaften der Nachrichtenauthentizität bzw. Integrität, um die Eigenschaft der Nicht-Abstreitbarkeit. Mit dem Einsatz von digitalen Signaturen ist es eindeutig möglich, den Ursprung einer Nachricht festzustellen [69]. Eine mit digitaler Signatur versehene Nachricht, kann ausschließlich der Urheber dieser Nachricht erstellen. Dagegen können n weitere Instanzen diese auf ihre Echtheit überprüfen. Des Weiteren variiert der Wert der digitalen Unterschrift, da diese auf Basis der zu signierenden Daten erstellt wird. Wird dagegen der juristische Unterschied betrachtet, gibt die beteiligte Person bei einer handschriftlichen Unterschrift eine Willenserklärung ab. Hingegen erstellt bei der digitalen Signatur stellvertretend das beteiligte System die Unterschrift. Im schlimmsten Fall könnte es passieren, dass durch Schadsoftware bzw. Angriff auf die IT-Sicherheit eine nicht beabsichtigte Willenserklärung vorgenommen wird [234].

Über gesetzliche Bestimmungen wie beispielsweise von der EU veröffentlichten eIDAS [235] (engl. electronic IDentification, Authentication and trust Services) müssen eingesetzte digitale Signaturen auf asymmetrischen Verfahren basieren [236]. Aus diesem Grund wird nachfolgend nur auf dieses näher eingegangen (s. Abbildung A.6). Zu Beginn erstellen die Entitäten Alice und Bob jeweils ein Schlüsselpaar. Für Alice wäre darin ein öffentlicher Schlüssel V_A sowie ein privater Schlüssel S_A enthalten. Beide hinterlegen die öffentlichen Schlüssel (V_A, V_B) in einer Datenbank. Signiert nun Alice eine Nachricht M durch Verschlüsseln mithilfe des eigenen privaten Schlüssels gemäß $C(M, S_A) = sig$ kann Bob die Nachricht nach dem Empfang durch die Nutzung des öffentlichen Schlüssels V_A aus der Datenbank durch $sig(M) = E(sig, V_A)$ verifizieren.

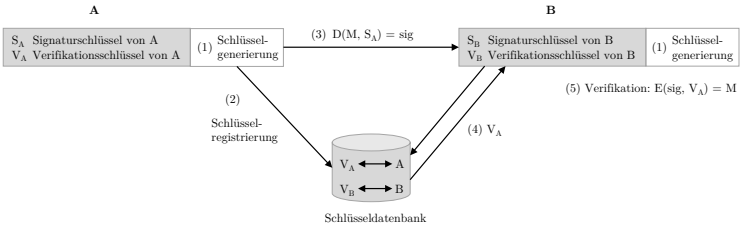


Abbildung A.6: Übersicht an Komponenten zum Erzeugen und Überprüfen einer digitalen Signatur basierend auf einem asymmetrischen Verfahren (basierend auf [68]).

Public Key Infrastruktur

Der Schlüsselaustausch bei asymmetrischen Verfahren kann über eine Infrastruktur für öffentliche Schlüssel (engl. PKI) erfolgen, die aufgrund der hohen Verbreitung [237] nachfolgend erläutert wird. Innerhalb dieser Umgebung wird mithilfe von Zertifikaten eine feste Zuordnung zwischen einem öffentlichen Schlüssel und einer Person oder Einrichtung digital bescheinigt [68]. Die Zertifikate enthalten beispielsweise Informationen über den Zertifikatsaussteller, die Gültigkeitsdauer, den öffentlichen Schlüssel sowie den Namen des Teilnehmers. Die Ausgabe der Zertifikate erfolgt über eine Zertifizierungsstelle (CA) (engl. Certification Authority) nachdem sich ein Teilnehmer davor eindeutig identifiziert hat. Das entsprechende Zertifikat wird mithilfe des pri-

vaten Schlüssels der CA signiert [23]. Damit ist jeder Teilnehmer in der Lage die Gültigkeit eines Zertifikats von einem anderen Teilnehmer mit dem öffentlich bekannten Schlüssel der zugehörigen CA zu verifizieren. Außerdem ist er in der Lage, verschlüsselte Nachrichten mithilfe des Zertifikats eines anderen Teilnehmers an dessen Inhaber zu versenden.

Dadurch wird über die beteiligte CA ein indirektes Vertrauen zwischen den Teilnehmern geschaffen. Zudem ist eine CA jederzeit in der Lage, bereits ausgegebene Zertifikate auch vor Ablauf der Gültigkeitsdauer als ungültig zu bezeichnen, falls ein Teilnehmer nicht mehr als vertrauenswürdig angesehen werden kann.

A.1.4 Arten von Firewalls

Paketfilter

Klassische Firewall-Paketfilter basieren auf den Schichten drei und vier (Netzwerk- und Transportschicht) des ISO/OSI Modells (s. Abbildung A.7). Dadurch sind die Filter in der Lage, auf Protokollinformationen dieser Schichten zuzugreifen, um eine definierte Sicherheitsstrategie umzusetzen. Da in der klassischen IT in privaten Netzwerken sowie dem Internet die Protokollkombinationen TCP/IP und UDP/IP zum Einsatz kommen, werden auf Basis von Protokollinformationen (Sender- und Empfänger IP-Adressen) den Filtern abzuarbeitende Regelwerke hinterlegt. Grundlegend kann dabei in statische und dynamische Filter unterschiedenen werden.

Statische Filter

Die Klasse der statischen Filter arbeitet auf Basis einer vordefinierten Filterliste, die verschiedene Einträge zu Protokolleigenschaften enthalten kann [68]. In Tabelle A.3 ist exemplarisch eine mögliches Regelwerk dargestellt. Trifft ein Datenpaket an der Firewall ein, werden die aktuellen Paketinformation wie Absender- und Ziel IP-Adresse sowie der Port mit den Einträgen in der Tabelle verglichen. Dabei arbeitet der implementierte Algorithmus die Einträge nacheinander aufsteigend ab. Gibt es eine Übereinstimmung, so werden die hinterlegten Aktionen (*accept, deny*) des Eintrags von der Firewall durch-

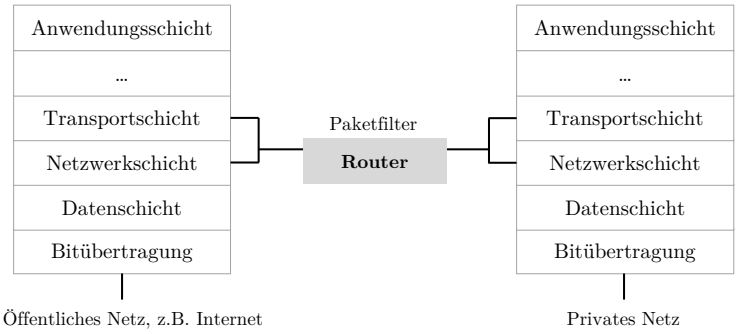


Abbildung A.7: Einordnung eines Firewall-Paketfilters in das ISO/OSI Modell [68].

gesetzt [238]. Dabei können verschiedene Arten der Einträge existieren. Zum einen ist es möglich einzelne Adressen zu definieren oder in Form eines Adressintervalls (mit * bezeichnet) zu hinterlegen. Beginnend mit Regel 1 würde gelten, dass die Quelladresse 140.192.37.20 von jedem Quellport mit beliebiger Zieladresse und Zielport 80 blockiert wird. Hingegen würde bei Regel 3 jedes Paket von beliebiger Quelladresse und beliebigem Quellport mit Zieladresse 140.192.37.40 und Zielport 80 akzeptiert.

Tabelle A.3: Auszug einer möglichen Filtertabelle für statische Paketfilter in Firewalls [238].

		Quelle		Ziel		
Nummer	Protokoll	Adresse	Port	Adresse	Port	Aktion
1:	tcp	140.192.37.20	any	*.*.*.*	80	deny
2:	tcp	140.192.37.*	any	*.*.*.*	80	accept
3:	tcp	*.*.*.*	any	140.192.37.40	80	accept
4:	tcp	140.192.37.*	any	140.192.37.40	80	deny
5:	tcp	140.192.37.30	any	*.*.*.*	21	deny

Darüber hinaus existieren noch weitere Optionen, die bei einer Übereinstimmung angewendet werden können. Für eine spätere Weiterverarbeitung ist es sinnvoll, bei einem blockierten Paket einen Eintrag in einem Log-File zu erstellen. Über Log-Files können wiederum umfassende Analysen über mögliche Angriffe durchgeführt werden, um die Sicherheitsstrategie kontinuierlich anpassen zu können. In der dargestellten Tabelle sind neben den Absender- und

Ziel IP-Adressen auch dazugehörige Portnummern hinterlegt, die im ISO/OSI Modell der Transportschicht zugeordnet werden. Diese Informationen erlauben es, nicht nur zwischen Absender- und Empfängersystemen zu unterscheiden, sondern zusätzlich mit Hilfe der Ports auch die auf dem System laufenden Dienste mit einzubeziehen. Auch hier gibt es die Möglichkeit, entweder einen spezifischen Port zu hinterlegen oder mit *any* alle Ports zu adressieren.

Dynamische Filter

Die Weiterentwicklung der Paketfilter brachte die Einführung von dynamischen Filtern [68]. Dadurch sind sie in der Lage, bereits analysierte Paketinformationen zu speichern, um diese in zukünftige Filterentscheidungen mit einbeziehen zu können. Diese Art wird auch als zustandsbehafteter Filter bezeichnet, da Entscheidungen von einem vergangenen Verhalten abhängen. Dies bringt gerade bei zustandslosen Netzwerkprotokollen wie UDP den Vorteil, dass die Firewall bei einer Client-Server Anfrage erkennen kann, ob es sich wirklich um das erwartete Antwortpaket für den zuvor angefragten Client handelt. In Abbildung A.8 ist eine beispielhafte Client-Server Anfrage dargestellt, die über einen dynamischen Paketfilter verläuft. Der Client stellt darin dem Server zu Beginn eine Anfrage. Die Header-Informationen des Paketes (Sender-IP, Empfänger-IP, Port) werden vom Paketfilter gespeichert. Der Server antwortet nun mit einem passenden UDP-Paket. Der Filter prüft nun, ob die Antwort zu der zuvor gestellten Anfrage passt. Eine ungültige Antwort (falscher Empfänger Port) wird vom Filter sofort verworfen [68].

Proxy-Firewall

Ein weitere Firewall-Variante stellen Proxy-Firewalls dar, die ebenfalls auf der Transportebene des ISO/OSI Modells arbeiten. Die Firewall arbeitet dabei als Vermittler zwischen zwei Netzen und stellt nur eine bestimmte Menge an Diensten bereit, die ein Client beispielsweise aus einem privaten Netzwerk gegenüber einem anderen Netz (z.B. Internet) nutzen kann. Dadurch kann die nutzbare Menge an Diensten für einen Client in der Firewall exakt definiert werden. Zudem können im Vergleich zu Paketfiltern spezielle Kontextinformationen in die Regeln mit aufgenommen werden [68], [23]. Es ist dadurch

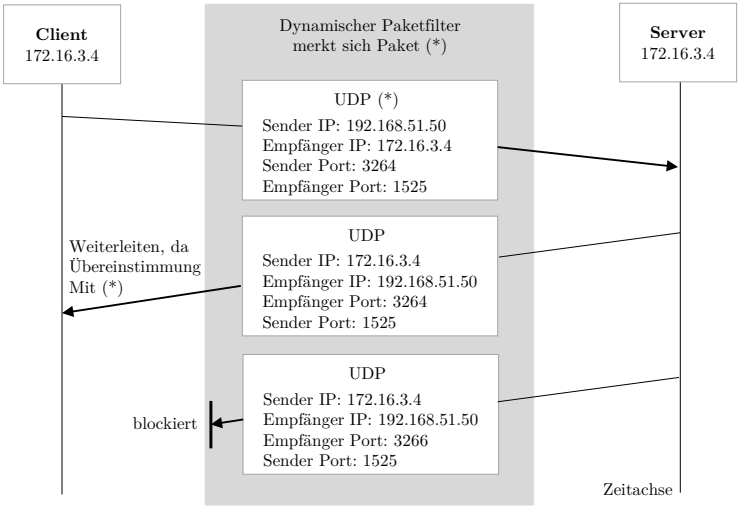


Abbildung A.8: Funktionsweise eines dynamischen Paketfilters auf Basis einer Client-Server Anfrage mit UDP-Protokoll [68].

möglich, die Anzahl an gleichzeitigen Verbindungen zu beschränken oder eine zeitliche Grenze vorzugeben, um wesentlich feingranularer zu filtern.

Applikationsfilter

Auf der obersten Ebene des ISO/OSI Modells sind die Applikationsfilter spezifiziert. Diese Art erweitern die Filtermöglichkeiten der Proxy-Firewall, indem diese exakte Informationen über die verwendeten Anwendungen besitzen. Dadurch sind diese Filter in der Lage anwendungsspezifische Nutzdaten zu analysieren, da für jede Anwendung ein dedizierter Proxy innerhalb der Firewall bereitgestellt wird [68]. Des Weiteren kann eine Authentifizierung zwischen dem Benutzer (Client) sowie dem Proxy erfolgen, um beispielsweise *Spoofing-Angriffe* zu erschweren [23].

A.1.5 Ausgewählte Maßnahmen zur Absicherung von Fahrzeugnetzwerken

Sichere On-Board Kommunikation

Zur Absicherung der Kommunikation in signal-orientierten Netzwerken hat das AUTOSAR Classic Gremium das SecOC-Modul [239] spezifiziert. Dieses ermöglicht die Gewährleistung der Security-Eigenschaften (s. Abschnitt 2.2.1) Authentizität sowie Integrität von übertragenen Nachrichten. Zusätzlich wird ein Zähler definiert, der jedem Frame einen Wert hinzufügt, mit dem der Empfänger die Aktualität (engl. freshness) der Botschaft prüfen kann. Dadurch wird ein Wiedereinspielen von aufgezeichneten Nachrichten verhindert (s. Definition A.1.8). Der Standard verwendet dafür Nachrichtenauthentifizierungscodes (s. Abschnitt 3.1.3), um die Header- bzw. Nutzdaten der protokollspezifischen *PDUs* abzusichern (s. Abbildung A.9). Der Standard umfasst zwei verschiedene Varianten der Absicherung (mit und ohne Header-Daten).

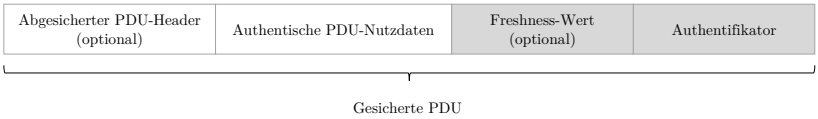


Abbildung A.9: Aufbau einer abgesicherten PDU auf Basis von SecOC (adaptiert von [239]).

Definition A.1.7 Protocol Data Unit

Als PDU wird eine Dateneinheit definiert, die alle Informationen eines bestimmten Kommunikationsprotokolls enthalten. Ausgehend von einer Datenquelle, bei der eine PDU die reinen Nutzdaten umfasst kommen je nach eingesetztem Kommunikationsprotokoll verschiedene Header-Daten auf den unterschiedlichen Schichten des ISO/OSI Modells hinzu.

Definition A.1.8 Angriff durch Wiedereinspielung

Beschreibt die Möglichkeit eines Angreifers, bestimmte Teile des Netzwerkverkehrs aufzuzeichnen und dem Empfänger in gleicher Weise oder mit geringfügigen Änderungen wiedereinzuspielen (engl. replay-attack), um eine bestehende Authentifizierung zu umgehen bzw. eine andere Identität vorzutäuschen [240]. Diese Art von Angriff wird dem Angriffstyp *Spoofing* (s. Abschnitt 2.2.1) zugeordnet.

Beschreibt die Möglichkeit eines Angreifers, bestimmte Teile des Netzwerkverkehrs aufzuzeichnen und dem Empfänger in gleicher Weise oder mit geringfügigen Änderungen wiedereinzuspielen (engl. replay-attack), um eine bestehende Authentifizierung zu umgehen bzw. eine andere Identität vorzutäuschen.

Security-Access

Der Security Access ist ein Authentifizierungs- und Autorisierungsverfahren für die Diagnose, das auf dem Challenge-Response-Prinzip basiert und u.a. im UDS-Protokoll (s. Abschnitt A.1.1) als Diagnosedienst spezifiziert ist. Der Tester sendet hierzu eine Anfrage an ein Steuergerät (s. Abbildung A.10).

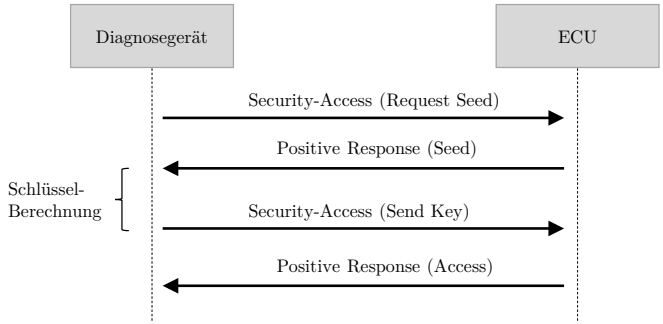


Abbildung A.10: Ablauf einer Diagnose-Authentifizierungssequenz auf Basis eines Security-Access.

Als Antwort erhält der Tester einen Initialisierungswert (Seed) und berechnet durch einen geheimen Algorithmus einen Schlüssel. Diesen Schlüssel (Key) sendet das Diagnosegerät anschließend an das Steuergerät, das die korrekte Berechnung und damit die Gültigkeit der Authentifizierung prüft und anschließend eine Menge an Berechtigungen vergibt [41].

A.2 A-ABAC Framework

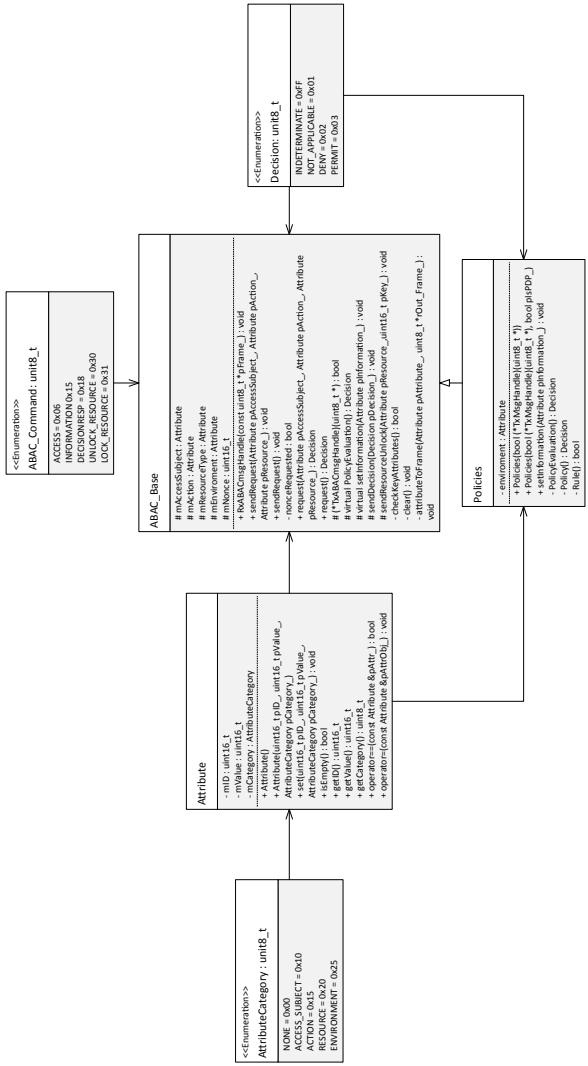


Abbildung A.11: ABAC UML Framework

A.2.1 Zugriffs-Policies

Anwendungsfall 1

Auflistung A.1: Zugriffsrichtlinie (Policy) zur Definition der Zugriffsberechtigungen von Anwendungsfall 1.

```
1
2  policy AccessDiagnosisData {
3  target clause
4  attribute.role == "engineer"
5  or  attribute.role == "technician"
6  or  attribute.role == "mechanic"
7  apply firstApplicable
8  ReadDiagnosis
9  }
10
11 rule ReadDiagnosis {
12 permit target clause
13 attribute.action         == "read"
14 and attribute.status     == "connected"
15 and attribute.status     == "vehicle at standstill"
16 and attribute.resourceID == "diagnosticsData"
17 }
```

Anwendungsfall 2

Auflistung A.2: Zugriffsrichtlinie (Policy) zur Definition der Zugriffsberechtigungen von Anwendungsfall 2.

```
1
2  policy AccessDiagnosisData {
3  target clause
4  attribute.role == "engineer"
5  or  attribute.role == "technician"
6  or  attribute.role == "mechanic"
7  apply firstApplicable
8  ReadDiagnosis
9  }
10
11 rule ReadDiagnosis {
12
13 permit target clause
14 attribute.resourceID == "diagnosticsData"
15 condition attribute.incoming_msg_time > 2 (in ms)
16 and attribute.status      == "connected"
17 and attribute.status      == "vehicle at standstill"
18 }
```

Anwendungsfall 3

Auflistung A.3: Zugriffsrichtlinie (Policy) zur Definition der Zugriffsberechtigungen von Anwendungsfall 3.

```
1
2  policy AccessDiagnosisData {
3  target clause
4  attribute.role == "engineer"
5  or  attribute.role == "technician"
6  or  attribute.role == "mechanic"
7  apply firstApplicable
8  Execute EOL
9  }
10
11 rule ExecuteEOL {
12 permit target clause
13 attribute.action         == "execute"
14 and attribute.resourceID == "EOLfunction"
15 condition
16 and attribute.seat_occupancy == "false"
17 and attribute.status     == "vehicle at standstill"
18 and attribute.belt_buckle == "false"
19 }
```

A.3 Übersicht UDS-Diagnoseservices

SID	Service Beschreibung
0x10	Diagnostic Session Control
0x11	ECU Reset
0x14	Clear Diagnostic Information
0x19	Read DTC Information
0x22	Read Data By Identifier
0x23	Read Memory By Address
0x27	Security Access
0x28	Communication Control
0x2A	Read Data by Periodic ID
0x2E	Write Data By Identifier
0x2F	Input Output Control By Identifier
0x31	Routine Control
0x34	Request Download
0x35	Request Upload
0x36	Transfer Data
0x37	Transfer Exit
0x3D	Write Memory By Address
0x3E	Tester Present
0x85	Control DTC Setting

A.4 Aufgezeichnete Diagnosedaten (Auszüge)

A.4.1 Fehlersuche - Motor ECU

Nr.	Zeitstempel	Protokoll	DLC	CAN-ID	CAN Nutzdaten
1	23.57877085	CAN	32	0x6F1	DF0322F150000000
2	26.04472759	CAN	32	0x6F1	DF0319020C000000
3	26.13503172	CAN	32	0x6F1	5630100500000000
4	26.2212738	CAN	32	0x6F1	630100500000000
5	26.30592345	CAN	32	0x6F1	1030100500000000
6	26.3919845	CAN	32	0x6F1	3730100500000000

7	26.48002954	CAN	32	0x6F1	6330100500000000
8	26.56631741	CAN	32	0x6F1	5D30100500000000
9	26.65661156	CAN	32	0x6F1	7830100500000000
10	26.75502188	CAN	32	0x6F1	6730100500000000
11	26.84539635	CAN	32	0x6F1	6130100500000000
12	26.93362701	CAN	32	0x6F1	6030100500000000
13	27.04792703	CAN	32	0x6F1	730100500000000
14	27.14836093	CAN	32	0x6F1	2230100500000000
15	27.27289615	CAN	32	0x6F1	2C30100500000000
16	27.36521337	CAN	32	0x6F1	1430100500000000
17	27.47142576	CAN	32	0x6F1	4030100500000000
18	30.31092744	CAN	32	0x6F1	EF041802FFFF0000
19	31.09412497	CAN	32	0x6F1	EF041802FFFF0000
20	31.87637928	CAN	32	0x6F1	EF041802FFFF0000
21	34.21702613	CAN	32	0x6F1	DF0322F150000000
22	68.53850759	CAN	32	0x6F1	DF0319020C000000
23	68.63171206	CAN	32	0x6F1	5630100500000000
24	68.71820411	CAN	32	0x6F1	630100500000000
25	68.80616228	CAN	32	0x6F1	6330100500000000
26	68.8908507	CAN	32	0x6F1	3730100500000000
27	68.97881787	CAN	32	0x6F1	5D30100500000000
28	69.07146315	CAN	32	0x6F1	1030100500000000
29	69.15725683	CAN	32	0x6F1	6730100500000000
30	69.24990137	CAN	32	0x6F1	6130100500000000
31	69.33618182	CAN	32	0x6F1	6030100500000000
32	69.4503339	CAN	32	0x6F1	7830100500000000
33	69.54891909	CAN	32	0x6F1	2230100500000000
34	69.67717465	CAN	32	0x6F1	2C30100500000000
35	69.7695875	CAN	32	0x6F1	1430100500000000
36	69.8757691	CAN	32	0x6F1	4030100500000000
37	72.70932688	CAN	32	0x6F1	EF041802FFFF0000
38	73.4924686	CAN	32	0x6F1	EF041802FFFF0000
39	74.27471317	CAN	32	0x6F1	EF041802FFFF0000
40	76.584007	CAN	32	0x6F1	DF0322F150000000
41	86.49744544	CAN	32	0x6F1	120322F150000000
42	88.21328278	CAN	32	0x6F1	120322F150000000
43	89.52793394	CAN	32	0x6F1	1201200000000000

44	95.09269917	CAN	32	0x6F1	120322F150000000
45	95.24614352	CAN	32	0x6F1	120322F150000000
46	96.6871934	CAN	32	0x6F1	120319020C000000
47	96.87873961	CAN	32	0x6F1	1201200000000000
48	103.1110075	CAN	32	0x6F1	120322F150000000
49	103.2443586	CAN	32	0x6F1	120322F150000000
50	119.2492859	CAN	32	0x6F1	120319020C000000
51	126.1967834	CAN	32	0x6F1	120414FFFFFF0000
52	127.4212397	CAN	32	0x6F1	120319020C000000

A.4.2 Fehlersuche - Kombiinstrument

Nr.	Zeitstempel	Protokoll	DLC	CAN-ID	CAN Nutzdaten
1	19.95537245	CAN	8	0x6F1	6002100300000000
2	20.08772344	CAN	8	0x6F1	600322F150000000
3	26.38137146	CAN	8	0x6F1	600322DA00000000
4	26.52383832	CAN	8	0x6F1	600322DA0A000000
5	26.65916755	CAN	8	0x6F1	600322DA0F000000
6	26.80958511	CAN	8	0x6F1	600322DA47000000
7	26.94792593	CAN	8	0x6F1	600322DA47000000
8	50.4389061	CAN	8	0x6F1	6002100300000000
9	50.61840324	CAN	8	0x6F1	600431010F040000

A.5 Untersuchungsergebnisse verschiedener N-Gramme

A.5.1 Test-Szenario 2

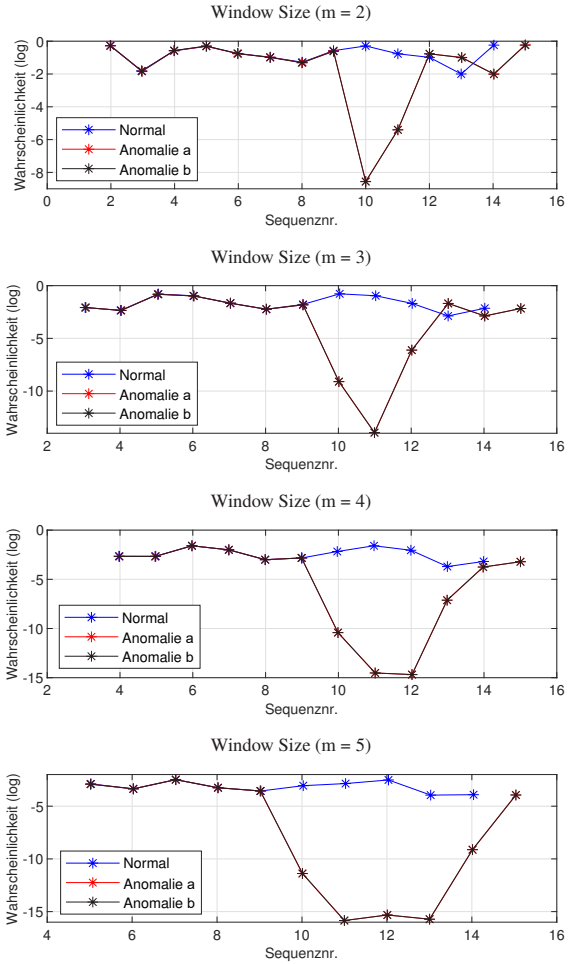


Abbildung A.12: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der Sliding Window Größe m für das 2. Szenario.

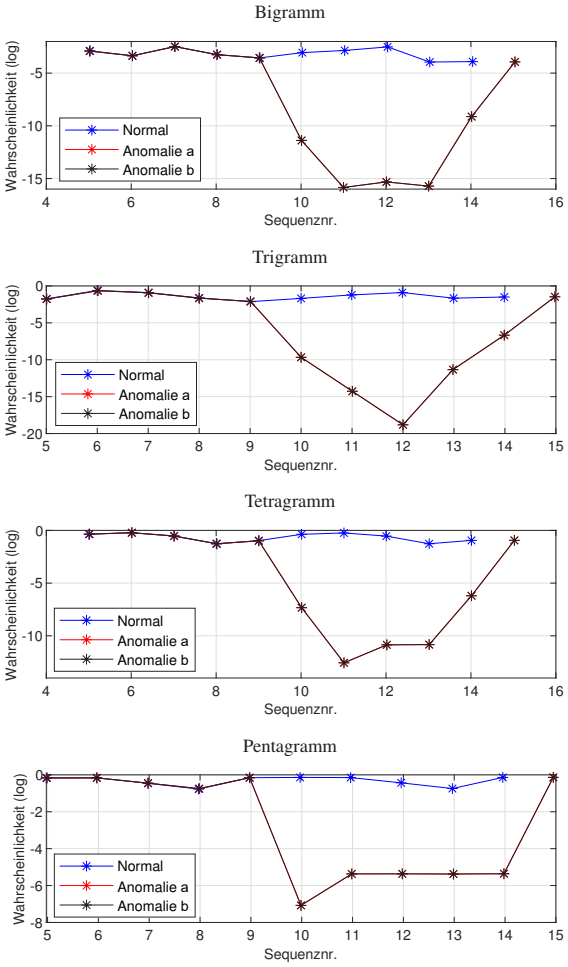


Abbildung A.13: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der n-Gramme für das 2. Szenario.

A.5.2 Test-Szenario 3

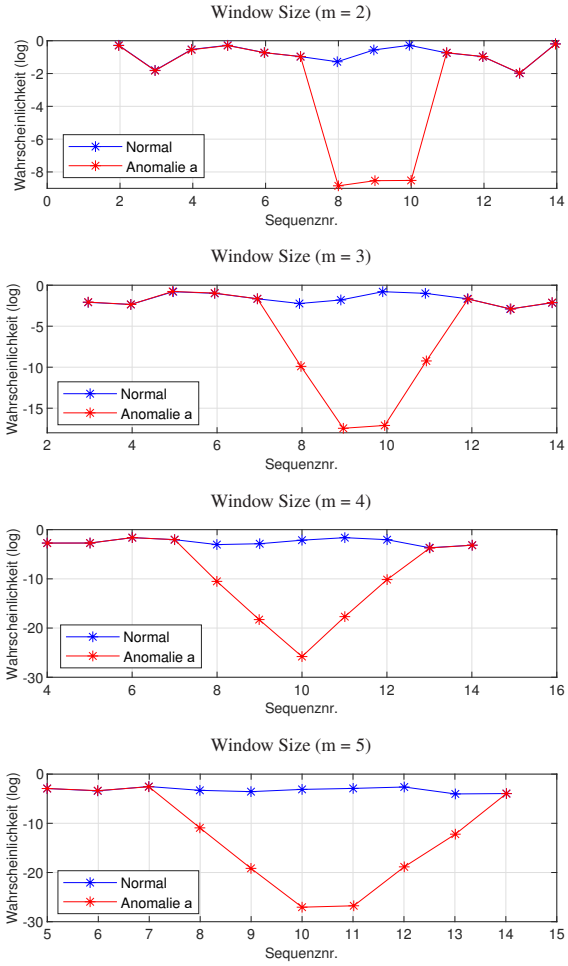


Abbildung A.14: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der Sliding Window Größe *m* für das 3. Szenario.

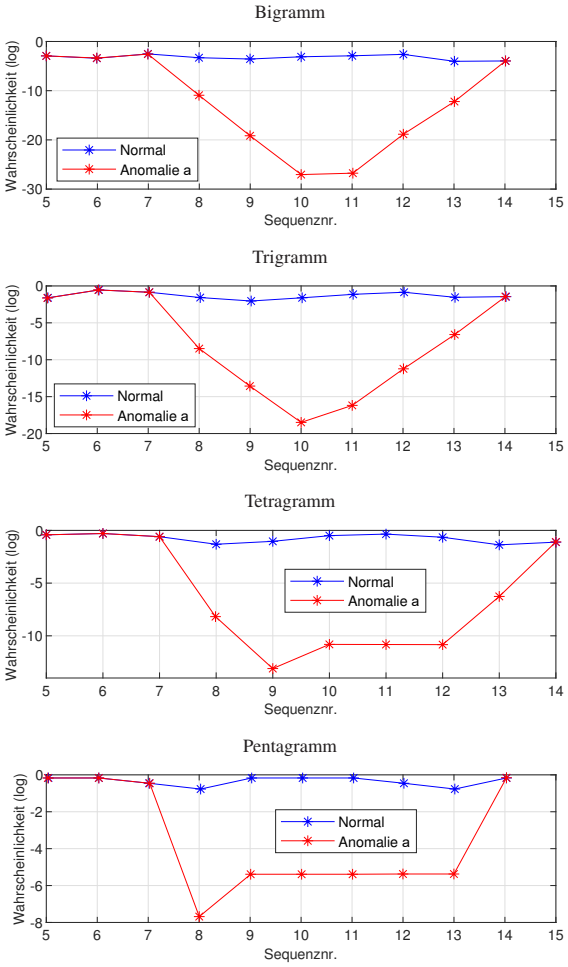


Abbildung A.15: Verläufe von berechneten Wahrscheinlichkeiten durch Variation der n-Gramme für das 3. Szenario.

Literaturnachweise

[1] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Die Lage der IT-Sicherheit in Deutschland 2020*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2020.pdf;jsessionid=7DED17ABD94C6FA12B65CCB7C59526E3.internet472?__blob=publicationFile&v=1, Abruf: 20.10.2022

[2] Wilkens, Andreas: *220 Milliarden Euro Schaden durch Ransomware und andere Cyber-Angriffe*. https://www.heise.de/news/220-Milliarden-Euro-Schaden-durch-Ransomware-und-andere-Cyber-Angriffe-6156111.html, Abruf: 20.10.2022

[3] Schmoll-Trautmann, Anja: *Studie: IT-Security in Deutschland 2018*. https://www.zdnet.de/88338859/studie-it-security-in-deutschland-2018/, Abruf: 14.05.2021

[4] Donath, Andreas: *Supermarktkette muss wegen Cyberangriff Läden schließen*. https://www.golem.de/news/kaseya-supermarktkette-muss-wegen-cyberangriff-800-laeden-schliessen-2107-157859.html, Abruf: 20.10.2021

[5] Donath, Andreas: *Cyberangriff sorgt für Katastrophenfall in Anhalt-Bitterfeld*. https://www.golem.de/news/malware-cyberangriff-sorgt-fuer-katastrophenfall-in-anhalt-bitterfeld-2107-158042.html, Abruf: 20.10.2021

[6] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Definition "Ransomware"*. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Gefaehrdungen/Fortschrittliche-Angriffe/Fortschrittliche-Angriffe_node.html, Abruf: 10.08.2022

[7] pwc and strategy& (Hrsg.): *The 2017 Strategy & Digital Auto Report: Fast and furious: Why making money in the "roboconomy" is getting*

harder. https://www.strategyand.pwc.com/media/file/2017-Strategyand-Digital-Auto-Report.pdf, Abruf: 11.12.2018

[8] Handelsblatt: *Selbstfahrende Autos - Das Silicon Valley schaltet einen Gang hoch*. https://www.handelsblatt.com/unternehmen/it-medien/selbstfahrende-autos-das-silicon-valley-schaltet-einen-gang-hoch/20345528.html, Abruf: 15.05.2021

[9] Handelsblatt: *Warum Daimler und BMW jetzt auf Kooperation setzen*. https://www.handelsblatt.com/unternehmen/industrie/autokonzerne-warum-daimler-und-bmw-jetzt-auf-kooperation-setzen/24373450.html, Abruf: 15.05.2021

[10] Ebert, Christof ; Favaro, John: Automotive Software. In: *IEEE Software* (2017), Nr. 3, S. 33–39

[11] Europäische Union (Hrsg.): *VERORDNUNG (EU) 2015/758, E-Call*. https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32015R0758&from=EN, Abruf: 14.04.2021

[12] Helmling, Markus: Service-orientierte Architekturen und Ethernet im Fahrzeug: Auf dem Weg zum fahrenden Rechenzentrum. In: *Elektronik automotive* Sonderausgabe Ethernet (2017), S. 18–21

[13] M. Wille ; U. Kleine: *Volkswagen goes Adaptive (Video incl. Slideset): Adaptive AUTOSAR as SW Framework for the new electric vehicle platform*. https://youtu.be/NxFflt_48yA, Abruf: 15.05.2021

[14] Tischer, Mirko: AUTOSAR Adaptive - Das Rechenzentrum im Fahrzeug. In: *Elektronik automotive Bordnetz* 2018 (2018), Nr. September, S. 30–33

[15] Schweiker, Markus ; Huck, Thorsten: CHASSIS ARCHITECTURES – Partitioning of cross-domain functions. Version: 2017. http://dx.doi.org/10.1007/978-3-658-14219-3{_}26. In: Pfeffer, Prof. Peter E. (Hrsg.): *7th International Munich Chassis Symposium 2016*. Wiesbaden : Springer Fachmedien Wiesbaden, 2017 (Proceedings). – DOI 10.1007/978–3–658–14219–3_26. – ISBN 978–3–658–14218–6, S. 367–379

[16] Traub, Matthias ; Maier, Alexander ; Barbehön, Kai L.: Future Automotive Architecture and the Impact of IT Trends. In: *IEEE Software* 34 (2017), Nr. 3, S. 27–32

[17] Maul, Mario ; Becker, Gerhard ; Bernhard, Ulrich: Serviceorientierte EE-Zonenarchitektur Schlüsselelement für neue Marktsegmente. In: *ATZelektronik* 13 (2018), Nr. 1, S. 36–41

[18] Burkacky, Ondrej ; Deichmann, Johannes ; Stein, Jan P.: *Automotive software and electronics 2030*. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/mapping-the-automotive-software-and-electronics-landscape-through-2030, Abruf: 07.04.2021

[19] UNECE: *United Nations Economic Commission for Europe (UNECE)*. https://www.unece.org/info/ece-homepage.html, Abruf: 16.04.2021

[20] Burkacky, Ondrej ; Deichmann, Johannes ; Klein, Benjamin ; Pototzky, Klaus ; Scherf, Gundbert ; McKinsey & Company (Hrsg.): *Cybersecurity in automotive*. https://www.mckinsey.com/~/media/mckinsey/industries/automotive%20and%20assembly/our%20insights/cybersecurity%20in%20automotive%20mastering%20the%20challenge/cybersecurity-in-automotive-mastering-the-challenge.pdf, Abruf: 14.04.2021

[21] IEC/TR 61508-0:2005-10: *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 0: Functional safety and IEC 61508*. 2005

[22] ISO: *ISO 26262 – Road Vehicles – Functional Safety*. 2011

[23] Harris, S. ; Maymi, F.: *CISSP All-in-One Exam Guide*. New York: McGraw-Hill Education, 2016

[24] Wolf, Marko: *Security engineering for vehicular IT systems: Improving the trustworthiness and dependability of automotive IT applications: Zugl.: Bochum, Univ., Diss., 2008*. 1. ed. Wiesbaden : Vieweg + Teubner, 2009. – ISBN 978–3–8348–0795–3

[25] Dürrwang, Jürgen ; Beckers, Kristian ; Kriesten, Reiner: A Lightweight Threat Analysis Approach Intertwining Safety and Security for the Automotive Domain. In: *International Conference on Computer Safety, Reliability, and Security*, 2017, 305–319

[26] Koscher, Karl ; Czeskis, Alexei ; Roesner, Franziska ; Patel, Shwetak ; Kohno, Tadayoshi ; Checkoway, Stephen ; McCoy, Damon ; Kantor, Brian ; Anderson, Danny ; Shacham, Hovav ; Savage, Stefan: Ex-

perimental Security Analysis of a Modern Automobile. In: *2010 IEEE Symposium on Security and Privacy*, 2010, S. 447–462

[27] Checkoway, Stephen ; McCoy, Damon ; Kantor, Brian ; Anderson, Danny ; Shacham, Hovav ; Savage, Stefan ; Koscher, Karl ; Czeskis, Alexei ; Roesner, Franziska ; Kohno, Tadayoshi u. a.: Comprehensive Experimental Analyses of Automotive Attack Surfaces. In: *USENIX Security Symposium*, 2011

[28] Miller, Charlie ; Valasek, Chris: Remote exploitation of an unaltered passenger vehicle. In: *Black Hat USA* 2015 (2015)

[29] Keen Security Lab: *Experimental Security Assessment of BMW Cars: A Summary Report*. https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf, Abruf: 25.04.2022

[30] Friedhelm Greis: *Wenn das Auto seinen Fahrer erpresst*. https://www.golem.de/news/fahrzeugsicherheit-wenn-das-auto-seinen-fahrer-erpresst-1710-130732.html, Abruf: 15.05.2021

[31] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Branchenlagebild Automotive: Cyber-Sicherheit in der Automobilbranche*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Branchenlagebild/branchenlagebild-automotive.pdf;jsessionid=89BE7B1ED41B38655955263DF0130295.internet082?__blob=publicationFile&v=7, Abruf: 20.02.2022

[32] Zürcher Hochschule für Angewandte Wissenschaften (Hrsg.): *Von der Mechatronik zu Cyber-physikalischen Systemen*. https://blog.zhaw.ch/industrie4null/2017/02/06/von-der-mechatronik-zu-cyber-physikalischen-systemen/, Abruf: 15.11.2021

[33] Broy, Manfred: *Cyber-physical systems: Innovation durch softwareintensive eingebettete Systeme*. Springer-Verlag, 2011

[34] Sommer, Florian ; Dürrwang, Jürgen ; Kriesten, Reiner: Survey and Classification of Automotive Security Attacks. In: *Information* 10 (2019), Nr. 4, S. 148. – ISSN 2078–2489

[35] AUTOSAR: *AUTOSAR 4.3.1 – Requirements on Crypto Stack*. 2017

[36] SAE J3061:2016-01: *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*. 2016

[37] ISO/SAE 21434:2021: *Road vehicles — Cybersecurity engineering*. 2021

[38] Tsolkas, Alexander ; Schmidt, Klaus: *Rollen und Berechtigungskonzepte: Identity- und Access-Management im Unternehmen*. 2. Auflage. Wiesbaden : Springer Vieweg, 2017. http://dx.doi.org/10.1007/978-3-658-17987-8. http://dx.doi.org/10.1007/978-3-658-17987-8. – ISBN 978–3–658–17987–8

[39] Valasek, Chris ; Miller, Charlie: *CAN Message Injection: OG Dynamite Edition*. http://illmatics.com/can%20message%20injection.pdf, Abruf: 13.04.2022

[40] Ring, Martin ; Rensen, Tobias ; Kriesten, Reiner: Evaluation of Vehicle Diagnostics Security: Implementation of a Reproducible Security Access. In: *Secureware* 2014 (2014)

[41] ISO 14229-1:2013-03: *Road vehicles - Unified diagnostic services (UDS) - Part 1: Specification and requirements*. 2013

[42] National Institute of Standards and Technology (Hrsg.): *The Five Functions of the Cybersecurity Framework*. https://www.nist.gov/document/thefivefunctionspptx, Abruf: 13.01.2021

[43] Reif, Konrad: *Automobilelektronik: Eine Einführung für Ingenieure ; mit 38 Tabellen*. 4., überarb. Aufl. Wiesbaden : Vieweg + Teubner, 2012 (ATZ/MTZ-Fachbuch). – ISBN 978–3–8348–1498–2

[44] Staron, Miroslaw: *Automotive Software Architectures: An Introduction*. Springer, 2017. – ISBN 978–3–319–58610–6

[45] Streichert, Thilo ; Traub, Matthias: *Elektrik-Elektronik-Architekturen im Kraftfahrzeug: Modellierung und Bewertung von Echtzeitsystemen*. 2012. 2012 (VDI-Buch). – ISBN 978–3–642–25477–2

[46] ISO 17458:2013: *Road vehicles — FlexRay communications system*. 2013

[47] ISO 17987-1:2016: *Road vehicles — Local Interconnect Network (LIN): Part 1: General information and use case definition*. 2016

[48] Buckl, Christian ; Camek, Alexander ; Kainz, Gerd ; Simon, Carsten ; Mercep, Ljubo ; Stahle, Hauke ; Knoll, Alois: The software car: Building ICT architectures for future electric vehicles. http://dx.doi.org/10.1109/IEVC.2012.6183198. In: *2012 IEEE International Electric Vehicle*. – DOI 10.1109/IEVC.2012.6183198, S. 1–8

[49] Brooks: No Silver Bullet Essence and Accidents of Software Engineering. In: *Computer* 20 (1987), Nr. 4, S. 10–19. http://dx.doi.org/10.1109/MC.1987.1663532. – DOI 10.1109/MC.1987.1663532. – ISSN 0018–9162

[50] Lock, Andreas ; Tracey, Nigel ; Zerfowski, Detlef: *Aufbruch in neue Welten: Neue E/E-Architekturen mit Vehicle Computern bringen neue Chancen!* https://www.etas.com/download-center-files/DLC_realtimes/RT_2019_2020_de_6_rgb_02-2020.pdf, Abruf: 15.04.2021

[51] Zimmermann, Werner ; Schmidgall, Ralf: *Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur*. 5., aktualisierte und erw. Aufl. 2014. 2014 (ATZ/MTZ-Fachbuch). http://dx.doi.org/10.1007/978-3-658-02419-2. – ISBN 978–3–658–02419–2

[52] Häckel, Timo: *Automobile Kommunikationsarchitekturen zur Unterstützung von Dienstgütevereinbarungen*, Hochschule für Angewandte Wissenschaften Hamburg, Masterthesis, 2018. https://reposit.haw-hamburg.de/bitstream/20.500.12738/8411/1/HaeckelMasterarbeit.pdf

[53] Melzer, Ingo: *Service-orientierte Architekturen mit Web-Services: Konzepte - Standards - Praxis*. 4. Aufl. Heidelberg : Spektrum Akad. Verl., 2010. – ISBN 978–3–8274–2549–2

[54] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *SOA-Security-Kompendium 2.0*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/SOA/SOA-Security-Kompendium_pdf.pdf?__blob=publicationFile&v=1, Abruf: 20.10.2020

[55] Stoll, Hannes: *Die (re-)konfigurierbare Fahrzeugarchitektur*, Karlsruher Institut für Technologie, Dissertation, 2021

[56] OASIS (Hrsg.): *Reference Architecture Foundation for Service Oriented Architecture Version 1.0*. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/cs01/soa-ra-v1.0-cs01.html#_Toc343761555, Abruf: 15.03.2022

[57] Microsoft (Hrsg.): *Was ist Middleware – Definition*. https://azure.microsoft.com/de-de/overview/what-is-middleware/, Abruf: 13.04.2022

[58] Tischer, Mirko: *The Computing Center in the Vehicle: AUTOSAR Adaptive*. https://assets.vector.com/cms/content/know-how/_technical-articles/AUTOSAR/AUTOSAR_Adaptive_ElektronikAutomotive_201809_PressArticle_EN.pdf, Abruf: 14.05.2022

[59] AUTOSAR (Hrsg.): *AUTOSAR Classic Platform*. https://www.autosar.org/standards/classic-platform/, Abruf: 14.05.2022

[60] AUTOSAR (Hrsg.): *AUTOSAR Adaptive Platform*. https://www.autosar.org/standards/adaptive-platform/, Abruf: 14.05.2020

[61] Varas, Marcelino: Service-orientierte Software-Architekturen: Auf dem Weg zum fahrenden Rechenzentrum. In: *Elektronik automotive* 11 (2019), S. 42–45

[62] ISO 15031-1:2010: *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics: Part 1: General information and use case definition*. 2010

[63] Europäische Union (Hrsg.): *Richtlinie 98/69/EG des Europäischen Parlaments und des Rates*. https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1998L0069:19981228:DE:PDF, Abruf: 14.05.2021

[64] Schäuffele, Jörg ; Zurawka, Thomas: *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 6., überarb. u. akt. Aufl. 2016. 2016 (ATZ/MTZ-Fachbuch). http://dx.doi.org/10.1007/978-3-658-11815-0. – ISBN 978–3–658–11815–0

[65] ISO 27001:2017-06: *Information technology - Security techniques - Information security management systems - Requirements*. 2017

[66] Sowa, Aleksandra: *Management der Informationssicherheit: Kontrolle und Optimierung*. 2017 (Studienbücher Informatik). – ISBN 978–3–658–15626–8

[67] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *BSI - Definition "Cyber-Sicherheit"*. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Glossar-der-Cyber-Sicherheit/Functions/glossar.html?nn=522504&cms_lv2=132798, Abruf: 14.05.2022

[68] Eckert, Claudia: *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. De Gruyter, 2018. – ISBN 978–3–11–055158–7

[69] Paar, Christof ; Pelzl, Jan: *Understanding Cryptography: A Textbook for Students and Practitioners*. 2010 http://dx.doi.org/10.1007/978-3-642-04101-3. – ISBN 978–3–642–04101–3

[70] Bedner, Mark ; Ackermann, Tobias: Schutzziele der IT-sicherheit. In: *Datenschutz und Datensicherheit-DuD* 34 (2010), Nr. 5, S. 323–328

[71] IONOS (Hrsg.): *Server/Client Definitionen*. https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-server-ein-begriff-zwei-definitionen/, Abruf: 14.05.2022

[72] Lauber, Andreas F.: *Testen von Datensicherheit in vernetzten und automatisierten Fahrzeugen durch virtuelle Steuergeräte*, Karlsruher Institut für Technologie, Dissertation, 2020. http://dx.doi.org/10.5445/IR/1000123951. – DOI 10.5445/IR/1000123951

[73] Klipper, Sebastian: *Information Security Risk Management: Risikomanagement mit ISO/IEC 27001, 27005 und 31010*. 2011 http://dx.doi.org/10.1007/978-3-8348-9870-8. – ISBN 978–3–8348–1360–2

[74] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Die Lage der IT-Sicherheit in Deutschland 2019*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2019.pdf?__blob=publicationFile&v=7, Abruf: 14.05.2021

[75] Microsoft: *Phasen des Security Lifecycle (SDL) von Microsoft*. https://blogs.technet.microsoft.com/voy/2013/06/10/security-series-2-how-to-bake-security-in-products-and-services-sdl/, Abruf: 14.01.2021

[76] OWASP: *Software Assurance Maturity Model (SAMM): Core Model 1.5*. https://github.com/OWASP/samm/raw/master/v1.5/Final/SAMM_Core_V1-5_FINAL.pdf, Abruf: 27.11.2018

[77] ISO/IEC 15408-1:2009: *Information technology — Security techniques — Evaluation criteria for IT security: Part 1: Introduction and general model*. 2009

[78] Shostack, Adam: *Threat modeling: Designing for security*. John Wiley & Sons, 2014

[79] Macher, Georg ; Sporer, Harald ; Berlach, Reinhard ; Armengaud, Eric ; Kreiner, Christian: SAHARA: A security-aware hazard and risk

analysis method. In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, S. 621–624

[80] Fredriksen, Rune ; Kristiansen, Monica ; Gran, Bjørn A. ; Stølen, Ketil ; Opperud, Tom A. ; Dimitrakos, Theo: The CORAS framework for a model-based risk management process. In: *International Conference on Computer Safety, Reliability, and Security*, 2002, S. 94–105

[81] Schmittner, Christoph ; Ma, Zhendong ; Smith, Paul: FMVEA for Safety and Security Analysis of Intelligent and Cooperative Vehicles. Version: 2014. http://dx.doi.org/10.1007/978-3-319-10557-4. In: Bondavalli, Andrea (Hrsg.) ; Ceccarelli, Andrea (Hrsg.) ; Ortmeier, Frank (Hrsg.): *Computer safety, reliability, and security* Bd. 8696. Heidelberg : Springer, 2014. – DOI 10.1007/978–3–319–10557–4. – ISBN 978–3–319–10556–7, S. 282–288

[82] Affia, Abasi-Amefon O. ; Matulevičius, Raimundas ; Nolte, Alexander: Security Risk Management in Cooperative Intelligent Transportation Systems: A Systematic Literature Review. In: *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems*, 2019, S. 282–300

[83] Homoliak, Ivan ; Toffalini, Flavio ; Guarnizo, Juan ; Elovici, Yuval ; Ochoa, Martín: Insight Into Insiders and IT. In: *ACM Computing Surveys* 52 (2019), Nr. 2, S. 1–40. http://dx.doi.org/10.1145/3303771. – DOI 10.1145/3303771. – ISSN 03600300

[84] Probst, Christian W.: *Advances in Information Security*. Bd. 49: *Insider Threats in Cyber Security*. Boston, MA : Springer Science+Business Media LLC, 2010 http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10406647. – ISBN 978–1–4419–7132–6

[85] Ponemon Institute (Hrsg.): *2020 Cost of Insider Threats Global Report*. https://cdw-prod.adobecqms.net/content/dam/cdw/on-domain-cdw/brands/proofpoint/ponemon-global-cost-of-insider-threats-2020-report.pdf, Abruf: 14.07.2021

[86] Cole, Eric ; Ring, Sandra: Insider Threat: Protecting the Enterprise from Sabotage. In: *Spying, and Theft, Canada: Syngress Publications* (2006)

[87] Pfleeger, Shari L. ; Predd, Joel B. ; Hunker, Jeffrey ; Bulford, Carla: Insiders Behaving Badly: Addressing Bad Actors and Their Actions. In: *IEEE Transactions on Information Forensics and Security* 5 (2010), Nr.

1, S. 169–179. http://dx.doi.org/10.1109/tifs.2009.2039591. – DOI 10.1109/tifs.2009.2039591. – ISSN 1556–6013

[88] Brackney, Richard C. ; Anderson, Robert H.: *Understanding the Insider Threat. Proceedings of a March 2004 Workshop*

[89] IEEE 802.11:2016: *IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2016

[90] Prabhu, Sunitha ; Thompson, Nik: A Unified Classification Model of Insider Threats to Information Security. In: *31st Australasian Conference on Information Systems*, 2020

[91] World Wide Web Consortium (Hrsg.): *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. https://www.w3.org/TR/2008/REC-xml-20081126/, Abruf: 20.11.2020

[92] Dürbeck, Stefan ; Kolter, Jan ; Pernul, Günther ; Schillinger, Rolf: Eine verteilte Autorisierungsinfrastruktur unter Berücksichtigung von Datenschutzaspekten. In: *Informatik-Spektrum* 34 (2011), Nr. 3, S. 265–275. http://dx.doi.org/10.1007/s00287-009-0411-0. – DOI 10.1007/s00287–009–0411–0. – ISSN 0170–6012

[93] FU-Berlin (Hrsg.): *Rechnersicherheit - Zugriffskontrolle*. https://www.mi.fu-berlin.de/inf/groups/ag-idm/teaching/Rechnersicherheit/RS15.pdf, Abruf: 04.08.2021

[94] Kessel, Johannes: *Benutzerverwaltung und Sicherheitskonzepte im Geschäftsprozessmanagement*, Universität Stuttgart, Diplomarbeit, 2013. ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/DIP-3476/DIP-3476.pdf

[95] Priebe, Torsten ; Dobmeier, Wolfgang ; Muschall, Björn ; Pernul, Günther u. a.: ABAC-Ein Referenzmodell für attributbasierte Zugriffskontrolle. In: *Sicherheit* Bd. 62, 2005, 285–296

[96] Klarl, Heiko: *Zugriffskontrolle in Geschäftsprozessen: Ein modellgetriebener Ansatz: Zugl.:Regensburg, Univ., Diss., 2010*. 1. Aufl. Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, 2011 (Vieweg+Teubner Research). http://dx.doi.org/10.1007/978-3-8348-9913-2. http://dx.doi.org/10.1007/978-3-8348-9913-2. – ISBN 978–3–8348–1465–4

[97] OASIS: *eXtensible Access Control Markup Language (XACML) Version 3.0*

[98] Hu, Vincent C. ; Ferraiolo, David ; Kuhn, Rick ; Friedman, Arthur R. ; Lang, Alan J. ; Cogdell, Margaret M. ; Schnitzer, Adam ; Sandlin, Kenneth ; Miller, Robert ; Scarfone, Karen u. a.: Guide to attribute based access control (ABAC) definition and considerations. In: *NIST special publication* 800 (2013), Nr. 162

[99] HL7 Deutschland e.V (Hrsg.): *XACML*. http://wiki.hl7.de/index.php?title=ihecb:XACML, Abruf: 15.06.2020

[100] Hu, Vincent C. ; Scarfone, Karen: *Guidelines for Access Control System Evaluation Metrics*. http://dx.doi.org/10.6028/NIST.IR.7874

[101] Shirey, R.: *Internet Security Glossary, Version 2*. http://dx.doi.org/10.17487/rfc4949

[102] Whitman, Michael E. ; Mattord, Herbert J.: *Principles of information security*. 4. ed., International ed. Stamford, Conn. : Course Technology Cengage Learning, 2012. – ISBN 978–1–111–13821–9

[103] Al-Jarrah, Omar Y. ; Maple, Carsten ; Dianati, Mehrdad ; Oxtoby, David ; Mouzakitis, Alex: Intrusion Detection Systems for Intra-Vehicle Networks: A Review. In: *IEEE Access* 7 (2019), S. 21266–21289

[104] Modi, Chirag ; Patel, Dhiren ; Borisaniya, Bhavesh ; Patel, Hiren ; Patel, Avi ; Rajarajan, Muttukrishnan: A survey of intrusion detection techniques in Cloud. In: *Journal of Network and Computer Applications* 36 (2013), Nr. 1, S. 42–57. http://dx.doi.org/10.1016/j.jnca.2012.05.003. – DOI 10.1016/j.jnca.2012.05.003. – ISSN 10848045

[105] Khraisat, Ansam ; Gondal, Iqbal ; Vamplew, Peter ; Kamruzzaman, Joarder: Survey of intrusion detection systems: techniques, datasets and challenges. In: *Cybersecurity* 2 (2019), Nr. 1. http://dx.doi.org/10.1186/s42400-019-0038-7. – DOI 10.1186/s42400–019–0038–7

[106] Hofmockel, Julia: *Anomalieerkennung in Kommunikationsdaten zur Datenselektion im Fahrzeug*, Karlsruher Institut für Technologie, Dissertation, 2019

[107] García-Teodoro, P. ; Díaz-Verdejo, J. ; Maciá-Fernández, G. ; Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. In: *Computers & Security* 28 (2009), Nr. 1-2, S.

18–28. http://dx.doi.org/10.1016/j.cose.2008.08.003. – DOI 10.1016/j.cose.2008.08.003. – ISSN 01674048

[108] Schacht, Sigurd ; Lanquillon, Carsten: *Blockchain und maschinelles Lernen: Wie das maschinelle Lernen und die Distributed-Ledger-Technologie voneinander profitieren*. Berlin Germany : Springer Vieweg, 2019. – ISBN 978–3662604076

[109] Jungmann, Alexander ; Lang, Christian ; Pinsker, Florian ; Kallweit, Roland ; Taubenreuther, Mirko ; Butenuth, Matthias: Artificial intelligence for automated driving – quo vadis? Version: 2020. http://dx.doi.org/10.1007/978-3-658-27990-5{_}11. In: Bertram, Torsten (Hrsg.): *Automatisiertes Fahren 2019*. Wiesbaden : Springer Fachmedien Wiesbaden, 2020 (Proceedings). – DOI 10.1007/978–3–658–27990–5_11. – ISBN 978–3–658–27989–9, S. 117–134

[110] Hindy, Hanan ; Brosset, David ; Bayne, Ethan ; Seeam, Amar K. ; Tachtatzis, Christos ; Atkinson, Robert ; Bellekens, Xavier: A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. In: *IEEE Access* 8 (2020), S. 104650–104675. http://dx.doi.org/10.1109/ACCESS.2020.3000179. – DOI 10.1109/ACCESS.2020.3000179

[111] Weber, Marc: *Untersuchungen zur Anomalieerkennung in automotive Steuergeräten durch verteilte Observer mit Fokus auf die Plausibilisierung von Kommunikationssignalen*, Karlsruher Institut für Technologie, Dissertation, 2019. http://dx.doi.org/10.5445/IR/1000092815. – DOI 10.5445/IR/1000092815

[112] Pfister, Beat (Hrsg.) ; Kaufmann, Tobias (Hrsg.): *Sprachverarbeitung*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2017. http://dx.doi.org/10.1007/978-3-662-52838-9. http://dx.doi.org/10.1007/978-3-662-52838-9. – ISBN 978–3–662–52837–2

[113] Carstensen, Kai-Uwe (Hrsg.) ; Ebert, Christian (Hrsg.) ; Ebert, Cornelia (Hrsg.) ; Jekat, Susanne (Hrsg.) ; Klabunde, Ralf (Hrsg.) ; Langer, Hagen (Hrsg.): *Computerlinguistik und Sprachtechnologie: Eine Einführung*. 3., überarb. und erw. Aufl. Heidelberg : Spektrum Akad. Verl., 2010. – ISBN 978–3–8274–2023–7

[114] Jurafsky, Daniel ; Martin, James H.: *Speech and language processing: An introduction to natural language processing, computational*

linguistics, and speech recognition. 2. ed. Upper Saddle River, NJ : Prentice Hall, 2008 (Prentice Hall series in artificial intelligence). – ISBN 978–0131873216

[115] Sandmann, Werner ; Uni Bamberg (Hrsg.): *Vorlesungsskript - Markovketten*. https://www.uni-bamberg.de/fileadmin/uni/fakultaeten/wiai_lehrstuehle/informatik_ktr/Dateien/MAKV-WS07-08/makv07-4Markovketten.pdf. Version: 2008

[116] Reichel, Uwe: *Statistische Sprachmodelle*. https://www.phonetik.uni-muenchen.de/~reichelu/kurse/stat_lm/script_reichel_stat_lm.pdf. Version: 2009

[117] Sommer, Florian ; Dürrwang, Jürgen: *IEEM-HsKA/AAD : Automotive Attack Database (AAD)*. https://github.com/IEEM-HsKA/AAD. Version: 22.07.2022

[118] *Common Vulnerabilities and Exposures (CVE)*. https://cve.mitre.org/, Abruf: 20.07.2020

[119] Deichmann, Johannes ; Klein, Benjamin ; Scherf, Gundbert ; Stützle, Rupert ; McKinsey&Company (Hrsg.): *The race for cybersecurity: Protecting the connected car in the era of new regulation*. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-race-for-cybersecurity-protecting-the-connected-car-in-the-era-of-new-regulation. Version: 2019, Abruf: 20.04.2020

[120] Upstream Security Ltd. (Hrsg.): *Upstream Security's 2020 Global Automotive Cybersecurity Report*. https://www.upstream.auto/upstream-security-global-automotive-cybersecurity-report-2020/, Abruf: 15.07.2020

[121] Steffelbauer, Markus ; Fahböck, Günter: Diagnose und Security: Schöne neue Welt. In: *HANSER automotive* 2019 (2019), Nr. 10, 60–63. https://automotive.softing.com/fileadmin/softfiles/pdf/ae/articles/2019/HANSERautomotive_Sichere_Remote_Diagnose_1910_DE.pdf

[122] Smith, Craig: *The car hacker's handbook: A guide for the penetration tester*. San Francisco, USA : No Starch Press, 2016

[123] GitHub: *CaringCaribou*. https://github.com/CaringCaribou/caringcaribou, Abruf: 20.05.2021

[124] Weigert, Sebastian: *Radio Frequency Identification (RFID) in der Automobilindustrie*. Deutscher Universitäts-Verlag, 2007. – ISBN 978–3–8350–0638–6

[125] Miller, Charlie ; Valasek, Chris: Adventures in automotive networks and control units. In: *DEF CON* 21 (2013), S. 260–264

[126] Rapid7: *Metasploit: Penetration Testing Software, Pen Testing Security*. https://www.metasploit.com/. Version: 2020, Abruf: 17.6.2020

[127] Offensive Security: *Kali Linux*. https://www.kali.org/, Abruf: 17.6.2020

[128] Luo, Feng ; Hou, Shuo: Cyberattacks and Countermeasures for Intelligent and Connected Vehicles. In: *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 12 (2019), Nr. 1. http://dx.doi.org/10.4271/07-12-01-0005. – DOI 10.4271/07–12–01–0005. – ISSN 1946–4622

[129] Chikofsky, E. J. ; Cross, J. H.: Reverse engineering and design recovery: a taxonomy. In: *IEEE Software* 7 (1990), Nr. 1, S. 13–17. http://dx.doi.org/10.1109/52.43044. – DOI 10.1109/52.43044. – ISSN 0740–7459

[130] Abowd, Gregory D. ; Dey, Anind K. ; Brown, Peter J. ; Davies, Nigel ; Smith, Mark ; Steggles, Pete: Towards a better understanding of context and context-awareness. In: *International symposium on handheld and ubiquitous computing*, 1999, S. 304–307

[131] ISO 11898-1:2015-12: *Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling*. 2015

[132] ISO 26021:2009: *Road vehicles – End-of-life activation of on-board pyrotechnic devices*. 2009

[133] Felderer, Michael ; Büchler, Matthias ; Johns, Martin ; Brucker, Achim D. ; Breu, Ruth ; Pretschner, Alexander: Security testing: A survey. In: *Advances in Computers* Bd. 101. Elsevier, 2016, S. 1–51

[134] Chickowski, Ericka ; Bitdefender (Hrsg.): *Tesla Sabotage Highlights Danger of Insider Threat*. https://businessinsights.bitdefender.com/tesla-sabotage-highlights-danger-insider-threat, Abruf: 21.02.2020

[135] Cimpanu, Catalin: *Mercedes-Benz onboard logic unit (OLU) source code leaks online*. https://www.zdnet.com/article/mercedes-be

nz-onboard-logic-unit-olu-source-code-leaks-online/, Abruf: 12.07.2020

[136] Wen, Haohuang ; Chen, Qi A. ; Lin, Zhiqiang: *Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT*. http://web.cse.ohio-state.edu/~lin.3021/file/SEC20a.pdf, Abruf: 17.04.2020

[137] Kriesten, Reiner ; Kuhlee, Lorenz: *Angriffe auf die Flotten-Daten abwehren*. https://www.faz.media/fileadmin/04_Sonderthemen/VS_Dienstwagen_und_Flottenmanagement.pdf, Abruf: 22.07.2020

[138] Malekian, Reza ; Moloisane, Ntefeng R. ; Nair, Lakshmi ; Maharaj, B. T. ; Chude-Okonkwo, Uche A. K.: Design and Implementation of a Wireless OBD II Fleet Management System. In: *IEEE Sensors Journal* 17 (2017), Nr. 4, S. 1154–1164. http://dx.doi.org/10.1109/JSEN.2016.2631542. – DOI 10.1109/JSEN.2016.2631542. – ISSN 1530–437X

[139] Seifert, Stefan ; Obermaisser, Roman: Secure automotive gateway—secure communication for future cars. In: *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, 2014, S. 213–220

[140] Pesé, Mert D. ; Schmidt, Karsten ; Zweck, Harald: Hardware/Software Co-Design of an Automotive Embedded Firewall. In: *SAE Technical Paper Series*, SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2017 (SAE Technical Paper Series)

[141] Holle, Jan ; Shukla, Siddharth: Torwächter der Bordnetzkommunikation. In: *ATZelektronik* 13 (2018), Nr. 6, S. 42–45

[142] Luo, Feng ; Hou, Shuo: Security Mechanisms Design of Automotive Gateway Firewall. In: *SAE Technical Paper Series*, SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2019 (SAE Technical Paper Series)

[143] Kreissl, Jochen: *Absicherung der SOME/IP Kommunikation bei Adaptive AUTOSAR*. Stuttgart, Universität Stuttgart, Masterthesis, 2017. https://elib.uni-stuttgart.de/bitstream/11682/9482/1/ausarbeitung.pdf

[144] Kim, Dae-Kyoo ; Song, Eunjee ; Yu, Huafeng: Introducing Attribute-Based Access Control to AUTOSAR. No. 2016-01-0069. In: *SAE Technical Paper* (2016)

[145] Hamad, Mohammad ; Nolte, Marcus ; Prevelakis, Vassilis: A framework for policy based secure intra vehicle communication. In: Altintas,

Onur (Hrsg.) ; Casetti, Claudio (Hrsg.) ; Kirsch, Nicholas (Hrsg.) ; Lo Cigno, Renato (Hrsg.) ; Meireles, Rui (Hrsg.): *2017 IEEE Vehicular Networking Conference (VNC)*. Piscataway, NJ and Piscataway, NJ : IEEE, 2017. – ISBN 978–1–5386–0986–6, S. 1–8

[146] Gupta, Maanak ; Sandhu, Ravi: Authorization framework for secure cloud assisted connected cars and vehicular internet of things. In: *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, 2018, S. 193–204

[147] Ruland, Christoph ; Sassmannshausen, Jochen: Firewall for Attribute-Based Access Control in Smart Grids. In: *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, 2018, S. 336–341

[148] Huh, Jun H. ; Bobba, Rakesh B. ; Markham, Tom ; Nicol, David M. ; Hull, Julie ; Chernoguzov, Alex ; Khurana, Himanshu ; Staggs, Kevin ; Huang, Jingwei: Next-Generation Access Control for Distributed Control Systems. In: *IEEE Internet Computing* 20 (2016), Nr. 5, S. 28–37. http://dx.doi.org/10.1109/MIC.2016.105. – DOI 10.1109/MIC.2016.105. – ISSN 1089–7801

[149] Hugot, Vincent ; Jousse, Adrien ; Toinard, Christian ; Venelle, Benjamin: *oMAC : Open Model for Automotive Cybersecurity*. Ruhr-Universität Bochum, 2019. http://dx.doi.org/10.13154/294-6674. http://dx.doi.org/10.13154/294-6674

[150] Grimm, Daniel ; Stang, Marco ; Sax, Eric: Context-Aware Security for Vehicles and Fleets: A Survey. In: *IEEE Access* 9 (2021), S. 101809–101846. http://dx.doi.org/10.1109/ACCESS.2021.3097146. – DOI 10.1109/ACCESS.2021.3097146

[151] Weber, Marc ; Klug, Simon ; Sax, Eric ; Zimmer, Bastian: Embedded Hybrid Anomaly Detection for Automotive CAN Communication. In: *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, 2018

[152] Kneib, Marcel ; Huth, Christopher: Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, S. 787–800

[153] Murvay, Pal-Stefan ; Groza, Bogdan: TIDAL-CAN: differential Timing based Intrusion Detection And Localization for Controller Area Net-

work. In: *IEEE Access* (2020), S. 1. http://dx.doi.org/10.1109/ACCESS.2020.2985326. – DOI 10.1109/ACCESS.2020.2985326

[154] Wasicek, Armin ; Pesé, Mert D. ; Weimerskirch, André ; Burakova, Yelizaveta ; Singh, Karan: Context-aware intrusion detection in automotive control systems. In: *5th ESCAR USA Conference, USA*, 2017, S. 21–22

[155] Kalutarage, Harsha K. ; Al-Kadri, M. O. ; Cheah, Madeline ; Madzudzo, Garikayi: Context-aware Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus. In: Hof, Hans-Joachim (Hrsg.) ; Fritz, Mario (Hrsg.) ; Krauß, Christoph (Hrsg.) ; Wasenmüller, Oliver (Hrsg.): *ACM Computer Science in Cars Symposium on - CSCS '19*. New York, New York, USA : ACM Press, 2019. – ISBN 9781450370042, S. 1–8

[156] Khreich, Wael ; Khosravifar, Babak ; Hamou-Lhadj, Abdelwahab ; Talhi, Chamseddine: An anomaly detection system based on variable N-gram features and one-class SVM. In: *Information and Software Technology* 91 (2017), S. 186–197. http://dx.doi.org/10.1016/j.infsof.2017.07.009. – DOI 10.1016/j.infsof.2017.07.009

[157] Zolotukhin, Mikhail ; Hämäläinen, Timo ; Juvonen, Antti: Online anomaly detection by using n-gram model and growing hierarchical self-organizing maps. In: *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2012, S. 47–52

[158] Hermsdorf, Kristian: *Inhaltliche Verwandtschaftsbeziehungen von Textdokumentenauf Grundlage von Selbstorganisierenden Lernverfahrenund prototypische Implementierung*. Dresden, Hochschule für Technik und Wirtschaft, Diplomarbeit, 2002. http://www.informatik.htw-dresden.de/~fritzsch/SF/Literatur-Datenbank/Bestand-Literatur/diplom_Kristian_Hermsdorf.pdf

[159] AUTOSAR: *AUTOSAR Adaptive R19-11: Explanation of IPsec Implementation Guidelines*. 2019

[160] AUTOSAR (Hrsg.): *Adaptive Platform 18.03*. https://www.autosar.org/standards/adaptive-platform/adaptive-platform-1803/. Version: 2018, Abruf: 02.07.2020

[161] DDS Foundation (Hrsg.): *OMG Data Distribution Service (DDS)*. https://www.dds-foundation.org/what-is-dds-3/, Abruf: 14.05.2021

[162] Garcia, Fernando ; RTI (Hrsg.): *Integrating DDS Into the AUTOSAR Adaptive Platform*. https://www.rti.com/blog/integrating-dds-into-the-autosar-adaptive-platform, Abruf: 14.05.2021

[163] UNECE: *UN Regulation No. 155*. https://unece.org/sites/default/files/2021-03/R155e.pdf. Version: 2021

[164] UNECE: *UN Regulation No. 156*. https://unece.org/sites/default/files/2021-03/R156e.pdf. Version: 2021

[165] UNECE: *Uniform provisions concerning the International Whole Vehicle Type Approval (IWVTA)*. https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2018/R000e.pdf, Abruf: 07.04.2020

[166] Europäische Union (Hrsg.): *Richtlinie 2007/46/EG des Europäischen Parlaments und des Rates*. https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32007L0046&from=DE, Abruf: 14.07.2020

[167] Kullmann, Hans J.: *ProdHaftG: Gesetz über die Haftung für fehlerhafte Produkte*. 5., neu bearb. Aufl. Berlin : Schmidt, 2006. – ISBN 9783503093557

[168] Elend, Bernd ; Walrant, Thierry ; Olma, Georg ; eeNews AUTOMOTIVE (Hrsg.): *Securing CAN Communication Efficiently With Minimal System Impact*. https://www.eenewsautomotive.com/news/securing-can-communication-efficiently-minimal-system-impact/page/0/4. Version: 2018, Abruf: 25.11.2020

[169] Dworkin, M. J.: *Recommendation for block cipher modes of operation*. http://dx.doi.org/10.6028/NIST.SP.800-38B

[170] CVSS Special Interest Group (Hrsg.): *Common Vulnerability Scoring Group v3.0 - Specification Document*. https://www.first.org/cvss/v3.0/cvss-v30-specification_v1.9.pdf, Abruf: 24.08.2020

[171] Zoppelt, Markus ; Kolagari, Ramin T. ; (Keine Angabe): Reaching Grey HavensIndustrial Automotive Security Modeling with SAM. In: *International Journal on Advances in Security* 12 (2019), Nr. 3&4, 223–235. http://www.iariajournals.org/security/

[172] Sitawarin, Chawin ; Bhagoji, Arjun N. ; Mosenia, Arsalan ; Chiang, Mung ; Mittal, Prateek: *DARTS: Deceiving Autonomous Cars with Toxic Signs*. https://arxiv.org/pdf/1802.06430. Version: 2018, Abruf: 25.08.2020

[173] Kapoor, Prateek ; Vora, Ankur ; Kang, Kyoung-Don: Detecting and Mitigating Spoofing Attack Against an Automotive Radar. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2018. – ISBN 978–1–5386–6358–5, S. 1–6

[174] Hartzell, Shawn ; Stubel, Christopher ; Bonaci, Tamara: Security Analysis of an Automobile Controller Area Network Bus. In: *IEEE Potentials* 39 (2020), Nr. 3, S. 19–24. http://dx.doi.org/10.1109/MPOT.2018.2837686. – DOI 10.1109/MPOT.2018.2837686. – ISSN 0278–6648

[175] Ray, Sandip ; Chen, Wen ; Bhadra, Jayanta ; Al Faruque, Mohammad A.: Extensibility in Automotive Security. In: *Proceedings of the 54th Annual Design Automation Conference 2017*. New York, NY, USA : ACM, 06182017. – ISBN 9781450349277, S. 1–6

[176] OASIS (Hrsg.): *Abbreviated Language for Authorization (ALFA) - Version 1.0*. https://www.oasis-open.org/committees/download.php/55228/alfa-for-xacml-v1.0-wd01.doc. Version: Â´, Abruf: 17.11.2020

[177] STMicroelectronics (Hrsg.): *STM32 Nucleo-64 board*. https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-f401re.html, Abruf: 18.11.2020

[178] FreeRTOS (Hrsg.): *Real-time operating system for microcontrollers*. https://www.freertos.org/, Abruf: 18.11.2020

[179] NXP (Hrsg.): *MPC5746CMPC5746C Microcontroller Datasheet*. https://www.nxp.com/docs/en/data-sheet/MPC5746C.pdf, Abruf: 25.09.2020

[180] Müter, Michael ; Groll, Andre ; Freiling, Felix C.: A structured approach to anomaly detection for in-vehicle networks. In: *2010 Sixth International Conference on Information Assurance and Security*, IEEE, 2010. – ISBN 978–1–4244–7407–3, S. 92–98

[181] Shashanka, Madhu ; Shen, Min-Yi ; Wang, Jisheng: User and entity behavior analytics for enterprise security. In: *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, 122016. – ISBN 978–1–4673–9005–7, S. 1867–1874

[182] Gowda, B. N. S. ; Lakshmikantha, Vibha: User Behavior Prediction using A Novel Sentence N-Gram Model. In: *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, IEEE, 32020. – ISBN 978–1–7281–4167–1, S. 391–397

[183] Zolotukhin, Mikhail ; Hämäläinen, Timo: Detection of Anomalous HTTP Requests Based on Advanced N-gram Model and Clustering Techniques. Version: 2013. http://dx.doi.org/10.1007/978-3-642-40316-3{_}33. In: *Internet of Things, Smart Spaces, and Next Generation Networking* Bd. 8121. Berlin, Heidelberg : Springer Berlin Heidelberg, 2013. – DOI 10.1007/978–3–642–40316–3_33. – ISBN 978–3–642–40315–6, S. 371–382

[184] Luber, Stefan ; Schmitz, Peter: *Definition User and Entity Behavior Analytics (UEBA)*. https://www.security-insider.de/was-ist-user-and-entity-behavior-analytics-ueba-a-983974/, Abruf: 13.03.2021

[185] Nguyen, Khanh: *N-gram language models: Part 2: Higher n-gram models*. https://medium.com/mti-technology/n-gram-language-models-70af02e742ad, Abruf: 20.06.2021

[186] Allison, Ben ; Guthrie, David ; Guthrie, Louise: Another look at the data sparsity problem. In: *International Conference on Text, Speech and Dialogue*, 2006, S. 327–334

[187] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Anomaly detection. In: *ACM Computing Surveys* 41 (2009), Nr. 3, S. 1–58. http://dx.doi.org/10.1145/1541880.1541882. – DOI 10.1145/1541880.1541882. – ISSN 03600300

[188] Rudzicz, Frank ; Robertson, Sean ; Jeblee, Serena: *Corpara, language models, and smoothing: Natural Language Computing - Vorlesungsfolien*. https://www.cs.toronto.edu/~frank/csc401/lectures/2_Corpora_and_Smoothing.pdf, Abruf: 14.06.2021

[189] Wu, Jun: *The beauty of mathematics in computer science*. Boca Raton, FL and London and New York : CRC Press, 2019. – ISBN 9781138049604

[190] Rossi, Richard J.: *Mathematical statistics: An introduction to likelihood based inference*. 1st edition. Hoboken, NJ : John Wiley & Sons, 2018. – ISBN 978–1–118–77104–4

[191] Stahl, Oliver ; Baecker, Roman ; Bartholdt, Michael ; Roth, Sebastian ; Sejdiu, Arber: *Cybersecurity as a Matter of Competitive Ad-*

vantage. https://www.porsche-consulting.com/en/home/news/cybersecurity-as-a-matter-of-competitive-advantage/, Abruf: 20.08.2021

[192] Brandt, Thiemo ; Tamisier, Théo: The Future Connected Car – Safely Developed Thanks to UNECE WP.29? Version: 2021. http://dx.doi.org/10.1007/978-3-658-33521-2{_}31. In: Bargende, Michael (Hrsg.) ; Reuss, Hans-Christian (Hrsg.) ; Wagner, Andreas (Hrsg.): *21. Internationales Stuttgarter Symposium*. Wiesbaden : Springer Fachmedien Wiesbaden, 2021 (Proceedings). – DOI 10.1007/978–3–658–33521–2_31. – ISBN 978–3–658–33520–5, S. 461–473

[193] Metzker, Eduard: *Cybersecurity: Anforderungen an Intrusion-Detection-Systeme im Auto*. https://www.all-electronics.de/automotive-transportation/cybersecurity-anforderungen-an-intrusion-detection-systeme-im-auto.html, Abruf: 20.06.2021

[194] AUTOSAR: *AUTOSAR FO R20-11: Specification of Intrusion Detection System Protocol*

[195] Olt, Christian: Aufbau eines Cyberabwehrzentrums für das Connected Car. In: *ATZ Elektronik* 14 (2019), Nr. 5, S. 44–47. http://dx.doi.org/10.1007/s35658-019-0038-0. – DOI 10.1007/s35658–019–0038–0

[196] Williams, Amrit ; Nicolett, Mark: *Improve IT Security With Vulnerability Management*. https://www.gartner.com/resources/127400/127481/improve_it_security_with_vul_127481.pdf. Version: 2005

[197] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Leitfaden „IT-Forensik“ (Version 1.0.1)*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf;jsessionid=415343C6AEEF18E2052CB790BFDA7BF1.internet082?__blob=publicationFile&v=1, Abruf: 20.07.2021

[198] Zhang, Yanan ; Shi, Peiji ; Liu, Yangyang ; Han, Shengqiang ; Mu, Baoying ; Zheng, Jia: Study on Incident Response System of Automotive Cybersecurity. Version: 2019. http://dx.doi.org/10.1007/978-3-030-21373-2{_}16. In: Li, Jin (Hrsg.) ; Liu, Zheli (Hrsg.) ; Peng, Hao (Hrsg.): *Security and Privacy in New Computing Environments* Bd. 284. Cham : Springer International Publishing, 2019. – DOI 10.1007/978–3–030–21373–2_16. – ISBN 978–3–030–21372–5, S. 198–209

[199] Hoppe, Tobias: *Prävention, Detektion und Reaktion gegen drei Ausprägungsformen automotiver Malware*, Universität Magdeburg, Dissertation, 2014. https://d-nb.info/1066295336/34

[200] Guissouma, Houssem ; Diewald, Axel ; Sax, Eric: A Generic System for Automotive Software Over the Air (SOTA) Updates Allowing Efficient Variant and Release Management. Version: 2019. http://dx.doi.org/10.1007/978-3-319-99981-4{_}8. In: Borzemski, Leszek (Hrsg.) ; Świątek, Jerzy (Hrsg.) ; Wilimowska, Zofia (Hrsg.): *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018* Bd. 852. Cham : Springer International Publishing, 2019. – DOI 10.1007/978–3–319–99981–4_8. – ISBN 978–3–319–99980–7, S. 78–89

[201] Bolz, Robin ; Kriesten, Reiner: Automotive Vulnerability Disclosure: Stakeholders, Opportunities, Challenges. In: *MDPI - JCP (Journal of Cybersecurity and Privacy)* 1 (2021), Nr. 2, S. 274–288. http://dx.doi.org/10.3390/jcp1020015. – DOI 10.3390/jcp1020015

[202] Bundesministerium für Bildung und Forschung (Hrsg.): *Mehr Sicherheit durch kontinuierliches Monitoring von Sicherheitsvorfällen autonom agierender Fahrzeuge (Projekt "UNCOVER").* https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/uncover, Abruf: 16.08.2021

[203] Saltzer, Jerome H.: Protection and the control of information sharing in multics. In: *Communications of the ACM* 17 (1974), Nr. 7, S. 388–402. http://dx.doi.org/10.1145/361011.361067. – DOI 10.1145/361011.361067. – ISSN 00010782

[204] AUTOSAR: *AUTOSAR AP R20-11: Requirements on Identity and Access Management*

[205] Sattler, Sven: *In-Car Gaming bei Mercedes-Benz.* https://www.daimler.com/magazin/technologie-innovation/in-car-gaming.html, Abruf: 20.07.2021

[206] Herold, Nadine ; Posselt, Stephan-A. ; Hanka, Oliver ; Carle, Georg: Anomaly detection for SOME/IP using complex event processing. In: *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, IEEE, 2016. – ISBN 978–1–5090–0223–8, S. 1221–1226

[207] Iorio, Marco ; Reineri, Massimo ; Risso, Fulvio ; Sisto, Riccardo ; Valenza, Fulvio: Securing SOME/IP for In-Vehicle Service Protection. In: *IEEE Transactions on Vehicular Technology* 69 (2020), Nr. 11, S. 13450–13466. http://dx.doi.org/10.1109/TVT.2020.3028880. – DOI 10.1109/TVT.2020.3028880. – ISSN 0018–9545

[208] Gehrmann, Tobias ; Duplys, Paul: Intrusion Detection for SOME/IP: Challenges and Opportunities. In: *2020 23rd Euromicro Conference on Digital System Design (DSD)*, IEEE, 82020. – ISBN 978–1–7281–9535–3, S. 583–587

[209] Alkhatib, Natasha ; Ghauch, Hadi ; Danger, Jean-Luc: SOME/IP Intrusion Detection using Deep Learning-based Sequential Models in Automotive Ethernet Networks. In: *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, IEEE, 2021. – ISBN 978–1–6654–0066–4, S. 0954–0962

[210] Moustafa, Nour ; Hu, Jiankun ; Slay, Jill: A holistic review of Network Anomaly Detection Systems: A comprehensive survey. In: *Journal of Network and Computer Applications* 128 (2019), S. 33–55. http://dx.doi.org/10.1016/j.jnca.2018.12.006. – DOI 10.1016/j.jnca.2018.12.006. – ISSN 10848045

[211] Auto-ISAC: *Automotive Information Sharing and Analysis Center*. https://automotiveisac.com/, Abruf: 09.07.2021

[212] Rosenstatter, Thomas ; Olovsson, Tomas: Open Problems when Mapping Automotive Security Levels to System Requirements. In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, SCITEPRESS - Science and Technology Publications, 2018. – ISBN 978–989–758–293–6, S. 251–260

[213] Metzker, Eduard: Reliably Detecting and Defending Against Attacks: Requirements for Automotive Intrusion Detection Systems. In: *Automobil Elektronik* (2020), Nr. 03. https://assets.vector.com/cms/content/know-how/_technical-articles/Security_Intrusion_Detection_AutomobilElektronik_202003_PressArticle_EN.pdf

[214] Reinhardt, Dominik ; Dannebaum, Udo ; Scheffer, Michael ; Traub, Matthias: High Performance Processor Architecture for Automotive Large Scaled Integrated Systems within the European Processor Initiative Research Project. In: *SAE Technical Paper Series*, SAE International, 2019

[215] ISO/IEC 7498-1:1994-11: *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. 1994

[216] Vector Informatik GmbH (Hrsg.): *Motivation für CAN FD*. https://elearning.vector.com/mod/page/view.php?id=63, Abruf: 25.02.2020

[217] Decker, Peter ; Garnatz, Oliver: CAN XL für zukünftige Fahrzeugarchitekturen. In: *HANSER automotive* 2020 (2020), Nr. 4-5, 40–43. https://assets.vector.com/cms/content/know-how/_technical-articles/CAN_XL_Introduction_Hanser_automotive_202007_Pressarticle_DE.pdf

[218] Colvin, Alan: CSMA with collision avoidance. In: *Computer Communications* 6 (1983), Nr. 5, S. 227–235. http://dx.doi.org/10.1016/0140-3664(83)90084-1. – DOI 10.1016/0140–3664(83)90084–1. – ISSN 01403664

[219] Röder, Jürgen: *Automotive Ethernet | VDI Wissensforum*. https://www.vdi-wissensforum.de/news/automotive-ethernet/, Abruf: 03.02.2020

[220] Hoedt, Jens: *Fahrdynamikregelung für fehlertolerante X-By-Wire-Antriebstopologien*. Darmstadt, Dissertation, 2013. https://tuprints.ulb.tu-darmstadt.de/3631/7/Dissertation_Hoedt.pdf

[221] Klauda, Matthias ; Schaffert, Michael ; Lagospiris, Athanassios ; Piel, Gunnar ; Kappel, Sven ; Ihle, Markus: Weichenstellung für 2020 Paradigmenwechsel in der E/E-Architektur. In: *ATZelektronik* 10 (2015), Nr. 2, S. 16–23

[222] Härter, Hendrik: *Die Entwicklung von Automotive Ethernet und was es im Fahrzeug leistet*. https://www.elektronikpraxis.vogel.de/die-entwicklung-von-automotive-ethernet-und-was-es-im-fahrzeug-leistet-a-706416/, Abruf: 24.08.2020

[223] Kozierok, Charles M.: *Automotive Ethernet: Definitive guide ; [TCP/IP, BroadR-Reach, Switch Technology, Real-Time Protocols, Audio Video Bridging, IEEE Physical Layers, Electromagnetic Compatibility et More]*. Ed. 1.2. Madison Heights : Intrepid Control Systems, 2014. – ISBN 978–0990538806

[224] IEEE 802.3bw-2015: *IEEE Standard for Ethernet Amendment 1: Physical Layer Specifications and Management Parameters for 100 Mb/s Operation over a Single Balanced Twisted Pair Cable (100BASE-T1)*. 2015

[225] ISO 15765-2:2016-04: *Road vehicles - Diagnostic communication over Controller Area Network (DoCAN) - Part 2: Transport protocol and network layer services*. 2016

[226] Elektronik Kompendium (Hrsg.): *Definition - Gateway*. https://www.elektronik-kompendium.de/sites/net/0901111.htm, Abruf: 26.02.2020

[227] ISO 10681-2:2010: *Road vehicles — Communication on FlexRay: Part 2: Communication layer services*. 2010

[228] ISO 15765-3:2004: *Diagnostics on Controller Area Networks (CAN): Part 3: Implementation of unified diagnostic services (UDS on CAN)*. 2004

[229] ISO 13400-2:2012: *Road vehicles - Diagnostic communication over Internet Protocol (DoIP): Part 2: Transport protocol and network layer services*. 2012

[230] ISO 27145-1:2012: *Road vehicles — Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements: Part 1: General information and use case definition*. 2012

[231] Kerckhoffs, Auguste: *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: Avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883

[232] Barker, Elaine ; Roginsky, Allen: *SP800-131A REV.2 - Transitioning the use of cryptographic algorithms and key lengths*. Gaithersburg, MD : National Institute of Standards and Technology. http://dx.doi.org/10.6028/NIST.SP.800-131Ar2. http://dx.doi.org/10.6028/NIST.SP.800-131Ar2

[233] Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): *Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Version 2019-01*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile, Abruf: 07.06.2020

[234] Schwenk, Jörg: *Sicherheit und Kryptographie im Internet: Theorie und Praxis*. 4., überarb. u. erw. Aufl. 2014. Springer Vieweg, 2014 http://dx.doi.org/10.1007/978-3-658-06544-7. – ISBN 978–3–658–06544–7

[235] Europäische Union (Hrsg.): *Verordnung (EU) Nr. 910/2014 des Europäischen Parlaments und des Rates vom 23. Juli 2014*

über elektronische Identifizierung und Vertrauensdienste für elektronische Transaktionen im Binnenmarkt und zur Aufhebung der Richtlinie 1999/93/EG. https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32014R0910&from=DE. Version: 2014, Abruf: 27.02.2020

[236] Rupp, Susanne: *Die Beweisführung mit privaten elektronischen Dokumenten*. Bd. 1. Nomos Verlag, 2018

[237] GlobalSign (Hrsg.): *Die GlobalSign PKI-Umfrage 2019*. https://www.globalsign.com/de-de/blog/globalsign-pki-umfrage-2019-ergebnisse/. Version: 2019, Abruf: 21.02.2020

[238] Al-Shaer, Ehab ; Hamed, Hazem: Design and implementation of firewall policy advisor tools. In: *DePaul University, CTI, Tech. Rep* (2002)

[239] AUTOSAR: *AUTOSAR Classic R19-11: Specification of Secure Onboard Communication*

[240] *CWE VIEW: Research Concepts*. https://cwe.mitre.org/data/definitions/1000.html. Version: 2020, Abruf: 21.07.2020

Eigene Publikationen

[BDR+20] Bolz, Robin ; Dürrwang, Jürgen ; Rumez, Marcel ; Sommer, Florian ; Kriesten, Reiner: Unterstützung der Security-Entwicklung von Fahrzeugen durch Angriffsanalysen. Version: 2020. https://www.hs-karlsruhe.de/fileadmin/hska/GOEM/Uebrige_Baeume/Baum_Forschung/Forschung_aktuell_2020_01.pdf. In: *Forschung aktuell 2020*. 2020, 28–33

[BRS+20] Bolz, Robin ; Rumez, Marcel ; Sommer, Florian ; Dürrwang, Jürgen ; Kriesten, Reiner: Enhancement of Cyber Security for Cyber Physical Systems in the Automotive Field Through Attack Analysis. In: *embedded world Conference 2020 Proceedings*. 2020

[DBR+18] Dürrwang, Jürgen ; Braun, Johannes ; Rumez, Marcel ; Kriesten, Reiner ; Pretschner, Alexander: Enhancement of Automotive Penetration Testing with Threat Analyses Results. In: *SAE International Journal of Transportation Cybersecurity and Privacy* 1 (2018), Nr. 2. http://dx.doi.org/10.4271/11-01-02-0005. – DOI 10.4271/11–01–02–0005. – ISSN 2572–1054

[DBRK17] Dürrwang, Jürgen ; Braun, Johannes ; Rumez, Marcel ; Kriesten, Reiner: Security Evaluation of an Airbag-ECU by Reusing Threat Modeling Artefacts. In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2017. – ISBN 978–1–5386–2652–8, S. 37–43

[DRBK17] Dürrwang, Jürgen ; Rumez, Marcel ; Braun, Johannes ; Kriesten, Reiner: Security Hardening with Plausibility Checks for Automotive ECUs. Version: 2017. http://www.thinkmind.org/download.php?articleid=vehicular_2017_2_40_30053. In: *VEHICULAR 2017* Bd. 6. 2017, 38–41

[NRT+21] Neubauer, Kevin ; Rumez, Marcel ; Tremmel, Huiying ; Hoppe, Augusto ; Kriesten, Reiner ; Nenninger, Philipp ; Sax, Eric

; Becker, Jürgen: Virtual Verification of E/E Architectures for Secure Automated Driving Functions. In: *7th IEEE International Symposium on Systems Engineering (ISSE 2021)*. 2021

[PRGS22] Puder, Andreas ; Rumez, Marcel ; Grimm, Daniel ; Sax, Eric: Generic Patterns for Intrusion Detection Systems in Service-Oriented Automotive and Medical Architectures. In: *Journal of Cybersecurity and Privacy* 2 (2022), Nr. 3, S. 731–749. http://dx.doi.org/10.3390/jcp2030037. – DOI 10.3390/jcp2030037

[PVR+22] Puder, Andreas ; Vetter, Andreas ; Rumez, Marcel ; Henle, Jacqueline ; Sax, Eric: A Mixed E/E-Architecture for Interconnected Operating Tables Inspired by the Automotive Industry. In: *2022 International Symposium on Medical Robotics (ISMR)*, IEEE, 2022. – ISBN 978–1–6654–6928–9, S. 1–8

[RDB+19] Rumez, Marcel ; Dürrwang, Jürgen ; Brecht, Tim ; Steinshorn, Timo ; Neugebauer, Peter ; Kriesten, Reiner ; Sax, Eric: CAN Radar: Sensing Physical Devices in CAN Networks based on Time Domain Reflectometry. In: *2019 IEEE Vehicular Networking Conference (VNC)*. 2019

[RDBK18] Rumez, Marcel ; Dürrwang, Jürgen ; Braun, Johannes ; Kriesten, Reiner: Security Hardening of Automotive Networks Through the Implementation of Attribute-Based Plausibility Checks. In: *International Journal on Advances in Security* 11 (2018), Nr. 1&2, 52–59. http://www.iariajournals.org/security/

[RDG+19] Rumez, Marcel ; Duda, Alexander ; Gründer, Patrick ; Kriesten, Reiner ; Sax, Eric: Integration of Attribute-based Access Control into Automotive Architectures. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 09.06.2019 - 12.06.2019. – ISBN 978–1–7281–0560–4, S. 1916–1922

[RGKS20] Rumez, Marcel ; Grimm, Daniel ; Kriesten, Reiner ; Sax, Eric: An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures. In: *IEEE Access* 8 (2020), S. 221852–221870. http://dx.doi.org/10.1109/ACCESS.2020.3043070. – DOI 10.1109/ACCESS.2020.3043070

[RLF+20] Rumez, Marcel ; Lin, Jinghua ; Fuchß, Thomas ; Kriesten, Reiner ; Sax, Eric: Anomaly Detection for Automotive Diagnostic Applications Based on N-Grams. In: *2020 IEEE 44th Annual*

Computers, Software, and Applications Conference (COMPSAC), IEEE, 13.07.2020 - 17.07.2020. – ISBN 978–1–7281–7303–0, S. 1423–1429

[SRS21] Schmidt, Kristina ; Rumez, Marcel ; Sommer, Florian: On the Development of Service-oriented Vehicle Networks based on CANoe. In: *Reports on Energy Efficient Mobility 2021*. 2021, S. 56–60

[SSG+22] Schindewolf, Marc ; Stoll, Hannes ; Guissouma, Houssem ; Puder, Andreas ; Sax, Eric ; Vetter, Andreas ; Rumez, Marcel ; Henle, Jacqueline: A Comparison of Architecture Paradigms for Dynamic Reconfigurable Automotive Networks. In: *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, IEEE, 2022. – ISBN 978–1–6654–1687–0, S. 1–7

[VOG+20] Vetter, Andreas ; Obergfell, Philipp ; Guissouma, Houssem ; Grimm, Daniel ; Rumez, Marcel ; Sax, Eric: Development Processes in Automotive Service-oriented Architectures. In: *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, 2020. – ISBN 978–1–7281–6949–1, S. 1–7

Betreute studentische Arbeiten

[ABZ18] Amann, Julian ; Breitschopf, Patrick ; Zerweck, Jonas: *Untersuchung und Bewertung von Firewall Modellierungstechniken sowie Sicherheitskonzepten für Serviceorientierte Architekturen*, Hochschule Karlsruhe - Technik und Wirtschaft, Projektarbeit, 2018

[Ama17] Amann, Julian: *Automatisierung einer experimentellen BUS-Analyse*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2017

[And17] Andjelkovic, Miroslav: *Entwicklung einer zustandsbasierten Firewall für automotive Netzwerke*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2017

[And21] Anderer, Fabian: *Entwicklung einer Android-App zur Steuerung von Fahrfunktionen*, Hochschule Karlsruhe, Projektarbeit, 2021

[BCG+17] Barthelmeß, Étienne ; Cinkara, Alpaslan ; Gadinger, Marcel ; Holl, Jonathan ; Klamm, Sarah: *Automotive Ethernet*, Hochschule Karlsruhe - Technik und Wirtschaft, Projektarbeit, 2017

[BS19] Brecht, Tim ; Steinshorn, Timo: *Erkennung von CAN-Teilnehmern auf Basis physikalischer Leitungseigenschaften*, Hochschule Karlsruhe - Technik und Wirtschaft, Projektarbeit, 2019

[Dud18] Duda, Alexander: *Entwicklung und Evaluierung eines verteilten Zugangskontrollsystems für automotive Netzwerke*, Hochschule Karlsruhe - Technik und Wirtschaft, Projektarbeit, 2018

[Grü17] Grünenwald, Justin: *Design und Implementierung einer serviceorientierten Fahrzeugkommunikation mittels SOME/IP auf der Basis von automotive Ethernet*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2017

[Gün22] Günster, Philipp: *Optimierung eines Intrusion-Detection-System- Ansatzes für Automotive-Diagnoseanwendungen*, Hochschule Karlsruhe - Technik und Wirtschaft, Masterthesis, 2022

[Hah21] Hahn, Niclas: *Evaluierung kontext-basierter Intrusion Detection-Ansätze für Automotive-Anwendungen*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2021

[Hän17] Hänle, Jan: *Integration, Darstellung und Steuerung von Diagnosebotschaften in einer Matlab-Umgebung*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2017

[Jei21] Jeitz, Roby: *Kontextbasiertes IDS anhand von Diagnoseprotokolle mittels neuronaler Netze*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2021

[Kol20] Kollross, Bastian: *Entwicklung eines Konzepts zur Generierung von Diagnosedatensets zum Testen von Intrusion Detection Systemen*, Hochschule Karlsruhe - Technik und Wirtschaft, Bachelorthesis, 2020

[Lin20] Lin, Jinghua: *Development & evaluation of a method for detecting insider attacks in diagnostic devices*, Hochschule Karlsruhe - Technik und Wirtschaft, Masterthesis, 2020

[SLM21] Simmann, Gabriel ; Lederer, Daniel ; Menking, Tobias: *Entwicklung einer Car-2-X Simulation und Aufbau eines Demonstrators*, Hochschule Karlsruhe, Projektarbeit, 2021

[SS21] Schmidt, Tim ; Schink, Alex: *Anbindung einer Fahrzeugsteuerungs-App an eine prototypische E/E-Architektur*, Hochschule Karlsruhe, Projektarbeit, 2021

Verzeichnisse

Abbildungsverzeichnis

Abkürzungsverzeichnis

UDS	Unified Diagnostic Services
OBD	On-Board Diagnostics
CAN	Controller Area Network
LIN	Local Interconnect Network
ECU	Electronic Control Unit
DoS	Denial of Service
IT	Informationstechnik
IP	Internet Protocol
OSI	Open Systems Interconnection
ID	Identifier
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
HTTP	Hypertext Transfer Protocol
SID	Service Identifier
LEV	Level Identifier
PID	Parameter Identifier
OEM	Original Equipment Manufacturer
PKI	Public Key Infrastructure
CA	Certification Authority
PCI	Protocol Control Information
NIST	National Institute of Standards and Technology
MAC	Media Access Control
ISO	International Organization for Standardization
ACC	Adaptive Cruise Control
SAE	Society of Automotive Engineers
ACC	Adaptive Cruise Control
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
AUTOSAR	AUTomotive Open System ARchitecture
SecOC	Secure Onboard Communication
PDU	Protocol Data Unit
	Message Authentication Code
EOL	End of Life
CVSS	Common Vulnerability Scoring System

RBAC	Role-based Access Control
ABAC	Attribute-based Access Control
MAC	Mandatory Access Control
DAC	Discretionary Access Control
SOA	service-orientierte Architektur
XACML	eXtensible Access Control Markup Language
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PIP	Policy Information Point
PAP	Policy Administration Point
BSI	Bundesamt für Sicherheit in der Informationstechnik
EU	Europäische Union
eCall	emergency Call
CPS	cyber-physisches System
SOME/IP	Scalable Service-Oriented Middleware over IP
IEC	International Electrotechnical Commission
OSI	Open Systems Interconnection model
XML	Extensible Markup Language
OASIS	Organization for the Advancement of Structured Information Standards
DLC	Data Length Code
ID	Identifier
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CAN FD	CAN Flexible Data-Rate
TDMA	Time Division Multiple Access
CRC	Cyclic Redundancy Check
ACK	Acknowledge
IEEE	Institute of Electrical and Electronics Engineers
SF	Single Frame
CF	Consecutive Frame
FC	Flow Control Frame
FF	First Frame
SDU	Service Data Unit
SID	Service Identifier
LEV	Subfunction Level
PID	Parameter Identifier

WWH-OBD	World Wide Harmonized On-Board-Diagnostic
HPC	High-Performance-Controller
ZCU	Zone Controller Unit
CA	Zertifizierungsstelle
HIDS	Host-basierte IDS
NIDS	Netzwerk-basierte IDS
E/E	Elektrik/Elektronik
ACL	Access Control List
WLAN	Wireless Local Area Network
RFID	Radio Frequency Identification
KNN	Künstlich Neuronalen Netz
GHSOM	Growing Hierarchical Self-Organizing Map
FPGA	Field Programmable Gate Array
IAM	Identity und Access Management
DDS	Data Distribution Service
IPSec	Internet Protocol Security
TLS	Transport Layer Security
ESP	Encapsulating Security Payload
AH	Authentication Header
UNECE	United Nations Economic Commission for Europe
LiDAR	Light Detection and Ranging
CapBAC	Capability-based Access Control
GPS	Global Positioning System
CVSS	Common Vulnerability Scoring System
OASIS	Organization for the Advancement of Structured Information Standards
ALFA	Abbreviated Language for Authorization
A-ABAC	Automotive Attribute-based Access Control
DoIP	Diagnostics over Internet Protocol
RPC	Remote Procedure Call
CL	Computerlinguistik
UEBA	User Entity Behavior Analytics
SLM	Statistical Language Model
MLE	Maximum Likelihood Estimation
KI	Künstliche Intelligenz
SIDS	Signatur-basiertes Intrusion Detection System

AIDS	Anomalie-basiertes Intrusion Detection System
VLAN	Virtual Local Area Network
Auto-ISAC	Automotive Information Sharing & Analysis Center
POSIX	Portable Operating System Interface
CCC	Central Computing Cluster
CAGR	Compound Annual Growth Rate
CSMS	Cyber Security Management System
SOC	Security Operations Center
SIEM	Security Information and Event Management
WER	Word Error Rate
SOTA	Software over-the-Air
VC	Vehicle Computer
HPC	High-Performance-Computer

Tabellenverzeichnis