Grundlagen

1

Zusammenfassung

Die zunehmende Komplexität der elektronischen Systeme im Fahrzeug sowie die Vernetzung mit der Außenwelt erhöhen das Risiko für Fahrzeuge, Ziel von Cyberangriffen zu werden. Automotive Cybersecurity identifiziert diese Risiken und führt Methoden und Maßnahmen ein, um sie zu reduzieren. In diesem Kapitel werden die grundlegenden Fachbegriffe und kryptographische Grundlagen eingeführt. Außerdem werden Bedeutung und Nutzen von Cybersecurity für den Automobilbereich erörtert und die größten Herausforderungen beleuchtet.

Die zunehmende Komplexität der elektronischen Systeme im Fahrzeug sowie die Vernetzung mit der Außenwelt erhöhen das Risiko für Fahrzeuge, Ziel von Cyberangriffen zu werden. Automotive Cybersecurity identifiziert diese Risiken und führt Methoden und Maßnahmen ein, um sie zu reduzieren.

In diesem Kapitel werden die wichtigsten Grundlagen über Security im Automobilbereich vermittelt. Zunächst werden die wichtigsten Fachbegriffe, die Schutzziele und kryptographischen Grundlagen eingeführt. Die Bedeutung von Cybersecurity für den Automobilbereich wird näher beleuchtet, indem auf vernetzte und automatisierte Fahrzeuge als aktuelle Entwicklungstrends eingegangen wird und indem erörtert wird, wie moderne Fahrzeuge aus der Perspektive möglicher Angreifer erscheinen.

Um ein möglichst vollständiges Bild von Automotive Security zu erhalten wird anhand eines Bedrohungsmodells und einer Aufstellung verschiedener Angreifertypen deren Motivation, Absichten und Fähigkeiten erläutert.

Schließlich wird auf verschiedene branchentypische Aspekte eingegangen, die die Automobilindustrie als Ganzes im Hinblick auf Cybersecurity vor große Herausforderungen stellen.

M. Wurm, *Automotive Cybersecurity,* https://doi.org/10.1007/978-3-662-64228-3_1

1.1 Security-Grundlagen

1.1.1 Fachterminologie und Zusammenhänge

1.1.1.1 Sicherheit

Unter dem Begriff Sicherheit versteht man im Allgemeinen die Gewissheit oder auch die Zuverlässigkeit, dass etwa für eine Person oder für ein Objekt keine Gefahr droht. Die Mehrdeutigkeit des deutschen Begriffs Sicherheit macht ihn allerdings ungeeignet für eine genaue, technische Definition.

Zur Verfeinerung des Sicherheitsbegriffs wird *Security* häufig mit dem Begriff *Zuverlässigkeit* in Beziehung gesetzt. Zuverlässigkeit (engl. dependability) ist ein Maß für die systematische Bewertung bestimmter Eigenschaften eines Systems – in diesem Zusammenhang eines Rechnersystems. Der Begriff *Zuverlässigkeit* beschreibt anhand mehrerer Merkmale das Vertrauen in ein System. [9] und [2] führen zur Begriffsklärung folgende *Attribute* an:

- Verfügbarkeit, im Sinne von Bereitschaft
- Ausfallsicherheit, im Sinne von Betriebszuverlässigkeit
- Safety, in Sinne von funktionaler Sicherheit
- Integrität, im Sinne von unzulässigen (engl. improper) Veränderungen
- Wartbarkeit

Bezogen auf die oben aufgeführten Eigenschaften deckt Security die Verfügbarkeit als auch die Integrität ab, wobei der Integritätsschutz aus der Security-Perspektive sowohl die ungewollten, zufälligen Veränderungen als auch alle beabsichtigten, böswillig gewollten Veränderungen (Angriffe) umfasst. Über die Zuverlässigkeitsattribute hinaus gehört die Vertraulichkeit zu den Standard-Schutzzielen von Security und wird ggf. um die Zurechenbarkeit bzw. Nicht-Abstreitbarkeit ergänzt.

Alles in allem stellt sich somit heraus, dass Security als alternativlose Systemeigenschaft zwingend erforderlich ist, um die Zuverlässigkeit und Sicherheit zu gewährleisten.

Im Kontext technischer Systeme eignen sich die engl. Begriffe Safety und Security besser für eine eindeutige Trennung der beiden Sachverhalte:

- Safety: Der Begriff Safety bezeichnet die funktionale Sicherheit, bzw. die Betriebssicherheit eines Systems. Ein System darf seine Umgebung etwa durch undefiniertes, unzulässiges Verhalten oder Zustände nicht gefährden. Safety schützt somit Mensch und Umwelt vor negativen Einflüssen des Systems, etwa durch Fehlverhalten und Ausfälle.
- Security: Der Begriff Security bezeichnet die Informations- und Datensicherheit bzw. die Angriffssicherheit eines Systems. Security umfasst alle Eigenschaften und Maßnahmen, die das System vor absichtlichen und unabsichtlichen Bedrohungen

von außen schützen. Security schützt somit das System vor negativen Einflüssen von Mensch und Umwelt, wie etwa Bedrohungen und Angriffe. Während sich die sog. klassische IT-Security auf die Absicherung der informationstechnischen Systeme eines Unternehmens wie etwa Computer, Server, Netzwerke und Internetanbindungen konzentriert, zielt die Cybersecurity im Kontext des Automotive Bereichs auf die Absicherung deren Produkte ab. In Fahrzeugen werden sog. *Deeply-Embedded Cyberphysical Systems* integriert, d. h. elektronische Systeme, die in mechatronischen Komponenten verbaut sind und spezifische Funktionen ausführen.

Security ist wie Safety eine eigenständige Disziplin und spielt für die Entwicklung von Fahrzeugen auch eine mindestens gleichwertige Rolle. Die jeweiligen Zielsetzungen erscheinen vollständig gegensätzlich, hinter beiden Disziplinen steht jedoch eine gemeinsame Intention, nämlich die Entwicklung robuster Systeme sowie die Prävention von Schäden.

In bestimmten Aspekten sind Safety und SOTIF sogar von Security abhängig, denn ohne Daten- und Informationssicherheit kann auch die Betriebssicherheit nicht gewährleistet werden. Ohne den Schutz der Security könnte ein Angreifer das System manipulieren und in einen Safety-kritischen Zustand bringen. Für die gesellschaftliche Akzeptanz automatisierter und autonom fahrender Fahrzeuge sind *sichere* Systeme im Sinne von *safe und secure* eine Voraussetzung. Security greift dabei i. d. R. auf wirksamere Werkzeuge zurück als Safety. Kryptographische Prüfsummen oder Hashwerte zur Absicherung der Integrität sind beispielsweise aufgrund ihrer Einweg-Bedingung sowohl gegen zufällige als auch gegen absichtliche Manipulationen geschützt, wohingegen CRC-Prüfsummen, wie sie typischerweise bei Safety-Integritätsschutzmaßnahmen zum Einsatz kommen, leicht rekonstruiert werden können, s. [31]. Zugleich sind Safety-Maßnahmen statisch und auf ein bestimmtes System ausgelegt, d. h. sie werden in aller Regel nach der Entwicklung nur noch für Fehlerkorrekturen angepasst. Security ist dagegen dynamisch und muss sich weit über die Entwicklungsphase hinaus an die sich stetig verändernden Bedrohungsszenarien anpassen.

Ansätze für eine gemeinsame Vorgehensweise von Safety und Security während der Entwicklung, das sog. Co-Engineering, wurden in [1] untersucht. Auf der Grundlage des Safety-Standards ISO 26262 [11] für die funktionale Sicherheit von Straßenfahrzeugen, sowie des Security-Standards SAE J3061 [25] schlugen Amorim et al. eine Vorgehensweise vor, wie „eigenständige, aber sich gegenseitig beeinflussende Disziplinen“ miteinander verknüpft werden könnten. Diese Verknüpfung beruht im Wesentlichen auf der Anwendung gemeinsamer bzw. kombinierter Design-Patterns. Eine gegenseitige Beeinflussung sollte dabei möglichst gering bleiben, um nicht etwa durch die Implementierung von Safety-Maßnahmen neue Security-Schwachstellen einzuführen – oder umgekehrt.

Skoglund et al. [28] kamen in ihrer Arbeit zu einem etwas anderen Ergebnis. Für die Designphase sahen sie nur wenige Gemeinsamkeiten. Der Vorteil des Co-Engineerings, d. h. der miteinander verwobenen Entwicklungsprozesse für Safety und Security beschränkte sich ihrer Ansicht nach auf das geringere Risiko für Lücken, etwa durch

fehlende Abstimmung und Informationsaustausch. Ein höheres Synergiepotential wird in der Verifikationsphase vermutet, etwa durch die (Wieder-)Verwendung gemeinsamer Testumgebungen und Testmethoden.

Trotz aller Unterschiede, Gemeinsamkeiten und möglicher Synergien dürfen und werden beide Disziplinen niemals von ihren jeweils eigenen Zielsetzungen abkommen. Da Safety im Automotive Bereich schon viel länger ein Thema und deshalb besser etabliert ist als Security, ist es umso kritischer, beide Disziplinen unter einen Hut zu stecken. Zur Förderung eines besseren Verständnisses sowie zur Stärkung einer Security-Kultur ist eine konsequente Abgrenzung bis auf weiteres ratsam.

1.1.1.2 Authentisierung, Authentifizierung, Autorisierung

Die Begriffe Authentisierung, Authentifizierung und Autorisierung werden häufig miteinander verwechselt. Im Bereich der Zugangskontrolle elektronischer Systeme, etwa des Diagnosezugangs einer ECU, besteht ein starker Zusammenhang zwischen diesen Begriffen. Umso wichtiger ist das Verständnis für die einzelnen Vorgänge und deren Bedeutung für die Security eines Systems.

Anhand des folgenden Szenarios werden die Begriffe näher erklärt: Ein Diagnosetester (Client) möchte eine Diagnosesitzung mit einem angeschlossenen Steuergerät (Server) starten, und auf Dienste zugreifen, die nur für bestimmte Benutzergruppen zugänglich sind.

Der Client authentisiert sich, indem er einen Nachweis seiner Identität an den Server übergibt. Dies kann etwa in Form einer individuellen, vertraulichen Information, z. B. eines Passworts, oder in Form eines Public-Key-Zertifikats geschehen. Der Server authentifiziert den Client, indem er die Echtheit und Glaubwürdigkeit (Authentizität) des Identitätsnachweises überprüft. Dieser Vorgang kann durch Überprüfung des Passworts bzw. durch Überprüfung der Signatur(-kette) des Zertifikats realisiert werden.

Nachdem der Client erfolgreich identifiziert und authentifiziert wurde, kann ihm der Server Zugriff auf Informationen und Dienste gewähren. Zuvor muss jedoch noch überprüft werden, ob der Client über die entsprechenden Berechtigungen verfügt, d. h. für die jeweiligen Dienste autorisiert wurde. Die Autorisierung ist die Vergabe von Berechtigungen für bestimmte Informationen, Zugänge, Dienste, etc. und für eine bestimmte Identität bzw. für bestimmte Gruppen. Technisch wird dies entweder über vordefinierte Gruppenberechtigungen oder über Rechtezertifikate gelöst.

Vergleich: Grenzkontrolle bei der Einreise

Die einreisende Person weist sich mit ihrem Reisepass bei der Grenzkontrolle aus (Authentisierung). Die Grenzbeamtin prüft die Echtheit dieses Identitätsnachweises, etwa anhand fälschungssicherer Merkmale, und gleicht das enthaltene Lichtbild mit dem Gesicht der Person ab (Authentifizierung). Nach der erfolgreichen Authentifizierung erfolgt die Autorisierung, d. h. es wird geprüft, ob die Person etwa aufgrund der Staatsangehörigkeit oder aufgrund eines ausgestellten Visums zur Einreise berechtigt ist.

1.1.1.3 Begriffe des Risikomanagements und deren Zusammenhänge

Ein essenzieller Bestandteil des Cybersecurity-Engineering-Prozesses [13] ist der systematische Umgang mit Gefahren und Risiken. Was ist im Zusammenhang mit Cybersecurity-Angriffen unter Bedrohungen, Schwachstellen und Risiken zu verstehen? Anhand des folgenden Narrativs werden die wichtigsten Begriffe des Risikomanagements und deren Zusammenhänge erklärt.

Im Mittelpunkt des Risikomanagements stehen die Assets des betrachteten Systems. Nach ISO 21434 kann ein kompromittiertes Asset Schäden für dessen Besitzer hervorrufen. Zu den Assets bzw. schützenswerten Gütern zählen unterschiedliche materielle und immaterielle Bestandteile des Systems, etwa sensible Informationen, Fahrzeugfunktionen, Softwarekomponenten, Hardwarekomponenten, Infrastrukturkomponenten und Kommunikationsverbindungen. Die verschiedenen, standardisierten Vorgehensmodelle für die Risikobewertung, s. [12, 24] und [7], legen für die einzelnen Assets ihre jeweiligen Security-Eigenschaften wie Vertraulichkeit, Integrität und Verfügbarkeit fest und definieren dadurch ihren Schutzbedarf.

Eine Bedrohung (engl. threat) ist eine konkrete, drohende Gefahr, die auf ein bestimmtes Asset ausgerichtet ist und ihre Schutzziele beeinträchtigen kann. Zu Bedrohungen zählen nicht nur absichtliche, boshafte Aktivitäten von Angreifern, etwa Manipulation von Informationen oder nicht-autorisierte Zugriffe, sondern auch unabsichtliche und unvorhersehbare Ereignisse, wie etwa Ausfälle der Mobilfunkverbindung oder physische Beschädigungen durch Vandalismus.

Angreifer (engl. attacker) führen Angriffe durch, indem sie vorhandene Schwachstellen (engl. vulnerabilities) ausnutzen. Sie bedrohen damit die Assets eines Systems. Häufig werden passive von aktiven Angriffen unterschieden. Passive Angriffe dienen der Informationsbeschaffung und sind gegen den Schutz der Vertraulichkeit und Privatsphäre gerichtet. Aktive Angriffe sind gegen die Integrität, Authentizität und Verfügbarkeit des Systems gerichtet und nutzen Möglichkeiten zur Manipulation von Daten und Abläufen.

Die Risiken einer Bedrohung werden durch Schwachstellen erhöht und von Gegenmaßnahmen und Schutzkonzepten reduziert. Das Risiko einer Bedrohung ist abhängig von der Wahrscheinlichkeit, dass ein bestimmter Angriff erfolgreich durchgeführt wird, und vom Schaden, der im schlimmsten Falle daraus entsteht. Im Rahmen einer Risikoanalyse werden die Eintrittswahrscheinlichkeiten der einzelnen Bedrohungen (Angriffsvektoren) ermittelt und die möglichen Auswirkungen eines erfolgreichen Angriffs auf das jeweilige Asset bewertet.

Gegenmaßnahmen (engl. security controls) verhindern mögliche Bedrohungen und reduzieren die Eintrittswahrscheinlichkeiten der Angriffsvektoren auf ein akzeptables Niveau. Gegenmaßnahmen werden als Security-Bausteine in einem für das jeweilige System maßgeschneiderten Security-Konzept strukturiert, was in diesem Buch ausführlich untersucht wird.

Als Angreifer kommt eine große Bandbreite verschiedener Personengruppen in Betracht – von einfachen Hobby-Hackern mit relativ beschränkten Mitteln und Kenntnissen über mit z. T. öffentlichen Mitteln geförderten Security-Forschern bis hin zu

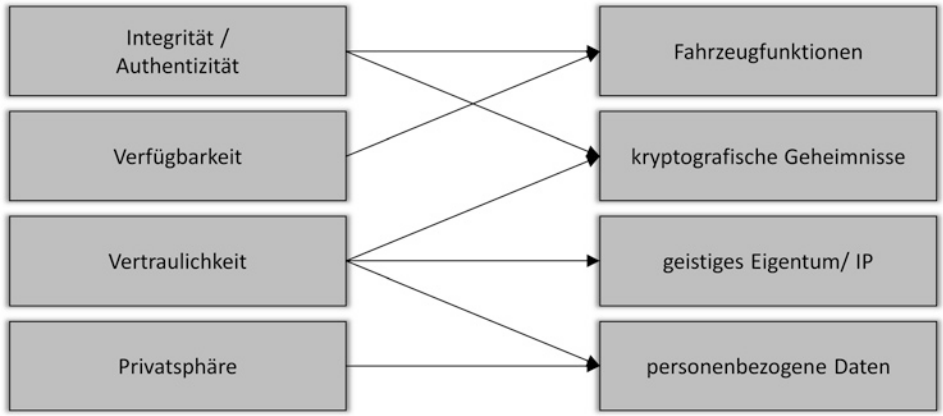

Abb. 1.1 Schutzziele mit Zuordnung zu Assets

staatlichen Organisationen wie Geheimdiensten, die Angriffe etwa zu Überwachungszwecken, Industriespionage oder Cyber-Kriegsführung ausführen. Häufig werden Angriffe jedoch vom aktuellen Fahrzeugbesitzer durchgeführt. Beispielsweise kann durch das illegale „Zurückdrehen" des Kilometerstandes der Wiederverkaufswert eines Fahrzeugs erhöht werden.

1.1.2 Schutzziele

Was sind Schutzziele?
Schutzziele legen die Security-Eigenschaften eines Assets fest – vor dem Hintergrund eines Bedrohungsszenarios, vor dessen Gefahren das Asset bewahrt werden soll, vgl. [13]. Basierend auf den Schutzzielen werden die erforderlichen Security-(Gegen-) Maßnahmen abgeleitet, sodass die Assets keinem oder nur einem akzeptabel geringen Risiko durch Cybersecurity-Bedrohungen ausgesetzt sind.

Die allgemeinen, klassischen Schutzziele der Kryptographie können in drei Kategorien aufgeteilt werden: Vertraulichkeit, Integrität und Verfügbarkeit. Im Embedded-/ Automotive-Bereich wird dieses Dreigestirn häufig um das Schutzziel Authentizität ergänzt. Nicht alle Schutzziele sind für jedes Asset relevant, s. Abb. 1.1. Zu diesen klassischen IT-Security-Schutzzielen kommt der Schutz der Privatsphäre sowie der Schutz, bzw. die Garantie der Zurechenbarkeit. Letzteres spielt allerdings nur für bestimmte Anwendungsfälle eine Rolle.

Für Fahrzeugfunktionen wie Motorsteuerung und X-by-Wire-Funktionen sind Integrität, Authentizität und Verfügbarkeit relevant. Kryptographische Geheimnisse müssen vor allem vertraulich behandelt werden, aber auch deren Integrität ist wichtig. Für den

Schutz geistigen Eigentums ist primär deren Vertraulichkeit von Belang. Bei personenbezogenen Daten müssen darüber hinaus Maßnahmen zur Gewährleistung der Privatsphäre getroffen werden.

Beschreibung der Schutzziele und deren technischen Umsetzung

- *Vertraulichkeit* beschreibt die Eigenschaft, dass ausschließlich berechtigte Personen bzw. Entitäten auf die zu schützenden Informationen zugreifen können. Dabei ist die Zugriffskontrolle nur eine Möglichkeit der technischen Realisierung. Verschlüsselung, d. h. die Abbildung des Klartextes in einen Ciphertext, sowie das Verstecken von Informationen in einem sog. verdeckten Kanal sind weitere Möglichkeiten. Zu den schützenswerten Informationen zählen etwa geistiges Eigentum und Firmengeheimnisse wie die ECU-Software, sowie kryptographische Geheimnisse wie Schlüsselmaterial. Darüber hinaus zählen auch personenbezogene Daten und Informationen, die die Pseudonymität oder Anonymität von Personen gefährden könnten, zu den vertraulichen Daten.
- *Integrität* beschreibt den Schutz von Informationen vor unbeabsichtigten oder böswilligen Veränderungen. Unbeabsichtigte Veränderungen können etwa durch Fehler bei der Übertragung oder Speicherung auftreten, wohingegen böswillige Veränderungen die absichtliche Manipulation eines Angreifers voraussetzt. Integrität schließt die Vollständigkeit mit ein, sodass auch ein Entfernen oder Hinzufügen von Informationen als Verstoß gegen diese Eigenschaft erkannt werden würde. Technisch wird die Integrität schützenswerter Informationen etwa mittels kryptographischer Checksummen geprüft. So kann ein Verlust der Integrität zwar nicht verhindert, aber zumindest zweifelsfrei und nicht kompromittierbar erkannt werden. Für praktisch die gesamte ECU-Software und insbesondere für safetyrelevante Informationen spielt der Schutz deren Integrität für eine korrekte Funktionsweise eine entscheidende Rolle.
- *Authentizität* bedingt, dass die Echtheit einer Information bzw. eines Absenders sichergestellt ist. Die Authentizität einer Information ist gegeben, falls dessen Urheber eindeutig identifizierbar und dessen Urheberschaft kryptographisch sicher überprüfbar ist. Authentizität wird im Kontext der Bedrohungsanalysen häufig in Kombination mit Integrität behandelt, weil sich deren technischen Lösungen gleichen oder überlappen. So kann mithilfe einer digitalen Signatur die Authentizität einer Information überprüft werden, indem die Signatur anhand des öffentlichen Schlüssels des Urhebers verifiziert wird. Eine erfolgreiche Authentizitätsprüfung impliziert stets auch die Integrität der jew. Informationen, weil eine verletzte Integrität zwangsläufig ein Scheitern der Authentizitätsprüfung nach sich ziehen muss.
- *Verfügbarkeit* (engl. availability) definiert die Anforderung an das System, seine Dienste und Funktionen innerhalb einer gewissen Zeitspanne (Echtzeitfähigkeit) nach Aufforderung zur Verfügung stellen zu können. Hardware-, Software- und Kommunikationsressourcen müssen abrufbereit sein, damit ein korrekter und unmittelbarer Betrieb gewährleistet werden kann. Oftmals ist die Einschränkung oder sogar vollständige Verweigerung des Dienstes (engl. denial of service, DoS) das Ziel

eines Angreifers. Eine häufige technische Lösung zum Aufrechterhalten der Verfügbarkeit ist die Implementierung redundanter Pfade. Ein redundantes (Teil-)System kann bei einem Ausfall die Aufgabe des Primärsystems übernehmen und so die Verfügbarkeit sicherstellen.

- *Zurechenbarkeit* bzw. Verbindlichkeit (engl. accountability) ist eine Eigenschaft, die dafür garantiert, dass die entsprechende Person oder Entität die Urheberschaft einer bestimmten Information bzw. eine bestimmte Aktion nicht von sich weisen kann. Der Begriff Nichtabstreitbarkeit (engl. non-repudiability) wird oftmals sinnverwandt benutzt, spielt jedoch insbesondere bei rechtlichen Sachverhalten wie Haftbarkeit und Gewährleistung eine Rolle. Im Kontext der V2X-Kommunikation wurde die Abstreitbarkeit im Rahmen der Bedrohungsanalyse als mögliche Schwachstelle identifiziert, s. Abschn. 5.4.2. Falls etwa der Empfang oder das Absenden bestimmter kritischer Botschaften, beispielsweise Stauende-Warnungen oder Geschwindigkeitsbegrenzungen, abgestritten werden kann, ist eine rechtssichere Zurechenbarkeit nicht möglich und damit eine eventuelle Strafverfolgung erschwert. Bei fehlendem Schutzziel Nicht-Abstreitbarkeit könnte jeder V2X-Teilnehmer abstreiten, eine bestimmte Botschaft abgesendet oder empfangen zu haben. Die technische Umsetzung kann mittels manipulationssicherer Log-Speicher erfolgen, anhand derer der Empfang bestimmter Botschaften protokolliert und nachgewiesen werden kann. Die Nicht-Abstreitbarkeit der Urheberschaft einer abgesendeten Botschaft ist durch das digitale Signaturverfahren in Verbindung mit einer vertrauenswürdigen Public-Key-Infrastruktur gewährleistet.
- Der Schutz der Privatsphäre (engl. privacy) wurde spätestens seit Inkrafttreten der DSGVO im Jahre 2018 zu einer ernst zu nehmenden Aufgabe für alle Automobilhersteller. Im Fahrzeug bzw. in der Teilnahme des Straßenverkehrs werden personenbezogene Daten erhoben, verarbeitet und z. T. auch gespeichert. Im Rahmen der V2X-Kommunikation veröffentlichen Fahrzeuge zyklisch ihre Positionsdaten, was bei fehlender Pseudonymität oder Anonymität das Erstellen von Bewegungsmustern der Insassen bzw. Fahrer erlauben würde. Unter Pseudonymität versteht man die Zuordnung einer Identität zu einer *Tarnidentität,* dem Pseudonym. Dabei gilt es, das Wissen über diese Zuordnung besonders zu schützen, weil damit die reale Identität aufgedeckt werden könnte. Darüber hinaus ist auch die Vergabe und Zuordnung mehrerer Pseudonyme zu einer einzigen Entität möglich, vgl. Abschn. 5.4.2, sodass bspw. ein Pseudonym nur für einen kurzen begrenzten Zeitraum benutzt wird und somit die Verfolgung und Verhaltensanalyse auf diesen Zeitraum beschränkt wird.

1.1.3 Kryptographische Grundlagen

1.1.3.1 Kryptographie

Was ist Kryptographie?
Der Begriff Kryptographie stammt vom griechischen *kryptos* (verborgen) und *graphein* (schreiben) ab und bezeichnet die Wissenschaft der Geheimschriften oder der Verschlüsselung von Informationen.

Kryptosystem und Sicherheit
Ein Kryptosystem besitzt eine kryptographische Funktion, die einen *Klartext* (=unverschlüsselter Text) in einen *Geheimtext* übersetzt (=Chiffre). Die Entschlüsselungsfunktion ist die Umkehrfunktion der Verschlüsselung. Mit ihrer Hilfe kann aus dem Geheimtext wieder der ursprüngliche Klartext berechnet werden.

Die Sicherheit antiker, einfacher Kryptosysteme, bzw. Chiffren, beruhte auf der Geheimhaltung des Verschlüsselungsverfahrens. Dieses Prinzip wird auch *Security by Obscurity* oder auf Deutsch etwa *Sicherheit durch Unklarheit* genannt. Die Sicherheit moderner Kryptosysteme beruht dagegen auf der Veröffentlichung und ausführlichen Prüfung der Verfahren, sodass Schwächen und Hintertüren vorab möglichst vollständig ausgeschlossen werden können. Gemäß Auguste Kerkhoffs' Prinzip darf die Sicherheit eines Kryptosystems lediglich auf der Geheimhaltung des geheimen Schlüssels beruhen. Zudem muss es aufgrund der schieren Anzahl der Schlüssel praktisch unmöglich sein, den geheimen Schlüssel durch systematisches Ausprobieren (Brute-force-Angriff) herauszufinden.

Beispiel

Die sog. *Cäsar-Chiffre* zählt zu den bekannten Vertretern klassischer Kryptosysteme. Sie diente Gaius Julius Cäsar gemäß historischen Überlieferungen zur Verschlüsselung seines Schriftverkehrs. Für die Verschlüsselung des Klartextes verschiebt sie jeden einzelnen Klartext-Buchstaben um eine vorgegebene Anzahl von Positionen im Alphabet. Die Anzahl dieser Verschiebeschritte definiert den geheimen Schlüssel. Mit dem Schlüssel "3" wird etwa A durch D und Z durch C ersetzt. Für die Entschlüsselung wird die Verschiebung in umgekehrter Richtung ausgeführt.

Die Cäsar-Chiffre bzw. deren Verallgemeinerung, die sog. monoalphabetische Substitution, besitzt gleich mehrere fundamentale Schwächen, die zu einem kryptoanalytischen Angriff einladen. Zum einen besteht der Schlüsselraum aus nur 26 Schlüsseln[1], was ein systematisches Durchprobieren aller Schlüssel selbst ohne

[1] Der Einfachheit halber werden zu Demonstrationszwecken die Alphabete der Klartext- und Ciphertexträume und damit auch der Schlüsselraum auf die 26 lateinischen Großbuchstaben beschränkt.

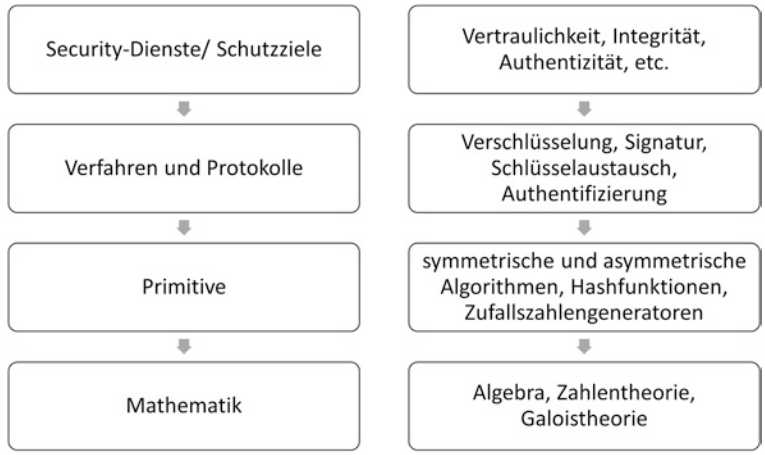

Abb. 1.2 Securitymodell nach ISO

technische Hilfsmittel zulässt. Zum anderen bleibt die statistische Buchstabenverteilung bei der monoalphabetischen Substitution unverändert. Da bei der Verschlüsselung jeder Buchstabe – abhängig vom Schlüssel – immer durch einen anderen, bestimmten Buchstaben ersetzt wird, ändert sich insgesamt an der Buchstabenhäufigkeit nichts. In Folge dessen kann mithilfe einer Häufigkeitsanalyse die Verschlüsselung (zumindest teilweise) rückgängig gemacht werden. ◀

In Anlehnung an das ISO-Securitymodell [10] werden mathematische Grundlagen, kryptographische Algorithmen und Verfahren unterschiedlichen Modellebenen zugeordnet, s. Abb. 1.2.

Auf der untersten Ebene befindet sich das mathematische Fundament der darüberliegenden kryptographischen Algorithmen. So bauen die kryptographischen Algorithmen auf verschiedene Theorien, Grundsätze und Probleme der Algebra (Fundamentalsatz), der Zahlentheorie (Faktorisierungsproblem) oder der Galoistheorie (diskreter Logarithmus) auf. Auf der Ebene der Algorithmen und Primitive sind die grundlegenden Bausteine wie Block- und Stromchiffren, RSA, Hashfunktionen und Zufallszahlengeneratoren definiert. Auf deren Basis werden auf der Ebene der kryptographischen Verfahren und Protokolle komplexere Security-Mechanismen entwickelt, etwa Verschlüsselungsverfahren, Signaturverfahren, Schlüsselaustauschprotokolle und Authentifizierungsverfahren. Ganz oben sind die von Bedrohungsanalysen motivierten Schutzziele wie Vertraulichkeit, Integrität, Authentizität und Verfügbarkeit definiert, die durch die Anwendung der geeigneten Verfahren und Protokolle erreicht werden können.

1.1.3.2 Prinzipien der symmetrischen und asymmetrische Kryptographie

1.1.3.2.1 Symmetrische Kryptographie

Zur immanenten Aufgabe der Kryptographie zählt die Verschlüsselung von Informationen. Intuitives und über mehrere Jahrhunderte hinweg alternativloses Vorgehen ist die sog. *symmetrische Verschlüsselung*. Aus der symmetrischen Eigenschaft folgt, dass sowohl für die Verschlüsselung als auch für die Entschlüsselung der gleiche Schlüssel verwendet wird.

Dieser Ansatz ähnelt der Benutzung eines Tresors. Analog zur Datenverschlüsselung kann eine Tresortüre mit einem Schlüssel verschlossen und mit dem gleichen Schlüssel wieder aufgeschlossen werden. Nur wer in Besitz des Schlüssels ist, kann die Tresortüre öffnen bzw. die Nachricht entschlüsseln.

1.1.3.2.2 Asymmetrische Kryptographie

Als Alternative zur symmetrischen Kryptographie kam in der Geschichte der Kryptographie erst vergleichsweise spät, und zwar in den 1970er Jahren die Entwicklung asymmetrischer Verfahren ins Spiel.

Die sog. *Public-Key-Verschlüsselungsverfahren* sehen die Verwendung von zwei verschiedenen Schlüsseln vor. Ein öffentlicher Schlüssel und ein privater, geheimer Schlüssel bilden dabei ein Schlüsselpaar. Mit dem öffentlichen Schlüssel wird ein Klartext in den Geheimtext umgewandelt (verschlüsselt) und (nur) mit dem privaten, geheimen Schlüssel kann der Geheimtext wieder in den Klartext zurücktransformiert (entschlüsselt) werden.

Die asymmetrische Kryptographie kann weitgehend in Analogie zum Briefkastenprinzip erklärt werden: Jeder kann einen Brief in den Kasten einwerfen, aber nur der Besitzer kann mit seinem (privaten) Schlüssel den Briefkasten öffnen und den Inhalt lesen. Nur eine Seite benötigt den geheimen, privaten Schlüssel.

Sogenannte Einwegfunktionen stellen in asymmetrischen Verfahren sicher, dass die mit dem öffentlichen Schlüssel durchgeführte Transformation nicht umkehrbar ist. Ansonsten könnte der Geheimtext mit Kenntnis des öffentlichen Schlüssels entschlüsselt werden. Die zentrale Anforderung an Einwegfunktionen ist, dass sie in eine Richtung einfach und schnell, in die umgekehrte Richtung aber praktisch nicht berechnet werden können. Bei asymmetrischen Verfahren werden deshalb Einwegfunktionen mit sog. Falltüren (engl. trapdoor) eingesetzt. Bezogen auf die asymmetrischen Kryptosysteme bedeutet das, dass sich mit Kenntnis des geheimen Schlüssels, die Umkehrfunktion (und damit die Entschlüsselung der Geheimtexte) durchaus effizient berechnen lässt.

Es existieren zwei Klassen von Einwegfunktionen, die auf unterschiedlichen mathematischen Problemen basieren:

- Das *Faktorisierungsproblem* ganzer Zahlen: Ein Produkt aus zwei Primzahlen, d. h. eine Multiplikation, ist einfach zu berechnen. In umgekehrter Richtung besteht die Aufgabe, aus der zusammengesetzten Zahl deren Teile, bzw. aus dem Produkt die Primfaktoren zu ermitteln. Obwohl für dieses Faktorisierungsverfahren einige Algorithmen existieren, etwa das sog. *Quadratische Sieb,* ist kein Verfahren bekannt, das mit heutigen Rechnersystemen in endlicher Zeit zur Lösung kommt und die Aufgabe effizient berechnen kann, s. *Hintergrundinformationen zu Post-Quantum Computing.*
- Das *diskrete Logarithmusproblem*: Ein weiteres mathematisches Problem, das als Einwegfunktion Verwendung findet, ist der diskrete, d. h. ganzzahlige Logarithmus. Während die diskrete Exponentialfunktion vergleichbar einfach zu berechnen ist, ist für dessen Umkehrfunktion kein effizientes Verfahren bekannt. Das diskrete Logarithmusproblem ist Grundlage verschiedener kryptographischer Verfahren, u. a. des Diffie-Hellman-Schlüsselaustauschs und der Elliptischen-Kurven-Kryptosysteme, s. unten.

1.1.3.2.3 Unterschiede in der Schlüsselverwaltung

Verschlüsselungsverfahren sorgen auf der einen Seite zwar für die Vertraulichkeit einer Information, erzeugen auf der anderen Seite jedoch Aufwände und Probleme hinsichtlich der Schlüsselverwaltung.

Ein wesentlicher Unterschied zwischen symmetrischen und asymmetrischen Verfahren liegt hierbei in der Anzahl der erforderlichen Schlüssel.

Symmetrische Verfahren: Für eine vertrauliche Kommunikation zwischen n Teilnehmern benötigt jeder Teilnehmer n-1 Schlüssel, um mit jedem anderen Teilnehmer einen vertraulichen Kanal aufzubauen. Insgesamt beträgt die Anzahl der erforderlichen Schlüssel $(n/2) * (n - 1)$.[2] Die Schlüsselanzahl wächst demnach quadratisch mit der Teilnehmerzahl und wird deshalb schnell nicht mehr beherrschbar.

Asymmetrische Verfahren: Für eine vertrauliche Kommunikation benötigt jeder Teilnehmer ein Schlüsselpaar, jeweils bestehend aus einem öffentlichen und einem privaten Schlüssel. Insgesamt beträgt die Anzahl der erforderlichen Schlüssel 2*n. Diese nur linear mit der Teilnehmerzahl ansteigende Schlüsselzahl ist weniger aufwendig und leichter beherrschbar.

Ein weiterer Unterschied besteht in den Anforderungen an den Schlüsselaustausch. Geheime Schlüssel müssen über einen vertraulichen und manipulationsgeschützten Kanal ausgetauscht werden, wohingegen öffentliche Schlüssel lediglich über einen manipulationsgeschützten Kanal ausgetauscht werden müssen.

[2] Jeder der n Teilnehmer benötigt zunächst n-1 Schlüssel. Für symmetrische Verfahren wird auf beiden Seiten jedoch der gleiche Schlüssel benötigt, deshalb der Faktor 0,5.

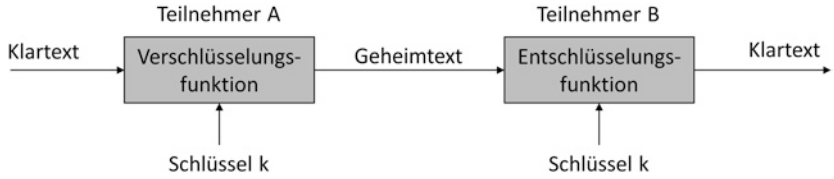

Abb. 1.3 Symmetrische Kryptographie

1.1.3.3 Kryptographische Verfahren und Protokolle

1.1.3.3.1 Symmetrische Kryptosysteme

Symmetrische Kryptosysteme werden unterteilt in Stromchiffre und Blockchiffre. Bei Stromchiffren werden einzelne Elemente des Klartexts, etwa Bits oder Bytes, ver- bzw. entschlüsselt. Bei Blockchiffren erfolgt die Ver- bzw. Entschlüsselung blockweise, d. h. mehrere Elemente des Klartextes, beispielsweise 128 Bit, werden zusammengefasst und in einem Durchgang verarbeitet.

Ein Klartext wird vom Teilnehmer A mit einer Verschlüsselungsfunktion und dem geheimen Schlüssel k verschlüsselt und als Geheimtext an Teilnehmer B übermittelt, s. Abb. 1.3. Teilnehmer B kann den Geheimtext mit der Entschlüsselungsfunktion und dem gleichen Schlüssel k wieder in den Klartext umwandeln bzw. entschlüsseln.

Funktionsweise und Eigenschaften von Stromchiffren

Der Klartext wird durch eine bitweise Verknüpfung des Klartextstroms mit einem Schlüsselstrom verschlüsselt. Ein *Schlüsselstromgenerator* erzeugt aus einem (endlichen) Schlüssel einen pseudozufälligen Bitstrom, den Schlüsselstrom, der zur Verschlüsselung genutzt wird. Im Unterschied zu den Blockchiffren verwenden Stromchiffren relativ einfache Verknüpfungsoperationen wie etwa bitweises XOR. Zur Entschlüsselung muss anhand des gleichen, geheimen Schlüssels der gleiche Schlüsselstrom erzeugt werden.

Die Sicherheit von Stromchiffren hängt maßgeblich von der Güte der verwendeten Zufallszahlen, d. h. des Schlüsselstroms ab. Idealerweise würde anstatt eines Pseudozufallsgenerators ein sog. One-Time-Pad verwendet werden. Ein One-Time-Pad ist prinzipiell ein Schlüssel, der aus einer Folge echter Zufallswerte besteht. Jeder Zufallswert wird nur für die Verschlüsselung eines einzigen Klartextzeichens, bzw. im umgekehrten Fall für die Entschlüsselung eines einzigen Geheimtextzeichens verwendet– daher die Bezeichnung *One-Time*-Pad. Dieses Verschlüsselungsverfahren ist informationstheoretisch sicher und kryptoanalytisch nicht brechbar, denn ein Durchprobieren aller Schlüssel würde als Ergebnis neben dem korrekten Klartext auch alle anderen sinnvollen und sinnlosen Texte derselben Länge ergeben. Der Angreifer hätte allerdings keine Möglichkeit, aus der Menge aller möglichen Texte den korrekten Text

zu erkennen. Der Nachteil des One-Time-Pads ist, dass der Schlüssel die gleiche Länge wie die Nachricht besitzen muss.

Ein Vorteil von Stromchiffren ist deren geringe Verzögerung, denn jedes eintreffende Klartextzeichen kann ohne Zwischenspeicherung verschlüsselt und weitergeleitet werden. Außerdem wirken sich (zufällige) Störungen im verschlüsselten Geheimtext, d. h. einzelne Bitfehler, die etwa bei der Übertragung durch Störungen verfälscht wurden, nach der Entschlüsselung auch nur auf die entsprechenden Bits im Klartext aus. Stromchiffren eigenen sich deshalb für die Anwendung im Mobilfunkbereich und für Übertragungen von Audiodaten.

Funktionsweise und Eigenschaften von Blockchiffren

Blockchiffren verarbeiten Blöcke fester Länge, d. h. n Klartextbits werden in einem Block gleichzeitig mit einem geheimen Schlüssel verschlüsselt. Typische Blocklängen variieren abhängig vom Verschlüsselungsverfahren zwischen 64 Bit und 256 Bit.

Der Mathematiker Claude Elwood Shannon definierte 1949 zwei wichtige Anforderungen, die Blockchiffren zum Erzielen einer starken Verschlüsselung erfüllen müssen, s. [27]:

- *Konfusion*: Die Operationen der Blockchiffre müssen die Beziehung zwischen Klartext, Schlüssel und Geheimtext verbergen, etwa mithilfe von Substitution. Der Geheimtext sollte keine (statistischen) Eigenschaften besitzen, die Rückschlüsse auf Schlüssel oder Klartext ermöglichen.
- *Diffusion*: Jedes Bit des Klartextes und jedes Bit des Schlüssels sollte möglichst viele Bits des Geheimtextes beeinflussen, etwa mithilfe von Permutation der einzelnen Bits in einem Block.

Zwei Designprinzipien, Permutation und Substitution, dienen dabei der Erhöhung der Konfusion bzw. Diffusion:

- Permutation: Ein Element des Klartextes wird innerhalb des Blocks umgestellt. In mehreren Runden wird dadurch die Anordnung bzw. Reihenfolge der Elemente verändert.
- Substitution: Ein Element des Klartextes wird anhand einer bestimmten Vorschrift ersetzt.

Moderne Blockchiffren wie AES (Advanced Encryption Standard) permutieren und substituieren den zu verschlüsselnden Block mehrmals hintereinander in mehreren Runden, um ein möglichst hohes Maß an Konfusion und Diffusion zu erreichen.

Blockchiffren können in verschiedenen Betriebsmodi genutzt werden. Der intuitivste Betriebsmodus ist die voneinander unabhängige, blockweise Verschlüsselung des Klartextes. Der beliebig lange Klartext wird dabei in mehrere Blöcke aufgeteilt und der letzte Block wird ggf. aufgefüllt (Padding). In diesem, sog. ECB-Modus (Electronic Codebook

Mode) wird jeder Block mit dem gleichen Schlüssel verschlüsselt. Identische Klartextblöcke werden zu identischen Geheimtextblöcke verschlüsselt. Sich wiederholende Muster im Klartext sind im ECB-Modus in den Geheimtextblöcken wiederzufinden, was Rückschlüsse auf den Klartext zulässt und ggf. eine Kryptoanalyse begünstigt. Darüber hinaus kann ein Angreifer einzelne Blöcke gezielt miteinander vertauschen, entfernen oder eigene Blöcke einschleusen. Indem sie dafür sorgen, dass die jeweiligen Geheimtextblöcke nicht nur vom Klartext und vom Schlüssel abhängen, sondern von einer weiteren Information, schaffen mehrere verschiedene Betriebsmodi Abhilfe gegen die oben genannten Schwächen. Im Counter Mode (CTR) dient ein einfacher, monotoner Zähler als zusätzlicher Parameter und im Cipher Block Chaining Mode (CBC) wird der vorherige Geheimtextblock gemeinsam mit dem Klartextblock verschlüsselt und für den allerersten Block wird ein Initialisierungsvektor verwendet. Durch diese Verkettung entsteht eine Abhängigkeit der einzelnen Blöcke voneinander. Wiederholungen und Muster im Klartext werden somit verschleiert, da identische Klartextblöcke mit unterschiedlichen Parametern verschlüsselt werden.

Der Advanced Encryption Standard (AES) ist der bedeutendste Vertreter symmetrischer Blockchiffren. AES verarbeitet jeden Block in einem Permutations- und Substitutionsnetzwerk. Dabei werden in Abhängigkeit der Block- und Schlüssellänge zwischen 10 und 14 Runden durchlaufen. Jede Runde besteht aus vier Transformationsschritten.

- *Substitution:* Jedes Byte des Blocks wird anhand einer sog. Substitutionsbox (S-Box) monoalphabetisch verschlüsselt. Dies dient der Vermischung der Bits innerhalb jedes Bytes.
- *ShiftRow* und *MixColumn:* Diese beiden Operationen dienen der Permutation, d. h. die einzelnen Bytes werden innerhalb des Blocks in jeder Runde neu angeordnet, um die Diffusion zu erhöhen.
- *KeyAddition:* In diesem Schritt wird der Block mit einer bitweisen XOR-Operation mit dem Schlüssel, bzw. dem davon abgeleiteten Rundenschlüssel verknüpft.

Seit seiner Einführung im Jahre 2000 wurden mehrere Schwächen des AES-Algorithmus veröffentlicht. Der bislang erfolgversprechendste Angriff, der sog. Biclique-Angriff, wurde 2011 veröffentlicht und reduziert den Berechnungsaufwand für eine vollständige Schlüsselsuche von 128 Bit auf 126,1 Bit bzw. von 256 Bit auf 254,4 Bit [4]. Die resultierende, effektive Schlüssellänge ist dennoch weiterhin ausreichend hoch, sodass weder Biclique-Angriffe noch andere Angriffe eine praktische Bedrohung für die Sicherheit des AES-Algorithmus darstellen. AES gilt stand heute als sicherer Algorithmus.

1.1.3.3.2 Hashfunktionen

Hashfunktionen bilden Nachrichten beliebiger Länge auf Hashwerte kurzer, fester Länge ab. Hashfunktionen arbeiten ohne Schlüssel, weshalb die Hashwerte, die häufig

auch als Fingerabdrücke bezeichnet werden, ausschließlich von der jeweiligen Nachricht, abhängen. Für ihre Anwendung in der Kryptographie müssen Hashfunktionen die folgenden Anforderungen erfüllen:

- Hashfunktionen sollten Einweg-Funktionen sein. D. h. die Hashfunktion sollte effizient aus einer Nachricht den zugehörenden Hashwert berechnen können, zu einem gegebenen Hashwert sollte es allerdings praktisch nicht möglich sein, eine zugehörende Nachricht zu finden. Im Englischen wird diese Eigenschaft *Pre-Image-Resistance* bezeichnet.
- Aufgrund der unterschiedlichen Größen von Urbildbereich und Bildbereich sind Kollisionen möglich, d. h. es existieren Paare unterschiedlicher Nachrichten, für die die Hashfunktion die gleichen Hashwerte berechnet. Die sog. *schwache Kollisionsresistenz* (engl. second pre-image resistance) fordert von kryptographischen Hashfunktionen, dass es praktisch nicht möglich sein darf, für eine gegebene Nachricht eine zweite, sich unterscheidende Nachricht zu finden, die denselben Hashwert liefert wie die erste Nachricht. Diese Eigenschaft spielt insbesondere für die Sicherheit digitaler Signaturen eine bedeutende Rolle, weil mithilfe eines sog. Second-Preimage-Angriffs für eine gegebene, signierte Nachricht eine zweite Nachricht gefunden werden könnte, die denselben Hashwert und damit dieselbe Signatur besitzt.
- Die *starke Kollisionsresistenz* (engl. collision resistance) fordert von kryptographischen Hashfunktionen, dass es praktisch nicht möglich sein darf, zwei beliebige Nachrichten mit demselben Hashwert zu finden. Ein Beispiel eines sog. Kollisionsangriffs wurde 2007 erfolgreich von [30] am inzwischen als unsicher eingestuften MD5-Hashalgorithmus demonstriert.

Als kryptographische Primitive kommen Hashfunktionen in verschiedenen kryptographischen Anwendungen zum Tragen. Sie sind beispielsweise ein fester Bestandteil der meisten Signaturverfahren, weil in der Regel nicht die Nachrichten selbst, sondern deren Hashwerte signiert werden. Ein weiteres, wichtiges Anwendungsgebiet sind Integritätsprüfungen. So kann etwa die Datenintegrität von Nachrichten oder Applikationsparameter der ECU-Software anhand der zugehörigen Hashwerte überprüft werden.

Zu den bekanntesten und am weitesten verbreiteten Hashfunktionen zählen der Message-Digest-Algorithmus und der Secure Hash Algorithm (SHA). Der MD5-Algorithmus wird seit 2005 aufgrund der Existenz eines effizienten Angriffs als unsicher eingestuft. Die Secure Hash Algorithmen (SHA) wurden von NIST spezifiziert. Für SHA-1 sind seit 2017 Schwachstellen bekannt, weshalb auch er seither nicht mehr eingesetzt werden sollte. Seine Nachfolger SHA-2 und SHA-3 gelten als sicher.

1.1.3.3.3 asymmetrische Kryptosysteme

Wie oben beschrieben, kommen beim asymmetrischen Kryptosystem (auch Public-Key-Kryptosystem) zwei Schlüssel zum Einsatz. Jeder Teilnehmer besitzt ein Schlüsselpaar,

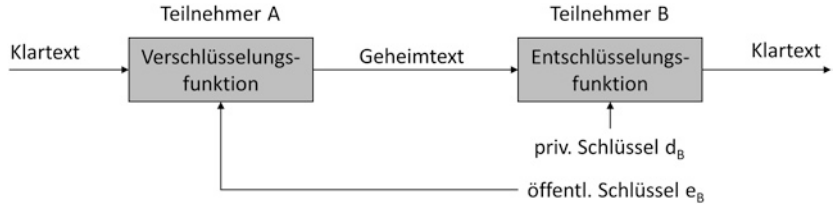

Abb. 1.4 Asymmetrisches Kryptosystem

bestehend aus einem öffentlichen Schlüssel e und einem privaten Schlüssel d. Wenn Teilnehmer A eine verschlüsselte Nachricht an Teilnehmer B senden möchte, muss der Klartext der Nachricht mit dem öffentlichen Schlüssel von B (e_B) verschlüsselt werden, s. Abb. 1.4. Zuvor muss Teilnehmer B seinen öffentlichen Schlüssel an Alice übermitteln. Dieser Vorgang kann, im Gegensatz zu symmetrischen Verfahren, über öffentliche, unverschlüsselte Kanäle erfolgen. Die verschlüsselte Nachricht kann nur der Besitzer des privaten Schlüssels d_B wieder entschlüsseln, also Teilnehmer B.

Eine wichtige Eigenschaft von Public-Key-Kryptosystemen ist, dass der private Schlüssel d zum öffentlichen Schlüssel e passt, sodass er die Verschlüsselung vollständig und korrekt umkehren kann. Andererseits darf es nicht möglich sein, aus dem öffentlichen Schlüssel Rückschlüsse auf den geheimen, privaten Schlüssel ziehen zu können.

Im Gegensatz zu symmetrischen Schlüsseln, die bestenfalls aus echten Zufallszahlen erzeugt werden, sind die Anforderungen an die Generierung asymmetrischer Schlüssel wesentlich komplexer.

Beim bekanntesten Vertreter asymmetrischer Kryptosysteme, dem RSA-Verfahren (benannt nach ihren Erfindern Ron Rivest, Adi Shamir und Leonard Adleman), sind große Primzahlen die Grundlage für die Schlüsselerzeugung. Die Sicherheit des Verfahrens basiert auf dem Faktorisierungsproblem, s. oben, sodass es praktisch unmöglich ist, die öffentlichen Schlüsselparameter zu faktorisieren, um damit wiederum den geheimen Schlüssel zu berechnen.

Die kürzeste Schlüssellänge für das RSA-Verfahren, die heute noch als sicher gilt (bis ca. 2023 [5]), beträgt 2000 Bit – eine Dezimalzahl mit rund 600 Dezimalstellen. Für die praktische Anwendung hat dies zu Folge, dass hier im Gegensatz zu symmetrischen Verfahren relativ große Bereiche für die Speicherung und Verarbeitung der RSA-Schlüssel vorgesehen werden müssen.

Zur Verbesserung der Effizienz werden für die öffentlichen Schlüssel häufig kleinere Werte gewählt – im Rahmen der zulässigen Anforderungen von RSA – sodass, der Speicherbedarf und demzufolge auch der Rechenaufwand für die kryptographischen Operationen mit dem öffentlichen Schlüssel schneller ausgeführt werden können. Für Anwendungen auf eingebetteten Systemen ist diese ressourcenschonende Möglichkeit die übliche Praxis. Als Alternative zum RSA-Kryptosystem benötigen *Elliptische-Kurven-Kryptosysteme* (engl. Elliptic Curve Cryptosystem, ECC) bei vergleichbarem

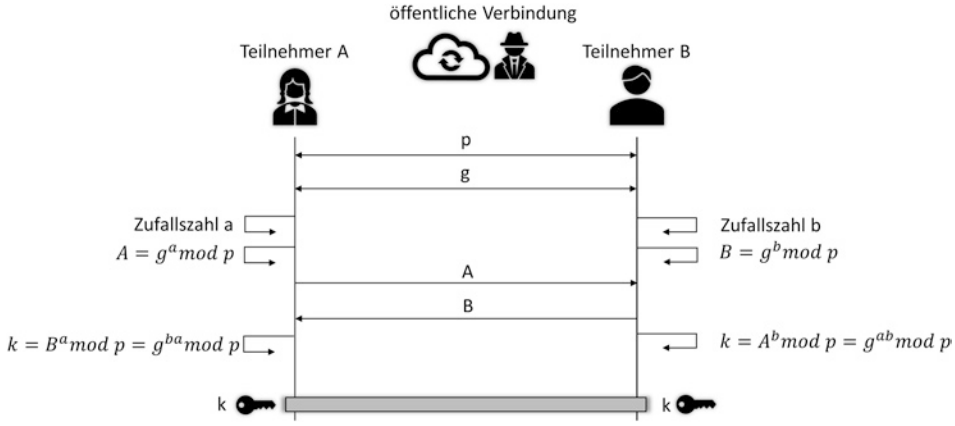

Abb. 1.5 Schlüsselaustauschprotokoll

Sicherheitsniveau wesentlich kürzere Schlüssel als RSA- oder DLP-Verfahren, s. unten unter *Empfohlene Schlüssellängen.* ECC gewinnt deshalb insbesondere für die ressourcenbeschränkten, eingebettete Systeme an Bedeutung.

Für eine ausführliche Erklärung der zugrunde liegenden Mathematik des RSA-Verfahrens, des DLP-Verfahrens und des Elliptischen-Kurven-Kryptosystems wird auf [23] verwiesen.

1.1.3.3.4 Schlüsselaustauschprotokolle

Symmetrische Kryptosysteme setzen voraus, dass alle Teilnehmer über einen gemeinsamen Schlüssel verfügen. Dieser geheime Schlüssel muss zuvor entweder über einen sicheren Kanal oder bei einem persönlichen Treffen ausgetauscht werden. In der Praxis ist diese Art des Schlüsselaustauschs häufig nicht machbar, da sich die Kommunikationsteilnehmer entweder nicht kennen oder über keinen sicheren Kanal verfügen (Internet). Wie können aber Teilnehmer über einen öffentlichen, unsicheren Kanal einen gemeinsamen Schlüssel austauschen? Das *Diffie-Hellman-Protokoll,* benannt nach Whitfield Diffie und Martin Hellman, beschreibt eine Möglichkeit zur Schlüsselvereinbarung und löst damit dieses Schlüsseltauschproblem, s. [6]. Das Ziel des DH-Protokolls ist es, einen gemeinsamen Schlüssel für die sichere Kommunikation zu vereinbaren – als Sitzungsschlüssel und zur Absicherung der Authentizität und Vertraulichkeit während der Kommunikationssitzung.

Die zugrunde liegende Mathematik beruht auf Galois-Körpern und bestimmten Eigenschaften des diskreten Logarithmus. Zum einen ist die diskrete Exponentialfunktion in Galoiskörpern eine Einwegfunktion, d. h. für deren Umkehrung, den diskreten Logarithmus, ist zurzeit kein effizientes Verfahren bekannt. Zum anderen ist die Exponentiation kommutativ, d. h. die Exponenten können miteinander vertauscht werden.

In Abb. 1.5 einigen sich Teilnehmer A und B mithilfe des DH-Protokolls über eine unsichere Verbindung auf ein gemeinsames Geheimnis k: Zunächst einigen sich

Teilnehmer A und B auf eine Primzahl p, typischerweise mit einer Länge eines aktuellen RSA-Schlüssels (z.Z. mindestens 2000 Bit) sowie eine ganze Zahl g[3]. Danach berechnet Teilnehmer A eine geheime Zufallszahl a und berechnet $A = g^a \bmod p$. Teilnehmer B berechnet ebenfalls eine geheime Zufallszahl b und berechnet $B = g^b \bmod p$. Teilnehmer A und B schicken sich gegenseitig ihre berechneten Werte A und B zu. Beide Teilnehmer können daraus den gemeinsamen, geheimen Schlüssel k berechnen: $k = B^a \bmod p = g^{ba} \bmod p = A^b \bmod p = g^{ab} \bmod p$. Ein Angreifer, der die Kommunikation zwischen Teilnehmer A und B mithören kann, kann aus den öffentlich ausgetauschten Parametern p, g, A und B das gemeinsame Geheimnis k nicht berechnen, denn hierzu müsste der diskrete Logarithmus von A oder B berechnet werden. Zur weiterführenden Lektüre der mathematischen Grundlage und Beweisführung wird auf [23] verwiesen.

Eine wichtige Voraussetzung für die Sicherheit des DH-Protokolls ist die Authentisierung der ausgetauschten Botschaften. Falls die Botschaften während der Schlüsselvereinbarung nicht authentisiert werden kann ein Angreifer einen MITM-Angriff durchführen.

Im Automotive-Bereich kommt das DH-Protokoll eher selten zum Einsatz, da die Kommunikationspartner wie etwa fahrzeuginterne Komponenten oder Backend-Server normalerweise im Voraus feststehen. Schlüsselzertifikate können demnach bereits im Vorfeld erstellt und ausgetauscht werden, sodass eine spontane Schlüsselvereinbarung mit unbekannten Teilnehmern in der Regel nicht erforderlich ist. Dennoch existieren für einige Sonderfälle Anwendungsszenarien der Schlüsselvereinbarung im Fahrzeug. Beispiel: Für die Schlüsselvereinbarung zwischen mehreren unbekannten ECUs, etwa zum Einlernen des Immobilisers bei der Fahrzeuginbetriebnahme oder für die Vereinbarung der SecOC-Schlüssel (initial bei der Fahrzeuginbetriebnahme oder nach einem Austausch einer Komponente), s. [16] und Abschn. 5.3.4.

1.1.3.3.5 Signaturverfahren

Wie kann der Empfänger einer Nachricht sicher sein, dass die Nachricht von einem bekannten Absender stammt (Authentizität) und unterwegs nicht verändert wurde (Integrität)?

Signaturen sind kryptographisch erzeugte Informationen, die Nachrichten (oder allgemein: digitale Daten) hinsichtlich dieser beiden Schutzziele absichern sollen. Für bestimmte Anwendungsbereiche wird zusätzlich die Zurechenbarkeit bzw. die Nichtabstreitbarkeit, s. Abschn. 1.1.2, als abzusichernden Schutzziel berücksichtigt.

Signaturverfahren können auf die gleiche Weise in symmetrische und asymmetrische Verfahren unterteilt werden wie Verschlüsselungsverfahren. Digitale, elektronische Signaturen werden mittels asymmetrischer Verfahren umgesetzt und symmetrische Verfahren dienen der Erstellung und Prüfung sog. Message Authentication Codes (MAC).

[3] An die Zahl g werden bestimmte Anforderungen gestellt, s. math. Grundlagen.

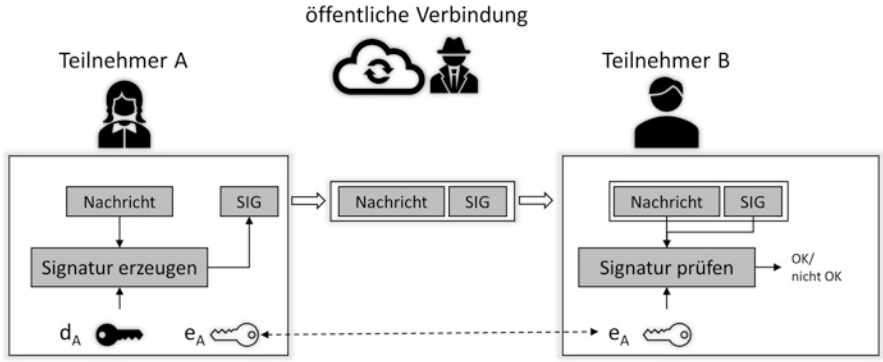

Abb. 1.6 Digitale Signatur

1.1.3.3.5.1 Digitale Signatur

Digitale Signaturen werden mithilfe asymmetrischer Verfahren wie RSA, ElGamal oder ECC erstellt. In Abb. 1.6 wird ihre grundsätzliche Funktionsweise dargestellt. Teilnehmer A möchte über eine öffentliche, unsichere Verbindung eine Nachricht an Teilnehmer B schicken. Teilnehmer B möchte prüfen, ob die empfangene Nachricht tatsächlich von Teilnehmer A stammt und unterwegs nicht verändert wurde. Voraussetzung für das folgende Protokoll ist die Verfügbarkeit eines Algorithmus zur Berechnung einer Signatur auf der Senderseite und eines Algorithmus zur Prüfung der Signatur auf der Empfängerseite. Außerdem muss Teilnehmer A über ein asymmetrisches Schlüsselpaar (e_A, d_A) verfügen und den öffentlichen Teil e_A an den Empfänger übermitteln.

Teilnehmer A erzeugt mithilfe des privaten Schlüssels d_A die Signatur SIG der Nachricht und hängt diese an die zu sendende Nachricht an. Teilnehmer B prüft mithilfe des öffentlichen Schlüssels e_A, ob die Signatur der empfangenden Nachricht gültig ist oder nicht. Bei einer gültigen Signatur sind Authentizität und Integrität der Nachricht sichergestellt. Bei einer ungültigen Signatur wurden Nachricht oder Signatur unterwegs (absichtlich oder zufällig) verändert, sodass die definierten Schutzziele für diese Nachricht verfehlt wurden.

Im Gegensatz zum Message Authentication Code ermöglicht eine digitale Signatur das Schutzziel Zurechenbarkeit, d. h. jeder Empfänger einer Nachricht mit gültiger Signatur kann die Urheberschaft dieser Nachricht eindeutig und nicht abstreitbar dem Besitzer des zugehörigen privaten Schlüssels zuordnen.

Anders als bei der Verschlüsselung wird für das Signaturverfahren das Schlüsselpaar des Senders verwendet, weil die kritische Aktion, die Signaturerzeugung, mit dem privaten Schlüssel durchgeführt werden muss. Die Signaturprüfung ist ein unkritischer Vorgang und kann von jedem Teilnehmer, der den öffentlichen Schlüssel des Senders kennt, durchgeführt werden.

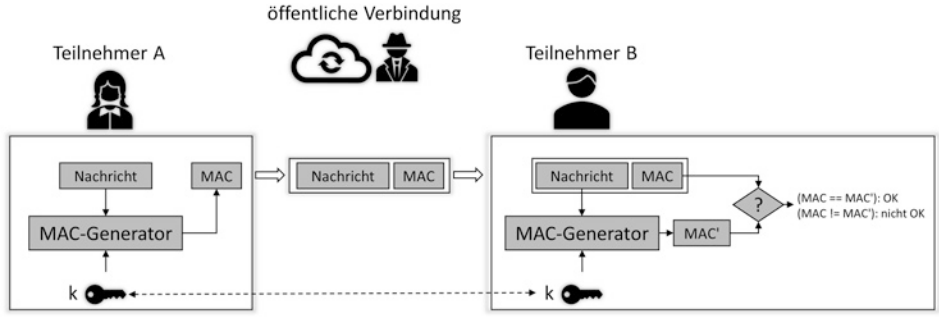

Abb. 1.7 Message Authentication Code

Abweichend von dem hier dargestellten prinzipiellen Ablauf werden üblicherweise nicht die Nachrichten selbst signiert, sondern deren Hashwerte. Somit können die Nachrichten eine beliebige Länge besitzen und sind nicht an Beschränkungen des Signaturverfahrens gebunden. Darüber hinaus sind in der Praxis zum Schutz gegen verschiedenartige Angriffe mehrere Maßnahmen erforderlich, die in modernen, standardisierten Signaturverfahren wie *RSA SSA PSS* in PKCS#1 bereits umgesetzt sind, s. [19].

1.1.3.3.5.2 Message Authentication Code

Message Authentication Codes werden mithilfe symmetrischer Verfahren wie Blockchiffren oder schlüsselbasierten Hashfunktionen erstellt. In Abb. 1.7 wird ihre grundsätzliche Funktionsweise dargestellt.

Teilnehmer A möchte über eine öffentliche, unsichere Verbindung eine Nachricht an Teilnehmer B schicken. Teilnehmer B möchte prüfen, ob die empfangene Nachricht tatsächlich von Teilnehmer A stammt und nicht verändert wurde. Voraussetzung für das folgende Protokoll ist die Verfügbarkeit eines MAC-Generators und eines geheimen Schlüssels auf beiden Seiten. Letzterer muss ggf. vorab über einen sicheren Kanal oder per Schlüsselvereinbarungsverfahren ausgetauscht werden.

Teilnehmer A (Sender) berechnet mithilfe des MAC-Generators aus der Nachricht und dem geheimen Schlüssel k den MAC und hängt diesen an die zu sendende Nachricht an. Teilnehmer B (Empfänger) berechnet MAC' aus der empfangenen Nachricht mithilfe des MAC-Generators und des Schlüssels k und vergleicht den empfangenen MAC mit dem berechneten MAC'. Bei Übereinstimmung sind Authentizität und Integrität der Nachricht sichergestellt. Bei Ungleichheit wurden Nachricht oder MAC unterwegs (absichtlich oder zufällig) verändert, sodass die definierten Schutzziele für diese Nachricht verfehlt wurden.

Im Gegensatz zur digitalen Signatur ermöglicht ein Message Authentication Code keine Zurechenbarkeit. Jeder Teilnehmer, der in Besitz des Schlüssels k ist, kann für eine

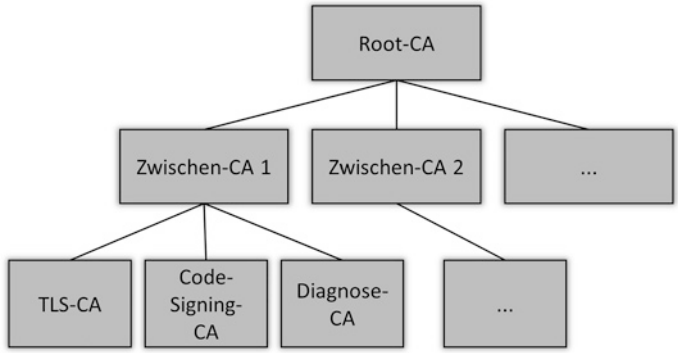

Abb. 1.8 Public-Key-Infrastruktur

Nachricht den zugehörige MAC-Wert berechnen. Trotz dieses Nachteils werden Message Authentication Codes wegen der viel besseren Performance häufig angewandt.

1.1.3.3.6 Zertifikate und PKI

Was sind Zertifikate und wozu dienen sie? Ein Zertifikat ist, vereinfacht ausgedrückt, eine Datenstruktur, die von einer vertrauenswürdigen Instanz signiert wurde und dessen Echtheit anhand dessen öffentlichen Schlüssels überprüft werden kann.

Die Datenstruktur der Zertifikate kann entweder proprietär spezifiziert werden oder es wird ein Standard-Format wie etwa X.509 oder CVC gewählt. Der X.509-Standard ist insbesondere für Anwendungen und Protokolle im Internet weit verbreitet. Ein X.509-Zertifikat macht unter anderem Angaben zu dessen Inhaber und enthält dessen öffentlichen Schlüssel. Für die Anwendung des Zertifikats sind Informationen über dessen Gültigkeitszeitraum sowie über den Aussteller und den verwendeten Algorithmus zum Signieren des Zertifikats enthalten. Der Aussteller eines Zertifikats, die sog. Certification Authority (CA), ist in der Regel Teil einer hierarchischen Vertrauensarchitektur, der sog. Public-Key-Infrastruktur (PKI).

In Abb. 1.8 wird die PKI idealisiert mit einer Baumstruktur dargestellt. Die Wurzelinstanz bildet die Root-CA. Keine Autorität steht über ihr, d. h. ihr muss vertraut werden. Deshalb ist auch die Einbindung des Root-Zertifikats mit dem öffentlichen Schlüssel der Root-CA in eingebettete Systeme zwingend abzusichern.

Unterhalb der Root-CA befinden sich eine oder mehrere Zwischenebenen. Die Zwischen-CAs (engl. Intermediate CA) dienen vor allem der Risikominimierung, etwa um die Root-CA nicht unnötig zu exponieren (die Root-CA signiert nur die wenigen Zwischen-CAs) und um ggf. verschiedene Geschäftsbereiche, Fahrzeug-Plattformen, Backend-Server oder Wirtschaftsräume voneinander zu trennen. Eventuell auftretende Security-Vorfälle beschränken sich in ihren Auswirkungen stets nur auf den jeweiligen Teilbereich.

Unter den Zwischen-CAs befinden sich CAs zur Ausstellung der Zertifikate für verschiedene Anwendungen, etwa für die TLS-Kommunikation, für den authentifizierten Diagnosezugang oder für die Erstellung von Codesignaturen.

PKIs schaffen außerdem die Möglichkeit, Zertifikate aus bestimmten Gründen wieder zurückziehen, d. h. als ungültig zu kennzeichnen. Diese sog. Revokation ist geboten, falls etwa der jeweilige Zertifikatinhaber aus dem Projekt oder dem Unternehmen ausscheidet, falls ein Gerät oder Fahrzeug außer Betrieb genommen wird (Decommission, s. Kap. 4) oder falls private Schlüssel kompromittiert wurden und in Folge dessen erneuert werden müssen. Certificate Revocation Lists (CRL) dienen zur Mitteilung aller Instanzen einer PKI über die zurückgezogenen Zertifikate.

Für welche Anwendungsfälle können digitale Zertifikate genutzt werden?
Zertifikate liefern eine Lösung für ein fundamentales Problem der asymmetrischen Verschlüsselungs- und Signaturverfahren: Wie kann sich ein Teilnehmer sicher sein, dass der öffentliche Schlüssel dem richtigen Teilnehmer gehört und nicht manipuliert wurde?

Ein Zertifikat ermöglicht eine beglaubigte Zuordnung eines öffentlichen Schlüssels zu einer bestimmten Identität. Ein Public-Key-Zertifikat kann hinsichtlich seiner Authentizität überprüft werden, indem die Signatur des Zertifikats mit dem Schlüssel der CA verifiziert wird.[4] Ohne diese Schutzmaßnahme könnte ein Angreifer einen *Man-In-The-Middle-Angriff* (MITM) auf den Schlüsselaustausch durchführen und bei Erfolg ohne Kenntnis der Teilnehmer – deren Kommunikation lesen und manipulieren.

Ein weiterer Anwendungsfall für digitale Zertifikate ist das sog. *Attributzertifikat* (engl. Authorisation certificate), das ebenfalls im X.509-Standard spezifiziert ist. Im Unterschied zum Public-Key-Zertifikat enthält ein Attributzertifikat selbst keinen Schlüssel. Vielmehr verweist es auf ein Public-Key-Zertifikat und ergänzt dessen Inhaber um zusätzliche Attribute wie etwa Berechtigungen oder Rollenzuordnungen. Auch das Attributzertifikat muss über eine von einer vertrauenswürdigen Instanz (PKI) ausgestellten Signatur verfügen. Die Trennung zwischen Identität/Schlüssel und zusätzlichen Attributen ermöglicht beliebige Erweiterungen der Attribute unter Beibehalten des Schlüsselzertifikats.

1.1.3.3.7 Zufallszahlen

Welche Rolle spielen Zufallszahlen für die Cybersecurity?
Die Grundlage vieler kryptographischer Verfahren und Protokolle sind gute Zufallszahlen. Zufallszahlen werden zum Erzeugen von Schlüsseln, für Challenge-Response-Verfahren oder als Initialisierungsvektoren für kryptographische Algorithmen benötigt.

[4] Um das höchste Maß an Sicherheit zu erreichen sollte die Zertifikatskette bis zur Root-CA der PKI überprüft werden.

Bei Cyberangriffen werden in den meisten Fällen nicht die kryptographischen Primitive oder Verfahren selbst angegriffen, denn die zugrunde liegende Mathematik ist bei modernen Verfahren praktisch nicht zu brechen – zumindest solange noch keine Quantencomputer verfügbar sind. Viel erfolgsversprechender sind hingegen Angriffe, wenn Implementierungsfehler vorhanden sind, insbesondere Fehler bei der Erzeugung von Zufallszahlen.

Ein prominentes Beispiel sind die digitalen Bürgerzertifikate in Taiwan (Citizen Digital Certificate). Sie wurden 2013 erfolgreich angegriffen, s. [3]. Das Problem bzw. die Schwachstelle lag dabei nicht im implementierten RSA-Algorithmus oder im (hinreichend guten) Zufallszahlengenerator der SmartCards, sondern in dessen fehlerhaften Implementierung.

Wie zufällig müssen gute Zufallszahlen sein?
Idealer Zufall setzt ein nicht-vorhersagbares und nicht beeinflussbares, unabhängiges Ergebnis voraus. Mit verschiedenen statistischen Tests, wie etwa dem *Autokorrelations-Test* oder dem *Chi-Quadrat-Test,* können die Eigenschaften von Zufallsfolgen, genauer gesagt deren Güte, überprüft werden. Korrelationen, Abhängigkeiten und systematische Fehler können damit gefunden und nicht-zufällige Zahlenfolgen können erkannt werden, s. [15].

Anforderungen an kryptographisch sichere Zufallszahlen
Die oben genannte Anforderung der Nicht-Vorhersagbarkeit lässt sich unterteilen in die sog. *Vorwärts-Unvorhersagbarkeit* (engl. forward unpredictability oder forward secrecy) und die *Rückwärts-Unberechenbarkeit* (engl. backward unpredictability oder backward secrecy). Beide Eigenschaften fordern, dass aus der aktuellen Ausgabe eines Zufallszahlengenerator, d. h. aus der aktuellen Zufallszahl, keinerlei Rückschlüsse auf die vorhergehenden bzw. nachfolgenden/zukünftigen Zufallszahlen gezogen werden können.

Erzeugung von Zufallszahlen
Problem: Rechnersysteme arbeiten deterministisch, d. h. bei gleichen Eingaben liefern sie die gleichen Ergebnisse. Ihre Funktionen sind vorprogrammiert, was hier wörtlich zu verstehen ist. Zum Erzeugen zufälliger Ergebnisse wurden Rechnersysteme nicht spezifiziert.

Pseudo-Zufallszahlengenerator Deterministische Zufallszahlengeneratoren erzeugen Zufallsfolgen, die nachvollziehbar und berechenbar sind. Die Ergebnisse sind keine echten Zufallszahlen, daher der Präfix *Pseudo.* PRNGs berechnen anhand eines vorgegebenen Algorithmus eine beliebig lange, aber endliche und sich wiederholende Zahlenfolge. Anhand des Seeds, dem initialen Entropiewert, wird der Einstiegspunkt in diese Zahlenfolge bestimmt. Die Schwächen von PRNGs liegen zum einen in ihrer deterministischen Eigenschaft, denn mit ausreichender Kenntnis des aktuellen Zustands können sowohl die nachfolgenden also auch die vorhergehenden Ausgaben vorhergesagt,

bzw. berechnet werden, s. [14]. Zum anderen verwenden PRNGs oftmals schwache Seeds, d. h. anstatt einer guten Entropiequelle werden Seeds häufig anhand einer deterministischen und deshalb vorhersagbaren Funktion berechnet.

Die oben genannten Anforderungen bzgl. kryptographischer Sicherheit werden von einem PRNG nicht erfüllt. PRNGs sind kryptographisch unsicher und mögen für bestimmte Anwendungsgebiete ausreichend sein – für eine Verwendung in der Kryptographie sind sie jedoch untauglich und sogar gefährlich.

TRNG Da deterministische Prozesse niemals alleinige Quelle echter Zufallszahlen sein können, erzeugen TRNGs ihre Zufallszahlen stattdessen anhand nicht vorhersagbarer Effekte bzw. Entropiequellen. Häufig werden mehrere Entropiequellen herangezogen und mittels einer kryptographischen Hashfunktion auf einen kürzeren Wert abgebildet. Einem Angreifer werden dadurch Rückschlüsse auf die Eingangswerte zusätzlich erschwert.

Echte Zufallszahlengeneratoren (engl. True Random Number Generator, TRNG) werden anhand ihrer Entropiequelle unterteilt in physikalische TRNG (PTRNG) und nicht-physikalische TRNG (NPTRNG), vgl. [14].

Zu physikalischen Entropiequellen zählen:

- Zeitintervalle zwischen zwei Ereignissen beim Zerfall radioaktiver Elemente
- Bauteiltoleranzen in Widerständen, Kondensatoren, Quarzen, etc.
- Halbleiterrauschen

Zu nicht physikalischen Entropiequellen zählen:

- Benutzereingaben, etwa über Maus, Tastatur, Mikrofon und Kamera
- Zeitdifferenzen zwischen eintreffenden Netzwerkpaketen
- Systemdaten wie verschiedene Zähler und die (lokale) Uhrzeit

Zukünftig könnte die Klasse der TRNGs um sog. Quanten-Zufallszahlengeneratoren (QRNG) bereichert werden. Diese nutzen quantenmechanische Effekte, die nach der Theorie (s. auch *Kopenhagener Deutung* und *Heisenbergsche Unschärferelation*) unbestimmbar und nicht vorhersagbar sind. Nguyen et al. [21] untersuchten bereits einen Prototypen eines QRNGs für eine Automotive ECU.

CSPRNG

Kryptographisch sichere PRNGs (CSPRNG:) sind eine spezielle Ausprägung von PRNGs. Ihre Ausgabe darf sich hinsichtlich ihrer statistischen Eigenschaften nicht von der Zahlenfolge eines TRNGs unterscheiden. Die Anforderungen hinsichtlich Forward- und Backward Secrecy, s. oben, müssen von CSPRNGs erfüllt werden.

Ihre Sicherheit basiert auf kryptographischen Primitiven und Einwegfunktionen, die einen Angriff auf das Verfahren unpraktisch machen. Der Seed, d. h. der Initialwert des Generators, ist wie beim PRNG der kritische Parameter. Mit Kenntnis des Seeds können auch bei CSPRNGs alle Zufallszahlen vorhergesagt werden. Der Seed muss zum einen eine ausreichend hohe Entropie besitzen und ausreichend lang sein und zum anderen muss er unbedingt geheim gehalten werden.

Praktische Lösung: HRNG
TRNGs besitzen aufgrund ihrer echten Entropiequellen einen entsprechend begrenzten Datendurchsatz, d. h. die Anzahl der Zufallszahlen, die ein TRNG pro Sekunde erzeugen kann, ist begrenzt. Der Datendurchsatz von PRNGs bzw. CSPRNGs hängt dagegen lediglich von der Performanz des jeweiligen Rechnersystems ab.

Hybride Zufallszahlengeneratoren (HRNGs) kombinieren die Vorteile verschiedener Typen von RNGs. Sie bestehen prinzipiell aus einer (guten) Entropiequelle, etwa einem TRNG, und einem deterministischen und deshalb performanteren CSPRNG, s. [15]. HRNGs verwenden für den Initialisierungswert (Seed) und zyklisch für das *Re-Seeding* die Zufallszahlen eines TRNGs und der CSPRNG erzeugt hochperformant die Zufallszahlen.

Standardisierung
Sowohl das *Bundesamt für Sicherheit in der Informationstechnik* (BSI) als auch das *American National Institute of Standards and Technology* (NIST) setzen mit ihren Standards für Zufallszahlengeneratoren und deren Design- und Testkriterien weltweite und branchenübergreifende Maßstäbe. Die BSI-Veröffentlichungen AIS20 und AIS31 spezifizieren Funktionsklassen und Evaluationsmethodologie für PRNGs bzw. TRNGs. [15] ergänzt diese Spezifikation um Vorgaben zur Vorgehensweise bei der Evaluierung und Zertifizierung.

1.1.3.3.8 Empfohlene Schlüssellängen
Welche Schlüssellänge sollte gewählt werden, um mittel- oder langfristig sicher zu sein?

Neben der eigentlichen algorithmischen Sicherheit der kryptographischen Verfahren, wie etwa der Resistenz gegen statistische Angriffe, ist die Schlüssellänge ein entscheidendes Maß für den Aufwand, eine Verschlüsselung zu brechen. Das sog. Sicherheitsniveau gibt den Aufwand eines Angreifers an, um ein Verfahren mit einer bestimmten Schlüssellänge zu brechen. Dies gilt unter der Voraussetzung, dass es keine effizientere Möglichkeit zum Brechen des Verfahrens gibt als das Durchprobieren des gesamten Schlüsselraums.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) gibt eine „jährlich zu überprüfende“ Empfehlung für die Schlüssellängen verschiedener Verfahren heraus, s. [5]. Aktuell wird für symmetrische Blockchiffren eine Schlüssellänge von mindestens 128 Bit empfohlen. Um ein vergleichbares Sicherheitsniveau zu erreichen wird für RSA

eine Schlüssellänge von mindestens 2000 Bit und für ECC eine Schlüssellänge von mindestens 250 Bit empfohlen.

1.2 Cybersecurity im Automobilbereich

1.2.1 Vernetzte und automatisierte Fahrzeuge

1.2.1.1 Strömungen in der Fahrzeugentwicklung

Rückblickend auf die vergangenen 10 bis 15 Jahre erfuhr die Fahrzeugindustrie einen kräftigen Impuls für die Entwicklung vernetzter Fahrzeuge (engl. connected vehicles) und automatisierter Fahrfunktionen (engl. automated driving functions). Diese beiden Strömungen zählen für die Automobilindustrie zu den wichtigsten Megatrends der ersten Hälfte des 21. Jahrhunderts und sind maßgeblich auch für den Fortschritt und Entfaltung des Themenbereichs Cybersecurity verantwortlich.

Ausgehend von zweckorientierten Steuerungssystemen wurden zunächst zur Erhöhung der Fahrsicherheit und des Fahrkomforts Funktionen wie etwa das Antiblockiersystem (ABS), das Elektronische Stabilitätsprogramm (ESP) mit mittlerweile für viele Fahrer unentbehrlichen Zusatzfunktionen wie der Berganfahrhilfe entwickelt. Hinzu kamen die Infotainmentsysteme, deren Funktionsumfang aufgrund der damals geringen Mobilfunkbandbreite schwerpunktmäßig zunächst auf Entertainmentfunktionen wie Radio und CD und auf Navigationssysteme mit der Anzeige der aktuellen Verkehrssituation (Staumeldungen) begrenzt war. Über die Jahre hinweg kamen mehr Fahrerassistenzsysteme (engl. Advanced Driver Assistance Systems, ADAS) hinzu, deren Automatisierungsgrade von *teilautomatisiert* (z. B. Spurhalteunterstützung) bis *autonom* (z. B. automatischer Notbremsassistent) reichen und damit einen fließenden Übergang in das vollautomatisierte, autonome Fahren (engl. autonomous driving, AD) abbilden [26]. Selbstfahrende Fahrzeuge, die keinen menschlichen Eingriff mehr erfordern und alle mögliche Fahrsituationen autonom bewältigen können, sind allerdings noch Gegenstand aktueller Forschung.

1.2.1.2 Vernetzte und automatisierte Fahrzeuge aus der Sicht von …

Fahrer/Mitfahrer Die Anbindung der Fahrzeuge an das Internet und die Vernetzung mit anderen Verkehrsteilnehmern ermöglichen oder verbessern Funktionen wie *Ridehailing* (Bestellung einer Fahrt), *Carsharing* oder der Integration Cloud-basierter Dienstleistungen wie etwa der Einblendung von Zusatzinformationen über den aktuellen Standort und der Umgebung für Werbung, Stadtführungen oder als Navigationshilfe. Die breitbandige Internetanbindung verwandelt Fahrzeuge zusätzlich in fahrende Hotspots, was alle möglichen Streaming- und Infotainmentanwendungen begünstigt.

Die automatisierten Fahrsysteme ermöglichen zudem komfortable Funktionen wie das autonome Ein- und Ausparken inkl. Parkplatzsuche. In Sachen Sicherheit und Wirtschaftlichkeit stellen vernetzte Fahrzeuge Teilnehmer eines sog. *kooperativen und intelligenten Transportsystems* (C-ITS, s. Abschn. 5.4.2) dar, was im Wesentlichen der Verbesserung der Verkehrssicherheit und der Optimierung des Verkehrsflusses und des Energieverbrauchs dient. Sie tauschen dafür über die Fahrzeug-zu-Fahrzeug-Kommunikation u. a. Safety-relevante Botschaften aus. Geisterfahrer, Raser, Schleicher und Drängler könnten dadurch zukünftig der Vergangenheit angehören. Warnungen vor Gefahrensituationen wie Stauenden, Unfällen, Glätte und Baustellen werden automatisiert. Wo Licht ist, ist auch Schatten. So bestehen auch berechtigte Sorgen um die Sicherheit dieser Fahrzeuge. Die Entmündigung des Fahrers bei (teil-)automatisierten Eingriffen ist rechtlich und gesellschaftlich umstritten. Unter welchen Bedingungen würden Sie sich in ein hoch- oder vollautomatisiertes Fahrzeug setzen und die Fahrt genießen können, wenn die Kontrolle über das Fahrzeug nicht bei Ihnen oder den Mitfahrern liegt und ein Eingreifen nicht oder nur noch eingeschränkt möglich ist? Diese Frage richtet sich an das Vertrauen und die Akzeptanz dieser technischen Systeme. Auf der technischen Ebene wird diese Aufgabe durch das Anlegen von entsprechend hohen Maßstäben an die Sicherheit und Zuverlässigkeit der zu entwickelnden Systeme gelöst. Auf der menschlichen bzw. gesellschaftlichen Ebene müssen Informationen bereitgestellt werden und Überzeugungs- und Aufklärungsarbeit geleistet werden.

Hersteller/Flottenbetreiber Pay-as-you-drive bzw. Car-Sharing sowie die Bereitstellung von Zusatzinformationen, s. oben, bedeuten für Flottenbetreiber und Drittanbieter neue Geschäftsmodelle. Zudem ermöglichen vernetzte Fahrzeuge eine erweiterte Flottenüberwachung und ggf. eine Fernsteuerung und Fernkonfiguration der Fahrzeuge. Allerdings entstehen aufgrund der hohen technologischen Komplexität hohe Entwicklungskosten und demzufolge ein erhöhtes Investitionsrisiko. In Zukunft könnte noch ein finanzielles Risiko durch Haftung für Schäden, die durch Fehlentscheidungen der automatisierten Fahrfunktionen entstehen, hinzukommen.

Angreifer/Hacker Aus Angreifersicht werden hier Fahrzeuge gebaut, die über das Internet erreichbar sind und Funktionen zur Überwachung und zur (Fern-)Steuerung besitzen. Sie bestehen aus hochkomplexen, elektronischen Systemen mit zahlreichen, unterschiedlichen Hardware- und Softwarekomponenten und unterschiedlichsten Schnittstellen. So betrachtet ist das eine Spielwiese für Hobby-Hacker und Angreifer, die sich und anderen ihr Können beweisen wollen, und ein attraktives Ziel für sog. Hacktivisten, die der Einführung dieser Technologie entgegenwirken wollen.

1.2.1.3 Wie wirkt sich die Entwicklung vernetzter und automatisierter Fahrzeuge auf die Cybersecurity aus?

Die in Abb. 1.9 dargestellte Funktionskette *Sense – Compute – Act und Connect* ermöglicht automatisiertes bzw. autonomes Fahren und damit den Ersatz des (menschlichen)

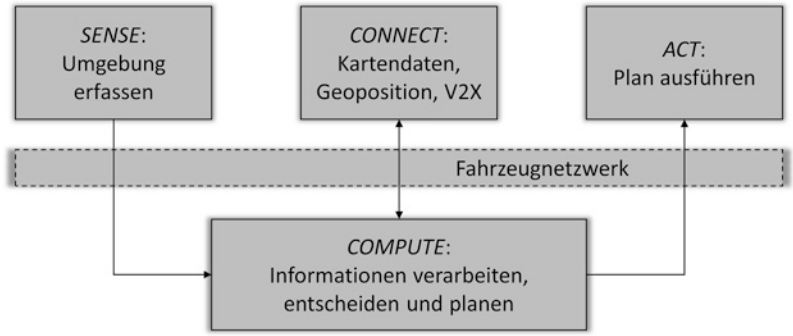

Abb. 1.9 Funktionskette für automatisiertes Fahren

Fahrers. Rechnergesteuerte Fahrfunktionen müssen grundsätzlich dieselben Aufgaben umsetzen, wie ein Mensch: sehen, erkennen, orientieren, planen, entscheiden und handeln. Was ein Mensch anhand seiner Sinne, seines Gehirns und seiner Motorik umsetzt, muss das rechnergesteuerte Fahrzeug mithilfe von Sensoren, Hochleistungsrechnern und Aktuatoren nachbilden. So wird die Umgebung des Fahrzeugs mit unterschiedlichen Sensoren wie RADAR, LIDAR, Kameras, Ultraschallsensoren, Beschleunigungs- und Neigungssensoren erfasst (SENSE), im Fahrzeugnetzwerk ausgetauscht und von Hochleistungsrechnern (engl. high-performance computer, HPC) verarbeitet. Sie haben die Aufgabe, die Umgebung zu erkennen (Perception), den aktuellen Standort zu verfolgen (Lokalisierung) und die optimale Vorgehensweise zu planen, um das gewünschte Fahrtziel zu erreichen. Die dafür zu treffenden Entscheidung, etwa über die Wahl der Verkehrsroute, fällt das System autonom, d. h. ohne menschlichen Einfluss. Schließlich erfolgt die Ansteuerung der Aktuatoren (ACT) wie etwa der Lenkung, der Bremse, des Antriebs und der Signaleinrichtungen (Blinker). Über externe Kommunikationskanäle (Connect) werden zusätzliche Informationen mit anderen Verkehrsteilnehmern und dem Backend-Servern ausgetauscht.

Diese einfach anmutende Funktionskette wird von einem komplexen Verbund aus rund 100 ECUs, die sich über das fahrzeuginterne Kommunikationsnetzwerk austauschen, umgesetzt. Jede Komponente erfüllt dabei spezifische Aufgaben, um in Summe die gewünschte automatisierte Fahrfunktionen zu ermöglichen. Gleichzeitig eröffnet ein hoher Grad der Vernetzung und Komplexität der Hardware-, Software und Funktionsarchitektur diverse Optionen für Cybersecurity-Angriffe. Dabei steigt das Risiko bei einem höheren Automatisierungsgrad des Fahrzeugs, weil einerseits die Kontrollmöglichkeiten des menschlichen Fahrers bzw. Insassen sinken und die des rechnergesteuerten Systems steigen.

Das Risiko für die funktionale Sicherheit lässt sich anhand des folgenden Szenarios erläutern: Ein hoch- oder vollautomatisiertes Fahrzeug fährt mit 130 km/h auf der Autobahn – mit aktiviertem *Highway-Pilot,* d. h. der Fahrer achtet nicht länger auf den

Verkehr sondern liest etwa ein Buch über Automotive Security. Obwohl sich das Fahrzeug schnell einem Stauende nähert, reagiert die automatische Bremsfunktion nicht – vielleicht weil die ACC-Funktion durch einen Hackerangriff übersteuert wird oder weil die Bremssteuersignale bei der Übermittlung an die Bremssteuerung von einem Angreifer verändert wurden. Dieses konstruierte, aber technisch denkbare Szenario könnte fatal enden. Es stellt allerdings klar, dass klassische Schutzmaßnahmen nicht ausreichend sind. Redundante Safety-Pfade sind unzureichend für den Schutz gegen Security-Angriffe. In [20] demonstrierten die Autoren die Wirkungslosigkeit mehrerer Safety-Mechanismen des AUTOSAR-Standards gegen Security-Angriffe.

Die Kritikalität beschränkt sich allerdings nicht auf Safety-Aspekte. Ohne ausreichenden Schutz vor Cybersecurity-Angriffen könnten vernetzte und automatisierte Fahrzeuge leicht von Diebstahl, Sabotage oder Hacktivismus betroffen werden. Letztere Bedrohung ist aufgrund der hohen Aufmerksamkeit in den Medien wahrscheinlich und könnte für das betroffene Unternehmen oder die gesamte Branche zu einem Reputations- und Vertrauensverlust führen.

1.2.2 E/E-Architektur

1.2.2.1 Was ist die E/E-Architektur?

Die Elektrisch/Elektronische Architektur, kurz E/E-Architektur, definiert die elektrischen und elektronischen Komponenten eines Fahrzeugs, sowie deren Energieversorgung und Vernetzung für die Signalverteilung und den Datenaustausch. In Bezug auf Cybersecurity sind vorrangig die jeweiligen Schnittstellen zwischen den Komponenten und das Kommunikationsnetzwerk von Interesse.

1.2.2.2 Evolution der E/E-Architektur

Früher, in den Anfangszeiten der Elektronifizierung, besaßen Fahrzeuge eine verteilte, *dezentrale E/E-Architektur* mit überwiegend voneinander isolierten ECUs, die ihre jeweiligen Funktionen weitestgehend unabhängig voneinander ausführten.

Es folgte der Zusammenschluss mehrerer ECUs desselben Funktionsbereichs zu *Domänen* (Antriebstrang, Komfort, Fahrwerk, Infotainment). Die Domänen bestanden aus eigenständigen Netzwerken mit z. T. unterschiedlichen Bussystemen. Für den Antriebstrang wird heute typischerweise der High-Speed-CAN verwendet, für die Komfortelektronik werden Low-Speed-CAN und LIN eingesetzt, für die echtzeitfähige Kommunikation der Fahrwerkkomponenten steht u. a. Flexray zur Verfügung und der MOST-Bus wurde eigens für die Infotainment-Domäne konzipiert.

Die Domänen-interne Kommunikation legte den Grundstein für das Verschieben bzw. Zusammenfassen von Funktionen, um die Anzahl der ECUs und die Kosten für die Verkabelung zu reduzieren. Der bis dahin noch geringe Domänen-übergreifende Informationsaustausch gewann mit der Einführung *zentraler Gateways* eine zunehmende Bedeutung. Das Gateway ist mit allen Subnetzen verbunden und

ermöglicht die Kommunikation zwischen ECUs verschiedener Domänen und damit komplexe Funktionalitäten wie etwa der Auffahrwarnung mit Notbremsfunktion. Assistenzsysteme wie Notbremsassistent oder Adaptive Cruise Control tauschen beispielsweise mit LIDAR, RADAR, Kamera, IMU-Sensoren, Drehzahlsensoren, Antriebstrangkomponenten, ABS, ESP und passive Sicherheitssysteme wie Airbag und Gurtstraffer Informationen aus.

Der nächste Schritt in der Entwicklung der E/E-Architektur ist die Einführung leistungsstarker *Domänencontrollern* (DC), die dank ihrer Ressourcen komplexere Funktionen umsetzen können und zudem bestehende Funktionen der jew. Domäne zur Kosten- und Performanceoptimierung zusammenführen. Ein Ethernet-Backbone sorgt für eine echtzeitfähige und breitbandige Kommunikation zwischen den Domänencontrollern.

In weiteren Konsolidierungsschritten könnten zukünftig die Funktionen der Domänencontroller miteinander verschmolzen und daran anknüpfend in einem Fahrzeugrechner zentralisiert werden. Letzterer wird das Domain-Gateway ersetzen bzw. einverleiben. Komplexe Hochleistungsrechner (engl. high-performance computer, HPC) machen den Weg für die genannten Fusionierungs- und Zentralisierungsschritte frei. Einerseits wird somit die Anzahl benötigter ECUs und der zugehörigen Verkabelung reduziert. Andererseits können Software-basierte Fahrzeugfunktionen in zentralisierten Architekturen leichter angepasst und für neue Fahrzeugplattformen übernommen und weiterentwickelt werden.

In einer sog. *zonalen Fahrzeugarchitektur* übernimmt einer leistungsstarker Zentralrechner wesentliche Fahrzeugfunktionen. In den verschiedenen Funktionsbereichen der E/E-Architektur (Zonen) befinden sich fast nur noch Sensoren, Aktuatoren und ggf. sog. Gateway-Zonenrechner, die als Bindeglied von Ethernet zu den klassischen Bussystemen fungieren. Die Fusionierung wesentlicher Fahrzeugfunktionen in einem zentralen Rechner ebnen darüber hinaus den Weg für einen Übergang zu einer *Cloud-basierten Architektur,* in der bestimmte Fahrzeugfunktionen in der Cloud berechnet und koordiniert werden könnten.

1.2.2.3 Welchen Einfluss besitzen heutige und zukünftige E/E-Architekturen auf Cybersecurity?

Die frühen Fahrzeugarchitekturen enthielten nur wenige ECUs, die wiederum nur vereinzelt oder gar nicht miteinander vernetzt waren. Eine (breitbandige) Internetanbindung gehörte nicht zur Serienausstattung. Für einen Angreifer stellten sich damalige Fahrzeuge als geschlossene Systeme mit nur wenigen systemübergreifenden Funktionen dar, was verglichen mit heutigen Fahrzeugen nicht besonders attraktiv erscheint.

Mit der Einführung von Domänencontrollern kam eine kritische Komponente ins Spiel, weil bei einem erfolgreichen Angriff die Kontrolle über die jeweilige Domäne erlangt werden kann. Mit zentralen Gateways verhält es sich noch problematischer, da im Falle einer Kompromittierung der Angreifer den Zugriff auf sämtliche Subnetze und ggf. die Kontrolle kritischer Fahrzeugfunktionen erhält.

Aufgrund der Verlagerung von Funktionen in Zentralrechner und Domainrechner verschwimmt die physische Zuordnung von Software und Daten zu einer physischen Komponente oder sie verschwindet ganz. Gleichzeitig gewinnt die Absicherung der Kommunikation an Bedeutung. Eine logische Trennung der Kommunikationsverbindungen etwa anhand virtueller Kanäle bzw. Tunnel ist erforderlich und mit steigendem Kommunikationsbedarf (Bandbreite) wächst auch die Anforderung an die Security-Performanz. Zudem erfordern neuartige Software- und Hardware-Architekturen der Hochleistungsrechner, wie etwa Multi-Prozessor- und Multi-Core-Architekturen und Virtualisierung, auch entsprechend angepasste Security-Technologien wie beispielsweise eine abgesicherte *Inter-Prozessor-Kommunikation* (IPC) und einen vertrauenswürdigen Hypervisor.

1.2.3 Automotive Security

1.2.3.1 Automotive Security vs. IT-Security

Fahrzeuge früherer Generationen waren eigenständige Systeme, ebenso voneinander isoliert wie deren Komponenten der E/E-Architektur. Security war kein Kriterium und wurde nicht *by-design* in der Entwicklung berücksichtigt. Die Systeme dieser Fahrzeuge wurden allein durch die Kontrolle des physischen Zugriffs vor Unbefugten geschützt.

Inzwischen sind Fahrzeuge ohne eine starke, interne Vernetzung und ohne Außenschnittstellen nicht mehr vorstellbar, s. oben. Mit zunehmender Vernetzung und einer steigenden Zahl von Schnittstellen werden Fahrzeuge aber auch zunehmend exponiert – ihre sog. *Angriffsoberfläche* steigt. Vernetzte und (teil-)automatisierte Fahrzeuge sind hochkomplexe Technologieträger und rücken allein aus diesem Grund zunehmend in den Fokus von Hackern und Angreifern.

Fahrzeuge werden angegriffen – genau wie Computer, Smartphones und Spielzeuge angegriffen werden. Weil ein Ausfall oder eine Fehlfunktion eines Fahrzeugs unmittelbare Auswirkungen auf die Sicherheit von Mensch und Umwelt haben kann, sind Cybersecurity-Angriffe auf Automotive Systeme als kritisch einzustufen.

Automotive Security beschäftigt sich einerseits mit den Risiken durch Cyberangriffe auf den Automotive Kontext, d. h. auf Fahrzeuge und deren Infrastruktur. Andererseits definiert Automotive Security Schutzmaßnahmen, die die Angriffsoberfläche verringert und mögliche Auswirkungen eines Angriffs beschränken.

Automotive Security erfindet dabei das Rad nicht neu. Mit der Übernahme von Technologien und Methoden aus dem „klassischen“ IT-Bereich wurden und werden auch bewährte und etablierte Vorgehensweisen des IT-Security-Bereichs auf den Automotive Bereich übertragen. Im Unterschied zum IT-Bereich gelten für den Automotive-Bereich jedoch andere Maßstäbe, was Echtzeitfähigkeit, Zuverlässigkeit und Safety-Kritikalität betrifft.

Tab. 1.1 Angreifertypen und deren Motivation

Angreifertyp	Motivation
Hobby-Hacker und Skript-Kiddies	Neugier, Spieltrieb, kurzzeitiger Ruhm und Anerkennung
Cyberkriminelle	Profit
Hacktivisten	Verfolgung politischer und ideologischer Ziele
Terroristen	Sabotage, Cyber-Kriegsführung
Staatliche Institutionen	Wirtschaftsspionage, Verfolgung geopolitischer Ziele, Cyber-Kriegsführung
Insider und ehemalige Mitarbeiter	Rache, Unzufriedenheit

1.2.3.2 Bedrohungsmodell

„Wenn du dich und den Feind kennst, brauchst du den Ausgang von hundert Schlachten nicht zu fürchten.“(Sunzi, Die Kunst des Krieges).

Ein Bedrohungsmodell dient im Rahmen einer *TARA* (Threat Assessment and Risk Analysis) zur systematischen Analyse der Angriffs- und Verteidigungsmechanismen des untersuchten Systems. Im Wesentlichen werden dabei die Angreifer selbst, deren Absichten und die Angriffsvektoren des Angriffs betrachtet.

1.2.3.2.1 Angreifer

Angreifer (engl. attacker oder threat actor) sind zwar die Hauptakteure eines Angriffs, denn von ihren aktiven, bewussten Handlungen geht die Gefahr für die Sicherheit eines Systems aus. Doch das wesentliche Problem dabei ist, dass der oder die Angreifer i. d. R. vorab nicht bekannt sind bzw. erst nachdem ein Angriff stattfand und erkannt wurde. In einem Angreifermodell werden potenzielle Angreifer deshalb u. a. anhand der Analyse früherer, bekannter Angriffe auf vergleichbare Systeme beschrieben. Angreifer werden anhand verschiedener Merkmale klassifiziert.

Motivation und Absichten Die Klassifizierung der Angreifer weist eine große Spannweite auf. An einem Ende stehen Hobby-Hacker und sog. *Skript-Kiddies*, die häufig durch Neugier und Experimentierfreude, aber auch durch das Streben nach Anerkennung motiviert sind. Am anderen Ende versuchen Kriminelle, Cyberterroristen und auch Geheimdienste etwa durch Sabotage und Spionage wirtschaftliche und politische Vorteile für sich und ihre Auftraggeber zu erwirken. Häufig geht die Gefahr von *internen Angreifern* aus, beispielsweise von (ehemaligen) Mitarbeitern, die von Rachegelüsten oder Frust angetrieben sind, s. Tab. 1.1.

Informationen über die Absichten eines Angreifers können unter Umständen Rückschlüsse auf seine Vorgehensweise und seine Methodik ermöglichen. So legen i. d. R. Fahrzeugdiebe keinen Wert auf die Zerstörung und Schädigung von Fahrzeugen oder deren Hersteller, wohingegen *Hacktivisten* zur Verfolgung ihrer Ideologien und die

organisierte Kriminalität zum Erzielen finanzieller Vorteile vor unrechtmäßigen Handlungen wie mutwillige Sabotage, Diebstahl von Informationen und Erpressung nicht zurückschrecken.

Ein weiterer Faktor ist die Ausdauer und Beharrlichkeit des Angreifers. Lässt er sich bei ersten Misserfolgen schnell vom Ziel abbringen? Oder wird er auch über einen längeren Zeitraum allen Rückschlägen zum Trotze wiederholt versuchen, seine Ziele zu erreichen?

Kenntnisse und Fähigkeiten Hier muss zwischen Angriffen unterschieden werden, für deren Vorbereitung und Durchführung ein Angreifer entweder nur öffentlich verfügbares Wissen benötigt oder sich (vertrauliches) Insiderwissen beschaffen muss. Für einfachere, eher wahllose Angriffe sind darüber hinaus lediglich oberflächliche technische Kenntnisse erforderlich, wohingegen für komplexere, zielgerichtete Angriffe oftmals ein Team aus technisch versierten Experten benötigt wird.

Werkzeuge und Ressourcen Auch bezüglich der zur Verfügung stehenden Werkzeuge und Ressourcen unterscheiden sich die Angreifertypen erheblich. Während elementare Werkzeuge und Ausrüstung bereits für ein geringes Budget über das Internet frei erworben werden können stehen aufwendige Laboreinrichtungen wie Röntgengeräte, Elektronenrastermikroskope oder einfach „nur“ ein Fahrzeug, das zu Versuchszwecken angegriffen werden kann, in der Regel nur größeren, organisierten Gruppen mit entsprechendem finanziellen Hintergrund zur Verfügung. Neben dem Budget spielen auch Einfluss auf bzw. Kontakte zu verschiedenen, einschlägigen Unternehmen eine Rolle – u. a. zur frühzeitigen und exklusiven Informationsbeschaffung über Sicherheitslücken in Betriebssystemen und Anwendungen wie etwa sog. Zero-Day-Exploits.

1.2.3.2.2 Angriffsvektoren und Schwachstellen

Angreifer führen mithilfe eines Werkzeugs, z. B. eines Skripts oder einer Einrichtung zum Abhören einer Kommunikationsverbindung, bestimmte Aktionen durch. Das Ensemble aus mehreren einzelnen Aktionen bildet den *Angriffsvektor* oder auch Angriffspfad, der den Angreifer zum gewünschten Ziel führt. Beispiel: Ein Angreifer zeichnet zunächst die Kommunikation der CAN-Busverbindung auf, analysiert anschließend die aufgezeichneten Nachrichten und führt schließlich einen Replay-Angriff durch, um einen bestimmten Steuerbefehl abzusetzen.

Das Vorhandensein einer oder mehrerer Schwachstellen wird damit zu einer wesentlichen Voraussetzung für eine erfolgreiche Aktion und damit eine wichtige Information für die Beschreibung eines Angriffs. So sind im Beispiel von oben die Aktionen „Buskommunikation lesen“ und „gefälschte Nachrichten auf dem Bus senden“ nur möglich, weil der CAN-Bus physisch zugänglich ist, keinen Schutz vor unbefugtem Zugriff bietet und weder die Integrität noch die Authentizität der Nachrichten schützt.

Mit der Absicht, bekannt gewordene Schwachstellen möglichst schnell und flächendeckend zu beseitigen, werden sie in Datenbanken gesammelt und veröffentlicht. Die

CVE-Datenbank (Common Vulnerabilites and Exposures) ist ein bekanntes Beispiel, s. [18, 29].

Eine vollständige und systematische Untersuchung aller möglichen und wahrscheinlichen Angriffsvektoren für ein konkretes System ist Aufgabe der Bedrohungsanalyse (TARA).

Basierend auf Erfahrungen und Security Best Practices in der Automobilindustrie, sowie Ähnlichkeiten der jew. Systeme (vernetzte und (teil-)automatisierte Fahrzeuge) kann in erster Annäherung ein Großteil der wahrscheinlichsten und deshalb kritischsten Angriffsvektoren angegeben werden.

- Die *OBD-II-Schnittstelle* ist in jedem Fahrzeug vorhanden, leicht zugänglich und besitzt eine standardisierte Steckerbelegung. Über diese Schnittstelle können mehrere Komponenten des Fahrzeugnetzwerks erreicht werden, weshalb sie im Falle eines schwachen oder fehlenden Security-Mechanismus ein hohes Gefährdungspotential aufweist.
- *Funkverbindungen* wie WiFi und Mobilfunk besitzen eine hohe Reichweite und machen die physische Anwesenheit eines Angreifers überflüssig. Sie sind typischerweise die erste Einsprungstelle in einem Angriffspfad auf vernetzte Fahrzeuge, s. unten. Hinzu kommt, dass über die externen Kommunikationsverbindungen z. T. sehr sensible Daten wie etwa OTA-Updates und Backend-Kommunikation übertragen werden.
- Die *Infotainment-Einheit* bzw. die *Head-Unit,* die häufig mit USB und Bluetooth zur Smartphone-Integration ausgestattet ist, enthält in vielen Fahrzeugen einen veralteten, weil ungepatchten Linux-Kernel, vgl. [17]. Zudem ist aufgrund der vergleichsweisen umfangreichen Software und der unterstützten Schnittstellen die Wahrscheinlichkeit für Schwachstellen hoch.
- Die im Fahrzeugschlüssel integrierten Transceiver für die *Wegfahrsperre- und Remote-Keyless-Entry-Funktionen* sind Teil notwendiger Sicherheitsfunktionen, die den unauthorisierten Zugang und Fahrzeugdiebstahl vereiteln sollen.

1.2.3.3 Fallbeispiele

In mehreren akademischen Angriffen auf Fahrzeuge demonstrierten Forscher, dass eine Fernsteuerung diverser Fahrzeugfunktionen über Funkschnittstellen machbar ist.

Eine Gegenüberstellung von drei prominenten und gut dokumentierten Angriffen auf Fahrzeuge von FCA [17], BMW [32] und Tesla [22] zeigt, dass es eine große Ähnlichkeit in der Struktur des Angriffs bzw. im Angriffspfad gibt.

Der erste Schritt des Angriffspfads zielt auf die Head-Unit ab. Über eine gefälschte GSM-Station, einem gefälschten WiFi-Hotspot oder dem Durchprobieren eines bestimmten IP-Adressbereichs konnten die Angreifer eine Verbindung zur Head-Unit aufbauen.

Im nächsten Schritt wurde eine Schwachstelle in der Head-Unit ausgenutzt, um die Kontrolle darüber zu erlangen. Beim Angriff auf den Jeep Cherokee bestand die

Schwachstelle aus einem offenen Diagnoseprotokoll, das Befehle über Telnet empfing und ausführte, ohne deren Authentizität zu prüfen. Beim Angriff auf den BMW konnte über die gefälschte GSM-Basisstation ein *Memory Corruption*-Fehler ausgelöst werden und infolgedessen die Ausführung eines beliebigen Codes erwirkt werden.

Ausgehend von der Head-Unit wurde im folgenden Schritt versucht, über das CAN-Gateway auf die CAN-Busse der Antriebstrang-, Body- und Chassis-Domänen zuzugreifen. Beim Jeep konnte die Software des CAN-Gateways heruntergeladen, reverse-engineert und wieder reprogrammiert werden- unter anderem weil keine Securityfunktion wie etwa eine Signaturprüfung implementiert war. Im Falle des angegriffenen BMWs konnte über eine Chip-to-Chip-Kommunikation (QNet) von der bereits kompromittierten Head-Unit auf das CAN-Gateway zugegriffen werden – ebenfalls aufgrund einer fehlenden Authentifizierung.

Zusammengefasst waren die Angreifer in der Lage, komplexe Angriffspfade zu erstellen, damit ausgehend von einer externen Funkverbindung u. a. beliebige Steuerbefehle auf den CAN-Bussen abgesetzt werden konnte. Unter anderem konnten auf diese Weise die Bremsen, das Automatikgetriebe, der Motor, die Infotainmenteinheit, die Klimaanlage und die Scheibenwischer ferngesteuert werden, vgl. [8].

In allen drei Fällen bedarf der eigentliche Angriff einer mehrmonatigen Vorbereitung durch ein mehrköpfiges Expertenteam. Nichtsdestotrotz weisen derartige Angriffe auf mehrere, neuralgische Punkte hin, die für zukünftige Fahrzeugentwicklungen zwingend abgesichert werden sollten.

1.2.4 Herausforderungen für Security im Automobilbereich

In diesem Abschnitt wird auf verschiedene, branchentypische Aspekte hingewiesen, die die Automobilindustrie als Ganzes vor große Herausforderungen stellen.

Mehrstufige Lieferkette
Die Lieferketten bestehen in der Automobilindustrie aus mehrstufigen, verschachtelten Netzwerken und logistischen Prozessen. Beispiel: Ein Systemlieferant beauftragt einen Dienstleister mit der Entwicklung einer Softwarekomponente und der Dienstleister integriert hierfür u. a. Softwarebibliotheken weiterer, externer Quellen. Eine vollständige Übersicht und Kontrolle werden somit zur Herausforderung. Hersteller müssen zur Umsetzung eines ganzheitlichen Securitykonzepts bestimmte Anforderungen an ihre Lieferanten übertragen. Damit alle Beteiligten, d. h. praktisch die gesamte Lieferkette, an einem Strang ziehen können setzt voraus, dass Zielvorgaben und Methodik rechtzeitig kommuniziert und verstanden wurden. Zulieferer und deren Vertragspartner müssen ihre Security-Expertise nicht nur für den Zeitraum der Produktentwicklung einbringen, sondern auch für eventuelle Korrekturen und Problemanalysen wie etwa Security-Incidents, die im Verlauf des Produktlebenszyklus auftreten.

Kosten- und Zeiteffizienz
Sowohl das finanzielle Budget als auch die zur Verfügung stehende Zeit für die Entwicklung sind kostbare und beschränkte Ressourcen. Das Abwägen und Ausloten, wieviel Security geradeso ausreichend für die Sicherheit und Zuverlässigkeit des Produkts ist, aber sich noch im zeitlichen und finanziellen Rahmen befindet, ist eine Gratwanderung.

Gesucht ist ein wirtschaftlich vernünftiges Securitykonzept, das auf der einen Seite ausreichende Security-Maßnahmen definiert, um das System gegen ein bestimmtes Bedrohungsszenario zu schützen, und das auf der anderen Seite gerade eben so effektiv ist, dass die Kosten, die ein Angreifer für die Vorbereitung und Durchführung eines Angriffs aufbringen muss, den finanziellen und ideellen Profit des Angriffsziels übersteigen.

Produkte mit übertriebenen Schutzmaßnahmen sind für diese Branche zu teuer und deshalb nicht wettbewerbsfähig. Produkte mit zu geringen Schutzmaßnahmen oder mit löchrigem Schutzkonzept werden mit hoher Wahrscheinlichkeit früher oder später Opfer eines Angriffs, was wiederum finanzielle Folgen nach sich ziehen könnte.

Eine innovationsgetriebene Entwicklung legt ihre Schwerpunkte naturgemäß auf die funktionalen Leistungsmerkmale wie etwa automatisierte, elektrisch angetriebene und vernetzte Fahrzeuge. Für Qualitätsmerkmale wie Security, die (noch) nicht zu den Unique-Selling-Points zählen, besteht die Herausforderung darin, mit dem Entwicklungstempo mitzuhalten und das Produkt trotzdem sicher zu machen.

Lebensdauer von Fahrzeugen
Fahrzeuge über den gesamten Zeitraum ihrer Lebensdauer gegen Gefahren von Cyberangriffen zu schützen stellt die Branche vor eine große Herausforderung. Einerseits sind Prognosen über zukünftige Bedrohungsszenarien schwierig, da Cybersecurity oftmals ein Wettrennen zwischen Angreifer und Verteidiger ist. Andererseits weil Ressourcen für zukünftige Änderungen, Erweiterungen und Korrekturen vorzuhalten aufwendig und teuer ist.

Long-Term-Support (LTS) in Verbindung mit Krypto-Agilität und der Möglichkeit, OTA-Updates auszurollen, werden zu wichtigen Teilen der Lösung. Aber auch hier gilt es, einige Fragen vorab zu klären: Wie wird sichergestellt, dass kritische Updates innerhalb einer festgelegten Frist auf den Fahrzeugen installiert werden? Wie werden Fahrzeugbesitzer bzw. Fahrer darüber informiert und welche Möglichkeiten zur Einflussnahme werden ihnen eingeräumt? Wie verändert sich beispielsweise die rechtliche Situation (bzgl. Haftung), falls ein Unfall geschieht nachdem der Fahrer einem Update nicht zugestimmt hat?

Ressourcenbedarf für Security
Die stringenten Echtzeitanforderungen und die knappen Speichergrößen und Bandbreiten eingebetteter Systeme stehen oftmals im Widerspruch zum Ressourcenbedarf von Securityfunktionen.

Ein Beispiel, das alle Aspekte umfasst ist die Absicherung der Buskommunikation (s. Abschn. 5.3.4): Für das kryptographische Verfahren wird sowohl auf der Sender-Seite als auch auf der Empfänger-Seite Speicherplatz für die Algorithmen benötigt. Die Berechnung vor dem Senden bzw. nach dem Empfangen verzögert die Vorgänge und erhöht damit die Latenz in der Buskommunikation. Zusätzlich müssen die Prüfsummen auf dem Bus übertragen werden, was einen erhöhten Bedarf der Bandbreite nach sich zieht. Einerseits kann durch performante Hardware, insbesondere durch Kryptobeschleuniger, Latenz und Speicherbedarf optimiert werden, andererseits zeigen die begrenzten Ressourcen die Grenzen der Machbarkeit auf, d. h. es können nicht beliebig viele Securityfunktionen integriert werden.

Wechselwirkungen und Abhängigkeiten von Safety und Security
Robuste, fehlertolerante Systeme erfordern den Schutz vor Safety- und Security-bedingten Risiken. Security-Maßnahmen sind explizit auch deshalb erforderlich, um Safety-Funktionen vor Cyberangriffen und deren Folgen zu schützen – „Keine Safety ohne Security!“. Bestimmte Safety-Maßnahmen wie etwa redundante Komponenten, Signalpfade, Schnittstellen, etc. können allerdings die Angriffsoberfläche vergrößern und damit zusätzliche Security-Vorkehrungen erfordern. Um einen Zirkelschluss zu vermeiden sollten in umgekehrte Richtung keine Safety-Anforderungen an die Security-Funktionen gestellt werden.

Literatur

1. Amorim, T., et al. (2017). Systematic pattern approach for safety and security co-engineering in the automotive domain. *Lecture Notes in Computer Science, 329–342*. https://doi.org/10.1007/978-3-319-66266-4_22.
2. Avizienis, A., et al. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing, 1*(1), 11–33. https://doi.org/10.1109/tdsc.2004.2.
3. Bernstein, D. J., et al. (2013). *Factoring RSA keys from certified smart cards: Coppersmith in the wild. International Conference on the Theory and Application of Cryptology and Information Security.* Springer.
4. Bogdanov, A., et al. (2011). Biclique cryptanalysis of the Full AES. *Lecture notes in computer science, 344–371*. https://doi.org/10.1007/978-3-642-25385-0_19
5. Bundesamt für Sicherheit in der Informationstechnik. (2021). *Technische Richtlinie BSI TR-02102–1 Kryptographische Verfahren: Empfehlungen und Schlüssellängen.*
6. Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*(6), 644–654. https://doi.org/10.1109/tit.1976.1055638

7. European Telecommunications Standards Institute. (2017). *TS 102 165–1: CYBER Methods and Protocols. Part 1: Method and Pro Forma for Threat, Vulnerability. Risk Analysis (TVRA).* Technical Specification.
8. Fröschle, S. & Stühring, A. (2017). Analyzing the capabilities of the CAN Attacker. Computer Security – ESORICS 2017, 464–482. https://doi.org/10.1007/978-3-319-66402-6_27
9. International Electrotechnical Commission. (2003). *IEC-60300–3–1: Dependability Management.*
10. ISO. (1989). ISO 7498–2. *information processing systems open systems interconnection basic reference model-part 2: Security architecture.*
11. ISO. (2011a). ISO 26262 – *Road vehicles – Functional safety, Part 1–10. ISO/TC 22/SC 32 – Electrical and electronic components and general system aspects.*
12. ISO. (2011b). ISO/IEC 27005:2011 – *Information technology, security techniques, information security risk management*.
13. ISO. (2020). ISO/SAE DIS 21434 *Road Vehicles – Cybersecurity engineering*.
14. Kelsey, J., et al. (1998). Cryptanalytic attacks on pseudorandom number generators. *Fast Software Encryption, 168–188*. https://doi.org/10.1007/3-540-69710-1_12.
15. Killmann, W., & Schindler, W. (2011). *A proposal for: Functionality classes for random number generators.* BSI.
16. Lee, Y. R., et al. (2004). Multi-party authenticated key agreement protocols from multi-linear forms. *Applied Mathematics and Computation, 159*(2), 317–331. https://doi.org/10.1016/j.amc.2003.10.018.
17. Miller, C. & Valasek, C. (2015). *Remote exploitation of an unaltered passenger vehicle*. Black Hat USA.
18. Mitre – Common Vulnerabilities and Exposures. (2005). MITRE – CVE. http://cve.mitre.org. Zugriffsdatum 2021-06-01.
19. Moriarty, K., et al. (2016). PKCS# 1: *RSA cryptography specifications version 2.2. Internet Engineering Task Force, Request for Comments, 8017*.
20. Nasser, A. M., et al. (2017). An approach for building security resilience in AUTOSAR based safety critical systems. *Journal of Cyber Security and Mobility, 6*(3), 271–304. https://doi.org/10.13052/jcsm2245-1439.633.
21. Nguyen, H. N., et al. (2019). Developing a QRNG ECU for automotive security: Experience of testing in the real-world. 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). Published. https://doi.org/10.1109/icstw.2019.00033.
22. Nie, S., et al. (2017). *Free-fall: Hacking tesla from wireless to can bus*. DEFCON. https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf. Zugriffsdatum 2021-06-01.
23. Paar, C., Pelzl, J. & Preneel, B. (2010). *Understanding cryptography: A textbook for students and practitioners*. Springer.
24. Ruddle, A., et al. (2009). *Security requirements for automotive on-board networks based on dark-side scenarios*. EVITA Project.
25. SAE International. (2016). J3061 – *Cybersecurity guidebook for cyber-physical vehicle systems.*
26. SAE on-Road Automated Driving Committee. (2016). SAE J3016. *Taxonomy and definitions for terms related to driving Automation systems for on-road motor vehicles.*
27. Shannon, C. E. (1949). Communication theory of secrecy systems*. *Bell System Technical Journal, 28*(4), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x.
28. Skoglund, M., et al. (2018). In search of synergies in a multi-concern development lifecycle: Safety and cybersecurity. *Developments in Language Theory, 302–313*. https://doi.org/10.1007/978-3-319-99229-7_26.

29. Sommer, F., et al. (2019). Survey and classification of automotive security attacks. *Information, 10*(4), 148. https://doi.org/10.3390/info10040148
30. Stevens, M., et al. (2007). Chosen-prefix collisions for MD5 and Colliding X.509 Certificates for different identities. *Advances in Cryptology - EUROCRYPT, 2007*, 1–22. https://doi.org/10.1007/978-3-540-72540-4_1
31. Stigge, M., et al. (2006). *Reversing CRC – Theory and practice.* HU Berlin.
32. Tencent Technology Co. (2018). *Experimental security assessment of BMW cars: A summary report*. https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf. Zugriffsdatum 2021-06-01.