Automotive Cybersecurity in hybriden E/E-Architekturen

Vertrauensplatzierung, Protokoll-Bridging und Safety/Security-Co-Engineering zwischen AUTOSAR Classic, AUTOSAR Adaptive und HPC-Plattformen

---

Abstract

Der Übergang von domänenbasierten zu zonalen E/E-Architekturen mit zentralen High-Performance-Computern (HPC) verlagert die Sicherheitsgrenze vom zentralen Gateway in die fahrzeuginternen Netze. Dieses Whitepaper untersucht, an welcher Stelle der Kommunikationskette zwischen HPC-Applikation und Aktor kryptografisches Vertrauen verankert werden sollte. Wir vergleichen vier Architekturvarianten (Re-Authentisierung in der Zonen-ECU, SecOC Ende-zu-Ende vom HPC zur Aktor-ECU, Ethernet bis zum Rand mit 10BASE-T1S und MACsec sowie CAN XL mit CANsec) anhand eines expliziten Angreifermodells und mehrerer Kriterien, darunter Vertrauensannahmen, Schlüsselbeziehungen, Overhead, Latenz und Fehlerverhalten aus Safety-Sicht. Ergänzend quantifizieren wir die Fälschungsresistenz trunkierter SecOC-MACs und den Platzbedarf post-quantensicherer Signaturen. Ergebnis ist eine Entscheidungsmatrix, die die TARA nach ISO/SAE 21434 nicht ersetzt, ihr aber eine begründete Vorauswahl von Architekturoptionen liefert.

---

1. Einleitung: Aufbrechen der Perimeter-Sicherheit

Klassische E/E-Architekturen verließen sich auf zentrale Gateways, die als Firewall zwischen externen Schnittstellen (Connectivity, OBD) und den internen Fahrzeugbussen (CAN, LIN, FlexRay) fungierten (Perimeter Security). Mit der Einführung des Software-Defined Vehicles (SDV) und Zonal-Architekturen verschiebt sich die Komplexität auf zentrale HPCs.

Auf diesen HPCs laufen dynamische, serviceorientierte Software-Plattformen (z. B. AUTOSAR Adaptive auf Linux oder QNX, häufig partitioniert durch einen Hypervisor), während Zonen-Steuergeräte auf Mikrocontrollern mit AUTOSAR Classic und harten Echtzeitanforderungen arbeiten. Die Sicherheitsgrenze verschiebt sich damit in die internen Netze. Ein Angreifer, der eine Schwachstelle in einem HPC-Dienst ausnutzt, kann versuchen, sich lateral in Richtung Zonen-Ebene zu bewegen (Lateral Movement) und dort Aktorik zu beeinflussen. Dass dies kein theoretisches Szenario ist, zeigen dokumentierte Angriffsketten von der Telematikeinheit bis auf den CAN-Bus [9, 10].

1.1 Fragestellung und Beitrag

Die Verlagerung der Sicherheitsgrenze ist gut dokumentiert [9, 10]. Offen ist in der Praxis weniger das Ob als das Wo: An welcher Stelle der Kette HPC, Zonen-ECU, Aktor-ECU wird Authentizität geprüft, und welche Komponente muss dafür vertrauenswürdig sein? Dieses Papier leistet dazu drei Beiträge:

- ein explizites Angreifermodell mit vier Angreiferklassen (Kap. 1.2),
- einen kriterienbasierten Vergleich von vier Varianten der Vertrauensplatzierung (Kap. 3.4, Tabelle 1),
- eine quantitative Abschätzung zu MAC-Trunkierung und zu Schlüssel- und Signaturgrößen post-quantensicherer Verfahren (Kap. 3.5 und 5, Tabelle 2).

1.2 Angreifermodell und Methodik

Wir unterscheiden vier Angreiferklassen:

- **A1 (Remote, unprivilegiert):** Codeausführung in einem nicht privilegierten Container oder Prozess auf dem HPC, z. B. über eine Schwachstelle in Connectivity- oder Infotainment-Diensten. Kein Zugriff auf Schlüsselmaterial.
- **A2 (HPC-privilegiert):** Kontrolle über Betriebssystem oder Hypervisor-Partition des HPC und damit Zugriff auf alle Krypto-Operationen, die der HPC über sein HSM anstoßen darf, nicht aber auf die Schlüssel selbst.
- **A3 (Zonen-ECU):** Codeausführung auf dem Applikationskern einer Zonen-ECU, z. B. über eine fehlerhafte Diagnose- oder Ethernet-Implementierung. Das HSM bleibt intakt, kann aber als Orakel benutzt werden.
- **A4 (physischer Buszugriff):** Der Angreifer schließt ein eigenes Gerät an einen Feldbus an, etwa über einen von außen erreichbaren Stecker, und sendet eigene Botschaften. Dieses Muster ist von realen Fahrzeugdiebstählen bekannt [20].

Seitenkanal- und Fault-Injection-Angriffe auf einzelne Steuergeräte betrachten wir nicht. Ebenfalls außerhalb des Rahmens liegen abgesicherte Diagnose, Debug-Schnittstellen und Audit-Prozesse. Die Analyse ist qualitativ-argumentativ: Zahlenwerte sind aus Spezifikationen abgeleitet, eigene Messungen liegen ihr nicht zugrunde.

2. Kryptografische Grundlagen der Architektur-Schichten

Die heterogene Struktur erfordert einen detaillierten Blick auf die eingesetzten Security-Stacks und Trust-Anchors der jeweiligen Domänen [1, 3].

2.1 AUTOSAR Classic (Zonal ECUs)

Zonen-Steuergeräte auf Mikrocontroller-Basis (z. B. Infineon AURIX, Renesas RH850) unterliegen starken Hardware-Restriktionen.

- **Trust Anchor:** Hardware Security Module (HSM) mit eigenem Kern und geschütztem Schlüsselspeicher. Die Ausprägungen orientieren sich an den HSM-Profilen des EU-Forschungsprojekts EVITA (Full/Medium/Light) [3] bzw. an der Secure Hardware Extension (SHE) [11]; konkrete Implementierungen sind herstellerspezifisch.
- **Kryptografie:** Für die laufende Kommunikation werden symmetrische Verfahren (AES-128-CMAC) eingesetzt. Asymmetrische Verfahren (ECDSA, RSA) sind auf aktuellen HSMs verfügbar und für Secure Boot und die Prüfung von Software-Updates Stand der Technik; zur Absicherung einzelner Echtzeitnachrichten sind sie wegen Rechenzeit und Signaturgröße ungeeignet.
- **Architektur:** Der Crypto Service Manager (CSM) verwaltet kryptografische Jobs, die über das Crypto Interface (CryIf) an den Crypto Driver des HSM delegiert werden. Schlüssel und Zertifikate verwaltet der Key Manager (KeyM).

2.2 AUTOSAR Adaptive & POSIX-Systeme (HPC)

- **Trust Anchor:** Hardware Root of Trust des SoC (integriertes HSM bzw. Secure Enclave), die Secure Boot und geschützten Schlüsselspeicher bereitstellt. Trusted Execution Environments (z. B. ARM TrustZone) ergänzen dies als isolierte Ausführungsumgebung, sind aber selbst kein Trust Anchor.
- Kryptografie: Vollumfängliche Unterstützung asymmetrischer Kryptografie (Public Key Infrastructure, PKI, X.509-Zertifikate) und dynamischer Schlüsselaushandlung (z. B. Diffie-Hellman).
- **Architektur:** Kryptografie über ara::crypto, Zugriffssteuerung über das Identity and Access Management (IAM) [4]. Das IAM entscheidet lokal auf der Maschine anhand von Manifesten, welche Applikation welchen Service nutzen darf, und trennt dabei Policy Enforcement Point (PEP) und Policy Decision Point (PDP). Es ist kein PKI-Mechanismus: Die Authentisierung gegenüber anderen Steuergeräten erfolgt getrennt davon über Zertifikate (TLS) oder symmetrische Schlüssel (SecOC).

3. Protokoll-Bridging: Sicherheitsmechanismen an den Schnittstellen

Ein zentraler Angriffspfad in zonalen Architekturen ist der Übergang von Automotive Ethernet (HPC) auf klassische Feldbusse (Zonen-ECU) [2, 5].

3.1 Feldbus-Sicherheit: AUTOSAR SecOC

Zum Schutz vor Manipulation und Replay-Angriffen spezifiziert AUTOSAR das Modul Secure Onboard Communication (SecOC) [2]; ob und für welche Signale es eingesetzt wird, entscheidet der OEM. SecOC arbeitet auf PDU-Ebene und ist damit unabhängig vom Transportmedium: Dieselbe gesicherte PDU kann über CAN, CAN FD, FlexRay oder Ethernet übertragen werden. An die Nutzdaten werden ein Message Authentication Code (MAC, in der Praxis meist AES-128-CMAC) und ein in der Regel

trunkierter Freshness Value (FV) angehängt. SecOC sichert Integrität, Authentizität und Aktualität, nicht die Vertraulichkeit.

- **Herausforderung:** Bei klassischem CAN (8 Byte Nutzlast) muss der MAC stark trunkiert werden; AUTOSAR beschreibt dafür Profile, etwa Profil 1 (24Bit-CMAC-8Bit-FV) mit 24 Bit MAC und 8 Bit FV [2]. Bei CAN FD (bis 64 Byte) besteht mehr Spielraum, die Trunkierung bleibt aber eine Abwägung zwischen Bandbreite und Fälschungsresistenz (Kap. 3.5). Das Freshness Value Management ist in AUTOSAR nicht normiert, sondern OEM-spezifisch, typischerweise über Trip- und Reset-Zähler und Synchronisationsnachrichten eines FV-Masters.

3.2 Ethernet-Sicherheit: MACsec, TLS/DTLS und SecOC

Auf HPC-Ebene wird Sicherheit auf unterschiedlichen OSI-Schichten umgesetzt:

- **Layer 2 (MACsec, IEEE 802.1AE [5]):** Sichert einzelne Links ab (Hop-by-Hop), wahlweise nur mit Integritätsschutz oder zusätzlich verschlüsselt (GCM-AES). Hardware-Offloading in PHYs und Switches macht MACsec leistungsfähig; die Schlüsselaushandlung erfolgt über MKA (IEEE 802.1X).
- **Oberhalb der Transportschicht (TLS über TCP, DTLS über UDP):** Sichert Verbindungen zwischen Endpunkten ab. Für SOME/IP, das überwiegend UDP nutzt, ist TLS weder vorgeschrieben noch die Regel; alternativ kommen SecOC auf PDU-Ebene, IPsec oder MACsec zum Einsatz. Der Handshake verursacht asymmetrischen Rechenaufwand beim Verbindungsaufbau, nicht bei jeder Nachricht.

3.3 Variante V1: Re-Authentisierung in der Zonen-ECU

Wird das HPC-Netz mit TLS oder MACsec und der Feldbus mit SecOC abgesichert, endet der kryptografische Schutz in der Zonen-ECU. Sie arbeitet dann als Security Gateway:

1. Das HPC sendet einen Befehl über eine authentisierte Verbindung (z. B. TLS oder MACsec).
2. Die Zonen-ECU terminiert den Schutz, prüft die Authentizität des Senders und gleicht den Befehl mit ihren lokalen Routing- und Filterregeln ab. Eine TLS-Terminierung setzt asymmetrische Kryptografie auf der Zonen-ECU voraus; das ist mit aktuellen HSMs machbar, kostet aber Ressourcen beim Verbindungsaufbau.
3. Die Zonen-ECU erzeugt die Feldbus-PDU, berechnet den SecOC-MAC mit einem eigenen Schlüssel und aktuellem FV und sendet sie an die Aktor-ECU.

Befund: Bei dieser Variante muss die Zonen-ECU vertrauenswürdig sein. Ein Angreifer der Klasse A3 kann gültige SecOC-Nachrichten an alle Aktoren der Zone erzeugen, ohne Schlüssel extrahieren zu müssen, weil er das HSM als Orakel nutzt. Die Zonen-ECU wird damit zum lohnendsten Angriffsziel der Zone. Die Variante ist jedoch nicht alternativlos, wie der folgende Vergleich zeigt.

3.4 Vergleich von vier Varianten der Vertrauensplatzierung

Tabelle 1 vergleicht vier Architekturvarianten für den Pfad vom HPC zum Aktor:

- **V1 Re-Authentisierung in der Zonen-ECU:** wie in Kap. 3.3 beschrieben.
- **V2 SecOC Ende-zu-Ende:** Der HPC erzeugt die SecOC-PDU mit einem Schlüssel, den er nur mit der Aktor-ECU teilt. Die Zonen-ECU leitet die PDU als PDU-Gateway unverändert von Ethernet auf CAN FD weiter und besitzt den Schlüssel nicht. Der Preis: Der HPC muss das PDU-Layout des Aktors kennen, die Dienstabstraktion der Zonenarchitektur wird an dieser Stelle durchbrochen.
- **V3 Ethernet bis zum Rand:** Aktoren werden über 10BASE-T1S angebunden und mit MACsec abgesichert; die Zonen-ECU wird zum Switch.
- **V4 CAN XL mit CANsec:** CANsec (CiA 613-2 [12]) überträgt die Sicherheitsfunktionen von MACsec auf die Sicherungsschicht von CAN XL (Integrität, Aktualität, Authentizität, optional Vertraulichkeit).

Tabelle 1: Vergleich der Varianten (A1 bis A4: Angreiferklassen aus Kap. 1.2)

Kriterium	V1 Zone re-authentisiert	V2 SecOC Ende-zu-Ende	V3 10BASE-T1S + MACsec	V4 CAN XL + CANsec
Vertrauen in Zonen-ECU	voll nötig	nur für Verfügbarkeit	nötig, falls Switch MACsec terminiert	nötig, falls Zone CANsec-Endpunkt ist
Integrität bei A3	nicht geschützt	geschützt	nur ohne Terminierung in der Zone	nur ohne Terminierung in der Zone
Integrität bei A2	nicht geschützt	nicht geschützt	nicht geschützt	nicht geschützt
Integrität bei A4	geschützt, wo SecOC eingesetzt	wie V1	geschützt	geschützt
Dienstabstraktion im HPC	bleibt erhalten	durchbrochen (HPC kennt PDU-Layout)	bleibt erhalten	bleibt erhalten
Schlüsselbeziehungen	wenige (je Zone)	viele (HPC zu jeder Aktor-ECU)	je Secure Channel bzw. Segment	je Segment
Overhead auf dem Feldbus	MAC + FV je PDU	wie V1	SecTAG (8 bis 16 Byte) + ICV (16 Byte) je Frame	CANsec-Header + ICV je Frame
Zusätzliche Latenz	Prüfen und neu erzeugen in der Zone	nur Prüfen in der Aktor-ECU	Hardware-Offload	Hardware im CAN-XL-Controller
Freshness	zweistufig (HPC-Zone, Zone-Aktor)	Ende-zu-Ende HPC-Aktor	im Protokoll integriert (Packet Number)	im Protokoll integriert
Migration Bestands-ECUs	einfach	Protokoll einfach, Schlüsselprozesse aufwendig	aufwendig (neue PHY)	aufwendig (neuer Controller)

Kriterium	V1 Zone re-authentisiert	V2 SecOC Ende-zu-Ende	V3 10BASE-T1S + MACsec	V4 CAN XL + CANsec
Werk und Werkstatt	Schlüssel je Zone; Anlernen an der Zone	Schlüssel je Aktor-ECU, mehrere Zulieferer; Anlernen gegen HPC	Schlüssel oder Zertifikat je Teilnehmer	Schlüssel je Teilnehmer
Reifegrad	in Serie etabliert	in Serie etabliert	im Aufbau	früh
Safety-Fehlerbild	Zone ist gemeinsame Fehlerursache	Verifikationsfehler lokal in der Aktor-ECU	abhängig von Terminierung	abhängig von Terminierung

Lesart: V2 verschiebt das Vertrauen von der Zonen-ECU auf HPC und Aktor-ECU und verkleinert so die Angriffsfläche gegenüber A3, zum Preis eines deutlich aufwendigeren Schlüsselmanagements. V3 und V4 lösen das Bandbreitenproblem, aber nicht automatisch das Vertrauensproblem: Terminiert die Zonen-ECU die Sicherung, gilt der Befund aus Kap. 3.3 unverändert. Gegen einen privilegierten Angreifer auf dem HPC (A2) hilft keine der Varianten. Wirksam sind dort nur Schlüsselnutzungsrichtlinien im HSM (welcher Prozess darf mit welchem Schlüssel MACs erzeugen), Plausibilisierung in der Aktor-ECU und Angriffserkennung (Kap. 4.3). V2 hat zudem zwei Preise, die in Protokollvergleichen gern fehlen: Der HPC verliert für die betroffenen Signale seine Dienstabstraktion, und die Schlüsselprozesse in Werk und Werkstatt werden deutlich aufwendiger, weil jede Aktor-ECU, auch nach einem Tausch, einen eigenen Schlüssel mit dem HPC teilen muss. An diesen Prozessen entscheidet sich die Machbarkeit eher als am Protokoll. Die Reifegrade in der letzten Zeile sind unsere Einschätzung, keine Markterhebung. Naheliegend ist eine Mischung: V2 für wenige sicherheitskritische Signalgruppen, V1 für den Rest.

Für A2 bietet sich ein Muster an, das aus der funktionalen Sicherheit bekannt ist: Linux wird nicht selbst als vertrauenswürdig behandelt, sondern von einer kleinen, durch einen Hypervisor isolierten Komponente überwacht. Elektrobit setzt dies für Safety-Anwendungen mit Ziel ASIL B bzw. SIL 2 um; eine externe Überwachungskomponente im Hypervisor fängt Aktionen des Kernels ab und prüft sie [18]. Auf Security übertragen hieße das, die Schlüsselnutzungsrichtlinie außerhalb des Linux-Systems durchzusetzen, denn das HSM selbst kennt keine Linux-Prozesse und kann nicht beurteilen, wer anfragt. Das ist als These zu verstehen; ein vergleichbarer Nachweis für Security liegt uns nicht vor.

3.5 Quantitative Abschätzung: Trunkierte MACs

Bei einem auf t Bit trunkierten MAC gelingt eine Fälschung pro Versuch mit der Wahrscheinlichkeit 2^-t. Entscheidend ist daher, wie viele Versuche ein Angreifer unbemerkt durchführen kann. Beispiel: Bei t = 24 Bit (SecOC-Profil 1) und 1.000 gefälschten Frames pro Sekunde auf einer Botschafts-ID beträgt die erwartete Zeit bis zur ersten erfolgreichen Fälschung 2^24 / 1.000 s, also rund 4,7 Stunden; eine Erfolgswahrscheinlichkeit von 50 % ist nach etwa 3,2 Stunden erreicht. Bei t = 64 Bit liegt der Erwartungswert bei rund 585 Millionen Jahren. 1.000 Frames pro Sekunde entsprechen auf einem CAN-Bus mit 500 kbit/s grob einem Viertel der Buskapazität und sind damit erreichbar, wenn auch nicht unauffällig.

Folgerung: Kurze MACs sind nur zusammen mit einer Überwachung fehlgeschlagener Verifikationen vertretbar. SecOC meldet Verifikationsfehler; diese sollten als Sicherheitsereignisse an den Intrusion Detection System Manager (IdsM) weitergegeben werden [14], damit ein Brute-Force-Versuch nach wenigen hundert Fehlversuchen erkannt wird und nicht nach Stunden Erfolg hat. Eine automatische Sperre nach Fehlversuchen ist aus Verfügbarkeitsgründen oft nicht gewollt; umso wichtiger ist die Meldung. Die Rechnung setzt voraus, dass der Angreifer den aktuellen FV kennt, was für einen Teilnehmer am Bus in der Regel zutrifft.

Messwerte zur Latenz der MAC-Erzeugung und -Prüfung auf Serien-Hardware liegen diesem Papier nicht zugrunde; die Einträge zu Latenz und Overhead in Tabelle 1 sind daher qualitativ zu lesen.

4. Normatives Risikomanagement (ISO/SAE 21434 & ISO 26262)

Die UN-Regelungen R155 (Cybersecurity Management System, CSMS) und R156 (Software Update Management System, SUMS) sind in der EU seit Juli 2024 für alle neu zugelassenen Fahrzeuge verbindlich und verlangen den nachweisbaren Schutz über den gesamten Lebenszyklus [6]. Methodisch stützen sich OEM und Zulieferer dabei auf ISO/SAE 21434 [1] und für Software-Updates auf ISO 24089 [13].

4.1 Threat Analysis and Risk Assessment (TARA)

ISO/SAE 21434 beschreibt in Kapitel 15 die Methoden der TARA: Identifikation von Assets und Schadensszenarien, Bewertung der Auswirkungen (Safety, Financial, Operational, Privacy), Bedrohungsszenarien, Angriffspfadanalyse, Bewertung der Angriffsdurchführbarkeit (Attack Feasibility) und Risikobestimmung. Die TARA wird für das jeweilige Item durchgeführt; eine bestimmte Methode zur Bedrohungsidentifikation schreibt die Norm nicht vor. STRIDE ist dafür eine verbreitete, aber optionale Hilfe. Für hybride Architekturen kommt es darauf an, Angriffspfade über die Domänengrenze hinweg zu modellieren, statt HPC und Zonen-ECU als getrennte Items isoliert zu bewerten.

- **Bewertung der Durchführbarkeit:** Beim Ansatz über das Angriffspotenzial, einem der in der Norm beschriebenen Bewertungsansätze, gilt: Je geringer das erforderliche Angriffspotenzial, desto höher die Durchführbarkeit. Ein Remote-Zugang über Mobilfunk in den HPC senkt das erforderliche Angriffspotenzial für den gesamten Pfad bis zur Zonen-Ebene, vor allem in den Faktoren Gelegenheitsfenster (Window of Opportunity) und Ausrüstung. Die Attack Feasibility steigt damit, nicht das Angriffspotenzial.

4.2 Co-Engineering: Safety (ISO 26262) und Security (ISO/SAE 21434)

Das Zusammenspiel von funktionaler Sicherheit und Security ist kein unbeachtetes Feld: ISO 26262-2:2018 enthält mit Anhang E eine Anleitung zu den möglichen Wechselwirkungen [7], und ISO/SAE 21434 fordert die Abstimmung mit angrenzenden Disziplinen. Offen ist die technische Auflösung konkreter Zielkonflikte, vor allem in zwei Punkten:

- **Zeitbudgets:** ISO 26262 legt keine festen Reaktionszeiten fest; maßgeblich ist das für das Item bestimmte Fault Tolerant Time Interval (FTTI). Für die laufende Kommunikation zählt, ob die symmetrische Prüfung (CMAC-Verifikation, MACsec-ICV) einschließlich der Warteschlangen im HSM in das Zeitbudget passt. Asymmetrische Operationen und TLS-Handshakes fallen dagegen beim Start oder Verbindungsaufbau an und belasten vor allem die Startzeit, nicht die Latenz jeder Nachricht.
- **Verfügbarkeit vs. Integrität:** Aus Security-Sicht ist eine nicht verifizierbare Nachricht zu verwerfen; aus Safety-Sicht kann das Ausbleiben eines Signals selbst gefährlich sein. Die Auflösung liegt nicht in einer Lockerung der Prüfung (etwa Nutzung vor der Verifikation), sondern in einem gemeinsamen Fehlerkonzept: Verifikationsfehler werden wie Kommunikationsausfälle behandelt (Ersatzwert, Degradation, Übergang in den sicheren Zustand), gezählt und als Sicherheitsereignis gemeldet. SecOC stellt dafür den Verifikationsstatus bereit. Sonderfälle wie das zeitweise Übersteuern der Prüfung, etwa in der Fertigung, müssen in TARA und Sicherheitskonzept gleichermaßen begründet sein.

4.3 Angriffserkennung und Reaktion

UN R155 verlangt, Angriffe auf Fahrzeuge zu erkennen und darauf zu reagieren [6]. Für hybride Architekturen heißt das: Sicherheitsereignisse der Zonen-ECUs (SecOC-Verifikationsfehler, Filterverstöße) und des HPC werden gesammelt, im Fahrzeug vorverarbeitet und an ein Vehicle Security Operations Center (VSOC) weitergeleitet. AUTOSAR stellt dafür den Intrusion Detection System Manager (IdsM) für Classic und Adaptive bereit [14]. Varianten mit kurzen MACs (Kap. 3.5) sind ohne diese Überwachung nicht vertretbar.

5. Implementierungs-Herausforderungen in der Praxis

Aus Spezifikationen und Integrationsprojekten ergeben sich drei Hürden für die aktuelle Fahrzeuggeneration:

1. **Schlüsselmanagement:** Die Verteilung und Erneuerung symmetrischer SecOC-Schlüssel über 15 Jahre und mehr, ohne dass jede Zonen- oder Aktor-ECU eine eigene Backend-Anbindung besitzt. Etablierte Bausteine sind das Schlüssel-Update-Protokoll der SHE-Spezifikation (Nachrichten M1 bis M5) [11], der AUTOSAR Key Manager und ein fahrzeuginterner Key Master, der vom Backend über einen authentisierten Kanal versorgt wird. Variante V2 aus Kap. 3.4 vervielfacht die Zahl der Schlüsselbeziehungen und macht dieses Thema zum eigentlichen Engpass.
2. **Post-Quantum-Kryptografie (PQC):** Symmetrische Verfahren wie AES-CMAC in SecOC sind nur durch den Grover-Algorithmus betroffen, der die effektive Schlüssellänge halbiert; mit 256-Bit-Schlüsseln ist dies abgedeckt. Betroffen sind die asymmetrischen Verfahren: Secure Boot, Signaturen von Software-Updates (R156), Fahrzeug-PKI, TLS und V2X. NIST hat ML-KEM, ML-DSA und SLH-DSA 2024 standardisiert (FIPS 203 bis 205) [15]. Das BSI empfiehlt

in TR-02102-1 hybride Verfahren und nennt neben ML-KEM auch FrodoKEM und Classic McEliece, für Signaturen ML-DSA, SLH-DSA sowie die zustandsbehafteten Hash-Signaturen LMS/HSS und XMSS [16, 8]. Das Problem ist nicht die CAN-Nutzlast, sondern die Größe von Schlüsseln und Signaturen im Boot- und Update-Pfad (Tabelle 2) sowie die Lebensdauer der Fahrzeuge: Ein heute in OTP-Speicher geschriebener Root-Schlüssel für Secure Boot muss bis in die 2040er Jahre halten. Krypto-Agilität und PQC-fähige Boot-Pfade sind daher eine Entscheidung zum Start of Production.

Tabelle 2: Schlüssel- und Signaturgrößen in Byte, zum Vergleich: CAN-FD-Nutzlast 64 Byte [15, 17]

Verfahren	Öffentlicher Schlüssel	Signatur	Bemerkung
ECDSA P-256	64	64	klassisch, nicht quantensicher
ML-DSA-44	1.312	2.420	FIPS 204
ML-DSA-65	1.952	3.309	FIPS 204
ML-DSA-87	2.592	4.627	FIPS 204
SLH-DSA-SHA2-128s	32	7.856	FIPS 205, langsame Signatur
LMS (h = 10, w = 8)	56	1.452	RFC 8554, zustandsbehaftet

Für Secure Boot ist die Signaturgröße unkritisch (einmal je Image); relevant sind Verifikationszeit, Codegröße im Boot-ROM und Hardware-Unterstützung im HSM. LMS und XMSS sind wegen ihrer gut verstandenen Sicherheitsannahmen für Firmware-Signaturen attraktiv, erfordern aber ein strenges Zustandsmanagement im Signatur-Backend.

3. **IAM und Zugriffsrechte:** Die konsistente Abbildung von Berechtigungen (welcher Prozess darf welchen Service nutzen und mit welchem Schlüssel MACs erzeugen) über HPC, Zonen-ECUs und Backend. Im HPC werden diese Regeln überwiegend statisch aus Manifesten abgeleitet. Die Herausforderung liegt in ihrer durchgängigen Ableitung aus der TARA und ihrer Kopplung an die Schlüsselnutzungsrichtlinien des HSM, nicht in der Verwaltung zur Laufzeit.

6. Zusammenfassung und Ausblick

Cybersecurity in zonalen E/E-Architekturen lässt sich nicht durch Netzwerk-Isolation allein erreichen. Die zentrale Architekturentscheidung ist, an welcher Stelle der Kette HPC, Zonen-ECU, Aktor Authentizität geprüft wird. Re-Authentisierung in der Zonen-ECU (V1) ist einfach, macht diese aber zum lohnendsten Angriffsziel. SecOC Ende-zu-Ende vom HPC (V2) verschiebt das Vertrauen und verlagert die Komplexität ins Schlüsselmanagement. 10BASE-T1S mit MACsec und CAN XL mit CANsec lösen Bandbreitenprobleme, nicht aber automatisch die Vertrauensfrage. Gegen einen privilegierten Angreifer auf dem HPC hilft keine Transportabsicherung; hier wirken nur Schlüsselnutzungsrichtlinien, Plausibilisierung und Angriffserkennung. PQC betrifft im Fahrzeug vor allem Secure Boot und Updates und erfordert Entscheidungen zum Start of Production.

Organisatorisch folgt daraus, dass klassische Steuergeräteentwicklung und HPC-Softwareentwicklung nicht getrennt arbeiten können. Dass etablierte und neue Fahrzeughersteller dabei voneinander lernen müssen, ist auch der Tenor von [19].

Offene Fragen für weitere Arbeiten sind (1) Messdaten zum Overhead der Varianten V1 bis V4 auf Serien-Hardware, (2) Verfahren, um HSM-Schlüsselnutzungsrichtlinien systematisch aus der TARA abzuleiten, und (3) Migrationspfade zu PQC-fähigen Boot-Ketten für bestehende Steuergeräteplattformen.

---

7. Literatur- und Quellenverzeichnis

1. **ISO/SAE 21434:2021**, Road vehicles: Cybersecurity engineering. ISO, Genf, 2021 (insbesondere Kap. 15).
2. **AUTOSAR**, Specification of Secure Onboard Communication, Classic Platform, Release R24-11 (bzw. das im Projekt verwendete Release).
3. **EVITA** (E-safety Vehicle Intrusion proTected Applications), EU-FP7-Projekt, 2008 bis 2011, HSM-Spezifikation.
4. **AUTOSAR**, Specification of Identity and Access Management, Adaptive Platform, Release R24-11.
5. **IEEE 802.1AE-2018**, Media Access Control (MAC) Security; **IEEE 802.1X-2020**, Port-Based Network Access Control (MKA).
6. **UNECE WP.29**, UN Regulation No. 155 (Cyber security and cyber security management system) und No. 156 (Software update and software update management system), 2021.
7. **ISO 26262-2:2018**, Road vehicles: Functional safety, Part 2: Management of functional safety, Anhang E.
8. **NIST SP 800-208**, Recommendation for Stateful Hash-Based Signature Schemes, 2020.
9. **S. Checkoway et al.**, Comprehensive Experimental Analyses of Automotive Attack Surfaces, USENIX Security Symposium, 2011.
10. **C. Miller, C. Valasek**, Remote Exploitation of an Unaltered Passenger Vehicle, Black Hat USA, 2015.
11. **AUTOSAR**, Specification of Secure Hardware Extensions (SHE), ursprünglich HIS, heute als AUTOSAR-Dokument veröffentlicht.
12. **CAN in Automation**, CiA 613-2, CAN XL add-on services, Part 2: CANsec, Version 1.0.0, 2026.
13. **ISO 24089:2023**, Road vehicles: Software update engineering.

14. **AUTOSAR**, Specification of Intrusion Detection System Manager (Classic und Adaptive Platform), Release R24-11.
15. **NIST FIPS 203, 204, 205**, Module-Lattice-Based KEM, Module-Lattice-Based Digital Signature, Stateless Hash-Based Digital Signature Standard, August 2024.
16. **BSI TR-02102-1**, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Version 2026-01.
17. **IETF RFC 8554**, Leighton-Micali Hash-Based Signatures, 2019.
18. **Elektrobit**, EB corbos Linux for Safety Applications, Produktbeschreibung, elektrobit.com, abgerufen am 02.10.2026.
19. **C. Stathatou, E. Huebner**, Mutual Learning for a Sustainable Automotive Future, UL Solutions (Software Intensive Systems), 2025.
20. **K. Tindell**, CAN Injection: keyless car theft, Canis Automotive Labs, April 2023.

ÜBER DEN HERAUSGEBER

Da Vinci Engineering GmbH (Stuttgart) ist spezialisiert auf Automotive Software Engineering. Als Entwicklungspartner für OEMs und Tier-1-Zulieferer unterstützt Da Vinci Engineering bei der Konzeptionierung hybrider E/E-Architekturen, der Integration von AUTOSAR Crypto-Stacks (Classic & Adaptive), der Durchführung von TARA-Methodiken nach ISO/SAE 21434 sowie dem domänenübergreifenden Performance- und Timing-Management sicherheitskritischer Netzwerke.

ÜBER DIE AUTOREN

Dr. Florian Schmidt: promovierter Software-Ingenieur, arbeitet seit 20 Jahren an Systemen für sicherheits- und security-regulierte Branchen, zuletzt mit Schwerpunkt Software-Defined Vehicle.

Martin Förster: Dipl.-Ing. Elektrotechnik, leitet seit 30 Jahren Projekte in der Entwicklung komplexer mechatronischer Systeme.