Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung

Jonas Schäufler

Security Architektur für Automotive Systeme

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Jonas Schäufler

Security Architektur für Automotive Systeme

Ausarbeitung eingereicht im Rahmen der Grundseminar

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 31. August 2018

Inhaltsverzeichnis

1 Hauptseminar Ausarbeitung

1.1 Automotive Network Security Architecture

Automotive Systeme sind heutzutage komplexe Systeme aus zahlreichen eingebetteten Sytemen die ein intra-vehicular Netzwerk bilden (IVN). Die Knoten des IVN können wiederrum Teilnehmer von anderen verschiedenenartigen Wide-Area Networks (WAN) sein, um mit Infrastruktur (V2I) oder anderen automotive Systemen (V2V) zu kommunizieren. Weitere Kommunikationspartner können direkt an das Vehikel angeschlossene Systeme sein (V2D) oder auch Personen die sich in der Nähe des Vehikels befinden (V2P) sowie Stromnetze (V2G). Diese Kommunikationtypen werden im Begriff der *Vehicle to X Kommunikation* (V2X) zusammengefasst.

Diese Netzwerke unterscheiden sich untereinander in ihrer Topologie und eingesetzter Technologie teilweise stark. Dies ist aber nur eine der Herausforderungen die sich bei dem Entwurf einer Architektur für automotive Systeme stellen. Im folgenden Abschnitt werden weitere Problematiken erläutert und aktuelle Lösungsvorschläge vorgestellt.

1.1.1 Herausforderungen

Zum einen gilt es das Intra-Vehichular Network (IVN) zu sichern, zum anderen müssen Sicherheitsvorkehrungen getroffen werden welche für das Automobil, als Teilnehmer von vielen verschiedenen anderen Netzwerken, gilt. Diese Netzwerke umfassen unter anderem die des Autoherstellers, Netzwerke von unabhängigen Serviceprovidern und Ad-Hoc Netzwerke aus Automobilen um untereinander Informationen auszutauschen, wie zum Beispiel im Bereich des autonomen Fahrens. Die vielfältigkeit der verschiedenen Netzwerke lässt auch schon erahnen, dass es auch verschiedene Interessengruppen gibt deren Güter es zu schützen gilt. Wobei auch die priorisierung der Schutzziele sich bei den verschiedenen Parteien unterscheiden können. Es gilt also eine Sicherheitsarchitektur zu entwerfen, welche diese verschiedenen Bereiche abdeckt, sowie erweiterbar und flexibel genug ist um bereits verwendete Technolgien als auch zukünftige Entwicklungen zu unterstützen und dazu mit ihnen skaliert.

1.2 aktueller Stand und Forschung

Aktuelle werden meist Einzellösungen für Subsysteme, Funktionalitäten oder Kommunikationskanäle in der Industrie verwendet. So spezifiziert der AUTomotive Open System ARchitecture (AUTOSAR) Standard eine Komponente, den Crypto Service Manager (CSM), die Standardisierte Kryptographische Funktionen bereitstellt, jedoch keine Dienste dergleichen auf höhere Abastraktionseben bietet. So müssen Sicherheitsmechanismen für jede Teil-Funktionalität auf

der Anwendungsebene implementiert werden, im Widerspruch zur etablierten AUTOSAR Methodologie welche auf Codegenerierung aus High-level Spezifikationen basiert.[1] Camek et al. argumentieren, dass durch neue Funktionalitäten, wie V2V (C2C) Kommunikation, neue *information and communication technlogy* (ICT) Architekturen im automotive Bereich entstehen, die nicht von Grund auf mit Rücksicht auf den Sicherheitsaspekt entworfen werden.[2] Deshalb eine Security Architektur nötig die adaptiv und verteilt ist, sowie unabhängig auf verschiedenen Ebene agieren kann. (Adaptive and Distributed Multiple Independent Levels of Security Architecture). Auch nennen Camek et al. Fragestellungen die es in diesem Rahmen zu erforschen und zu beantworten gilt und in das Design einer Sicherheitsarchitektur mit einfließen müssen. Hierzu zählt das Konfigurationsmanagement, Policy Management, Intrusion Detection, Audits, Product-Lifecycle und Business Model. Auch muss die Architekur Robust und Zuverlässig sein um für automotive Systeme geignet zu sein. Device Reliability und Road Safety sind Faktoren die berücksichtigt werden müssen.[3]

Synergien findet man im Forschungsbereich des *Internet der Dinge* (IoT), denn hier gibt es ähnliche Einschänkungen und Anforderungen. Begrenzte Resourcen, die Netz- und Raum-Mobilität der Teilnehmer und dass diese sich an Orten befinden können, bei denen jeder physisch mit dem Objekt interagieren kann, sind gemeinsame Problematiken.

1.2.1 EVITA Projekt

Das **E**-Safety **V**ehicle **I**ntrusion **P**rotected **A**pplications (EVITA) Projekt war ein Forschungsprojekt der EU mit dem Ziel eine Sicherheitslösung für automotive Netzwerke zu entwerfen und prototypisch zu implementieren. Eine vorraussetzung für eine sichere V2X Kommunikation ist ein sicherens Intra Vehicular Network (IVN). Deshalb konzetriert sich das EVITA Projekt hauptsächlich auf Kanale im on-board Netzwerk, um so eine Basis zu schaffen auf welcher sichere V2V und V2I Kommunikation aufgesetzt werden kann.

1.2.1.1 Security Engineering Prozess

Anhand einer Referenzarchitekur sind die Kommunikationskanäle und Nachrichten der Kommunikationspartner für verschiedene Szenarien abgeleitet worden. Neben V2V und V2I wurden auch *Nomadic Devices* sowie Diagnose und Nachrüstung beleuchtet. Jeder Use-Case wurde auf den Sicherheitsaspekt untersucht und technische sowie funktionale Anforderungen formuliert. [4]

Für die Ableitung der *Security Requirements* wurden die Uses-Cases mit Angriffsmotivationen kombiniert und *dark side* Szenarios verfasst welche als Angriffsbäume modeliert worden sind. Die *Security Requirements* werden den Use-Cases zugeordet und Semi-Formal notiert.[5] Weiterhin wird ein metamodel, das auf verschiedene Trust/Access-Models abgebildet werden kann, formal definiert und auf die *Security Requirements* angewandt. Siehe Abbildung 1.1. Da die formale Darstellung der Requirements sich nicht gut eignet um Mechanismen zu beschreiben, werden hierfür sogennante Security Building Blocks (SeBBs) mit cryptographischen Primitiven definiert. Die spezifischen und formalisierten Requirements wurden generalisiert und zu einer liste aus neun *top-level Requirements* zusammengefasst:

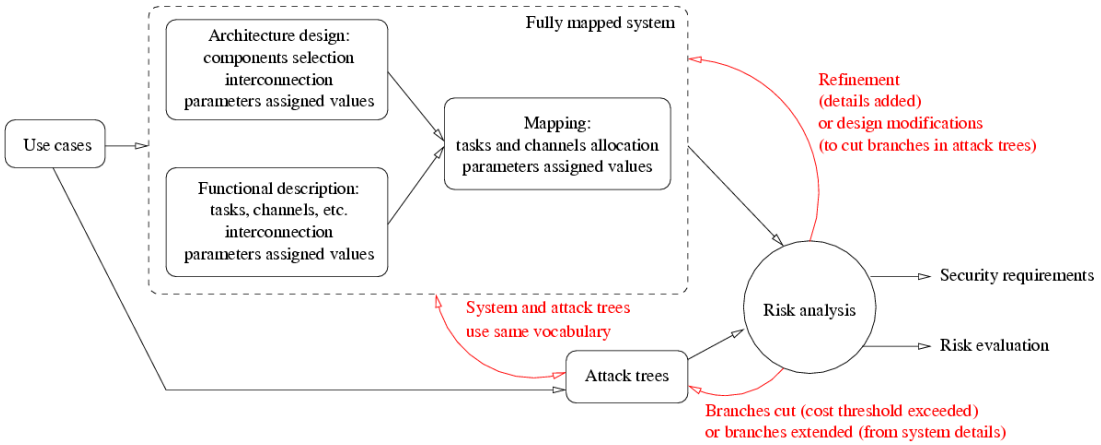

Abbildung 1.1: Security Requirement Engineering Prozess [5]

- **SR.1:** Integrität/Authentizität von e-Safety relevanten ereignissen.
- **SR.2:** Integrität/Authentizität der ECU/firmware installation/konfiguration
- **SR.3:** Sicheres *Execution Environment*
- **SR.4:** Physische Zugangskontrolle
- **SR.5:** Trusted ECU Platform. Integrität und Authentizität der Software
- **SR.6:** Sicheres Dateisystem
- **SR.7:** Vertraulichkeit der internen und externen Kommunikation
- **SR.8:** Geheimhaltung sensitiver Daten
- **FR.9:** keine Beeinflussung anderer Systeme

1.2.1.2 Architektur

Die *Root of Trust* der Architektur muss in der Hardware lokalisiert sein um das sichere Booten, Kontextwechsel und Bus-Nutzung zu gewährleisten zu können. Und um der Anwendung Speicher, cryptographische Funktionen mit Hardware-Beschleunigern und einer gehärteten CPU zur Verfügung zu stellen. Die Spezifikation der Architektur umfasst deshalb Hardware-Bauteile, Software-Module und Kommunikationsprotokolle. [6]

1.2.1.3 Hardware

Die *Hardware Security Module* (HSM) genannte Bauteile werden auf der Platine einer ECU verbaut, und stellen dem Kern der ECU ein Hardware Interface und Shared Memory zur Verfügung auf welcher Operationen des HSM ausgeführt werden können.

In Abbildung 1.2 ist eine generalisierte HSM-Architektur dargestellt. Aus flexibelitätsgründen enthält ein HSM einen programmierbaren sicheren Kern der alle angebotenen Funtionen

ausführt. Manche Funktionen könnten aber auch auf der Application CPU ausgeführt werden, wenn keine sensitiven Daten verwendet werden. Durch die Verwendung des eigenen Kerns bleiben im Falle einer Erweiterung die Änderung der Hardware der ECU minimal.

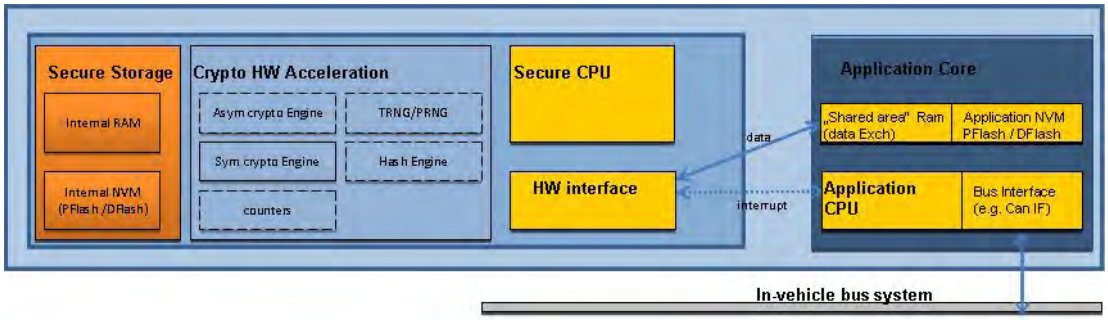

Abbildung 1.2: HSM-Architekur Übersicht [6]

Nicht jede Applikation benötigt den vollen Umfang an Funktionen. So muss ein HSM einer Head-Unit, welche für V2X Kommunikation zuständig ist, Beschleuniger für Asymmetrische Kryptographische Verfahren enthalten, für einen Sensor werden unter umständen nur symmetrische Verfahren verwendet da ein hoher Durchsatz an Daten vorliegt. Deshalb werden in der Architektur drei verschiedene HSM-Level spezifiziert:

full für V2X Kommunikation, *medium* für ECU-Eebene, und *light* für Sensoren/Aktoren.

	ECC-256	WHIRLPOOL	AES-128	PRNG	COUNTER	CPU	RAM	NVM
light			x	x			(x)	(x)
medium			x	x	x	x	x	x
full	x	x	x	x	x	x	x	x

Tabelle 1.1: Vergleich der HSM-Level Hardware-Komponenten

1.2.1.4 Software

Um mit Standards wie AUTOSAR oder SHE kompatibel zu sein und die flexibilität der Architektur zu erhalten werden Softwarefunktionen in Module ausgelagert, strikt getrennt und einer Domäne zugeordnet. Domänen werden auf dem Chip durch einen Microkernel getrennt, siehe Abbildung 1.3.

Das *vEPTM* dient als Abstraktionslayer für das HSM und bietet die Möglichkeit das HSM einer anderen ECU zu virtualisieren um dessen Funktionen zu nutzen, ermöglicht sicheres Booten, und gewährleistet Bi-Direktional Integrität zwischen Hardware-Bausteinen und Software-Modulen.

- **Entity Authentication Module**: Identifikation und Authentifizierung von Geräten, Nutzern und Rollen, SSO, Pseudonymisierung

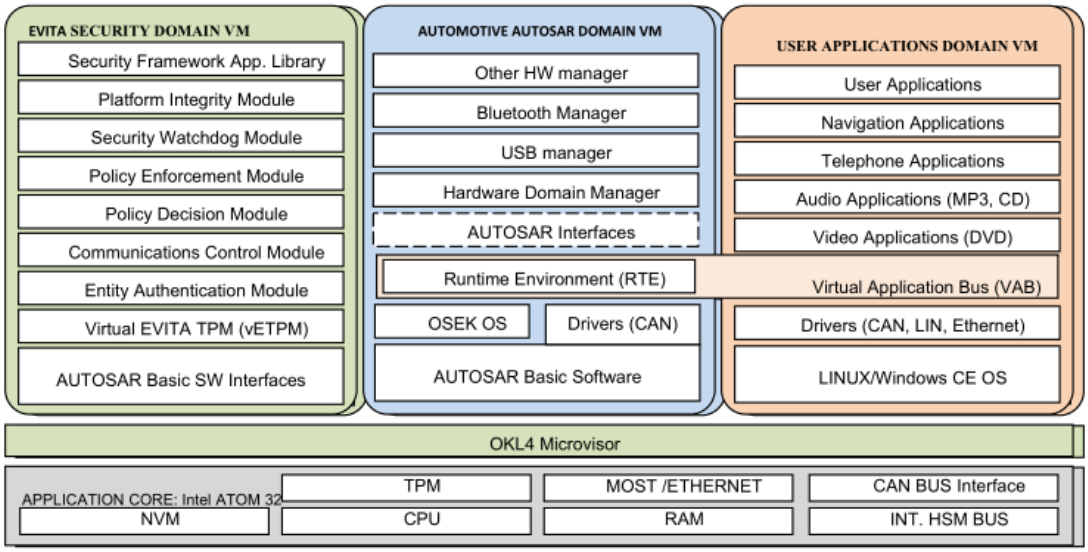

Abbildung 1.3: Software-Komponenten einer Head Unit ECU [6]

- **Communication Control Module**: Sichere Kommunikation entsprechend Policies für Interne Prozesskommunikation als auch IVN und externe Kommunikation, Plugins, Filter
- **Policy Decision Module**: Zugriffskontrolle auf Resourcen durch lokale oder remote Policy, verteilt
- **Security Watchdog Module**: Intrusion Detection / Prevention, Aktives und passives Monitoring
- **Platform Integrity Module**: Initialisierung bei Boot, Integritätsüberprüfung der Hard- und Software, Attestierungs-Service, UUID für Platform, Integritätskette zur Root-of-Trust
- **Secure Storage Module**: Zugriffskontrolle, Integrität, Geheimhaltung, Aktualität für nicht-flüchtigen Datenspeicher
- **Cryptographic Services**: Interface für Kryptographische funktionen, Freie Funktionen ohne Zustand, Applikation muss Schlüssel, Passwörter etc. verwalten.
- **Security Framework Application Library**: Interface zum Sicherheits-Framework um Operationen auf Applikationsebene zu implementieren

1.2.1.5 Ergebnisse und weitere Forschung

Aus sehr präzisen und formal definierten Anforderungen, abgeleitet durch die Methoden der Risiko- und Anforderungsanalyse adaptiert für die Automobilindustrie, wurde eine Sicherheitsarchitekur entworfen, prototypisch implementiert und getestet. Ein wichtiges Resultat des Projektes war die oben genannte Ableitung der Anforderungen. Diese bieten einen guten Satz

an klassischen Use-Cases und auch neue, noch nicht existierenden Funktionalitäten. Die daraus formal abgeleiteten Anforderungen und die im Prozess entstandenen Artefakte (Attack-Trees, Trust-Model, SeBBs) bieten eine gute Basis für weitere Forschungen. Deshalb wurde 2011, basierend auf den Ergebnissen von EVITA, ein weiteres EU-Forschungsprojekt gestartet: PRESERVE (Preparing Secure V2X Communication Systems), das sich auf die externe Kommunikation spezialisiert. Im folgenden Abschnitt wird ein kurzer Überblick über die Anforderungen und Architekur des Projektes gegeben.

1.2.2 PRESERVE

PRESERVE vereint Funktionalität und Wissen aus verschiedenen vorherigen Projekten. So wurden als Use-Cases wurden vier, vom Car-to-Car Communication Consortium (C2C-CC) als *C2C Phase 1* spezifizierten Fällen, verwendet: *Collision Warning*, *EV Warning*, *Hazardous Location Notification*, *Enhanced Route Guidance and Navigation*. Basierend auf diesen Szenarien und den Ergebnissen der Risiko- und Anforderungsanalyse von vorherigen Projekten (EVITA, SeVeCom, ETSI, sim^{TD}, PRE-DRIVE, C2C-CC) sind die *Security-Requirements* für das PRESERVE Projekt formuliert, und den entstammenden Projekten zugeordent worden. [7]

1.2.2.1 Anforderungen

PRESERVE betrachtet, wie man dem Namen entnehmen kann, hauptsächlich V2X Kommunikation. Deshalb wurden in den Anforderungen diverse Metriken und Obergrenzen definiert, da es sich dabei natürlich um ein unsicherers Medium handelt (Wireless) als im IVN. Diese Metriken beschreiben nicht nur das Zeitverhalten der Kommunikation, sondern auch die der Cryptographischen Funktionen wie *Certificate Cache Lookup Effectiveness*, *Signature Generation Delay*, *Pseudonym Change Delay*, *Signature Verification Delay*.

1.2.2.2 Architektur

Nicht nur Anforderungen wird auf oben genannte Projekte zurückgegriffen. Auch Komponenten der Architekturen von SeVeCom und EVITA werden über die definition einer *Abstract Architecture* den PRESERVE Architektur-Komponenten zugeordnet, welche eine Obermenge der Teil-Funktionalitäten bildet. V2X-Funktionalität basieren hauptsächlich auf Modulen des SeVeCom Projektes während die Funktionalität für interne Kommunikation und On-Chip Security von EVITA Modulen abgebildet wurden, siehe 1.2.1.4 und 1.4. Speziell für die Geheimhaltung der Identiät in der V2X-Kommunikation sind werden Mechanismen des PRECIOSA Projektes (Privacy Enabled Capability In co-Operative systems and Safety Applications) verwendet.[8]

Die abstrakte Architekturvereint mehrere verschiedene Funktionalitäten ein einer Komponente. Sie basiert auf der Referenzarchitekur des *Intelligent Transport Systems Communications Architecture* Standards [9].

- **Cryptographic Operations**: Cryptographic Services, HSM oder OpenSSL-Fallback
- **Secure Information**: Sicherer Speicher und Software, Konsistenz und Plausbilitätsprüfung, Privacy

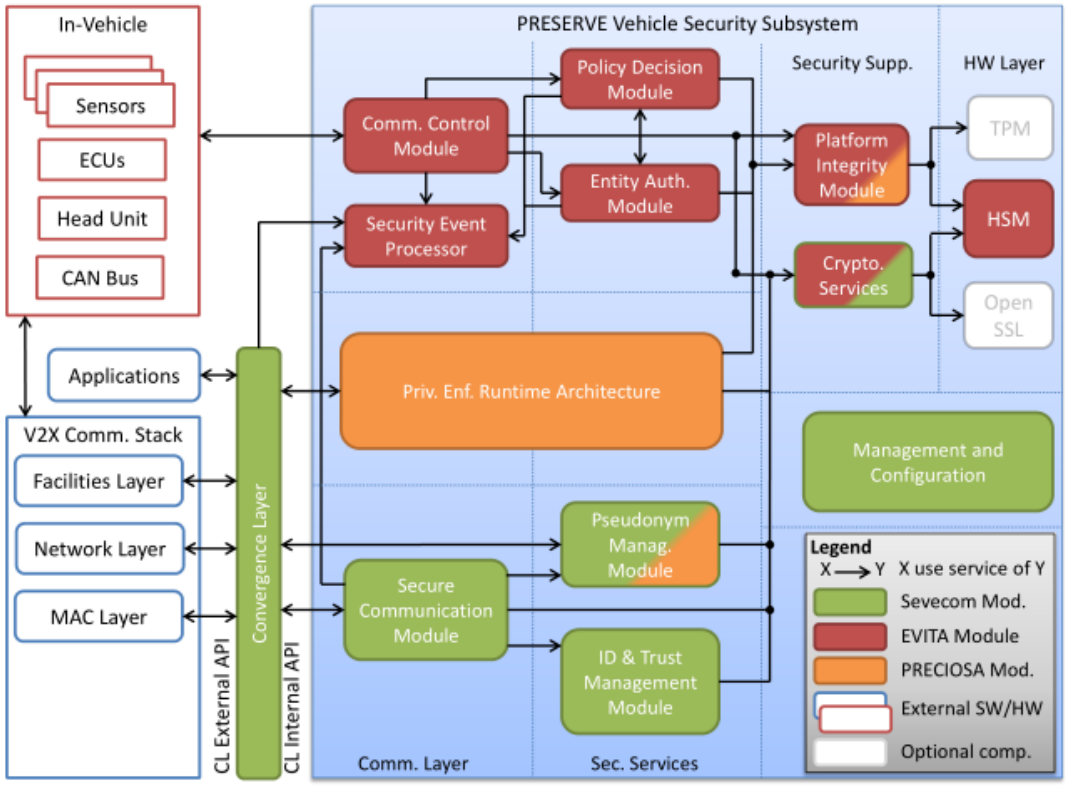

Abbildung 1.4: PRESERVE Architektur Komponenten[8]

- **Security Management**: Management der Zugangsdaten und Authentifizierung
- **Security Policies**: Verteilung und Anwendung der Policy
- **Security Analysis**: Audit und Monitoring
- **Secure Communication**: externe und interne Kommunikation

Zwei Kernkomponenten der Architektur sind das *Convergence Layer* (CL) und das Secure Communications Module (SCM). Das CL sorgt für kompatibilität zu verschiedenen Link-Layern und stellt die Schnittstelle zum Vehice Security Subsystem (VSS) für Applikationen bereit. Siehe 1.4. Alle V2X Nachrichten (außer PKI management) laufem über den SCM, er ist der Entry-Point des *Secure Communciation* Stacks. Für den *Secure Communications Stack* werden die Funktionalität andere Komponenten des VSS verwendet: *Cryptographic Services* (CRS), *Security Event Processor* (SEP), *Pseudonym Manager Module* (PMM), *ID and Trust Management Modle* (IDM). Das SCM dazu bietet ein Interface für andere Komponenten über den Stack Nachrichten zu senden. Applikationen können den Stack über eine API der CL verwenden. Diese ist zwecks kompatibilität sehr minimal gehalten und unterstüzt zwei Modus Operandi: Abwicklung über

das VSS mit Event-basierter Schnittstelle, jeweils ein Hook für eingehende und ausgehende Nachrichten, oder *Stack controlled*, Security Stack (sign, verify, encrypt, decrypt) wird im *Communication Stack* abgearbeitet, auf Netzwerk-Schicht (siehe Abbildung 1.5. Des weiteren bietet die API verschiedene Hilfsfunktionen für Pseudonym-Management und Logging.

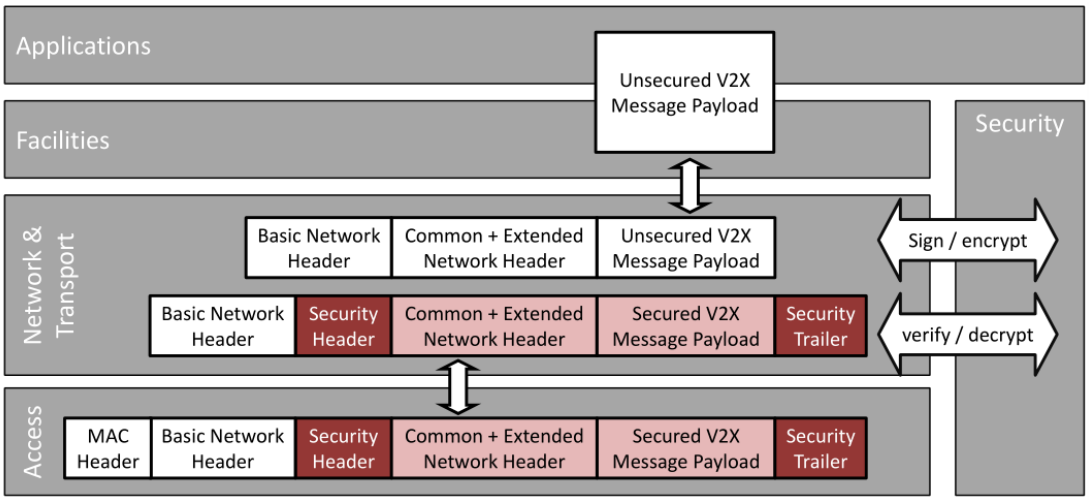

Abbildung 1.5: *Security Layer* im Kommunikations-Stack[8]

Das Security-Layer ist Vertikal zum Kommunikations-Stack, es können also aus jeder Netzwerk-Schicht die im oberen Abschnitt genannten Funktionen verwendet werden. Applikationen haben kein direktes Interface zum CL, PRESERVE hat sich dafür entschiedene auf Netzwerkebene zu ver-/entschlüsseln und zu verifizieren. Dies ist ein wichtiger Schritt im Vergleich zu bisherigen AUTOSAR API, in welchem die Implementation der Sicherheitsfunktionen Teil der Anwendung war.

1.3 Fazit und Ausblick

Der Einsatz von neuer Technologie im Automobil, wie LIDAR und Machine Learning, veranlasst Erstausrüster die Entwicklung voranzutreiben. Die zentralen Steuergeräte werden zu diesem Zweck mit immer stärkeren CPUs und GPUs ausgestattet. So besitzt das, im Oktober 2017 eingeführte, NVIDIA Drive PX Pegasus 16 spezialgefertigte ARM64 Prozessoren und 4 GPUs. Dies macht es möglich einerseits aufwendige Operationen zu delegieren, als auch die Machine Learning Kapazität für Diagnosefunktionen, wie die Erkennung von Anomalien im Netzwerk, zu nutzen.

Smart Ethernet Switches[10], speziell für die Automobilindustrie entwickelt, erlauben es das Interne Netzwerk weitgehend zu homogenisieren. So entstehen Netzwerk-Architekturen die auf Zonen (lokalität) basieren und nicht wie bisher auf Domänen (Gerätetyp/Link-Layer). Während der Entwicklung der Architektur für das EVITA Projekt wurde auch wegen des inhomogenen Netzwerks hohen Wert auf Flexibilität gelegt. Das Design der EVITA Architektur ist sehr modular und wie sie mit einer Architekur, die speziell für Zonen-basierte Netzwerke mit Ethernet-Backbone entworfen worden ist, vergleichbar wäre, ist bleibt offen. Das Ziel anlaufende SecVI Projektes ist es eine Netzwerkarchitektur, mit Ethernet-Backbone, zu entwickeln die robuster per Design ist. Also werden einerseits derzeitig nötige Mechansimen dort eventuell nichtmehr benötigt und es können Konzepte und Mechansimen die in anderen Ethernet basierten Netzwerken, wie dem Internet, alt eingessen sind. Zum Beispiel im bereich des Monitoring, Intrusion Detection, Firewalls, Key Management.

Ein Vorteil ist, dass durch das hohe Maß an Flexibilität neue Funktionalitäten ohne viel Komplexität hinzugefügt werden können. Jedoch könnte vielleicht durch eine verminderung der nötigen Flexibilität und Ethernet auf Link-Layer eine effizientere und günstigere Lösung entstehen. Auch in betracht der Hardware, denn dadurch das die Rand-ECUs in einer Zonen-basierten Architektur ein Gateway darstellen und diese, im Vergleich mit einzelnen ECUs, mit stärkeren Kernen ausgerüstet sind könnte die *light*, *medium*, *full* Unterteilung der EVITA HSM's nicht die effizienteste Aufteilung der Resourcen darstellen. Ein sehr wichtiger Apsekt in langlebigen Serienprodukten wie in der Automobilindustrie. Durch die verwendung Zentraler Knoten enstehen natürlich auch neue Flaschenhälse an den Rändern der Zonen und um die nötigen Zeitanforderungen auch in Zukunft einhalten zu können muss vielleicht eine angepasstere Lösung verwendet werden.

Literatur

[1] C. Bernardeschi, M. Di Natale, G. Dini und D. Varano, "Modeling and generation of secure component communications in autosar", in *Proceedings of the Symposium on Applied Computing*, Ser. SAC '17, Marrakech, Morocco: ACM, 2017, S. 1473–1480, ISBN: 978-1-4503-4486-9. DOI: 10.1145/3019612.3019682.

[2] A. G. Camek, C. Buckl und A. Knoll, "Future cars: Necessity for an adaptive and distributed multiple independent levels of security architecture", in *Proceedings of the 2Nd ACM International Conference on High Confidence Networked Systems*, Ser. HiCoNS '13, Philadelphia, Pennsylvania, USA: ACM, 2013, S. 17–24, ISBN: 978-1-4503-1961-4. DOI: 10.1145/2461446.2461450.

[3] S. Ray, W. Chen, J. Bhadra und M. A. A. Faruque, "Extensibility in automotive security: Current practice and challenges", in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Juni 2017, S. 1–6. DOI: 10.1145/3061639.3072952.

[4] E. Kelling, M. Friedewald, T. Leimbach, M. Menzel, P. Säger, H. Seudié und B. Weyl, "Specification and evaluation of e-security relevant use cases", EVITA project, EVITA Deliverable D2.1, Dez. 2009.

[5] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet und G. Pedroza, "Security requirements for automotive on-board networks based on dark-side scenarios", EVITA project, EVITA Deliverable D2.3, Dez. 2009.

[6] B. Weyl, M. Wolf, F. Zweers, T. Gendrullis, M. S. Idrees, Y. Roudier, H. Schweppe, H. Platzdasch, R. E. Khayari, O. Henniger, D. Scheuermann, A. Fuchs, L. Apvrille, G. Pedroza, H. Seudié, J. Shokrollahi und A. Keil, "Secure on-board architecture specification", EVITA project, EVITA Deliverable D3.2, Aug. 2011.

[7] J. P. Stotz, N. Bißmeyer, F. Kargl, S. Dietzel, P. Papadimitratos und C. Schleiffer, "Security requirements of vehicle security architecture", PRESERVE project, PRESERVE Deliverable 1.1, Juni 2011.

[8] N. Bißmeyer, S. Mauthofer, J. Petit, M. Lange, M. Moser, D. Estor, M. Sall, M. Feiri, R. Moalla, M. Lagana und F. Kargl, "V2x security architecture", PRESERVE project, PRESERVE Deliverable 1.3, Jan. 2014.

[9] E. T. S. Institute, "Intelligent transport systems (its) communications architecture", Techn. Ber., 2010.

[10] M. Ziehensack und M. Kunz, "Smart ethernet switch architecture", 2017 IEEE Standards Association (IEEE-SA) Ethernet & IP @ Automotive Technology Day, Techn. Ber., 2017.