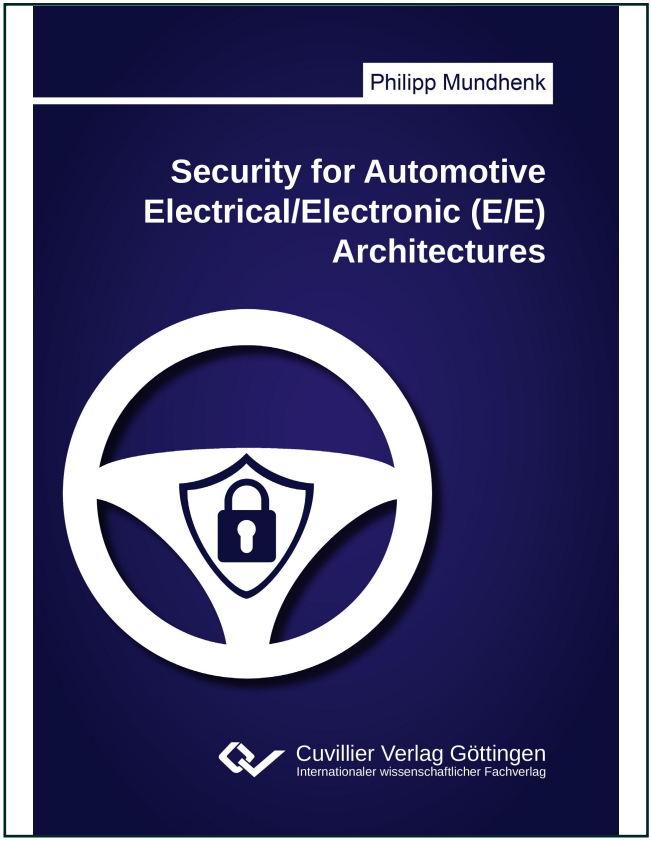
Philipp Mundhenk (Autor)

Security for Automotive Electrical/Electronic (E/E) Architectures



https://cuvillier.de/de/shop/publications/7598

1 Introduction

Vehicles today contain a large number of assistant and entertainment functions. These functions are realized as electronic components running control software. Such electronics are used throughout the vehicle, covering all functional domains. Advanced Driver Assistance Systems (ADASs) such as lane keeping and braking assistants, as well as autonomous vehicles rely heavily on software, processing the input of sensors distributed around the vehicle and computing actions to take for the actuators in the vehicle. In case of a lane keeping assistant, the sensor input might originate from a camera system. An ECU processes this input, computes how much the steering needs to be actuated and sends appropriate commands to the steering wheel motor. Today, such functions are core elements in the vehicle, both from a safety, as well as a business perspective. On the one hand, these systems can avoid accidents, on the other, they heavily influence the decision of the customer to buy or forgo a vehicle [142].

To achieve the required functionality, multiple sensors, actuators and computational units (ECUs) need to be interconnected. The sensors and actuators are typically attached to ECUs as well, for filtering, pre-computation and encoding of data [139]. The networks, or Electric/Electronic (E/E) architectures, existing in vehicles today have been designed at times when vehicles were single, non-interconnected units. In recent years, this has changed significantly, as vehicles are equipped with Internet connections, WiFi, cellular (3G, 4G) and vehicle-to-X (v2X) connections, among others. Nowadays, most vehicles on the market have some sort of interconnection. While the internal networks slowly develop to adjust to these new requirements, they have not been designed with security in mind. Attacks on single vehicles, as well as attacks on vehicle fleets over Internet connections are becoming a reality [107]. The interconnection of insecure vehicle networks to the Internet is reminiscent of the interconnection of the first

computer networks to the Internet and the resulting security issues in the 1980s. However, the security in vehicle networks is on many levels more concerning than the security in personal computer systems, as the vehicle networks and the connected ECUs have direct influence on the safety of the vehicle and its passengers.

The goal of this thesis is to advance secure communication in vehicles by contributing approaches for analysis and design of secure communication. The security of vehicles is analyzed and improvements are suggested based on the experience gained by designing and constructing the electric taxi EVA. EVA is a purpose-built electric taxi for tropical megacities developed and built in TUM CREATE. The lessons learned when securing the vehicle networks in EVA will motivate the remainder of the thesis, showing how to secure the communication in vehicle networks and how to evaluate this security. Furthermore, the challenges with security in the existing communication system FlexRay are outlined and one approach to solve these is shown.

In this chapter, the basic knowledge of automotive communication systems and network security is conveyed. Further, the challenges existing when combining these two domains are laid out. The detailed contributions of this thesis and the integration with existing work are shown.

1.1 Automotive Electrical/Electronic (E/E) Architectures

To understand the challenges to security in vehicles, the basic networking mechanisms need to be understood. In the following, an overview over E/E architectures in vehicles with a focus on existing and upcoming automotive bus systems is given.

Typically, automotive communication systems follow a bus structure (see Figure 1.1(c)). In many cases, these buses are interconnected by a central gateway in a star architecture (see Figure 1.1(a)). This concept is slowly changing, however, due to the introduction of functional *domains*, headed by a domain controller and interconnected over a backbone, typically in conjunction with a central gateway (see Figure 1.1(b)). The chosen architecture depends on the complexity, bandwidth and real-time requirements of the vehicle (see Section 1.1.1). Only recently, bus systems with sufficient bandwidth for backbone networks have been developed (see Section 1.1.3) [142].

The ECUs in vehicle architectures take up diverse tasks, from sensing tire pressure over user inputs via buttons to computations for systems such as Anti-lock Braking System (ABS) or Electronic Stability Program (ESP), and actuation of motor and brakes, among many others. ECUs will be shortly explored in Section 1.1.2.

1.1.1 Requirements

Based on the functions the vehicle network is to perform, the requirements on software and hardware for ECUs and communication systems can be defined [103]. Note that some of the requirements depend on the type of application. Some examples will be given in the following.

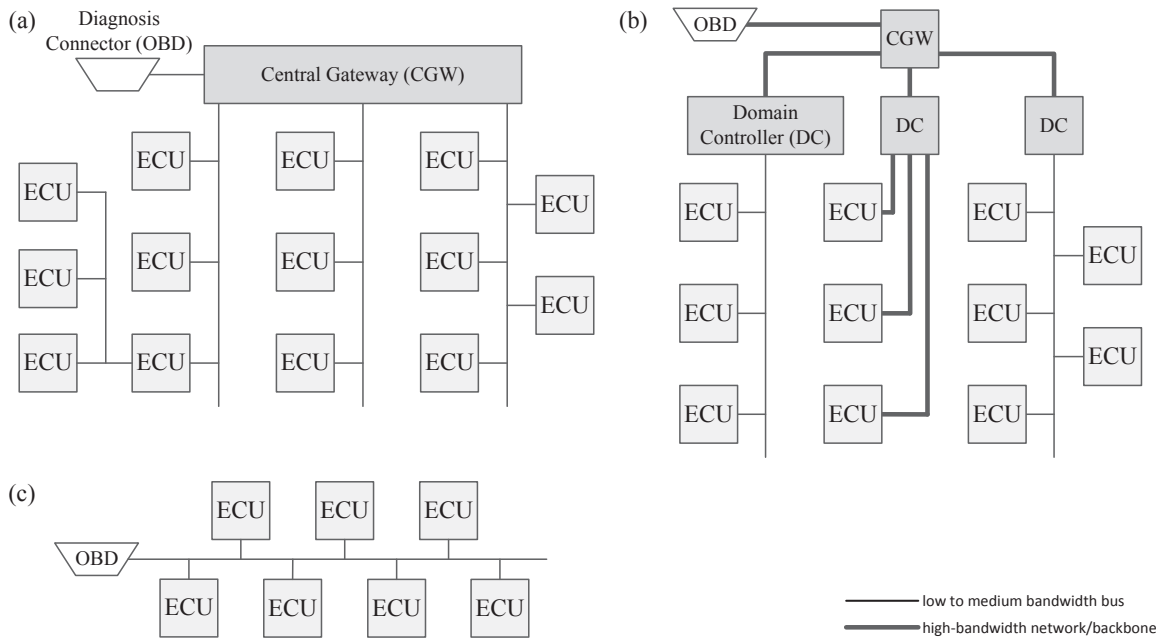


Figure 1.1: *Comparison of architecture variants: (a) Hierarchical bus systems with central gateway, (b) division into functional domains with domain controllers (DC), high-bandwidth backbone and central gateway (CGW) and (c) traditional single bus system.*

Cost. The overarching concern for vehicle architectures is cost. The minimization of cost for all components is key from the perspective of business model of the Original Equipment Manufacturers (OEMs) where revenues rise with lower production cost. Within this work, however, cost is not the main consideration. Whenever possible, cost is considered, e.g. by the reuse of existing elements, but this thesis is focused on the technical point of view on vehicle networks.

From the technical point of view, automotive architecture requirements can be characterized by their bandwidth demand, real-time capabilities, as well as number of devices, or, on a higher abstraction level, number of functions. In the following, we will quickly explain these requirements, before describing implementations addressing these requirements in more detail in the next section.

Bandwidth. Note that bandwidth demand strongly depends on the organization of the architecture. The drivetrain domain, e.g., typically requires less bandwidth than the infotainment domain. Overall, bandwidth demand in vehicle architectures is rising sharply and strongly driven by ADASs and infotainment systems. With an increasing amount of cameras and other data intensive sensors such as Lidar, the required bandwidth for data transmissions rises quickly. When transmitting High-Definition (HD) video data, as required by assistance systems such as lane keeping assistants, data rates of 1 to 18 MBit/s are required for a single camera, depending

on resolution, encoding, frame rate, etc. Traditional bus systems, such as shown in Figure 1.1(c) are in many cases not able to cope with such bandwidth demand. These networks have been designed to transmit control data, which in most cases is limited to messages of less than 2 Bytes length per application and data rates of less than 1 kBit/s per stream. Thus, new architectures and networks have been developed, allowing to segregate traffic and use these sensors and systems, without interfering with safety-critical control data. The most common of these networks will be discussed in Section 1.1.3.

Real-Time. Automotive systems require real-time data transmissions and processing, such that signals arrive and actions are taken guaranteed within a given time. This is to ensure that safety-critical functions can be executed as intended, without delay. An ABS, e.g., requires response times in the range of milliseconds to ensure that the vehicle remains steerable. These requirements originate in the potentially high speed of the vehicle and the frequency of actions required to fulfill a function. With increasing speed, longer response times or retransmissions of messages translate to a longer distance traveled before an action is taken. This can have potentially fatal consequences, e.g. when emergency braking is applied to stop the vehicle before crashing into the end of a traffic jam. Other applications, e.g., those where user interaction is required, require less rigorous response times, as the user reacts in slower time intervals. The infotainment is one example for such applications.

Size. Last but not least, the architecture requirements are defined by the number of functions and, corresponding to this, the number of devices. In short, the size of the architecture has a large influence on its structure. While for low-end cars with a minimal amount of electronics, a single Controller Area Network (CAN) bus might be sufficient, high-end cars with requirements for bandwidth and real-time varying with the function, might require a hierarchical system, including some high bandwidth buses. It is important to note that with multi-core processors and generally more powerful ECUs being introduced into the automotive domain, the trend of *ECU consolidation* is picking up speed. Due to the large number of ECUs, the overhead in weight, energy consumption and cost in vehicles is not negligible anymore. With ECU consolidation, the OEMs move away from the concept of ECU per function and start integrating multiple functions onto one ECU. Thus, the number of ECUs is remaining stable or even reduced, while the number of functions is increasing.

After a quick overview of the automotive domain, highlighting the bandwidth, real-time and size requirements on architectures, the following section will discuss ECUs in more detail.

1.1.2 Computation

Due to the distributed nature of sensors, controllers and actuators and the high degree of concurrency, automotive networks can be considered complex, heterogeneous, and distributed computers [139]. Each computation component is running on a single ECU and ECUs are dis-

tributed spatially in the vehicle. The computational capabilities of these ECUs vary widely. On the higher end are engine control units, infotainment systems and the central controllers for ADASs, often containing modern multi-core processors with large amounts of main memory. These systems are, in their computational capabilities, comparable to modern consumer electronics [121, 123]. Unix-based Operating Systems (OSs) such as QNX [136] represent the main OSs for infotainment systems.

Devices with lower computational capabilities reach down to 8-Bit processors [122, 138]. Core clock frequencies of these devices are in the range of two-digit Megahertz or below. Memory reaches down to single-digit Kilobytes. These devices are typically used for small switching tasks, such as recognizing or triggering simple on/off switches, triggering of motor controllers, etc. Often, such devices are used in subsystems, which might be attached to the main networks via bridges.

Between these two extremes, nearly all sizes and types of devices can be found in vehicles today [142]. When dimensioning software to be used on all devices in the vehicle, such as security mechanisms, any performance estimation must take this diverse network nature into account. This makes it difficult to exactly quantify the computational capabilities of the overall vehicle network.

ECU consolidation. As the number of ECUs in current vehicles is in the upper double-digit range, OEMs have started to combine ECUs [22]. This process of reducing the number of ECUs by combining multiple tasks on a single, more powerful ECU is called ECUs consolidation. It is especially useful for pure controller ECUs. ECUs which need to access hardware components, such as a switches or motors can not be integrated easily without cabling overhead. By removing the medium size ECUs and integrating them with high powered ECUs, the distribution of ECUs in automotive networks is changing. In the future, one can expect networks with more high-powered and less medium size ECUs, with the number of low-power ECUs remaining in similar numbers, possibly rising slowly in the high-end market.

Security in ECUs. Not all computation of an ECU is performed in the Central Processing Unit (CPU). Additional computation units allow the efficient computation of specific functions. Examples for this are, e.g., graphics accelerators (Graphics Processing Unit (GPU)) for devices with screens and cryptographic accelerators. Especially cryptographic accelerators are important in the context of this thesis. Larger and partially also mid-range ECUs are often equipped with cryptographic co-processors. While hardware accelerators allow storage of cryptographic keys and can accelerate some encryption functions [62], co-processors provide a full computation environment, mirroring the main system, including secure memory, external device connections, etc. [5]. As modern microcontroller cores often integrate hardware accelerators and co-processor functions, many ECU CPUs are available with cryptographic hardware support at similar price points [5, 123].

1.1.3 Current and Upcoming Networks

This section will introduce the interconnections between these ECUs. The most common communication systems used in vehicles are introduced here. While this list is not exhaustive, it covers the largest part of automotive systems in use today.

Controller Area Network (CAN). CAN is the most popular among the automotive bus systems [142]. It has been developed in the 1980s and has been standardized and extended by the International Organization for Standardization (ISO) in ISO 11898 parts 1-3 [57, 58, 59]. With bandwidths of between 125 kBit/s and 1 MBit/s and up to 8 Bytes per data packet, it allows the transmission of small amounts of status and control data in the vehicle. The CAN bus extension ISO 15765-2 defines segmentation of messages, thus allowing to transmit larger messages than 8 Bytes [66]. This is especially useful for diagnosis information. The success of CAN is in not small part founded in its cost, which is significantly lower than for most other systems on the market today.

CAN does not use direct addressing of receivers or identification of senders. Message frames do not include sender or receiver addresses and senders of messages can not be easily identified on the bus. Instead, receivers filter the traffic on the bus for accepted CAN message identifiers (IDs). Thus, the message ID is used as an indirect address.

CAN uses an arbitration mechanism to ensure access to the bus. Arbitration is achieved through the electrical characteristics of the bus, where a logical 0 on the bus is called dominant and overrides a logical 1. This way, the IDs of messages transmitted at the same time will be arbitrated automatically and the lowest message ID will pass.

While CAN is still the main prevailing system, the changing requirements towards more electronic functions, such as advanced driver assistance functions, require an increasing amount of bandwidth that CAN cannot cope with. Here, new communication systems are required.

Local Interconnect Network (LIN). While CAN is already on the lower end in terms of cost, it is still undercut by Local Interconnect Network (LIN) [142]. LIN has been developed by the LIN Consortium in the 1990s as a very inexpensive communication system for simple switching operations or for the transmission of minimal diagnosis information. LIN is currently a standard under development by the ISO as ISO 17987 parts 1-7 [77]. With up to 20 kBit/s, the available bandwidth is significantly lower than CAN. A single message frame holds between 2 and 8 Bytes of data. The bus access is controlled by a single master node, requesting slave nodes to transmit as required. Similar to CAN, identification is achieved via message IDs. LIN is often used to connect a single or a small set of nodes in a subnetwork to a CAN device.

FlexRay. FlexRay is a mixed system of time-triggered and event-triggered communication [142]. It is standardized as ISO 17458 parts 1-5 [69, 70, 71, 72, 73]. With a bandwidth of 10 MBit/s, it offers significantly higher transmission rates than CAN. Its time-triggered part,

the static segment, allows transmissions to be aligned to a common time, synchronized across the whole network. This can significantly reduce the worst-case response time and is critical for systems that need to react fast, e.g. to guarantee the safety of vehicles. Due to the time in the static segment being divided into timeslots, the bus access is already defined at design time. The dynamic segment of FlexRay uses a Flexible Time Division Multiple Access (FTDMA) approach, providing access to the bus in timeslots, which ECUs might extend as required, thus implementing a priority scheme.

Through the timeslots in the static and dynamic segment, an implicit addressing scheme is implemented. As all devices are time synchronized, transmitters and receivers can send and receive in timeslots assigned at design time.

FlexRay has gained popularity quickly, due to it being able to ensure fast response times in the network. In domains such as the drivetrain of the vehicle, where worst-case response times are crucial for safety, FlexRay offers significant advantages. However, due to the precise timing requirements and the complexity of FlexRay, the price per controller is relatively high, compared to CAN. Furthermore, depending on the configuration of static and dynamic segment, as well as the utilization of both segments, the net bandwidth can be significantly lower than 10 MBit/s.

In 2013, FlexRay has been standardized as ISO 17458-1 to -5 and is now under the administration of ISO. FlexRay will be introduced in more detail in Chapter 5, where it is extended to carry large messages, allowing to implement the security measures proposed in this thesis.

Media Oriented Systems Transport (MOST). In the infotainment domain, the Media Oriented Systems Transport (MOST) bus is sometimes used [142]. MOST is specified by the MOST Cooperation specifies bandwidths of 25 and 150 MBit/s over optical cables and 50 and 150 MBit/s over copper cables [109]. As MOST can offer a relatively large bandwidth, it is ideal to be used in the infotainment domain, where larger amounts of data, e.g. audio and video streams, need to be transmitted across the network.

MOST uses a ring architecture and for safety-critical environments can be configured in a double ring structure to achieve redundancy. Each MOST bus contains a timing master, generating frames for timing synchronization for other nodes. The bus access is also controlled by this timing master, leaving space in the frames for asynchronous or synchronous data to be transmitted in so called channels.

The MOST bus is a rather complex communication system, which results in high effort and cost for design and implementation. In recent years, MOST is increasingly under pressure by Automotive Ethernet, which offers similar bandwidth at a lower price point.

While the automotive networks in the previous section have been around for many years, the automotive networking landscape is changing. The rise of ADASs, as well as infotainment functions, which require a high amount of data, led to the development of faster networks. Cameras are becoming more ubiquitous in vehicles and the use of their data in control systems often