Mathematical Biosciences and Engineering

MBE, 20(9): 17057–17095.
DOI: 10.3934/mbe.2023761
Received: 08 April 2023
Revised: 20 May 2023
Accepted: 24 June 2023
Published: 29 August 2023

http://www.aimspress.com/journal/MBE

Research article

State-of-the-art survey of in-vehicle protocols and automotive Ethernet security and vulnerabilities

Aida Ben Chehida Douss[1], Ryma Abassi[1] and Damien Sauveron[2,*]

[1] Innov'COM, Sup'com, University of Carthage, Tunis, Tunisia
[2] MathIS, XLIM, UMR CNRS 7252, University of Limoges, Limoges, France

* **Correspondence:** Email: damien.sauveron@unilim.fr.

Abstract: With the help of advanced technology, the automotive industry is in continuous evolution. Modern vehicles are not only comprised of mechanical components but also contain highly complex electronic devices and connections to the outside world. Today's vehicle usually has between 30 and 70 ECUs (Electronic Control Units), which communicate with each other over standard communication protocols. There are different types of in-vehicle network protocols and bus systems, including the Controlled Area Network (CAN), Local Interconnected Network (LIN), FlexRay, Media Oriented System Transport (MOST), and Automotive Ethernet (AE). Modern cars are also able to communicate with other devices through wired or wireless interfaces such as USB, Bluetooth, Wi-Fi or even 5G. Such interfaces may expose the internal network to the outside world and can be seen as entry points for cyber-attacks. In this paper, the main interest is in the AE network protocol. AE is a special Ethernet design that provides the bandwidth needed for today's applications, and the potential for even greater performance in the future. However, AE is a "best effort" protocol, which cannot be considered reliable. This implies that it is not trustworthy in terms of reliability and timely deliveries. The focus of this paper is to present a state-of-the-art survey of security threats and protection mechanisms relating to AE. After introducing and comparing the different protocols being used in the embedded networks of current vehicles, we analyze the potential threats targeting the AE network and describe how attackers' opportunities can be enhanced by the new communication abilities of modern cars. Finally, we present and compare the AE security solutions currently being devised to address these problems and propose some recommendations and challenges to deal with security issue in AE protocol.

Keywords: modern vehicles; ECU; in-vehicle network communication; automotive Ethernet; AE security

1. Introduction

In the last two decades, there has been a significant evolution of automotive systems. Luxury, comfort and safety are the crucial parameters for moving vehicles. To deal with this evolution, automotive industries have replaced purely mechanical or hydraulic versions of many functionalities with Electronic Control Units (ECUs). ECUs represent the basic units of computation that can be connected to any device in the vehicle, such as the engine, automated operations in the car, and passenger comfort systems [1].

Recently, the number of ECUs in vehicles has increased to enable a variety of requirements and features, including those related to safety, fuel efficiency, convenience, and entertainment. Today, in high-end cars, it is common to have around 70 ECUs exchanging up to 2500 signals between them. Two of the most common ECUs in vehicles are the Engine Control Module (ECM) and the Electronic Brake Control Module (EBCM). The ECM handles processes such as fuel injection and ignition. The EBCM is used in the ABS (Antilock Braking System).

Given that ECUs must communicate with each other, they have to be interconnected in order to exchange signals and to carry out the different functions of a vehicle. In earlier vehicle systems, this type of communication was handled via dedicated wires through point-to-point connections. However, without serial networking, point-to-point cabling results in bulky, expensive, complex, and difficult-to-install wire harnesses. The addition of wiring considerably increases the difficulty of manufacturing a vehicle, its weight and its price, and also weakens its performance and poses problems of interoperability between the different components.

Hence, in-vehicle networking provides a more efficient approach to today's complex in-vehicle communications. This has led to the development of bus-based ECU networks, where communications between multiple ECUs are multiplexed over one or more shared buses. This also gave rise to the need for different communication protocols specifically targeting automotive communication systems. Today, the most commonly used protocols in automotive systems include the Controller Area Network (CAN) with its extended version, CAN FD (Flexible Data), the Local Interconnection Network (LIN), FlexRay, Media Oriented System Transport (MOST) and Automotive Ethernet (AE). In addition, embedded interfaces in modern cars can currently use wireless (e.g., Wi-Fi, Bluetooth) as well as wired (USB) systems to communicate with the outside world.

The primary objective of this paper is to provide a state-of-the-art survey of the main in-vehicle protocols—CAN, LIN, FlexRay, MOST and AE. The advantages and disadvantages of these protocols are presented and a comparison between them is also made based on a range of attributes. After the comparative study was carried out between the different in-vehicular protocols, we concluded that the AE protocol meets future bandwidth requirements for multimedia applications, autonomous driving and safety, such as the Advanced Driver-Assistance System (ADAS). Other features of this protocol include efficient communication, lower latency, scalability, and reduced wiring harnesses.

Hence, in the second part of this paper, we focus on the AE protocol and specifically on its security problems. Security means that a vehicle's systems are free from danger or threat of malicious use, which means that there should be measures to prevent malicious activity [2]. In fact, one of the

significant problems with AE is that it is a "best effort" communication protocol, meaning that it lacks fundamental security features by design and it is not trustworthy in terms of reliability and timely deliveries. This can introduce an array of security threats attempting to penetrate the system. The AE protocol was not designed with security in mind. It has several vulnerabilities, and may be affected by a variety of external attacks that can compromise its confidentiality (i.e., when unauthorized third parties intercept or record data), its integrity (i.e., when a malicious node modifies data on the network), its authenticity (i.e., when a node is not sure that the message is from a legitimate sender) and its availability (i.e., when the system availability is not guaranteed).

Adversaries can use these vulnerabilities to launch sophisticated attacks that may lead to loss of life and damage to property. Thus, vehicle security should be handled carefully and should not compromise the safety of the driver or passengers. Safety and security both need to be prioritized when designing a vehicle network because there are human lives at stake. Therefore, protection against external attacks is very important where automated driving functions are concerned.

Thereby, in the second part of this paper we propose a detailed survey of AE vulnerabilities and security to address the security issues in AE protocols. In this second section, we first present the security requirements and the constraints in vehicle networks based on AE. Then, we discuss the vulnerabilities and the results of attacks on AE. We then identify threats targeting AE-based in-vehicle networks. Finally, we provide a survey of recent work on existing AE security solutions that deal with these attacks. This survey is based on dividing the proposed solutions into three categories: 1) encryption-based solutions, 2) intrusion detection security-based solutions and 3) automotive firewall-based solutions. All security defenses presented in the paper are then summarized in two tables listing their advantages and disadvantages, their security attacks, security requirements and validation strategies.

At the end of the paper, recommendations and challenges are proposed in order to find possible solutions to secure the AE protocol, taking into account the specific constraints of vehicles (including resource-limited nodes and real-time deadlines).

This study aims to answer seven research questions illustrated in Figure 1. Obviously, several other questions can also be posed, but we believe that these are sufficient to expose a vast variety of research in the studied field.

RQ1. What are the basic concepts related to communication protocols currently being implemented in today's vehicles and what are the advantages and disadvantages of each one?

RQ2. What are the attributes that can be considered to compare the various protocols used in in-vehicle network communications?

RQ3. What are the security requirements and constraints in in-vehicle networks, based on the AE protocol?

RQ4. What security vulnerabilities can be found in an AE network, and what are the potential results of attacks on vehicle systems?

RQ5. What are the attacks targeting AE-based in-vehicle networks?

RQ6. What are the existing defenses against such attacks?

RQ7. What are the recommendations and challenges to find possible solutions to secure the AE protocol?

The remaining part of this paper is organized as follows. Section 2 provides background on in-vehicle network protocols including CAN, LIN, FlexRay, MOST and AE and conducts a comparative study of the in-vehicle protocols, based on various attributes. Section 3 deals with the security

requirements and constraints of AE. Section 4 presents the vulnerabilities of AE and the potential results of attacks on vehicle systems. Section 5 discusses the various types of security attacks launched against AE. Section 6 discusses existing solutions for AE security and proposes some recommendations and challenges to deal with security issues in the AE protocol. Finally, Section 7 concludes the paper.

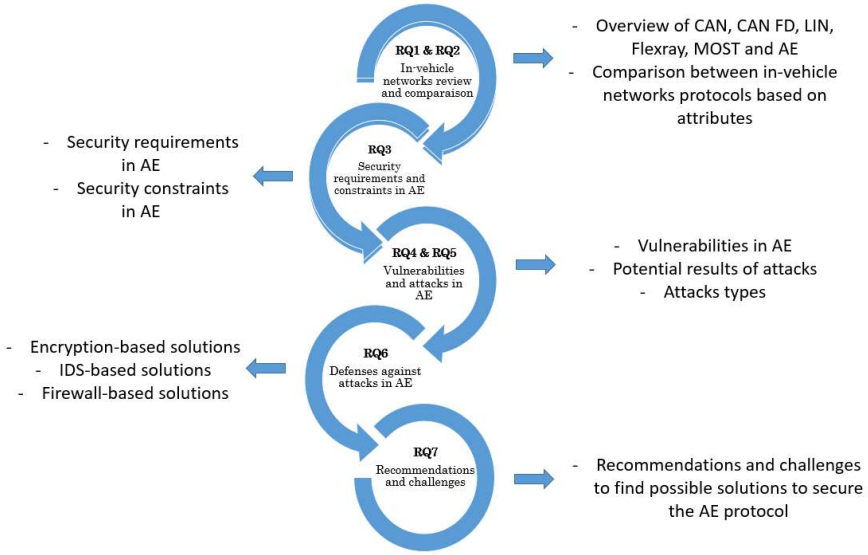


Figure 1. key structure of our review.

2. In-vehicle protocols: Description and comparison

In this section, a description of the main in-vehicle communication networks that have been traditionally used in automobiles is provided, as well as a comparison between these protocols based on a set of attributes. The considered attributes are derived from the Society for Automotive Engineers (SAE) classification [3]. The SAE classifies communication protocols into four categories, ranging from A to D, according to their rates and offered features. Table 1 presents the SAE classification of automotive networks with the four classes: A, B, C and D.

1) Class A, low speed (less than 10kb/s) for convenience features such as body and comfort.
2) Class B, medium speed (between 10 and 125kb/s) for general information transfer, such as emission data, instrumentation.
3) Class C, high speed (between 125kb/s and 1Mb/s) for real-time control such as traction control, and brake by wire.
4) Class D, very high speed (greater than 1Mb/s) for strictly real-time control and multimedia systems.

Table 1. SAE classification protocols.

Class	Rate	Automotive networks
A	<10kb/s	LIN
B	10kb/s → 125kb/s	CAN-B
C	125 kb/s → 1Mb/s	CAN-C
D	>1Mb/s	MOST, FlexRay, AE

Figure 2 shows the progression of in-vehicle networks over time [4].

In the following section, CAN and its extension CAN FD, LIN, MOST, FlexRay and AE protocols are described in detail, with a focus on the advantages and disadvantages of each.

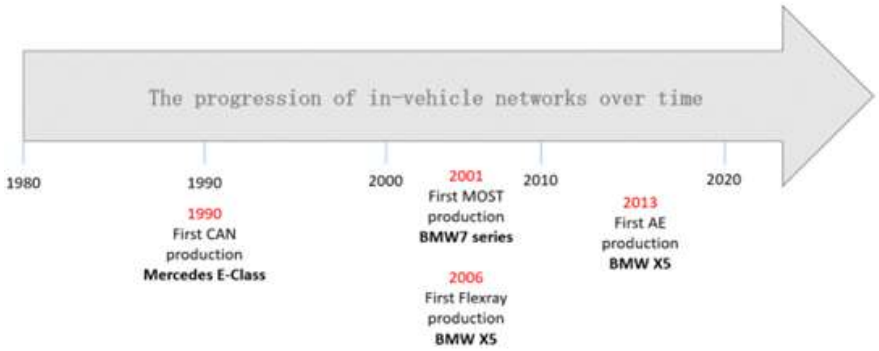


Figure 2. The progression of in-vehicle networks over time [4].

2.1. CAN—An overview

In this section, we will focus first on the classical version of CAN. Then, we will present an extension to the original CAN bus, called CAN FD (CAN Flexible Data).

A CAN (Controller Area Network) is an event-triggered communication protocol and a multi-master serial data bus, developed by Robert Bosch GmbH in the early 1980s [4]. It is a serial bus allowing the ECUs of a vehicle to communicate with each other.

A CAN bus requires only 2 wires to connect to other nodes. These wires are the CAN-H (High-speed CAN) and the CAN-L (Low-speed CAN). The high-speed CAN (0.5–1 Mb/s) is used for driving-related data while the low-speed variants (125–250 kb/s) are mostly used for body and comfort functionality within the passenger compartment.

The CAN protocol is internationally standardized as ISO 11519 [5] for Low-speed CAN and ISO 11898 [6] for High-speed CAN. It is also classified as class B for Low-speed CAN and class C for High-speed CAN in the SAE classification. A CAN bus supports a maximum length of 40 meters and up to 64 nodes.

The CAN protocol standardizes the physical and data link layers, which are the two lowest layers of the Open Systems Interconnection (OSI) communication model. For most systems, higher-layer protocols are needed to enable efficient development and operation.

The CAN protocol therefore belongs to the class of protocols denoted as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), which means that the nodes listen to the network in order to avoid collisions.

An example of a CAN bus with five nodes is provided in Figure 3.

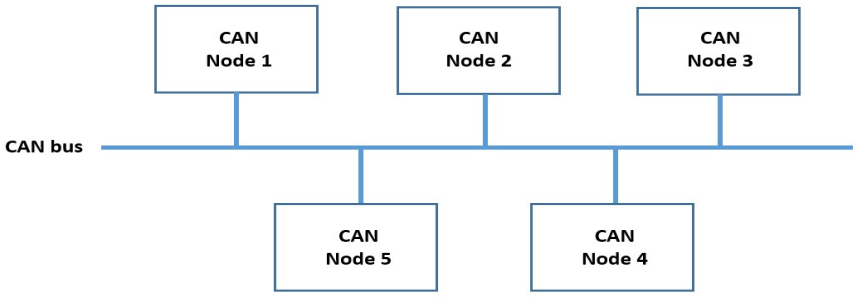


Figure 3. Five nodes connected through a CAN bus.

In a CAN protocol, there are four types of CAN frames [7]: 1) a data frame for data transmission, 2) a remote frame for requesting a data frame to a destination node, 3) an error frame to notify an error in the currently transmitted frame, and 4) an overload frame to delay the next message until the current message has been processed.

Figure 4 shows the basic CAN data frame structure for standard and extended arbitration identifier versions [8].

Figure 4. The structure of a CAN data frame.

Each field contained within the CAN data frame structure is described as follows [8]:

Start of frame (SOF), which specifies the beginning of a CAN message with a dominant bit and notifies all nodes of the start of a CAN message transmission.

Arbitration, which comprises an 11-bit identifier (ID) and can be extended up to 29 bits. Arbitration includes also a remote transmission (RTR) bit. The RTR bit simply distinguishes between data frames (RTR = 0) and remote frames (RTR = 1).

Control, also known as the check field, provides information for the receiver to check whether all intended packets have been received successfully. An IDE field is a dominant single identifier (ID) Extension bit, which means that a standard CAN identifier with no extension is being transmitted. RB0 is a Reserved Bit (for possible use in future standard amendments). The Data Length Code (DLC) is usually set to a value between 0 and 8, indicating a data field length between 0 and 8 bytes.

Data, a field that contains actual information for CAN nodes to perform actions. It can be 0 to 8 bytes long.

CRC, the Cyclic Redundancy Code or safety field, a 15-bit fault detection mechanism, which checks for packets validity. The CRC Del (Delimiter Bit) is always recessive, i.e., 1, following right behind the CRC Segment, allows for CRC processing time.

Acknowledge (ACK). The Acknowledgement Field, also known as the confirmation field, contains a 1-bit Acknowledgement Slot plus the ACK Delimiter Bit (which is always recessive). This field ensures that the receiver nodes receive the CAN frames correctly. Whenever it detects an error during the transmission process, the transmitter will be notified immediately by the receiver to send the data frames again.

End of frame (EOF). This field indicates the end of the CAN frame by a recessive bit's flag.

The CAN protocol provides several benefits over other communication protocols [9] because it:

is **standardized**. ISO has standardized the CAN protocol via ISO-DIS 11898 (for high-speed applications) and ISO-DIS 11519-2 (for low-speed applications). The CAN protocol is also classified by industry organizations, such as the SAE.

significantly reduces the wiring and is one of the lowest-cost networks. When the CAN protocol was first created, its primary goal was to enable faster communication between electronic devices in vehicles while reducing the amount of wiring (and the amount of copper) necessary by using a bus topology.

is **easy to implement**. For decades, CAN has been the primary network of choice for nearly every automobile manufacturer worldwide.

supports multi-master (any node can access the bus) and multicast features (messages can be sent to one/many/all nodes).

is **robust**. The robustness of CAN may be attributed in part to its abundant error checking procedures. Error detection is done in five different ways in CAN (bit monitoring and bit stuffing, as well as frame check, ACK check, and CRC check)

is **very flexible**. The CAN bus protocol is known as a message-based communication protocol. In this type of protocol, nodes on the bus have no identifying information associated with them. As a result, additional elements (ECUs) are easy to integrate or remove, without the need to change other nodes, software, or other network parameters.

provides **high fault tolerance**. The CAN protocol is based on CRC to check the integrity of transmitted data upon receipt at any node and has an automatic retransmission system for lost data.

conduct **fault confinement**. CAN nodes are able to distinguish short disturbances from permanent failures. Defective transmitting nodes are switched off, meaning the node is logically disconnected from the network (bus-off). Hence, faulty nodes do not disturb communication.

enables data **scheduling**. High priority data will be prioritized by ID in the CAN protocol in order to get immediate bus access.

The CAN protocol uses a differential wiring mode, represented by CAN_H and CAN_L, which **enhances immunity to noise and electrical interference**.

All these advantages have driven the use of the CAN bus as a standard for vehicle networking. Nevertheless, it has been recognized that CAN also has many disadvantages:

CAN supports a **maximum length of 40 meters** and **up to 64 nodes** due to electrical loading.

CAN is **low performance.** Throughput needs are constantly increasing worldwide and CAN's maximum 1 Mb/s speed is not able to handle many modern applications.

In order to optimize the signal quality, the ends of the bus line must be « terminated » with **load resistors**. A cable end which is not correctly terminated can make the entire bus inoperative, especially with a high transmission speed.

A CAN **data packet is very small**. At 1 kb to 1 Mb, with a data size of 0-8 bytes, this is sufficient only for certain lightweight applications. In addition, CAN transmits more bits in its header and footer

fields than it does bits of data.

The **cost of software development and maintenance is high**.

Due to the bus-topology, only a single ECU can allocate the CAN to transmit messages. Since nodes may wait for several periods for bus allocation, **inter-node dependency and communication delays arise**. Such an issue is hazardous for future automotive technologies, where real-time constraints are crucial.

CAN is a message-based protocol. As a result, **each device on the network listens to every message transmitted on the bus** and determines what action it needs to take.

Because the receiving node cannot verify the origin of a CAN message, many network traffic injection attacks are possible. As a result, various **network attacks** are easily performed and practically deployable on the CAN bus.

CAN-FD - An extension to the classical CAN

CAN FD is an extension to the original CAN bus protocol that was specified in ISO 11898-1 [10]. The protocol was developed in 2011 and released by Bosch (with industry experts) in 2012. Bosch developed both these technologies (CAN and CAN FD) to support the ever-growing need for data and technology in the automotive industry. Today, CAN FD is used in modern high-performance vehicles.

CAN FD is a data communication protocol compatible with existing CAN networks, allowing the new protocol to function on the same network as the classical CAN. It uses Flexible Data (FD); i.e., it can dynamically switch to different data rates, with larger or smaller message sizes.

The major differences between the classical CAN (Controller Area Network) and CAN FD are:

1) Increased length: The Classical CAN offers 8 data bytes, whereas CAN FD offers flexible data rates, ranging from 0-64 bytes per frame without changing the CAN physical layer. This reduces the protocol overhead and leads to an improved protocol efficiency.
2) Increased speed: A standard CAN network is limited to 1 Mb/s. CAN FD increases the effective data rate to 8 Mb/s (8 times faster than classic CAN). Hence, CAN FD has the flexibility to switch between faster and slower data rates.
3) CAN FD allows up to 30 times more efficient and faster communication between multiple ECUs.
4) Better reliability: One way to ensure reliability is to use a cyclic redundancy check (CRC). Another difference between classical CAN and CAN FD is the decrease in the number of undetected errors as a result of the increased performance of the CRC algorithm.

The frame formats for classical CAN Bus and CAN FD are not very different. However, a few added fields in the CAN FD frame format are not present in the classical CAN bus, as depicted in Figure 5.

5) **RRS**: Remote Request Substitution, which is always a dominant 0, because the remote frames are not supported in CAN FD (in classical CAN, there is RTR for identifying the data frames and remote frames).
6) **FDF**: Flexible Data Rate Format (always a recessive 1) used to indicate flexible data frame format usage.
7) **BRS**: Bit Rate Switch helps to determine the bit rate of a data frame. Dominant 0 signifies that the arbitration rate for the CAN FD data frame is up to 1Mb/s. Recessive 1 signifies a higher/faster arbitration rate for the CAN FD data frame ranging up to 5Mb/s.
8) **ESI**: Error State Indicator. A dominant 0 indicates the error-active mode. A recessive 1 indicates the error-passive mode.

Figure 5. The structure of a CAN FD data frame.

2.2. LIN—An overview

The Local Interconnect Network (LIN) [11] is a broadcast serial bus designed by the LIN Consortium (founded by 5 automakers: BMW, Volkswagen Group, Audi, Volvo Cars and Mercedes-Benz) in 1998. LIN was accepted by the ISO as the ISO 17987 [12] standard, which was released in 2016 and it is classified as class A in the SAE classification.

The number of LIN nodes that are connected through a physical cable is known as the LIN cluster. There are two types of nodes in every cluster: one Master node and up to 16 subsequent Slave nodes (Figure 6). The LIN bus does not need to manage bus collisions because only one message is allowed on the bus at a time.

The LIN bus transmission only requires one wire with a 12V signal at a communication speed of 20 kb/s with 40m bus length.

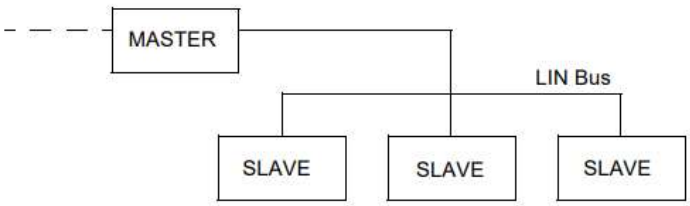


Figure 6. The structure of a LIN bus.

The LIN specification classifies LIN frames into six types [13]: 1) unconditional frames to carry signals (data); 2) event-triggered frames to conserve bus bandwidth by requesting an unconditional frame response from multiple slaves within one frame time slot; 3) sporadic frames to carry signals (data). The header of a sporadic frame should be sent only in its frame slot when the master task knows that a data value (signal) within the frame has been updated; 4) diagnostic frames to carry diagnostic or configuration data; 5) user-defined frames to carry any type of information; and 6) reserved frames, which are reserved for future purposes in the LIN specification.

The LIN protocol consists of frames, where each frame has two parts: the Header and the Response. The header is always from the master and the data response is from a single slave. The frame consists of the following fields (Figure 7).

The header consists of:

1) A ‘Break’ field, which alerts the slaves to an incoming message.
2) A ‘Sync’ field, which allows LIN nodes to measure the time between signal edges and

consequently determine the master node's baud rate.

3) An 'Identifier' field (6 bits), which identifies a slave or slaves and specifies an action to be undertaken.

Upon reception and interpretation of the ID, the polled slave begins the message response, which consists of one to eight bytes of data and an 8-bit checksum.

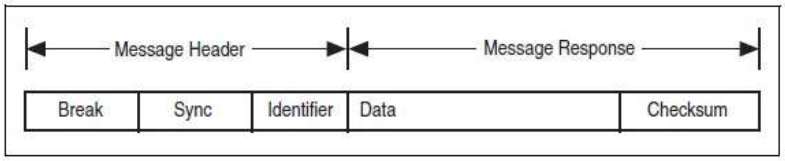


Figure 7. The structure of a LIN data frame.

The LIN bus is an inexpensive serial communications protocol, which effectively supports remote applications within a car's network. LIN's simplicity and low cost make it an attractive option for automotive subsystems where speed and bandwidth are not the primary concerns. Automotive applications where LIN is used today include: comfort cluster, powertrain, engine cluster, car doors, and seats.

The LIN protocol provides several advantages as listed below [14,15]:

1) LIN is **standardized**. ISO has standardized the LIN protocol via ISO-17987.
2) The LIN interface is **simple** and very **low-cost** to implement and use: The single-wire implementation contributes to the low cost and ease of implementation (less harness cabling).
3) The LIN bus is **flexible**. Extension is easy to implement.
4) The 20 kb/s maximum data rate and 40m maximum line length help to **diminish Electromagnetic Interference** (EMI).
5) The LIN bus is **deterministic**. Because the master node delegates pre-determined frame slots on the LIN bus, message frames are predictable, and collisions should not be introduced.
6) The LIN bus is **self-synchronized.** The sync field within a LIN frame allows follower nodes to always stay in sync with the leader, eliminating the need for external oscillators.
7) The LIN bus **guarantees latency times**.

However, LIN protocol has also many disadvantages [14,15] because the LIN bus is:

8) **slow speed**. Hence, it is not ideal for any safety or other important systems inside the vehicle.
9) based on a **master-slave concept**. The master controls all communication on the bus, there is not the possibility for event-driven communication. The other big problem with the master-slave scheme is that if the master is lost, the whole cluster becomes useless, as there is nothing to drive the communication on the bus.
10) low in **fault-tolerance**.

2.3. Flexray—An overview

FlexRay is a deterministic, fault-tolerant, high-speed, real-time system defined by ISO 17458-1 [16] and 17458-5 [17] standards, which is used in a range of cars. FlexRay is a class D network under the SAE classification. It was developed by the FlexRay consortium [17] and was presented in 2006 in the

BMW X5 (see Figure 2).

FlexRay can transmit data over unshielded twisted pair cables, in a flexible configuration, with support topologies including bus, star, and hybrid types. Designers can configure distributed systems by combining two or more of these topologies [18].

In addition to operating as a single-channel system, like CAN and LIN, FlexRay can operate as a dual-channel system with each channel having up to a 10Mb/s bandwidth. The dual-channel option makes data available via a redundant network.

Figure 8 shows a dual channel bus configuration. A node can be connected to both channels A and B (nodes 1, 3 and 5), only to channel A (node 4), or only to channel B (node 2). The FlexRay communication network can also be a single channel. In this case, all nodes are connected to the same bus.

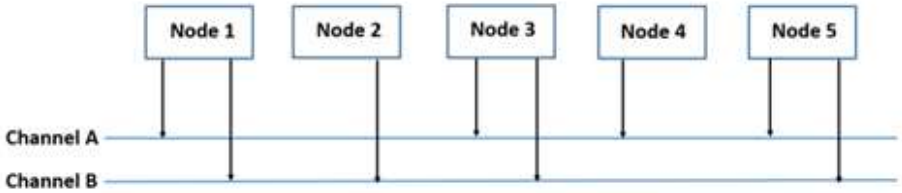


Figure 8. Dual channel FlexRay bus configuration.

There are many possible hybrid topologies. Figure 9 shows an example of one type of hybrid topology. In this figure, some nodes (A, B, C and D) are connected using star topologies. Other nodes (E, F and G) are connected to each other using a bus topology. This bus is also connected to a star.

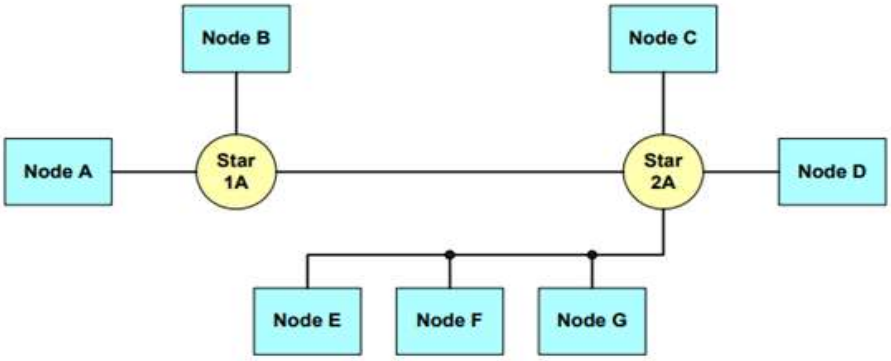


Figure 9. Single channel hybrid configuration.

The FlexRay communication protocol is defined by the frame format, consisting of Header segment, Payload segment and Trailer segment, as shown in Figure 10 [19].

The Header segment is 5 bytes (40 bits) long and includes five fields: 1) The first five bits define the basic features of the frame; 2) the Frame ID defines the slot in which the frame should be transmitted and is used for prioritizing event-triggered frames; 3) the Payload length contains the number of words that are transferred in the frame; 4) the Header CRC is used to detect errors during

the transfer; and 5) the Cycle count contains the value from a counter that advances incrementally each time a communication cycle starts.

The Payload segment contains the actual data transferred by the frame. The length of the FlexRay payload or data frame is up to 127 words (254 bytes), which is over 30 times greater than the CAN.

The trailer segment contains three 8-bit CRCs to detect errors.

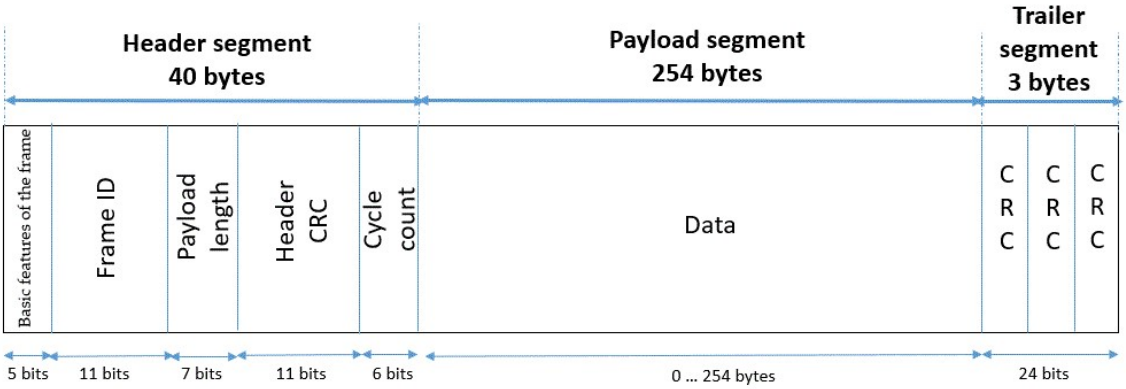


Figure 10. The structure of a FlexRay frame [18].

FlexRay uses a communication cycle consisting of static time-triggered windows using Time Division Multiple Access (TDMA) for each node and a dynamic window using Flexible Time Division Multiple Access (FTDMA) divided into several mini-slots [20].

FlexRay can transmit time-triggered and event-triggered information in the same cycle.

FlexRay also allows both synchronous (real-time) and asynchronous data transfer to meet the demand for various systems in vehicles.

The protocol fulfills the requirements for x-by-wire applications such as drive-by-wire, steer-by-wire, and brake-by-wire.

The development cost of a FlexRay network is very high so it is not commonly used.

The FlexRay protocol presents some advantages [21,22]:

1) FlexRay is **standardized**. ISO has standardized the FlexRay protocol via 17458-1 and 17458-5.

2) FlexRay provides a **higher data rate** than other protocols.

3) FlexRay allows **versatile configurations**, with support for topologies such as bus, star and hybrid types. Designers can configure distributed systems by combining two or more of these topologies.

4) Because FlexRay **provides two independent channels**, it can be used either to double the bandwidth or for redundancy to improve fault-tolerance. This feature increases safety and reliability for control applications in vehicles.

5) FlexRay allows both **synchronous** (real-time) and **asynchronous** data transfer to meet the demand for various systems in vehicles.

6) To meet diverse communication requirements, FlexRay also provides both static and **dynamic communication segments** within each communication cycle.

7) FlexRay is a **deterministic** system, which gives high reliability of communications.

8) FlexRay supports **important payload**.

9) FlexRay is a highly **flexible** protocol. It supports various topologies, time triggering, and event triggering.

10) Within the physical layer, FlexRay provides **fast error detection and signaling**, as well as error containment through an independent Bus Guardian. The Bus Guardian is a mechanism that protects a channel from interference caused by communication that is not aligned with the cluster's communication schedule.

11) FlexRay enables the **possibility of real-time applications**.

There are also some disadvantages [20]:

FlexRay is **expensive** compared with other protocols. However, several aspects of its design are intended to help control costs. For example, it uses inexpensive unshielded twisted pair (UTP) cabling and differential signaling on each pair of wires to reduce the effects of external noise on the network.

FlexRay is **complex to implement**.

FlexRay has a **low operating voltage level** (differential voltage of +2.0 v).

2.4. MOST—An overview

In 1998 the MOST (Media Oriented System Transport) Corporation was founded. Companies including BMW, Audi, Daimler, Harman and microchip technology companies, being core partners, worked to standardize the MOST technology as a global standard for multimedia network. MOST is in the class D network SAE classification and is standardized as ISO 21806 [23].

MOST [24] is a high-speed serial communication system for transmitting audio, video and control data via fiber-optic cables. It can manage up to 64 MOST devices in a ring configuration. Due to its plug and play functionality it is not very difficult to either add or remove a MOST device.

The MOST ring topology is shown in Figure 11 [24].

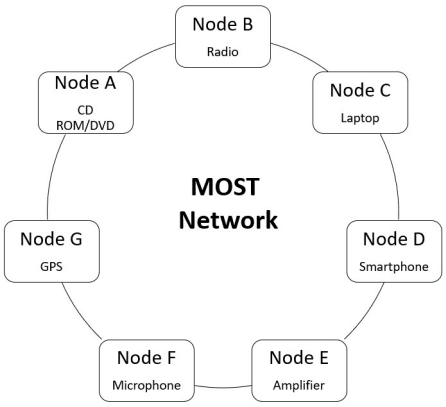


Figure 11. MOST ring topology [24].

MOST supports different physical layers, including fiber optics, Unshielded Twisted Pair (UTP), and Ethernet with three variable baud rate transmission options (25, 50 and 150 Mb/s). MOST25 only uses optical transmission over plastic optical fiber (POF). MOST50 uses electrical transmission over UTP and MOST150 is an enhancement for 100Base-T Ethernet.

The MOST specification not only defines the physical layer and data link layer; it covers all seven layers of the ISO/OSI Reference Model for data communication.

MOST offers, like FlexRay, two freely configurable, static and dynamic time segments for synchronous (up to 24 Mb/s) and asynchronous (up to 14 Mb/s) data transmission.

Unlike most automotive bus systems, MOST messages always include a clear sender and receiver address.

Figure 12 shows the MOST protocol frame structure for the three versions: MOST25, MOST50 and MOST150.

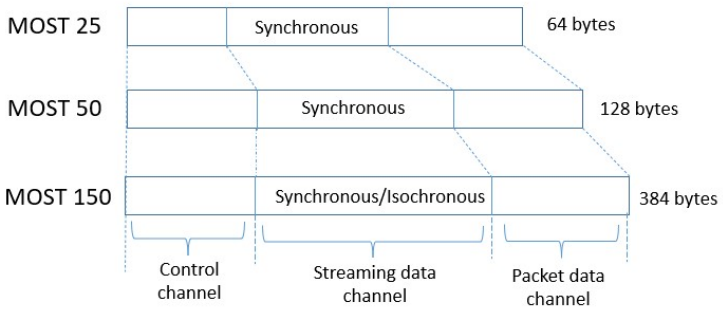


Figure 12. MOST protocol frame structure.

The MOST50 frame is twice as long as MOST25 (64 bytes) and contains 128 bytes. MOST25 contains 4 bytes of control data arranged in two locations of the frame. MOST50 contains 11 control bytes located in the header only at the beginning of the frame. In the framework of the MOST150 protocol, the transceiver speed is three times higher than in the MOST50 version. The frame is also three times longer at 384 bytes.

The MOST system offers some advantages, in that it:

1) is **standardized**. ISO has standardized the MOST protocol via ISO 21806.
2) is **flexible** and a **high-speed** multimedia network topology.
3) supports **variable baud rates** of 25, 50 and 150Mb/s.
4) uses **optical fiber**:
    - high data rate transmission
    - lighter and more flexible compared to shielded electric data lines
    - meets strict Electromagnetic Compatibility (EMC) requirements
    - does not cause any interference radiation
    - insensitive to electromagnetic interference irradiation

The disadvantages of the MOST system are:

1) Very high cost

2) Low fault-tolerance.

2.5. Automotive Ethernet—An overview

AE is a special type of Ethernet that fulfills the requirements for radio frequency resistance and includes protocols to enable real-time communication. Although Ethernet has been around for over 30 years, it has only recently been applied in vehicles.

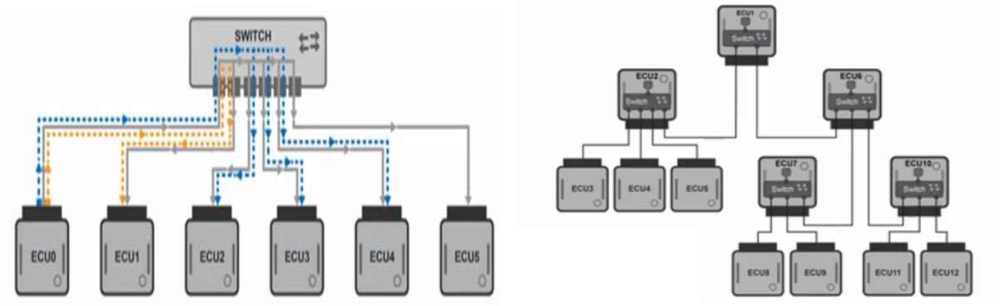
AE is classified as class D in the SAE classification. It was created in 1980 by Robert Metcalfe

and has been continually enhanced since that time [25].

AE gives transmission limits of up to 10 Gb/s and can use coaxial, copper or fiber optic links as the transmission medium. It supports multi-access full duplex transmission and is used for transmitting video information, which requires higher transmission capacity.

Ethernet uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) strategy.

AE with more than two nodes requires a switch to interconnect its end devices (star topology). A star topology network can be easily expanded by adding more nodes, limited by the number of ports on the switch (Figure 13a). The topology can also be extended into a « star of star » or tree topology by using multiple switches connected to each other (Figure 13b).



(a). Ethernet star topology **(b).** Ethernet tree topology

Figure 13. AE topologies.

AE uses several different frame formats. One of the most common formats is Ethernet II, as shown in Figure 14, with a payload of 1500 bytes.

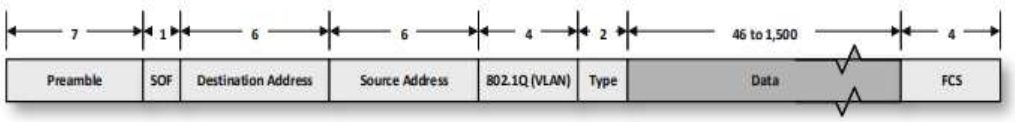


Figure 14. Ethernet II frame structure.

The various protocols used for AE include 100-Base-TX, BroadR-Reach (100Base-T1), Audio Video Bridge-Time-Sensitive Network (AVB/TSN), Diagnostics over Internet Protocol (DoIP), and Scalable Service-Oriented Middleware over Internet Protocol (SOME/IP).

AE offers some advantages:

1) AE is **standardized**. ISO has standardized the AE protocol via ISO 21111.

2) AE provides **higher bandwidth data transmissions** compared with LIN (20 kb/s), CAN (1 Mb/s), FlexRay (10 Mb/s), and MOST (25, 50, or 150 Mb/s) and it has a **low delay in reply**.

3) The most common network standard worldwide is the Transport Control Protocol / Internet Protocol (TCP/IP), which is commonly applied over Ethernet. The fact that Ethernet is the most common network interface protocol means that it has been **well tested** and all of the software interfaces for the upper layers of the Ethernet stack are exactly the same as for standard Ethernet.

4) AE provides **high flexibility**. It has plug and play capabilities, permitting components to be connected and disconnected as needed with automatic detection and configuration.

5) AE **meets the stringent EMC** (Electromagnetic Compatibility) **and EMI** (Electromagnetic Interference) **requirements**, as well as the temperature-grade requirements of the automotive application space.

6) **Each frame has a source and either one or multiple destinations**. In the other automotive network types, the transmitter of the frame is not defined and all nodes always receive all frames and then decide whether or not to use the frame contents.

7) Ethernet technology is **scalable**. It meets the scalability requirement imposed by today's automotive systems (i.e., the number of nodes to interconnect steady increases).

However, AE also has many disadvantages:

1) AE has **low fault-tolerance**.

2) With Ethernet's star topology, every node must be linked back to a single connection point (a switch). This constitutes **more restrictive requirements in term of cabling.**

3) Adding more switches to the network can lead to a tree topology. This creates **more flexibility but added cost**.

4) One of the significant problems with Ethernet is that is it a so-called **'best effort'** communication protocol. This implies that is it not trustworthy with regard to reliability and timely deliveries.

5) Another issue is that Ethernet is **affected by Radio Frequency Interference (RFI)** and thus it **cannot be used in a safety-critical, real-time system**.

As a result of this last disadvantage, many evolutions have been designed to the Ethernet standard, to be able to support real-time features, over the last decade. Among promising solutions able to guarantee hard delays are Time-Triggered Ethernet (TTE) [26], Flexible Time-Triggered Ethernet (FTTE) [27,28], Time-Sensitive Networking (TSN) [29,30] and Wireless Time-Sensitive Networking (WTSN) [31].

TTE [26]: Standardized by the SAE in the SAE6802 standard. It is used for safety-related applications, primarily in transportation industries and industrial automation, and is compatible with IEEE 802.3 Ethernet and integrates transparently with Ethernet network components. It is a scalable networking technology using time scheduling to deliver deterministic real-time communication over Ethernet. It can realize deterministic communication by combining the mature fault-tolerance and real-time mechanisms of the time-triggered technology based on the standard Ethernet. TT Ethernet network devices implement OSI Layer 2 services, and can support various physical layers.

FTTE [27,28]: Efficiently supports hard real-time operation in a seamless, flexible way, over shared or switched Ethernet. Like CAN, it uses a source-addressing scheme for real-time traffic. FTT Ethernet protocol employs an efficient master/multi-slave transmission control technique and combines online scheduling with online admission control, to guarantee continued real-time operation under dynamic communication requirements, together with data structures and mechanisms that are tailored to support dynamic QoS management.

TSN [29,30]: This is a set of Ethernet standards developed by the Institute of Electrical and Electronics Engineers (IEEE) working group IEEE 802.1, with the aim of achieving time guarantees for transmission of data. It is an extension of standard Ethernet that regulates data communication in OSI Layer 2. TSN systems can be supported over multiple variants of the Ethernet physical layer, such

as 10/100Base-T, unshielded twisted pair (e.g., Broadcom's BroadR-Reach transceiver chip) for automotive applications, or even standard 1G/10G Ethernet. It enables Ethernet to become a deterministic networking technology and provides the mechanisms to allow multiple types of traffic to share the same network, providing a basis for convergence. The advantage of TSN is that it provides end-to-end data transmission with extremely low delay and high reliability; therefore, it can be used for new space vehicles. To reach the targeted capabilities, the TSN technology is based on four key features: 1) time synchronization of network elements, 2) controlled and accountable delay (latency), 3) selection of communication paths, path reservations and fault-tolerance and 4) redundancy.

WTSN [31]: Extends the concept of TSN to wireless connectivity to deliver guaranteed wireless services. It offers many benefits, including flexible deployments, reduced maintenance costs, reconfigurability and mobility, which are crucial in manufacturing environments as operations become more mobile. Accelerated adoption of the WTSN infrastructure can be expected over the next decade for industrial robots, automated guided vehicles, and other industrial applications. WTSN will thus be a key technology to make Industry 4.0 implementations more agile and efficient.

Table 2 provides a summary of the advantages and disadvantages of each of the in-vehicle protocols.

2.6. Comparison of in-vehicle protocols

In this section, Figure 15 summarizes the bandwidths of the main automotive networks with their relative cost per node. Then, a comparison study of in-vehicle protocols; i.e., CAN, LIN, FlexRay, MOST and AE is undertaken. Several attributes have been taken into consideration in this comparison study, including the SAE classification, the topology used, the protocols' applications, and the communication architectures. Table 3 summarizes this comparison.

Based on the comparative study summarized in Table 3, it appears that AE has the necessary characteristics to satisfy the increasing automotive industrial demands. In fact, this protocol includes efficient communication, lower latency, scalability and a reduced wiring harness compared with other in-vehicle protocols. Hence, we chose to focus on the AE protocol; specifically, on its security issues. One of the significant problems with AE is that is it a so-called 'best effort' communication protocol, meaning that it lacks fundamental security features by design and is not trustworthy in terms of reliability and timely deliveries.

In the next part of this paper, we will focus on AE security.

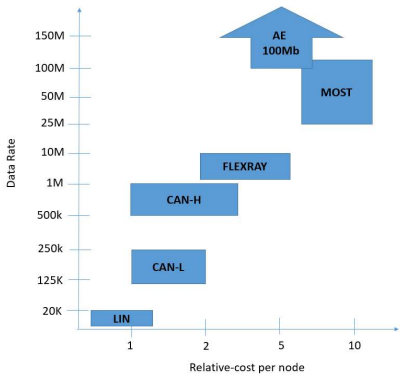


Figure 15. Bandwidth vs. Relative cost per node of the main in-vehicle networks.

Table 2. In-vehicle protocols, advantages and disadvantages.

	Advantages	Disadvantages
CAN	- Standardized - Low use of wiring - Low cost - Easy to implement - Multi master and multicast feature - Robustness - Flexibility - Highly fault-tolerant - Fault confinement - Efficiency	- Limited length (40 meters) - Limited number of nodes (up to 64) - Low performance - Noise and electrical interference - High software and maintenance expenditure - Bus must be terminated with load resistors - Data packet size is too small - Communication delays - Each device listens to every message transmitted on the bus - Security
LIN	- Standardized - Easy to implement and use - Very low cost - Flexibility - Deterministic - Self-synchronization - EMI mitigation - Guaranteed latency times	- Low data rate - Less effective bus access scheme with the master-slave configuration - Low fault-tolerance
FlexRay	- Standardized - High data rate - Supports different topologies - Provides high payload - High reliability of communications - Allows both synchronous and asynchronous data transfer - Allows static and dynamic communication segments within each communication cycle - Provides two independent channels - High fault-tolerance - Provides fast error detection and signaling - Possibility of real time applications	- High cost - High complexity - Low operating voltage levels
MOST	- Standardized - Flexible - High data rate - Supports variable baud rate of range - Meets strict EMC requirements - Uses optical fiber	- Very high cost - Low fault-tolerance
AE	- Standardized - High data transmission rate - High flexibility - Low delay in reply - Low latency - Meets the stringent EMC, EMI and temperature-grade requirements - Packet contains source and destination of the message - Meets the scalability requirement - Well tested	- High cost - Low fault-tolerance - Affected by Radio Frequency Interference (RFI) - Security

Table 3. Comparison of in-vehicle protocols.

Attributes	CAN	LIN	FlexRay	MOST	AE
SAE classification	-CAN-L: Class B -CAN-H: Class C	Class A	Class D	Class D	Class D
Creation	1980, Bosch	1998, LIN Consortium	Foundation FlexRay Consortium	1998, MOST association	1980, Robert Metcalfe
ISO standard	-CAN-L: ISO 11519 -CAN-H: ISO 11898	ISO 17987	ISO 17458-1 and 17458-5	ISO 21806	ISO 21111
Bandwidth (data rate)	CAN-L (125 kb/s-250kb/s) -CAN-H (500kb/s-1Mb/s)	20 kb/s	10 Mb/s	3 levels of bandwidth (MOST25, MOST50 and MOST150 Mb/s)	100Mb/s 1000Mb/s 2.5Gb/s, 5Gb/s, 10Gb/s
Topology	Serial bus	Serial bus	Bus, star, or hybrid	Ring	Point to point, star, or tree
Number of communication nodes (max)	64 nodes	16 nodes	Bus - 22 nodes Star- 22/64 nodes Hybrid- 64 nodes	64 nodes	Limited only by the number of switch ports
Maximum wire length	1Mb/s→40m 250kb/s→250m 125kb/s→500m	40m	22m	1280m	15m per link
Communication architecture	Multi-master bus	Single master bus	Multi-master	Multi-master	Multi-access full duplex transmission
OSI layers	1,2	1,2,7	1,2	1,2,3,4,5,6,7	1,2
Wires (cabling)	Twisted-pair copper wires (2 wires)	1 wire bus	1 or 2 unshielded twisted pair wires (2 or 4 wires)	Fiber optics, Unshielded twisted Pair	Single unshielded twisted pair wires or optical fiber
Bit encoding	NRZ with bit stuffing	NRZ	Copper or optical NRZ with start/stop bits	BiPhase	PAM (Pulse amplitude modulation)
Cabling Impedance	120 ohms	1k ohms	Between 80 and 110 ohms	50 ohms (Varies with physical layer)	100 ± 10 ohms
Operating voltage	3.3v	12v	Differential voltage of +2.0v	3.3v	3.3v
Duration of cycle	~ 240 μs	52μs	1–5 ms	2.7 ms	< 3.2μs ± 0.1μs>
Duplex mode	Half	Half	Full	Data stream	Full

Continued on next page

Attributes	CAN	LIN	FlexRay	MOST	AE
Applications	CAN-L: used for body and comfort functionality within the passenger compartment CAN-H: used for more driving related data	Comfort functions such as mirrors and seats	Hard real-time applications: x-by-wire applications (drive-by-wire, steer-by-wire, and brake-by-wire)	Audio and video streaming capabilities and other applications (Global Positioning System and entertainment systems)	Video information
Fault-tolerant	High	Low	High	Low	Low
Latency	Load dependent	Constant	Constant	Data stream	Losw
Media access	CSMA/CA	Polling	TDMA-FDMA	TDMA-FDMA	CSMA/CD
Cost	Very low	Very low	High	Very high	High
Error mechanisms	- Bit monitoring - Bit stuffing - Frame check - ACK check - CRC	- CRC - Diagnostic - Parity check	- CRC	- CRC - Positive/negative acknowledgements	- CRC
Flexibility	Very flexible	Very flexible	Flexible	Very flexible	Very flexible
Data packet size (Payload)	Max 8 bytes	Max 8 bytes	Up to 254 bytes	- 64 bytes (MOST25) - 128 bytes (MOST50) - 384 bytes (MOST150)	46 to 1500 bytes
Identifier ID	Standard format:11 bits Extended format: 29 bits	6 bits	11 bits	16 bits	No ID
Message transmission type	Asynchronous	Synchronous	Synchronous and Asynchronous	Synchronous, Asynchronous and isochronous (MOST 150)	Asynchronous
Redundant channel	Not supported	Not supported	2 channels	Not supported	Not supported
Frames	Data frame, remote frame, error frame, overload frame	Unconditional, event-triggered, sporadic, user-defined, diagnostic and reserved frames	Data frame	Data frame	Ethernet II frame
Communication (Schedule approach)	Event triggered	Time triggered	Time triggered + event triggered	Time triggered + event triggered	Time triggered + event triggered

3. Automotive Ethernet—Security requirements and constraints

The AE protocol was designed to be fast, lightweight and robust. Indeed, it must have satisfactory performance to meet time constraints in a real-time environment such as in automotive industries. However, by design, the AE protocol has several vulnerabilities. These vulnerabilities allow adversaries to easily gain access to the network by injecting fake messages for different purposes.

To have secure communication, a security solution should meet six criteria: confidentiality, authenticity, integrity, privacy, availability and non-repudiation. These specifications can also be linked to other in-vehicle protocols.

1) Confidentiality: Refers to the security of data exchanged between different nodes. It aims to ensure that sensitive information is kept secret and only authorized people can have access to this information. In AE, a large number of safety-critical messages are transmitted among ECUs. An attacker who is able to eavesdrop data frames can easily obtain important information. Therefore, encrypting every data frame in AE is necessary to provide confidentiality. The plaintext form of the data frame should be available only to a legitimate ECU.

2) Authenticity: Message authentication allows the receiver to confirm that the message was sent by a legitimate sender. Message authentication is crucial for many applications in AE. An attacker can easily inject messages; hence, the receiver needs to be sure that the data used in any decision-making process is from a legitimate sender. Failure to ensure the authentication of a system can cause extreme degradation in a communication network.

3) Integrity: Received information should be exactly the same as sent information, with no alteration. In AE, there is no default verification of frame integrity, meaning that interference, malicious users, or malfunctioning devices may cause frames to change from their intended structure.

4) Privacy: Privacy is another significant challenge in intelligent vehicle systems. It ensures that unauthorized users should not have access to a vehicle or a driver's personal information. Vehicles may have to share information (e.g., information about their geographical locations) with other vehicles or RSUs in the Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) modes of communication. Shared information can be used maliciously to track users. Hence, sensitive information must be protected in an intelligent vehicle.

5) Availability: Availability ensures that system availability is guaranteed despite differing circumstances. AE is a 'best effort' protocol, meaning that the frame will be delivered as soon as possible, and in some scenarios, that may be never.

6) Non-repudiation: Non-repudiation proves that the parties in the communication cannot deny the authenticity of the message that they have sent. In AE, the right to authenticate should not be denied by any broadcasting node.

Table 4. Security requirements of AE network.

Security requirements	Description
Confidentiality	Only authorized people can have access to the information
Authenticity	The receiver can confirm that the message was sent by a legitimate sender
Integrity	The data are not modified when transferring
Privacy	The information of each entity is confidential
Availability	The services provided are operational
Non-repudiation	The actions of entities are undeniable

Table 4 summarizes the security requirements for an AE network.

Moreover, when designing a security protocol for AE, a set of constraints must be considered [32–34]:

1) Resource-limited nodes: Vehicular computational performance is generally limited, compared to the computational performance of a computer. The processing and storage capabilities of the ECU are limited. Because of their computational disadvantage, vehicles are more likely to be hacked than computers. Further, a cybersecurity solution that requires a large number of processing or storage capabilities is not feasible for an in-vehicle system.

2) Limited software update: Most vehicles do not yet have the capability to update their software through Over-the-Air (OTA) updates despite the increased external connectivity of vehicles. Hence, vehicles would not always be able to be protected against the latest cyber-attacks.

3) Real-time deadlines: Several ECUs must perform tasks with fixed real-time constraints, which are often safety-critical. Therefore, any security measure must not impact these tasks.

4) Autonomy: Since the driver must be focused on driving, the vehicle should be as autonomous as possible when protection mechanisms are used.

5) Accessibility: All network components are clustered inside a vehicle and not physically distributed among different geographical locations, as in most common networks. Whoever is in control of the automobile has full access to the whole network and can then attack the whole structure.

6) Safety compliance: Unlike a regular Ethernet network, AE networks are safety compliant. Even if just a few sensors are misinformed or only a small number of illegitimate messages are sent, a vehicle could experience malfunctions that place the lives of drivers, passengers, cyclists and pedestrians at risk. Hence, it is important that applied security does not impact the safety of the vehicle, as safety takes priority over security in the vehicle domain.

7) Part cost: The automotive industry is extremely sensitive to parts costs, making expensive hardware-based security solutions difficult to achieve.

8) Security diversity: A vehicular network is a very heterogeneous system where most devices are unique in the network. Therefore, it can be tricky to apply a mutual security mechanism for the whole system.

9) Increased number of potential attacks: Ethernet is well-known. Therefore, the number of potential attackers (malicious or not) may cause problems in an Ethernet-based network.

10) Gained attention: Because it fulfills necessary requirements in in-vehicle networks, Ethernet will be the key technology for future vehicles. Hence, attacking a vehicle based on AE is attractive to numerous interest groups.

The constraints explained below can be related to all in-vehicle protocols. Only the last two constraints concern the AE protocol.

4. Automotive Ethernet−Vulnerabilities and potential results of car hacking

Before looking at the potential results of car hacking on vehicles, we must first identify the vulnerabilities of the AE protocol. According to Martin Lang [35], the AE protocol has many vulnerabilities, which were inherited from Ethernet.

1) Lack of confidentiality: By default, AE networks lack any encryption method. In fact, unauthorized parties can read all the frames that are sent.

2) Lack of authentication: When a message is being sent in AE, there is no reliability concerning

frame authenticity. This leads to the receiver not knowing whether the frame originated from the source MAC address or whether it was an unknown sender using a spoofed MAC address.

3) Lack of integrity: In AE, there is no default verification of frame integrity. Hence, malicious users may change frames from their intended structure.

4) Lack of availability: AE is a best effort protocol (it inherits this vulnerability from the Ethernet protocol). Hence, information should be accessible by authorized parties.

5) Non repudiation: In AE, it is not possible for an ECU to ensure that no party can deny that it sent or received a given message.

6) Network and system access: To perform an attack, it is necessary to access the network. Network access is achieved in AE in two different ways: 1) by connecting devices to the network, and 2) by gaining control over existing devices.

In [35], this vulnerability has seven different aspects: (a) "unauthorized joins" i.e., with Ethernet, anyone can easily connect to any switch port. This can lead to easy access to the switch and therefore to easy access to the entire network; (b) "unauthorized expansion of the network" i.e., Ethernet allows any user to connect their own switch to the network. This creates a security breach since the network can be easily attacked; (c) "VLAN join" i.e., an attacker can behave like a switch by sniffing VLAN management protocols. It will thus be able to join all the VLANs available on the Ethernet network; (d) "VLAN tagging and hopping" i.e., a malicious node without the necessary access can create and inject Ethernet frames into VLANs using various attacks (e.g., VLAN hopping attack); (e) "Remote access to the LAN" i.e., a malicious node can remotely access the Ethernet device by sending and receiving frames over the remote network based on methods such as social engineering; (f) "Topology and vulnerability discovery" i.e., once a malicious node has access to the Ethernet network, it can easily scan it (passive attack) in order to detect its topology and its various vulnerabilities; and (g) "Switch control" i.e., one of the vulnerabilities of switch is that it ships without credentials. Therefore, if an attacker takes control of the switch, traffic can be handled maliciously.

More details about these seven aspects of vulnerabilities can be found in [35].

Having listed the various vulnerabilities of the AE protocol, we must identify the possible existing motivations for an attacker to launch an attack against the embedded vehicular network [36]. These motivations are based on Studnia et al.'s investigation [37].

The primary motivation for launching an attack against a vehicle is to steal it. An attacker can exploit the vulnerability of the vehicle in order to deactivate the alarm, unlock it or start it remotely. Another motivation for attacking a car can be simply to damage it. This damage can be done for personal satisfaction or for reasons of sabotage. Sabotage aims, for example, to reduce the capabilities of a vehicle by disabling computers, modifying software or even carrying out denial of service (DOS) attacks. The consequences of sabotage can range from minor problems to fatal accidents or to the loss of customer confidence in the car manufacturer.

Another type of unauthorized modification that can be likened to an attack on a vehicle is electronic tuning. The purpose of tuning is to adapt the overall performance of a vehicle in response to the specific needs of the user. For example, the owner of the vehicle can reduce the recorded mileage of his vehicle in order to sell it at a higher price, or he can apply adjustments to the engine to gain more power output.

An attacker can also launch an attack on a given vehicle in order to obtain personal information transmitted in the car concerning the owner (for example, the phone book of the driver, the history of his calls or his GPS history). This attack results in a loss of driver privacy.

5. Automotive Ethernet—Attacks types

Several studies have evaluated threats to in-vehicle networks, especially with regard to the CAN bus. In contrast, there is a lack of research identifying threats to AE-based networks [37,38]. In this section, we identify threats targeting AE-based in-vehicle networks. Hence, we assume that the attacker has access to a target in-vehicle network and can invade the in-vehicle network by plugging an additional device into the targeted network or through the OBD (On Board Diagnostics) port. Such attack scenarios therefore imply a previous security breach, used by the attacker to open the vehicle and plug a device into the network. Once an attacker gains such access, he is capable of 1) passive attacks and 2) active attacks.

5.1. Type 1—Passive attacks

Passive attacks are performed by eavesdropping on the network traffic on ECU. The goals of a passive attack include identifying available services, vehicle status, and the nature of in-vehicle communications. As they do not affect any data within in-vehicle networks, passive attacks are difficult to detect. Fortunately, the impact of passive attacks in AE is very limited compared with the effect on a CAN bus. Indeed, in CAN, all nodes are connected in a single CAN bus and all packets are broadcasted and received by all connected nodes. In contrast, in AE, nodes are connected to a switch, which transmits only broadcast packets to the monitoring node, whereas other packets reach only their designated destinations. However, monitoring of data via passive attacks can be used to perform more complex active attacks.

5.2. Type 2—Active attacks

Active attacks are performed by attackers to disturb the in-vehicle network and to damage its basic functionalities. Nine types of plausible active attacks on AE networks can be distinguished [38].

1) **Frame injection**: Refers to the injection of fake frames into the in-vehicle network through malicious nodes or software programs.

2) **Spoofing attack**: Spoofing refers to a type of attack in which a malicious node will send messages with a fake ID to a corresponding legitimate ECU.

3) **Impersonation attack**: Impersonation is a type of attack in which a malicious node integrates the Ethernet network, impersonates a legitimate ECU, and takes over its functionality to completely disable the operation of the legitimate node. This type of attack can make the vehicle perform in unintended ways, endangering the vehicle and its surroundings.

4) **DoS attack**: DoS is a type of attack in which a specific ECU is flooded with invalid data to hinder normal communications. This attack can cause total loss of service or degradation of service. There is by default no protection against DoS attacks in Ethernet. AE is vulnerable to such an attack because it prioritizes network traffic and ensures a certain quality of service. Thus, the packets injected by an attacker can override other legitimate packets transmitted to the target ECU. This causes packet loss at the network level; other ECUs cannot transmit any packets to the target ECU during the DoS attack.

5) **Address Resolution Protocol (ARP) cache poisoning**: ARP cache poisoning is where the attacking node transmits spoofed ARP requests or responses in order to compromise the ARP cache of

the node under attack. Thus, this attack allows the attacker to read packets intended for another device.

6) **Dynamic Host Configuration Protocol (DHCP) Poisoning**: The attacker detects DHCP broadcast requests for the IP address and replies faster than the legitimate DHCP server. This allows the attacker to choose their own preference of gateway and Dynamic Host Service (DNS) server, making it possible for the attacker to control the victim's traffic.

7) **Man in the middle (MITM) attack**: The previously mentioned ARP and DHCP poisoning attacks can be deployed to redirect traffic to go through the attacker's host. The attack can modify the traffic if it is not protected by an authentication mechanism or just by eavesdropping it.

8) **Replay attack**: Replay attacks are a variant of man-in-the-middle attacks, in which valid transmission data is repeated or delayed. By default, there is no protection against replay attacks in Ethernet. A replay attack uses already sent frames to have an effect when sending them again.

9) **Fuzzing attack**: Fuzzing attacks are used by attackers to inject massive amounts of random data, or fuzz, into a target ECU and observe the outcomes. In this case, the attacker can scan active services, disrupt communications, execute unexpected commands for a designated service or change parameters that are crucial for vehicle operation. This type of attack is easy to perform because it requires less effort to affect the target vehicle. A fuzzing attack conducted by an attacker with prior knowledge may cause a critical accident.

6. Security solutions for Automotive Ethernet networks

The in-vehicle AE is still in its development phase. Hence, it has not been widely adopted by vehicle manufacturers. As a result, little research has been conducted to secure in-vehicle AE networks. Based on the research by Khatri et al. [39], we classified the countermeasures for AE networks into three categories: 1) encryption-based solutions, 2) intrusion detection systems-based solutions, and 3) firewall-based solutions.

6.1. Encryption-based solutions

Encryption is a security mechanism for making data and signals confidential. Confidentiality is an important property to protect a network from spoofing attacks. In some studies, cryptographic algorithms have been used for in-vehicle networks to secure AE protocol [40–44].

Yang et al. proposed a network security architecture system for vehicle Ethernet based on the combination of the advanced security encryption algorithm AES-128 and the key hash authentication algorithm HMAC-SHA1. The AES-128 encryption algorithm and HMAC-SHA1 authentication algorithm are combined and applied to video communications to address the network security problems of vehicle Ethernet [40].

After the encryption and authentication algorithms are added to the video file, normal communication is maintained, and the security and reliability of the information of the Ethernet communication network are further improved in the real-time communication environment. Furthermore, the proposed solution effectively prevents external intrusion and data hopping.

The authors also simulated the encryption authentication and transmission of video information in the vehicle Ethernet system using the CANoe.Ethernet experimental platform. The results demonstrated that there was no stuck and delayed phenomenon in the video transmission and playback process of encryption and authentication processing. Moreover, the AES-128 encryption operation and

the HMAC-SHA1 authentication operation in the video file improved the security of video communication and ensured normal playback of the video.

Another proposition based on encryption and dealing with securing AE in vehicles was introduced by Li et al., who proposed to improve the AES-128 encryption algorithm and the MD5 hash algorithm for AE network security. The combination of an improved AES-128 encryption algorithm and the MD5 hash algorithm have greatly improved security [41].

The authors also used the CANoe.Ethernet platform to verify the performance of the improved AES encryption algorithm and MD5 authentication algorithm. The experimental simulation using CANoe.Ethernet demonstrated that the improved AES-128 encryption algorithm proposed was 15% more efficient than the traditional encryption algorithm, and the improved MD5 hash algorithm was 4 times faster than the traditional authentication algorithm. Thus, the active network security performance of AE was further improved.

Zhu et al. proposed a post-quantum enhanced session key negotiation process for the in-vehicle Ethernet context and evaluated the negotiation overhead for the first time [42]. NTRUEncrypt (Number Theory Research Unit) was chosen as the post-quantum algorithm; with ECDH (Elliptic Curve Diffie–Hellman) and RSA (Rivest–Shamir–Adleman), as comparative algorithms, being implemented in separate processes. The three kinds of algorithms were analyzed in terms of time and memory overheads. The result showed that, besides NTRUEncrypt's particular attribute of resisting quantum computer attacks, the execution speed of session key negotiation using NTRUEncrypt was 66 times faster than ECDH, and 1530 times faster than RSA at the 128-bit security level. The memory occupation of the algorithms was the same order of magnitude as ECDH and RSA. As the transport layer security (TLS) protocol can fulfill most performance requirements of the automotive industry, post-quantum enhanced session key negotiation will probably be widely used for in-vehicle Ethernet communication.

Another scheme dealing with designing an authentication and secure communication scheme for the in-vehicle Ethernet context was proposed by Ma et al. [43]. In this paper, the authors designed an efficient secure scheme, including an authentication scheme using the Scalable Service-Oriented Middleware over IP (SOME/IP) protocol. The authentication scheme uses a safety and security controller as the Key Management Center (KMC) of the in-vehicle networks and is implemented based on symmetric cryptography. The session keys are regularly updated and distributed to the domain controllers to resist brute force attacks on the key. The secure communication scheme, based on the authentication scheme, modifies the payload field of the SOME/IP data frame to provide integrity and confidentiality protection for the communication process without changing the basic structure of the data frame. A security analysis based on Proverif (an automated cryptographic protocol verifier) was also carried out in this case to evaluate the security of the proposed authentication scheme. The results showed that a designed authentication scheme can provide mutual identity authentication for both communicating parties and can ensure the confidentiality of the temporary session key. An informal security analysis showed that a designed authentication and secure communication scheme can resist common malicious attacks, such as eavesdropping, replay, man-in-the-middle, and impersonal attacks. The performance experiments based on embedded devices showed that the additional overhead introduced by the secure scheme was very limited.

Silva et al. [44] proposed using cryptographic schemes for Ethernet-based layer-2 communication to provide authenticated encryption to safety-critical automotive control data. The proposed architecture combines an Advanced Encryption Standard (AES) with a Hash-based Message

Authentication Code (HMAC) in an automotive safety-critical system, and is based on a non-deterministic isolated network for control data. Confidentiality, integrity and authenticity are provided by combining AES with HMAC.

Different measures, such as limiting the Maximum Transfer Unit (MTU) of the message, were applied along with a careful examination of the worst case delays to guarantee determinism. An experimental evaluation was performed using low-cost hardware, supported by statistical tests on the results, which showed that, despite the introduced cryptographic overhead, latency requirements were comfortably met for this type of communication.

6.2. IDS security—Based solutions

Intrusion detection (IDS) technology, as a network security enhancement method, is low-cost and offers convenient deployment [45]. A few studies have been conducted recently on the use of intrusion detection technologies for in-vehicle CAN networks [46–48], but many studies have proposed IDS for AE [49–56].

Jeon et al. first examined Ethernet technology in vehicle networks and its security threats [49]. Then, they investigated the traffic characteristics of the AE and a testbed was built. They then designed an anomaly detection system using machine learning techniques for Ethernet-based in-vehicle networks. Based on the user's behavior, normal data are learned through machine learning and the learned model is applied to the vehicle. The proposed anomaly detection system consists of four stages, including 1) a data analysis module that collects and analyzes traffic data; 2) a feature extraction module extracting significant features via data analysis; 3) a feature processing module preprocessing data to facilitate data analysis from packets reached in real time; and 4) the actual anomaly detection module, which makes normal/abnormal judgments for preprocessed data using a model built using a machine learning algorithm. A number of features were also discussed and extracted from Ethernet traffic: Duration, Protocol Type, Src_bytes, Dst_bytes, Count, and Srv_count.

Grimm et al. presented an extension of their hybrid anomaly detection system for ECUs to improve the security and safety of vehicles using AE [50,51]. The proposition was designed for in-vehicle ECUs but it can be adapted to other application domains. The proposed system combined specification and machine learning-based anomaly detection methods. The features, necessary for the machine learning part, are selected to enable the detection of anomalies in real time and with respect to an automotive-specific communication scheme. The authors considered irregularities in message occurrences as the criterion for anomalous behavior in AE networks. Message irregularities may occur if the sending ECU cannot transmit data to a given receiver, or if the message consists of data from unknown protocols. Finally, detection performance and the applicability of different machine learning algorithms were evaluated in a simulation environment using CANoe.Ethernet based on synthetic and well-defined anomalies.

Another intrusion detection method was proposed by Jeong et al. using a deep learning model for detecting Audio-Video Transport Protocol (AVTP) stream injection attacks in AE-based networks [52]. The proposed intrusion detection model is based on a feature generation process and a Convolutional Neural Network (CNN). The feature generator was designed by considering the observed characteristics of real AVTP traffic. The detection model distinguishes whether AVTP packets transmitted over AE are benign or injected on a packet-by-packet basis. To evaluate the proposed intrusion detection system, the authors used real AVTP packets captured from a BroadR-Reach

network (an Ethernet physical layer standard designed for automotive connectivity applications). The evaluation results showed that the IDS correctly classified almost all AVTP packets, with very few false negatives. Furthermore, the experimental results showed that the model exhibited outstanding performance: the F1-score and recall were greater than 0.9704 and 0.9949, respectively. In terms of the inference time per input and the generation intervals of AVTP traffic, the CNN model can readily be employed for real-time detection.

Alkhatib et al. proposed in [53] to detect anomalies in AE-based in-vehicle networks using anomaly-based approaches. They described an anomaly detection-based IDS Convolutional AutoEncoder (CAE) for offline detection of anomalies in the Audio Video Transport Protocol (AVTP). The CAE consists of an encoder and a decoder with convolutional neural network CNN structures that are asymmetrical; i.e., the encoder architecture is not similar to the decoder architecture. Anomalies in AVTP packet streams, which may lead to critical interruption of media streams, are therefore detected by measuring the reconstruction error for each sliding window of AVTP packets transformed to images as an anomaly score. In order to detect intrusions on AVTP, the authors used the "AE Intrusion Dataset" [55] containing benign and malicious AVTP packets captured via their physical AE testbed. The CAE model proposed in this work was trained using Keras, a deep learning framework. The proposed approach was also compared with other state-of-the art traditional anomaly detection and signature-based models in machine learning. The numerical results showed that the proposed model outperformed the other methods and excelled in predicting unknown in-vehicle intrusions, with 0.94 accuracy. The model also has a low level of false alarm and missed detection rates for AVTP replay attacks.

Another scheme studied an intrusion detection technology based on vehicle-embedded Ethernet, relying on rule matching. This scheme was proposed by Zihan et al. The authors designed a binary intrusion detection rule format for storing rules, which can be converted to and from the current mainstream Snort/Suricata rule set. This binary rule format stores the same rules but requires smaller ROM resources [54]. The authors also designed an intrusion detection system, ETH-IDS, which fully complies with AUTOSAR specifications [55], and applied the binary rules to ETH-IDS. A multi-level comprehensive evaluation model was also proposed in the embedded environment to undertake quantitative evaluation of the intrusion detection systems. Related experiments were carried out in an automotive-embedded environment, and the performance advantages of ETH-IDS were verified compared with Suricata [52]. The validity of the evaluation model was also verified. ETH-IDS has advantages over Suricata in terms of CPU utilization and memory usage. The experiment also verified the effectiveness of the evaluation model.

Qui et al. looked at vehicle Ethernet as the research object, constructed machine learning samples for a neural network, applied self-coding network technology combined with the original characteristics to the network intrusion detection algorithm, and studied a self-learning vehicle Ethernet intrusion detection algorithm [56]. The algorithm generated in this study can be used for vehicle terminals with Ethernet communication functions, and can effectively resist 34 kinds of network attacks in four categories. This method effectively improves the network security defense capability of vehicle Ethernet, provides technical support for the network security of intelligent vehicles, and can be widely used in mass-produced intelligent vehicles with Ethernet.

Lindwall et al. [57] proposed a concept for a host-based IDS relying on two different detection methods. They suggested a combination of specification-based detection, focusing on message sequencing and allowed elapsed time between a request and its respective response, and anomaly-based detection, evaluating the frequency, payload length and timeout for request-response pairs. To

evaluate the proposed IDS, the authors implemented and tested its capabilities by launching a series of attacks against the IDS and measuring its detection performance. Five different attack scenarios were executed and the binary classification metrics and classification speed were measured. The evaluation showed that the proposed IDS successfully detected malicious events such as delay, packet injection, exhaustion and two different flooding attacks. The IDS with this hybrid approach is a promising security solution for in-vehicle AE networks.

6.3. Automotive firewall—Based solutions

In recent years, firewall technology has also become interesting for the automotive domain. A set of solutions has been proposed in the literature to secure AE based on firewalls [58–60].

Holle and Shukla presented a gateway firewall design for an AE switch, including packet and application filters [58]. They provide an overview of future Ethernet architectures and illustrate the necessity for integrating firewalls as a core component of embedded IT security in vehicles. Hence, according to the authors, a vehicle platform's firewall, which has excellent performance and is future-proofed, should ideally meet the following four requirements: 1) equally usable for multiple Ethernet-based E/E (Electrical/Electronic) architectures; 2) blocks illegitimate communication both with the entire electrical system and with individual ECUs; 3) forces Ethernet communication to flow in accordance with rules in the fashion of a router; and 4) provides a high level of configurability, and is cheap and easy to implement with a wide range of automotive E/E (Electrical/Electronic) architectures.

Another firewall approach was proposed by Pesé et al. [59], addressing the introduction of automotive firewalls into the next-generation domain architecture with a focus on partitioning of its features into hardware (HW) and software (SW) with respect to specific automotive requirements. An HW firewall is a simple packet filter, taking the Ethernet packets from the attacker as an input and immediately dropping packets that are not allowed on the internal network according to a pre-defined whitelist policy. Only packets matching one of the rules are allowed to be shunted to the Domain Controller (DC), which runs an SW firewall for further classification. The SW firewall is a stateful packet filter, which analyzes packets that have passed the first firewall.

A conceptual model for a firewall and Intrusion Detection and Prevention System (IDPS) to secure in-vehicle communication over AE was proposed by Yilmaz [60]. The proposed model consists of a firewall that acts as an initial filter of incoming packets, where mostly header data is checked for malicious signatures. The firewall is equipped with a ternary content-addressable memory (TCAM) and a rate meter for fast initial processing of packets at wire-speed and it allows or drops the packet according to its header information. The action on a packet is decided based on stateless and stateful filtering, both of which use the pre-set rules of the firewall. A packet passing these checks arrives at the IDPS unit for an extended search in the packet payload, where appropriate measures are taken to filter or block malicious packets. The inspection is carried out by checking the payload against a large number of stored signatures. Both subsystems interchange information about detected threats and create dynamic rules as they adapt to newly discovered threats.

Table 5. Security solutions for AE networks.

Method	Key idea	Advantages	Disadvantages
Yang et al. [40]	The AES-128 encryption and HMAC-SHA1 authentication algorithms are combined and applied to video communication to address the network security problems of vehicle Ethernet.	The combination of AES-128 and HMAC-SHA1 operations on the video file improve the security of video communication information while ensuring normal playback of the video.	The proposed approach was only validated on a video communication.
Li et al. [41]	An improved AES encryption algorithm and MD5 hash algorithm for the network security of AE.	The proposition improves the real-time, security and reliability performance of the AE.	The CANoe platform was only used to verify the performance of the improved AES and MD5 algorithms. No attacks were simulated to verify the security of the approach.
Zhu et al. [42]	A NTRUEncrypt enhanced session key negotiation for the in-vehicle Ethernet context.	The NTRUEncrypt's resists quantum computer attacks. The execution speed of session key negotiation using NTRUEncrypt is 66 times faster than ECDH, and 1530 times faster than RSA.	No simulation was done to verify and validate the security of the approach.
Ma et al. [43]	An authentication and secure communication scheme for the in-vehicle Ethernet context based on the SOME/IP protocol.	- Authentication scheme can provide mutual identity authentication for communicating parties and ensure the confidentiality of the issued temporary session key. - Scheme can resist the common malicious attacks conjointly. - Performance experiments showed that the additional overhead introduced by the secure scheme is very limited.	No simulation was done to verify and validate the security of the approach.
Silva Junior et al. [44]	A proposition and an evaluation of an architecture that deploys link-layer security combining AES with HMAC code on an automotive safety-critical system.	Confidentiality, integrity and authenticity are provided by combining AES with HMAC.	Authors mainly focused on the end-to-end delay but they did not evaluate the security aspect of the proposed architecture. - No simulation was done to validate the security of the approach.
Jeon et al. [49]	An anomaly detection system using machine learning techniques for Ethernet based in-vehicle network.	A good study on traffic characteristics for anomaly detection in Ethernet-based in-vehicle networks.	- More investigations should be made for machine learning used in this context because of the resource constraints in vehicles.

Continued on next page

Method	Key idea	Advantages	Disadvantages
Grimm et al. [50]	A system combining specification and machine learning-based anomaly detection methods to improve the security and safety of vehicles using AE.	The use of specification-based anomaly detection and machine learning algorithms sequentially within the embedded software of an ECU showed good detection rates and few false alarms.	High memory consumption and computing power are required for the training process.
Jeong et al. [52]	An intrusion detection method for detecting audio-video transport protocol stream injection attacks in AE-based networks.	- The model achieves a high detection performance and remarkably high recall for real stream AVTPDUs captured from the BroadR-Reach-based testbed. - The model is suitable for real-time detection in autonomous vehicles.	Experiments were limited to stream AVTPDUs. It would also be interesting to consider other AVB-related protocols and automotive diagnostics communication protocols.
Alkhatib et al. [53]	An anomaly detection-based Convolutional Autoencoder for offline detection of anomalies on the Audio Video Transport Protocol in the in-vehicle network AE.	- The model outperforms the other methods and excels at predicting unknown in-vehicle intrusions, with 0.94 accuracy. - It also has a low level of false alarm and missed detection rates for AVTP replay attack.	The authors only used datasets representing AVTP replay attacks. They should have explored more datasets covering other attacks.
Zihan et al. [54]	An in-vehicle embedded Ethernet intrusion detection technology based on rule matching, and a new binary rule format for intrusion detection.	The method combines the characteristics of the embedded environment to propose an evaluation system for embedded intrusion detection systems.	The rule set of the IDS can be increased and optimized to fit to more complex network environments.
Qui et al. [56]	An analysis of real-time heterogeneous data generated by key parts of intelligent vehicles in on-board Ethernet through machine learning to detect attack behavior compared to normal behavior.	The IDS method developed in this study can be widely used in key parts with Ethernet communication, and can provide strong support for active vehicle defense, and improve the information security defense level of intelligent vehicles.	No simulations have been conducted to validate the approach.
Lindwall and Ovhagen [57]	A concept for an IDS over AE relying on specification-based and anomaly-based detection methods.	The IDS evaluation showed that the two detection methods can operate together as a hybrid method to detect previously known and unseen threats.	The attack scenario was limited to a few types of attacks. It is important to test additional attack scenarios.
Holle and Shukla [58]	A gateway firewall design for an AE switch including packet and application filter	The need to integrate firewalls for AE is well presented in this paper.	The authors have not explained the precise details of the packet and application filtering techniques used by the firewall.

Continued on next page

Method	Key idea	Advantages	Disadvantages
Pesé et al. [59]	A combined HW and SW firewall. The HW firewall handles a simple and generic ruleset to restrict the traffic between domains. The SW firewall focuses on a small filter ruleset for custom-made rules and stateful packet inspection.	- The work prevents DoS attacks such as SYN flooding attacks. - Firewall was implemented on an Infineon AURIX TriCore and Altera Cyclone V FPGA to analyze automotive requirements metrics such as latency, jitter, CPU load and memory consumption.	The traffic model has been simplified for analysis purposes and the deployed hardware was reduced to just run the firewall.
Yilmaz [60]	A conceptual model for a Firewall and Intrusion Detection and Prevention System that aims to act as a solid security mechanism for the vehicular communication network.	The model is based on the combination of a firewall and an IDPS system.	TCAM has its limitations (e.g., high resource consumption, expensive solution).

Table 6. Security attacks, requirements and validation strategies for AE security solutions.

Security solutions	Category	Main mitigated attacks	Authentication	Confidentiality	Integrity	Privacy	Availability	Intrusion detection	Validation Strategies
Yang et al. [40]	Encryption	External intrusion and data hopping attack	Yes	Yes	Yes	Yes	Yes	N.C	Simulation (CANoe simulator)
Li et al. [41]	Encryption	N.A	Yes	Yes	Yes	Yes	No	N.C	Simulation (CANoe simulator)
Zhu et al. [42]	Encryption	Quantum computer attacks	Yes	Yes	No	No	No	N.C	Experiments (AURIX TC397 microcontroller)
Ma et al. [43]	Encryption	Eavesdropping, replay, man-in-the-middle, impersonal and masquerading attacks	Yes	Yes	Yes	Yes	No	N.C	Proverif, informal security analysis and performances evaluation
Silva Junior et al. [44]	Encryption	N.A	Yes	Yes	Yes	No	No	N.C	Experimental prototype consisting of an Ethernet switch and two nodes
Jeon et al. [49]	IDS	N.A	No	No	No	No	No	Yes	N.A
Grimm et al. [50]	IDS	Replay attack	No	No	No	No	No	Yes	Simulation (CANoe simulator)

Continued on next page

Security solutions	Category	Main mitigated attacks	Authentication	Confidentiality	Integrity	Privacy	Availability	Intrusion detection	Validation Strategies
Jeong et al. [52]	IDS	AVTP stream injection attacks	N.C	N.C	N.C	N.C	N.C	Yes	Experiments (Python library Keras and Google Colaboratory on an NVIDIA Tesla P100 GPU)
Alkhatib et al. [53]	IDS	AVTP replay attack	N.C	N.C	N.C	N.C	N.C	Yes	Numerical experiments
Zihan et al. [54]	IDS	Port scanning, denial of service, strong attacks and weak passwords, protocol anomaly	N.C	N.C	N.C	N.C	N.C	Yes	Experiments (Raspberry Pi 4 platform)
Qui et al. [56]	IDS	Partial tampering and injection attacks	N.C	N.C	Yes	N.C	N.C	Yes	N.A
Lindwall and Ovhagen [57]	IDS	Delay, packet injection, exhaustion and flooding attacks	No	No	No	No	Yes	Yes	Implementation (tests)
Holle and Shukla [58]	Firewall	N.A	N.C	N.C	N.C	N.C	N.C	N.C	N.A
Pesé et al. [59]	Firewall	DoS attacks	No	No	No	No	Yes	N.C	Firewall implementation an Infineon AURIX TriCore and Altera Cyclone V FPGA
Yilmaz [60]	Firewall+ IDPS	Eavesdropping on the communication, replay, intercept, modify, and inject messages	No	Yes	Yes	Yes	Yes	Yes	Switch under test approach equipped with a packet processor that consists of a packet parser, a search engine, a metering engine, and an action resolution unit. Hardware acceleration is realized by the use of the TCAM technology

N.A = Not Available. N.C = Not Concerned.

Table 5 summarizes the key ideas, as well as the advantages and disadvantages of each one of the security solutions for AE networks.

Table 6 lists all security defenses presented in this section and associates them with security attacks, security requirements and validation strategies.

Based on comparative studies described in Tables 5 and 6, we suggest the following challenges and recommendations for securing AE-based in-vehicle networks in the three categories (encryption-based solutions, IDS-based solutions, and firewall-based solutions) proposed in Section 7:

Category 1: Encryption-based security solutions:

Most of the security solutions proposed in this category ([40–42] and [44]) do not really take into consideration the limited resources of ECUs in terms of memory and computing power. However, cryptographic and hashing algorithms require high performance and storage capabilities to be able to perform the various cryptographic calculations.

The only solution that takes into account the limited resources of computers is the work proposed by Zhu et al. [42]. This paper proposed a post-quantum enhanced session key negotiation process for the AE context and based the solution on a post-quantum NTRUEncrypt algorithm. The results demonstrated that NTRUEncrypt is faster than ECDH and RSA in term of calculation speed. However, the memory occupation of the algorithm was the same order of magnitude as ECDH and RSA.

One of the strategies that we suggest in the category of encryption-based solutions is the use of lightweight security solutions such those applied for smart cards and in the IoT (Internet of Things), which are resource-constrained networks like vehicular networks.

Lightweight security mechanisms are based on Elliptic Curve Cryptography (ECC). ECC, an alternative technique to RSA, is a powerful cryptography approach for encrypting data. It generates security between key pairs for public key encryption by using the mathematics of elliptic curves. ECC has several advantages over RSA. In effect, ECC features smaller ciphertexts, keys, and signatures than RSA. It also generates keys and signatures faster than RSA and its decryption and encryption speeds are moderately fast. Hence, ECC offers high security with faster, shorter keys compared to RSA. ECC may therefore be a good alternative to secure AE protocol in resource-constrained in-vehicle networks [61].

Category 2: IDS-based security solutions:

With regard to the proposed works related to IDS, we note that almost all the IDS-based security solutions presented above apply AI (Artificial Intelligence) based on machine learning or deep learning methods to secure the AE protocol in a vehicular context.

However, these methods require a lot of computation time and memory storage for the learning phase. Therefore, these methods allow the detection of an attack (without stopping it) and can provide prompt feedback to administrators; but on the other hand, they do not take into account the limited resources of the ECUs in vehicles.

Another problem with AI algorithms is that they have various problems that are not explainable. Unexplainability in AI can be defined as the impossibility of providing an explanation for certain decisions made by an intelligent system that is both 100% accurate and comprehensible. In recent years, we have seen a rising quest for AI explainability in machine learning, and deep-learning.

One of the challenges of AI-based IDS in vehicle security is to have explainable AI solutions (XAI solutions). Such solutions can avoid incomprehensible decision problems in an intelligent system, which could be fatal for the driver and the vehicle's passengers.

Category 3: Firewall-based security solutions:

Based on the studies presented in Section 7.3, implementing firewalls in the vehicle context can effectively prevent unauthorized ECUs from sending safety-critical messages. However, in order to ensure that the firewalls work correctly and do not produce errors, it is absolutely necessary to carry out modeling.

One of the challenges of firewall-based security solutions is to make formal specification and validation. Modeling provides proof that the firewall, and more precisely its ACL (access-control list), are well configured and do not present any errors or conflicts.

7. Conclusions

With the ongoing development of automotive electronics and technologies, vehicles have provided drivers more driving entertainment by becoming connected to smart phones, Bluetooth and the Internet.

Vehicle bus systems and multimedia systems are connected to each other through in-vehicle network protocols such as CAN, LIN, FlexRay, MOST, and AE. However, due to the increasing amount of information exchanged within in-vehicle networks, and because these protocols lack fundamental security features by design, this can introduce an array of security issues, which cannot be ignored. These issues could seriously affect vehicle drivers' safety, personal privacy, and even endanger public safety. In this paper, we firstly investigated the in-vehicle network protocols CAN, LIN, FlexRay, MOST and AE, then we focused on AE network security and vulnerabilities. Hence, in the first part of this study, in-vehicle network protocols were studied and a comparison between them was made based on a set of attributes. Then, we focused on AE network protocols. An AE system can satisfy the network broadband needs of smart vehicles because it offers efficient communication, lower latency, scalability and a reduced wiring harness. However, the AE network is not secure, and adversaries can mount attacks to completely take over a vehicle's controls. However, some researchers have proposed security solutions to deal with AE security problems.

Therefore, in the second part of this review, the in-vehicle network communication protocol AE was studied and its vulnerabilities and potential attacks were described. Then, we presented a set of solutions aiming at enforcing security in AE networks. These security solutions were divided into three categories: encryption-based solutions, intrusion detection systems-based solutions, and firewall-based solutions. Each scheme has its advantages and limitations in terms of detection coverage, learning time, computational complexity, detection times, and robustness. We then provided some recommendations and challenges for comprehensive solutions to secure AE protocol.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Tariq, S. Y. Lee, H. K. Kim, S. S. Woo, CAN-ADF: The controller area network attack detection framework, *Comput. Secur.*, **94** (2020), 101857. https://doi.org/10.1016/j.cose.2020.101857
2. C. Corbett, E. Schoch, F. Kargl, P. Felix, Automotive Ethernet: Security opportunity or challenge?, 2016 (2016), 45–54.
3. S. Jadhav, D. Kshirsagar, A survey on security in automotive networks, in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, (2018), 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697772
4. C. M. Kozierok, C. Correa, R. B. Boatright, J. Quesnelle, Automotive ethernet: The definitive guide, *Intrepid Control Syst.*, **2014** (2014).
5. I. ISO, *Road Vehicles—Low-Speed Serial Data Communication—Part 1: General and Definitions*, International Organization for Standardization, 1994.
6. I. ISO, *Road Vehicles–Controller Area Network (CAN)*, International Organization for Standardization, 2015.
7. H. Zhang, X. Meng, X. Zhang, Z. Liu, CANsec A practical in-vehicle controller area network security evaluation tool, *Sensors*, **20** (2020), 4900. https://doi.org/10.3390/s20174900
8. S. F. Lokman, A. T. Othman, M. H. Abu-Bakar, Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review, *J. Wireless Comput. Network*, **184** (2019). https://doi.org/10.1186/s13638-019-1484-3
9. Total Phase, What is CAN bus protocol?, 2019. Available from: https://www.totalphase.com/blog/2019/08/5-advantages-of-can-bus-protocol/.
10. ISO, *Road vehicles—Controller Area Network (CAN)—Part 1: Data link layer and physical signaling*, International Organization for Standardization, 2015.
11. H. Qiang, L. Feng, Review of secure communication approaches for in-vehicle network, *Int. J. Autom. Technol*., **19** (2018), 879–894. https://doi.org/10.1007/s12239−018−0085−1
12. ISO, *Road Vehicles Local Interconnect Network (LIN)*, International Organization for Standardization, 2019.
13. J. M. Ernst, A. J. Michaels, LIN bus security analysis, in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, (2018), 2085–2090. https://doi.org/10.1109/IECON.2018.8592744
14. J. Huang, M. Zhao, Y. Zhou, C. Xing, In-vehicle networking: Protocols, challenges, and solutions, *IEEE Network*, **33** (2018), 92–98. https://doi.org/10.1109/MNET.2018.1700448
15. E. Hacket, LIN protocol and physical layer requirements, 2018. Available from: https://www.ti.com/lit/an/slla383a/slla383a.pdf?ts=1668071732342&ref_url=https%253A%252F%252Fwww.bing.com%252F.
16. ISO, *Road Vehicles—FlexRay Communications System, Part 1: General Information and Use Case Definition*, International Organization for Standardization, 2013.
17. ISO, *Road Vehicles—FlexRay Communications System—Part 5: Electrical Physical Layer Conformance Test Specification*, International Organization for Standardization, 2013.
18. Y. Wang, H. Liu, B. Huang, N. Zhang, Y. Wu, Reliability-based parameter design for FlexRay network in vehicles, *Adv. Mech. Eng.*, **2019** (2019). https://doi.org/10.1177/1687814019839905
19. J. Pradeep, S. R. Sebasteen, R. Dineshkrishn, Comparison of CAN and flexray protocol for automotive application, *Int. J. Pure Appl. Math.*, **119** (2018), 1739–1745.

20. Q. Hu, F. Luo, Review of secure communication approaches for in-vehicle network, *Int. J. Autom. Technol.*, **19** (2018), 879–894. https://doi.org/10.1007/s12239−018−0085−1
21. L. Pike, J. Sharp, M. Tullsen, P. Hickey, J. Bielman, Secure automotive software: The next steps, *IEEE Software*, **34** (2017), 49–55. https://doi.org/10.1109/MS.2017.78
22. M. Meier, D. Reinhardt, S. Wendzel, *Sicherheit 2016, Lecture Notes in Informatics (LNI)*, Gesellschaft fur Informatik, Bonn, 2016.
23. ISO, *Road vehicles—Media Oriented Systems Transport (MOST)—Part 1: General information and definitions*, International Organization for Standardization, 2020.
24. H. Rajeshwari, K. Siddarth, K. S. Gurumurthy, The impact of network topologies on the performance of the in-vehicle network, *Int. J. Comput. Theory Eng.*, **5** (2013). https://doi.org/10.7763/IJCTE. 2013.V5.719
25. K. van Cleave, A survey of automotive ethernet technologies and protocols, in *CSE570S: A Survey of Automotive Ethernet Technologies and Protocols*, (2019).
26. V. Eramo, F. G. Lavacca, M. Listanti, S. Caporossi, Definition and performance evaluation of an Advanced Avionic TTEthernet Architecture for the support of Launcher Networks, *IEEE Aerosp. Electron. Syst. Magaz.*, **33** (2018). https://doi.org/10.1109/MAES.2018.170161
27. V. Eramo, F. G. Lavacca, F. Valente, A. Pisculli, S. Caporossi, Simulation and experimental evaluation of a flexible time triggered ethernet architecture applied in satellite Nano/Micro Launchers, *Aerospace*, **5** (2018). https://doi.org/10.3390/aerospace5030084
28. V. Eramo, T. Fiori, F. G. Lavacca, F. Valente, A. Baiocchi, S. Ciabuschi, et al., A max plus algebra based scheduling algorithm for supporting time triggered services in ethernet networks, *Comput. Commun.*, **198** (2023). https://doi.org/10.1016/j.comcom.2022.11.014
29. L. Lo Bello, W. Steiner, A perspective on IEEE time-sensitive networking for industrial communication and automation systems, *Proceed. IEEE*, **107** (2019), 1094–1120. https://doi.org/10.1109/JPROC.2019.2905334
30. J. Sanchez-Garrido, B. Aparicio, J. G. Ramírez, R. Rodriguez, M. Melara, L. Cercós, et al., Implementation of a time-sensitive networking (TSN) Ethernet bus for microlaunchers, *IEEE Trans. Aerosp. Electron. Syst.*, **57** (2021), 2743–2758. https://doi.org/10.1109/TAES.2021.3061806
31. K. A. Mahin, M. Raheeb, O. Seijo, I. Val, H. P. Bernhard, When IEEE 802.11 and 5G meet time-sensitive networking, *IEEE Open J. Ind. Electron. Soc.*, **3** (2022), 14–36. https://doi.org/10.1109/OJIES.2021.3135524
32. M. Scalas, G. Giacinto, Automotive cybersecurity: Foundations for next-generation vehicles, in *2nd International Conference on new Trends in Computing Sciences (ICTCS)*, (2019), 1–6. https://doi.org/10.1109/ICTCS.2019.8923077
33. L. Pike, J. Sharp, M. Tullsen, P. Hickey, J. Bielman, Secure automotive software: The next steps, *IEEE Software*, **34** (2017), 49–55. https://doi.org/10.1109/MS.2017.78
34. M. Meier, D. Reinhardt, S. Wendzel, 45, in *Sicherheit 2016, Lecture Notes in Informatics (LNI)*, Gesellschaft fur Informatik, Bonn, 2016.
35. M. Lang, *Secure Automotive Ethernet Balancing Security and Safety in Time-Sensitive Systems,* Master thesis, Blekinge Institute of Technology, 2019.
36. T. Kiravuo, M. Sarela, J. Manner, A survey of Ethernet LAN security, *IEEE Commun. Surv. Tutorials*, **15** (2013), 1477–1491. https://doi.org/10.1109/SURV.2012.121112.00190

37. I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, Y. Laarouchi, A survey of security threats and protection mechanisms in embedded automotive networks, in *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop*, (2013).
38. P. Kleberger, T. Olovsson, E. Jonsson, Security aspects of the in-vehicle network in the connected car, in *2011 IEEE Intelligent Vehicles Symposium (IV)*, (2011), 528–533.
39. N. Khatri, R. Shrestha, S. Y. Nam, Security issues with in-vehicle networks, and enhanced countermeasures based on blockchain, *Electronics*, **10** (2021), 893. https://doi.org/10.3390/electronics10080893
40. H. Yang, M. Liu, Y. Xu, Y. Wu, Y. Xu, Research of automotive ethernet security based on encryption and authentication method, *Int. J. Comput. Theory Eng.*, **11** (2019), 1–5. https://doi.org/10.7763/IJCTE.2019.V11.1230
41. J. M. Li, F. Shuo, Y. Wu,Y. Xu, High-efficiency encryption and authentication network security for automotive Ethernet, *Int. J. Model. Optim.*, **12** (2022), 36–42. https://doi.org/10.7763/IJMO.2022.V12.797
42. Y. Zhu, Y. Liu, M. Wu, J. Li, S. Liu, J. Zhao, Research on secure communication on in-vehicle Ethernet based on post-quantum algorithm NTRUEncrypt, *Electronics*, **11** (2022), 856. https://doi.org/10.3390/electronics11060856
43. B. Ma, S. Yang, Z. Zuo, B. Zou, Y. Cao, X. Yan, et al., An authentication and secure communication scheme for in-vehicle networks based on SOME/IP, *Sensors*, **22** (2022), 647. https://doi.org/10.3390/s22020647
44. E. Silva, P. F. Araujo-Filho, D. R. Campelo, Experimental evaluation of cryptography overhead in automotive safety-critical communication, in *IEEE 87th Vehicular Technology Conference (VTC Spring)*, (2018), 1–5. https://doi.org/10.1109/VTCSpring.2018.8417610
45. W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, et al., A survey of intrusion detection for in-vehicle networks, *IEEE Trans. Intell. Trans. Syst.*, **21** (2020), 919–933. https://doi.org/10.1109/TITS.2019.2908074
46. M. Markovitz, A. Wool, Field classification, modeling and anomaly detection in unknown CAN bus networks, *Veh. Commun.*, **9** (2017), 43–52. https://doi.org/10.1016/j.vehcom.2017.02.005
47. M. Bresch, N. Salman, *Design and Implementation of an Intrusion Detection System (IDS) for in-Vehicle Networks*, Master Thesis, Chalmers University of Technology and University of Gothenburg, 2017.
48. M. J. Kang, J. W. Kang, A novel intrusion detection method using deep neural network for in-vehicle network security, *PloS One*, **11** (2016), e0155781. https://doi.org/10.1371/journal.pone.0155781
49. B. Jeon, H. Ju, B. Jung, K. Kim, D. Lee, A study on traffic characteristics for anomaly detection of Ethernet-based IVN, in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, (2019), 951–953. https://doi.org/10.1109/ICTC46691.2019.8940022
50. D. Grimm, M. Weber, E. Sax, An extended hybrid anomaly detection system for automotive electronic control units communicating via Ethernet-efficient and effective analysis using a specification- and machine learning-based approach, in *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport (VEHITS)*, (2018), 462–473. https://doi.org/10.5220/0006779204620473

51. M. Weber, S. Klug, E. Sax, B. Zimmer, Embedded hybrid anomaly detection for automotive CAN communication, in *9th European congress on embedded real time software and systems (ERTS 2018)*, (2018).
52. S. Jeong, B. Jeonb, B. Chungb, H. Kang Kim, Convolutional neural network-based intrusion detection system for AVTP streams in automotive Ethernet-based Networks, *Veh. Commun.*, **29** (2021), 100338. https://doi.org/10.1016/j.vehcom.2021.100338
53. N. Alkhatib, M. Mushtaq, H. Ghauch, J. L. Danger, AVTPnet: Convolutional autoencoder for AVTP anomaly detection in automotive ethernet networks, preprint, arXiv: 2202.00045.
54. Z. Zihan, C. Lirong, Z. Haitao, Z. Fan, Research on intrusion detection technology based on embedded Ethernet, in *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 587–600. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674069
55. Autosar, Standards. Available from: https://www.autosar.org/standards/.
56. B. Qiu, K. Chen, K. He, X. Fang, Research on vehicle network intrusion detection technology based on dynamic data set, in *IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, **2021** (2021), 386–390. https://doi.org/10.1109/ICFTIC54370.2021.9647072
57. H. Lindwall, P. Ovhagen, *A Concept for an Intrusion Detection System over Automotive Ethernet*, Master thesis, Lund University, 2020.
58. J. Holle, S. Shukla, Gatekeeper for in-vehicle network communication, *ATZelektronik Worldwide*, **13** (2018), 40–43.
59. M. Pesé, K. Schmidt, H. Zweck, *Hardware/software co-design of an automotive embedded firewall*, 2017-01-1659, SAE Technical Paper.
60. E. Yilmaz, *Firewall and IDPS Concept for Automotive Ethernet*, Master thesis, Uppsala Universitet, 2019.
61. Y. Genç, M. Habek, N. Aytaş, A. Akkoç, E. Afacan, E. Yazgan, Elliptic curve cryptography for security in connected vehicles, in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, (2022), 1–4. https://doi.org/10.1109/SIU55565.2022.9864762

AIMS Press