sensors

MDPI

Review

Evaluation of CAN Bus Security Challenges †

Mehmet Bozdal *, Mohammad Samie, Sohaib Aslam and Ian Jennions

IVHM Centre, Cranfield University, Cranfield MK43 0AL, UK; m.samie@cranfield.ac.uk (M.S.); S.Aslam@cranfield.ac.uk (S.A.); i.jennions@cranfield.ac.uk (I.J.)

* Correspondence: mehmet.bozdal@cranfield.ac.uk

† This paper is an extension version of the conference paper: Bozdal, M.; Samie, M.; Jennions, I. A Survey on CAN Bus Protocol: Attacks, Challenges, and Potential Solutions. In Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 16–17 August 2018.

Received: 31 January 2020; Accepted: 19 April 2020; Published: 21 April 2020



Abstract: The automobile industry no longer relies on pure mechanical systems; instead, it benefits from many smart features based on advanced embedded electronics. Although the rise in electronics and connectivity has improved comfort, functionality, and safe driving, it has also created new attack surfaces to penetrate the in-vehicle communication network, which was initially designed as a close loop system. For such applications, the Controller Area Network (CAN) is the most-widely used communication protocol, which still suffers from various security issues because of the lack of encryption and authentication. As a result, any malicious/hijacked node can cause catastrophic accidents and financial loss. This paper analyses the CAN bus comprehensively to provide an outlook on security concerns. It also presents the security vulnerabilities of the CAN and a state-of-the-art attack surface with cases of implemented attack scenarios and goes through different solutions that assist in attack prevention, mainly based on an intrusion detection system (IDS).

Keywords: CAN network; CAN security; ECU; in-vehicle communication

1. Introduction

The vehicle industry has evolved drastically over the last couple of decades into extensive automation of cars with a mesh of sensors and computational systems. These sensors are controlled by embedded electronic control units (ECUs), designed for the optimal management of a wide array of functions ranging from engine control to Anti-lock Braking (ABS) and Advanced Driver-Assistance Systems (ADAS), respectively. According to [1,2], a modern automobile is fitted with more than a hundred ECUs, and this number is envisaged to increase in the future. These ECUs are distributed all around the vehicle and communicate with each other via in-vehicle communication networks such as a Controller Area Network (CAN). Being the most common in-vehicle communication protocol for vehicle applications, CAN offers advantages such as cost-effective wiring, immunity to electrical interference, self-diagnosing, and error correction.

However, despite these functional benefits, the rising inter- and intra-vehicle communications render CAN vulnerable to cyber-attacks. The existing built-in security features of the CAN bus are primarily designed for ensuring reliable communication, and not for cybersecurity; therefore, it cannot prevent the network from cyberattacks. As a result, far-reaching implications of cyberattacks on CAN are anticipated. For instance, the safety of the driver and passengers can be jeopardised by the attack on airbag [3] or ABS systems. Eventually, it may affect the reputation of the car manufacturer with substantial financial implications like recalls [4]. Tampering of ECUs (e.g., used-cars' odometers [5]) is yet another example that may result in dire consequences for consumers and manufacturers.

Equally alarming is the lack of encryption in CAN, which has a strong bearing on individual data privacy. By design, CAN is a broadcast network that allows nodes to capture messages going through the network. As the broadcasted data is not encrypted, an adversary can acquire the desired data. This may lead to an invasion of privacy, mainly when modern cars are capable of acquiring the driver's personal information.

According to the 2019 industry survey [6], safety and security are the highest short-term and mid-term challenges for the automotive industry. Therefore, extensive studies have been carried out to find possible solutions [7,8] to the vulnerabilities of CAN. Some of these studies have performed successful experimental attacks on passenger cars [9–14] and heavy-duty vehicles [15,16]. At the same time, researchers have also proposed preventative methods for such known attacks. These include network segmentation, encryption, authentication, and intrusion detection systems (IDSs).

In light of the above, this paper provides a comprehensive literature review with the following main contributions:

a. Identification of the state-of-the-art and the most-probable security challenges associated with modern vehicles, covering a number of implemented physical and remote access attacks.
b. Highlighting the attack surfaces of modern vehicles with a critique on possible future attacks.
c. An in-depth analysis of the current research on CAN security issues to facilitate their effective and optimal mitigation.

Accordingly, the rest of the paper is organised as follows. Section 2 provides a background study on the CAN, followed by Section 3, which presents a detailed vulnerability assessment of the CAN. In Section 4, we give an in-depth account of the attacks that have been implemented on the CAN network, followed by a critique of the existing and proposed solutions in Section 5. Finally, the research challenges are discussed in Section 6, with our conclusions given in Section 7.

2. Overview of the Controller Area Network (CAN)

The CAN bus is a multi-master broadcast communication protocol developed by Robert Bosch GmbH in the early 1980s. A traditional CAN interface can provide up to 1 Mbps [17]. In 2012, Bosch released the CAN FD (flexible data-rate), which can achieve 5 Mbps in practice and has a 64-byte payload compared to 8 bytes in the classical CAN [18]. CAN FD is backward compatible and can coexist with classical CAN nodes. Classical CAN and CAN FD are both standardised under ISO 11898-1:2015.

The single two-wire bus architecture of CAN, as shown in Figure 1, reduces cabling. The distributed architecture of the network provides easy-maintenance and decreases the overall system cost. Moreover, the protocol uses differential wiring mode, represented by CAN_H and CAN_L, which enhances the immunity to noise and electrical interference. From a logic point of view, signals have two states (voltage levels): A dominant logic '0' and a recessive logic '1', meaning that the bus signal remains '0', the dominant logic, as long as one of the nodes releases logic '0' to the bus.

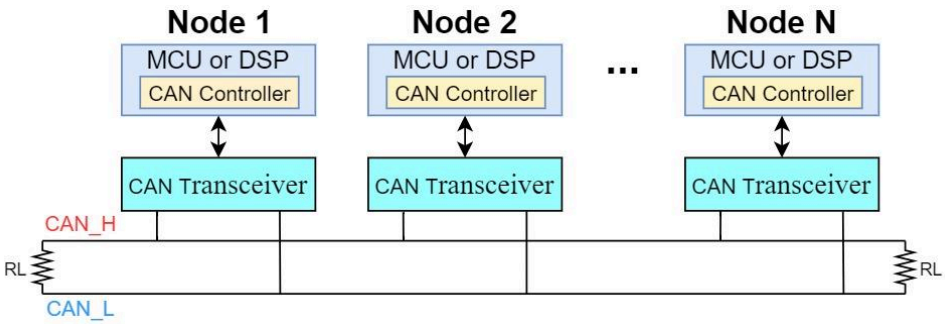


Figure 1. An example of a single two-wire Controller Area Network (CAN).

The CAN protocol has message-based communication provided via frames, as shown in Figure 2. Each frame has a message identifier field, data field, cyclic redundancy checksum (CRC), and some control bits. Every node listens to each frame and processes the relevant ones based on the message identifier field, which is also used for the arbitration.

SOF	Message Identifier	RTR	IDE	r0	DLC	Data	CRC	ACK	EOF	IFS
1-bit Dominant	11-bit or 28-bit (Arbitration Field)	1-bit	1-bit	1-bit	4-bit	0 to 8 Bytes	15-bit Checksum 1 - bit Delimeter	1 - bit Acknowledgement 1 - bit Delimeter	7-bit Recessive	-

Figure 2. Classical CAN frame structure.

Reliable Communication in CAN

The CAN protocol has a set of built-in features that provide robust communication. If two nodes start transmitting at the same time, the non-destructive arbitration mechanism resolves the conflict by allowing the highest priority node to continue the transmission without any interruption (e.g., Node 1 wins arbitration in Figure 3, without any disruption, as the dominant bit overrides the recessive one). Another feature is carrier sense multiple access with collision avoidance (CSMA/CA), which rules that the nodes have to wait for a certain amount of inactivity before the transmission. This assists in sensing if the bus is idle for ensuring that a collision will not occur.

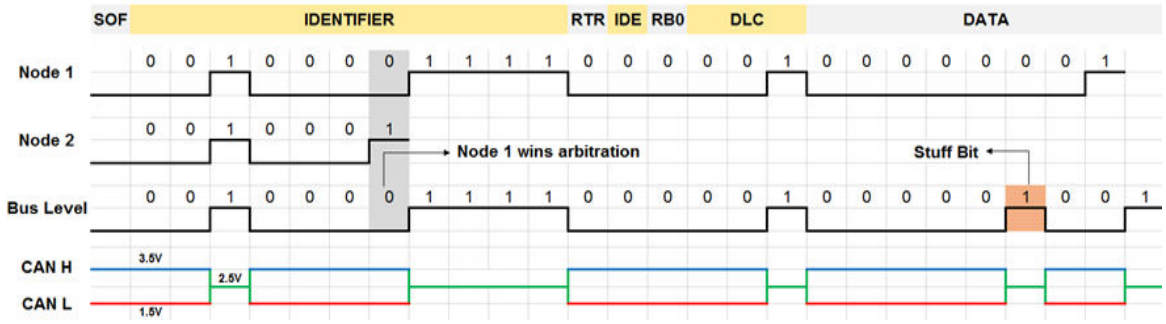


Figure 3. Signalling in CAN; Node 1 wins arbitration without any disruption.

The CAN bus has some bit-level and message-level error checking mechanisms. In the bit-level, the transmitter node monitors the bus. An error arises if there is a difference between the transmitted bit and the one observed on the bus. On the other hand, the message-level CAN-bus error check mechanism includes frame check over acknowledgment (ACK), cyclic redundancy checksum (CRC), and end of frame (EOF) fields. After the transmission of a frame, the transmitter node writes a recessive bit to the ACK field. If a node receives a message correctly, it overwrites the ACK field with a dominant bit; otherwise, the ACK field stays recessive, which indicates a transmission error. There is up to a 21-bit CRC field in a CAN frame for data integrity. If any node calculates a different CRC than the transmitter node, an error flag will be sent. The CRC delimiter, ACK delimiter, and EOF bits have fixed values and must always be recessive. During the frame form check, if these bits are dominant, an error is generated.

CAN also prevents the physical errors by disabling the faulty nodes from the bus traffic with an error confinement mechanism (ECM), as shown in Figure 4. The ECM is facilitated in each node using two error counters known as the received error counter (REC) and transmitted error counter (TEC). The TEC increases by eight if an error occurs during the transmission, and REC increases by one if the error comes during the reception. Every successful transmission or reception of a frame decreases the responsible counter by one. The counters' default values are zero, and nodes start at the error active state. A node will enter the error passive state if the value of the node's counter exceeds 127. In the error passive state, the node can only write recessive error flags, which will not affect the bus traffic.

The node turns to the bus off state if the TEC counter exceeds 255, meaning that the affected node will no longer take part in the bus traffic.

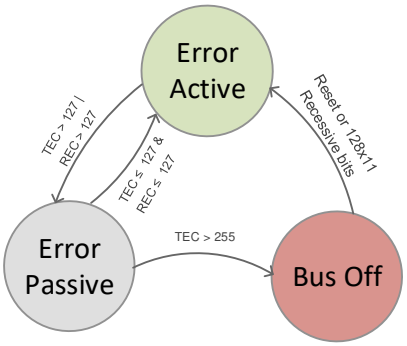


Figure 4. The state diagram of the error confinement mechanism (ECM) in the CAN bus.

3. Vulnerability Assessment of the CAN Protocol

It is essential to have a vulnerability assessment of a network to highlight security problems. Therefore, the vulnerability assessment of the CAN protocol can be carried out based on confidentiality, integrity, and availability.

Confidentiality means providing the data only to authorised people. However, the CAN protocol does not have inherent cryptographic methods to ensure confidentiality. This allows an intruder to access sensitive user data and cause an invasion of privacy.

Integrity is the accuracy, completeness, and validity of the data. The CAN bus has a CRC for verification of integrity against the transmission errors, but it cannot prevent data injected by malicious parties, which breaks the integrity. The protocol does not have a comprehensive integrity check and fails to sustain integrity.

Availability means that authorised users can use the system at all times. Given the nature of priority-based messaging, if a message with the highest priority is transmitted/inserted, the network will be inaccessible by the lower priority nodes, and availability is violated.

The CAN bus failed to pass all three essential security criteria. Thus, it is a clear indication that the CAN protocol does not have any security measurements against the attacks.

4. Automotive Attack Surface and Existent Attacks

In the 1950s, automotive electronics cost only 1% of the total car cost, while it is currently 35% and is expected to rise to 50% in 2030 [19]. Although the rise in electronics has improved comfort, functionality, and driving safety, it has created new attack surfaces, as shown in Figure 5. The protocol itself is defenceless to attacks; therefore, any exploit in the current/future telematics unit or infotainment system can disrupt the network, as summarised in Table 1.

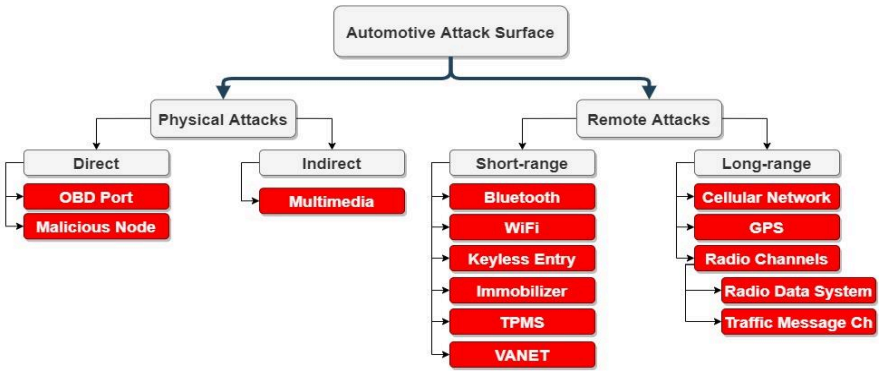


Figure 5. The automotive attack surface.

Table 1. Summary of the Controlled Area Network (CAN) bus attacks.

Reference	DoS	Modification [1]	Access Type	Notes/Root Cause
[11]	Y	N	OBD II	Does not require full CAN messages
[20]	N	Y	OBD II, CD, Bluetooth, GSM	Systematical experimental attacks. Indirect access via the car service computer
[21]	N	Y	In-direct OBD II	Attack via a smartphone app
[22]	Y	Y	Multiple remote sources	Remote attack analysis of 21 commercial cars
[13]	N	Y	Wi-Fi, GSM	Access CAN network via a browser exploit
[16]	Y	N	OBD II, compromised ECU	SAE J1939 data-link layer exploits
[23]	N	Y	Wi-Fi, GSM	Ransomware attack over the air
[24]	N	Y	TPMS	Remotely sending false TPMS data

[1] The modification includes replay, impersonation, and bogus information attacks.

The first CAN bus attack was performed on the power window by Hoppe and Dittman in 2007 [7,25]. Since then, numerous attacks have been performed. These attacks can be categorised as physical access attacks, where the attacker should access the vehicle physically, or remote attacks, which are implemented via wireless communication interfaces. Although attacks in the literature are mainly physical access ones, some experts have argued that physical access to the CAN network is not practical [26]. Therefore, current research is mainly focusing on remote access attacks.

4.1. Physical Access Attacks

Physical access attacks require direct or indirect access to the CAN bus network. Direct access can be obtained by the On-Board Diagnostic (OBD) port or a malicious node. The OBD port is the primary attack surface; hence, it has access to all of the nodes, even though network segmentation is used.

Koscher et al. [10] manipulated the CAN and controlled various modules including essential brake control and engine control modules through the On-Board Diagnostics II (OBD-II) port. They released the brake and prevented its activation while the car was running 40 mph by the continuous fuzzing method. The attack also includes the manipulation of the instrument cluster with false data, changing engine parameters, and disabling the engine.

Due to the CAN architecture, any malicious node can listen or send a message to disrupt the network. The attacks implemented through the OBD port can be replicated using a malicious node. Palanca et al. [11] applied a selective denial-of-service (DoS) attack on an unmodified 2012 Alfa Romeo Giulietta. The research showed that any person who has physical access to the network can disrupt it, even with a simple tool. This attack does not require a full message transmission; instead, it overwrites to the recessive bits and generates a transmission error. The contribution of this research is that it exploited the vulnerability of the CAN standard. After this research, an alert (ICS-ALERT-17-209-01) [27] was announced by the U.S. government. A similar research analysis was carried out by Murvay and Groza [28] to show the limitations of the attack on different bit rates and to breach the authentication methods.

Mukherjee et al. [16] implemented DoS attacks on the SAE J1939 standard [29], which is used in heavy-duty commercial vehicles. They performed three separate DoS attacks: (i) sending too many request messages for a supported Parameter Group Number (PGN) to overload the recipient ECU, (ii) sending manipulated false request to send (RTS) and causing overflow at the recipient buffer, and (iii) keeping the connections open via Clear to Send (CTS) messages and occupying the whole network. This work was one of the first studies to exploit the SAE J1939 specification. Murvay and

Groza [15] implemented impersonation and DoS attacks on SAE J1939. These works showed that SAE J1939 is vulnerable to protocol-specific attacks in addition to all CAN bus attacks.

There can also be indirect physical access attacks. These attacks require a physical object to be inserted into the car, but adversaries do not necessarily have direct access to the network. Checkoway et al. [20] developed an indirect access attack model, which included hacking the IT system of the car service and accessing the CAN via computer. The attack model also included attacking via multimedia devices (CD, USB, or MP3 player). Hoppe et al. [12] implemented an attack with a multimedia disc. Although the attack did not breach the CAN, it may scare the driver by flashing a warning on the screen and playing an alarm signal.

4.2. Remote Access Attacks

Nowadays, modern vehicles contain different types of wireless interfaces needed for communicating with systems such as passive anti-theft, tire pressure monitoring system (TPMS), Bluetooth, radio data, telematics, and so on. These wireless interfaces need to communicate with the CAN, usually via a gateway ECU to protect the network. However, there are studies that have demonstrated the hacking of a gateway ECU and gain accessed to the isolated CAN [12].

Checkoway et al. [20] compromised the TPMS, Bluetooth, FM channel, and a cellular network of a car through reverse engineering and claimed that thieves could steal vehicles easily as doors could be unlocked through the CAN messages. Woo et al. [21] proposed a remote attack via a malicious self-diagnostic app. If someone uses a malicious app to monitor/diagnose the vehicle's situation, the adversary takes control of the vehicle remotely and performs its attack from a long-distance.

Valasek and Miller [22] carried out a remote attack survey on 12 car brands and 21 commercial cars and identified the remote attack surfaces and their difficulties in compromising each vehicle. The attack was three-staged. The first stage was to compromise the ECU responsible for a wireless interface. The second stage was to inject messages to communicate with the safety-critical ECU. The last stage was to modify the ECU to behave maliciously. While the researchers believed that the increasing number of cyber-physical systems in the cars would increase their vulnerabilities, they could not practically verify this because of the high number of different applications in the vehicles. Furthermore, they also hacked a Jeep Cherokee remotely and disabled the engine in 2014 [9]. After this attack, a public announcement that stated the vulnerability of motor vehicles against remote attacks was published [30].

Savage and his team [31] took control of a Chevrolet Corvette's brakes and windshield wipers via a commercial telematics control unit in 2016. This attack indicates that the vulnerability of the CAN can be penetrated by the aftermarket equipment and cannot be entirely addressed by the manufacturer [32].

Nie et al. [13] implemented a remote attack on a Tesla Model S in 2016 via a wireless and cellular interfaces. The Keen Security Lab of Tencent [14] discovered multiple attack surfaces on BMW vehicles, which showed that even high-end commercially available cars could suffer from cyber-attacks.

Another wireless attack method is over-the-air (OTA) software updates. OTA is a cost-effective and scalable solution that allows the manufacturers to deliver software updates remotely. However, it is another attack surface where hackers can dive into the vehicle's communication network. Beek and Samani [23] implemented a ransomware attack via an OTA update.

The remote attack surface of the modern car is more substantial than the physical one, and with the rising connectivity in cars, the number of wireless attack surfaces is increasing day by day. In the near future, cars will be equipped with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, which build vehicular ad hoc networks (VANETs). VANETs aim for traffic optimisation and collision avoidance. To provide these benefits, VANETs use car sensors and have wireless connectivity. In VANETs, spoofed messages can be received or transmitted, and as a result, the in-vehicle communication network may be disrupted.

4.3. Privacy in the CAN

Acquiring CAN network data not only causes safety issues, but also the invasion of privacy. The modern vehicle collects data related to the driver, which passes through the vulnerable CAN network. An investigation [33] revealed that it was possible to obtain the precise location history of the car and other personal data (log of phone calls, list of contacts, email addresses, and photos) from the connected phone. An adversary can steal personal information only by passively listening to the bus. Furthermore, researchers [34,35] have shown that it is possible to identify the driver based on the sensory data travelling through the CAN bus. Therefore, monitoring the in-vehicle network can invade personal privacy.

5. Counter Measures for CAN Attacks

The attacks on CAN clearly show that the protocol is very vulnerable and requires cyber defence mechanisms for safe driving. The studies to solve this problem have mainly focused on four defence mechanisms: network segmentation, encryption, authentication, and intrusion detection, which are summarised in Table 2.

Table 2. Methods to secure the CAN bus.

Proposed Method	Benefits	Disadvantages
Network Segmentation	Limit access to the end-user	Increased cost, Difficulty in maintenance
Encryption	Hardened attacks, Confidential data transmission	Increased computational power, Increased traffic, Weak encryption due to frame size
Authentication	Secure data transmission	Increased computational power, Increased traffic
Intrusion Detection	Detect anomalies and attacks	Complicated algorithm design, Cannot guarantee the security

5.1. Network Segmentation

The most straightforward protection mechanism is separating the CAN network into multiple subnetworks. The segmentation provides control over who can access particular subnetwork and reduce the damage of the attack by limiting its spread. The interconnection between subnetworks is controlled via a gateway ECU. This model currently exists in commercial vehicles. The method is simple to implement, but it is not effective if the gateway ECU is compromised or manipulated like the hacking exhibited in [12]. Kammerer et al. [36] addressed this issue and proposed a star coupling router with security features. The paper ignored the security inside a subnetwork, but it is possible to implement a replay attack in a subnetwork and attack the other subnetworks bypassing the security check of the router.

Researchers at TU München proposed an automotive service bus architecture [37] where their two-layer architecture was designed to prevent external attacks. The infotainment system and all vital functions were separated from each other. All components could send and receive messages, but by default, they could not send any data as the central ECU allows whom to write to the automotive service bus.

Network segmentation increases the security level, but it is not a sufficient method to protect the CAN. It also makes the maintenance of the system more difficult, along with the increased cost.

5.2. Encryption

The CAN protocol uses a shared broadcast network without a built-in encryption mechanism. This allows an adversary to eavesdrop all the nodes and understand the communication. To prevent this data breach, a light-weight encryption system should be implemented. Although there are

commercial software-based encryption methods (e.g., Trillium [38], CANcrypt [39]) and manufacturers have proprietary encryption techniques implemented in cars, there have been reports claiming that encryption mechanisms in commercially available cars can be broken [40,41].

The limited data field is one of the problems for secure CAN encryption. This problem can be overcome by sending multiple CAN frames for a single message, and may solve the problem on low traffic networks, but it is not a solution for the currently rising traffic in automobile CAN networks. Another issue is the limited computational power of ECUs. If we consider the lifetime of a vehicle, it is possible to crack a static encryption key. Therefore, dynamic key exchange is required. However, this is harder to implement and is computationally expensive. The dynamic key can also cause latency on resource-constrained ECUs, and it is not acceptable for safety-critical real-time systems.

The different encryption mechanisms proposed are shown in Table 3. Doan and Ganesan [42] implemented hardware-based AES-128 encryption on FPGA chips for the CAN system. The hardware implementation of the method decreases latency and increases throughput. However, the method changes the legacy ECU and is not backward compatible. Another study used physical unclonable functions (PUFs) [43]. This method can obtain the private key from the physical characteristics of the ECUs; thus, hiding the key is not a problem. Although the method solves the problem for generating encryption keys, it also requires modifying the ECU.

Table 3. Encryption methods for the CAN bus.

Reference	Encryption Method	Traffic Effect	Key
[42]	AES-128 and SHA-1	Increased	Static Symmetric
[44]	XOR	No Change	Dynamically Synchronised
[43]	AES-256 and Elliptic-curve Diffie–Hellman	Increased	Symmetric
[45]	XOR	No Change	Static Symmetric
[46]	Tiny Encryption Algorithm	Increased	Static Symmetric
[47]	Triple DES	Increased	Dynamically Synchronised

Encryption hardens attacks and provides privacy; however, it is not sufficient to protect the CAN. Even the unbreakable encryption mechanism cannot prevent replay attacks.

5.3. Authentication

It is not possible to identify the sender of a CAN message. If an adversary has access to the network, they can send malicious messages and all the nodes accept them as authentic. This can be prevented via authentication.

VeCure [48] authentication, which has an acceptable 50 us processing delay, is based on trust groups where high-trust groups share a symmetric secret key. The method has a major advantage with fewer key numbers, which corresponds in size to the number of trust groups rather than the ECU number. However, it sends an authentication message after every transmitted frame, which doubles the network traffic. Another drawback of the method is that it cannot protect the system if a node from the trust group is compromised. LiBrA-CAN [49], proposed by Groza et al., splits the authentication keys between groups of multiple nodes to improve efficiency. Although the method is quite successful, it requires high bandwidth and is not compatible with traditional CAN.

Nowdehi et al. [50] identified five criteria for an authentication method to be implemented commercially: cost-effectiveness, backward compatibility, support for vehicle repair and maintenance, sufficient implementation details, and acceptable overhead. They analysed ten authentication methods in the literature using them. Not surprisingly, none of the methods could pass all five criteria.

There are also off-the-shelf products providing hardware-based authentication like the S32K family from NXP [51]. The S32K family has Cryptographic Service Engine (CSE), which has a Cipher-based Message Authentication Code (CMAC) to provide secure authentication, and is a hardware-based system that accelerates the process drastically. For instance, public-key authentication can be achieved

in less than 100 us [52] with hardware acceleration, while software authentication takes more than 10 ms, depending on the key size. However, the industry is currently concerned with the cost of ECUs. With the enhancement of hardware technology, it is possible to see more hardware-based methods to secure the CAN.

5.4. Intrusion Detection System (IDS)

Implementing security features on a safety-critical real-time system is a difficult task. Strong cryptographic methods are not feasible due to the limited resources (memory, bandwidth, and computational power) and time constraints, from which research on intrusion detection system (IDS) for CAN has emerged. The main advantage of IDS is that it usually does not modify the current CAN controller and the bus traffic does not increase.

Intrusion detection methods can be categorised as signature-based (misuse) detection and anomaly-based detection [53]. Signature-based detection checks for known attacks on the database; therefore, it requires regular updates for new attacks. Although it is quite successful in detecting known attacks, it fails to detect unknown attacks. Anomaly-based IDS analyses the behaviour of the network and recognises the deviation from normal behaviour. Accuracy is usually lower than that of the signature-based. In contrast to signature-based detection, anomaly-based IDS may detect unknown attacks.

There are different parameters that an IDS system can assess on the CAN. Müter et al. [54] defined eight anomaly detection sensors, as shown in Table 4, to identify the anomalies in a structured way. All these detection sensors were inspired by the typical behaviour of the CAN bus. Deviation from the normal behaviour of these sensors is the sign of the intrusion, and different IDS solutions use these sensors to detect intrusions. These solutions can be categorised as time/frequency-based, physical system characteristic, specification-based, and feature-based.

Table 4. Automotive anomaly detection sensors [54].

Sensor	Description
Formality	Correct message size, header and field size, field delimiters, checksum, etc.
Location	The message is allowed with respect to the dedicated bus system
Range	Compliance of payload in terms of data range
Frequency	Timing behaviour of messages is approved
Correlation	Correlation of messages on different bus systems adheres to the specification
Protocol	The correct order, start-time, etc. of internal challenge-response protocols
Plausibility	Content of message payload is plausible, no infeasible correlation with previous values
Consistency	Data from redundant sources is consistent

5.4.1. Time/Frequency-Based IDS

The automobiles have rigid safety rules and most of the ECUs transmit periodic signals. Any change in the frequency can be interpreted as abnormal behaviour, in other words, an intrusion. The basic IDS analyses the frequency of the CAN messages as presented in [55,56].

Offset ratio and time interval based IDS [57], as proposed by Lee et al., analyses the response time of the transmitted remote frame where the simple effective algorithm can detect attacks and type of attacks, however, the method increases bus traffic by injecting remote frames for analyses.

The time/frequency analysis provides useful information about the CAN. However, the vehicle's situation (e.g., idle, running) and the priority scheme of the CAN may significantly change the timing information and affect the result of time/frequency-based IDS. The method also cannot detect attacks where the frequency is not changed, like a masquerade attack in [58].

5.4.2. Physical Characteristic Based IDS

The physical characteristic of the CAN network can be used to detect intrusions; hence each transceiver has a different signal shape even though they transmit the same data. This can be caused by random manufacturing variations, cabling, and aging.

In [59], Choi et al. proposed VoltageIDS, which uses unique electrical characteristics of the CAN signal like a fingerprint. The different locations of the ECUs with different lengths of wire results in different resistance [60] and the resistance changes the signal features. They analysed eight of the signal features like positive and negative slope values and voltage value at a dominant level. The method has zero false-positive rates and can differentiate between attacks and errors; however, it requires an oscilloscope to gather the network signal and has heavy signal processing.

The CAN does not have a shared master clock, and each ECU uses its own quartz crystal. Cho and Shin [58] suggested the use of clock skew to detect intrusions. Although ECUs run the same frequency, they may have random drifting exceeding 2400 ms in a day [61]. They fingerprinted the transmitter ECU via the clock skew and detected the intrusions. Although they could reach 97% of the anomaly detection with a false-positive rate of 0.55%, the method only worked for the periodic messages. However, this method can be tricked by mimicking the clock skew, as shown in [62].

The physical characteristics of the CAN provides substantial information about ECUs. However, environmental factors like temperature and humidity and aging of the components can change the physical characteristics; therefore, the IDS may fail. They can also not detect the attacks from the software layer because the authenticated ECU will transmit the malicious messages, and the IDS does not find any changes to the signal characteristics. Similarly, the physical characteristic-based IDS requires heavy signal processing. As a result, it may cause latency or require expensive hardware.

5.4.3. Specification-Based IDS

Larson et al. [63] suggested specification based attack detection and implemented specification rules based on the CAN Open protocol. This method has limited attack detection capability and requires all the ECUs to have detectors. The method also is not powerful enough to prevent attacks; hence there are protocol compliant attacks like in [64].

Studnia et al. [65] proposed a language-based intrusion detection and derived the language characteristic of the network from the ECUs' specifications and generated the forbidden sequences. If one of these sequences occurs, an intrusion is detected.

5.4.4. Feature-Based IDS

Feature-based system analysis examines the network parameters like busload, frequency, number of dropped messages, and other parameters like abnormal messages and payload. This is usually based on artificial intelligence techniques.

Generative adversarial nets (GAN) based IDS [66] was proposed by Seo et al., who used the deep-learning model. The method is easy to expand and difficult to manipulate by an attacker, hence the detection mechanism has a black-box characteristic. Bloom filtering [67], proposed by Groza and Murvay, analysed the periodicity and payload of CAN messages. This method provides a memory-efficient analysis of data. Although both methods require heavy computation, they look promising in terms of tackling the CAN security problem.

Table 5 presents the comparison of the IDSs. Each method has a unique feature to suppress other methods, but also comes with a cost. For example, physical characteristic-based IDS can easily detect an inauthentic node, but it fails to detect an attack from a software layer. The best IDS system should be a hybrid system that takes advantage of different methods. Although IDS can mitigate a security problem, it cannot provide confidentiality. To have complete security, cryptography is required.

Table 5. Comparison of the intrusion detection system (IDS).

Reference	Algorithm Analyses	Parameters	Advantages	Downsides
[66]	Generative Adversarial Nets	A pattern of CAN ID	CAN train itself for unknown attacks	Expensive hardware
[68]	Adaptive Network-based Fuzzy Inference System	Busload, message frequency analysis	Detect attack type, simple solution	Works for simple attacks, updated each second, needs a feature database
[69]	Entropy-based	Entropy of IDs, payload	Does not require much information about traffic data	Very vulnerable to some attacks which include random bits
[70]	Long Short-term Memory Networks	Payload	Does not require pre-knowledge	Does not understand the natural change
[63]	Specification-based	Protocol policy	Less dependency	IDS should be placed at every ECU
[71]	Hamming Distance	Payload	Low computation	Low detection
[57]	Offset ratio and time interval	Remote frame timing	Simple efficient algorithm with low-cost hardware	Increased traffic
[72]	Analysis of ID Sequence	Sequence of ID	Low memory and computation requirement, detection of inserted few malicious messages	Very vulnerable to attacks which have a similar sequence of normal traffic
[59]	Support Vector Machine and Boosted Decision Tree	Electrical signal	Robust to some attack types, first IDS to differentiate between an error and an attack	High cost and vulnerable to environmental changes
[58]	Recursive Least Squares	Clock skew	Robust to some attack types,	Only works on periodic signals
[67]	Bloom Filtering	Message identifier, payload	Low memory usage for membership testing	Complex algorithm
[56]	Probability Density Function	Reception cycle period (frequency analysis)	Online learning	Hard to authenticate a non-periodic message
[55]	Flow-based	Message frequency	Simple algorithm	Only works on periodic signals

6. Discussions on CAN Security Research

Automotive security is getting more attention, and standardisations are coming to tackle cybersecurity problems. Cybersecurity guidebook for cyber-physical vehicle systems [73] and the fundamental principles of automotive cybersecurity specification (PAS 1885:2018) [74] were published by SAE in 2016 and British Standards Institute in 2018 consecutively. ISO 21434 Automotive Cybersecurity [75] is under development and expected to be released by 2020.

The CAN protocol has also gained attention from the industry to its vulnerabilities, and companies are now manufacturing high-end secure ECUs. The Secure Hardware Extension (SHE) [76] specification developed by the Hersteller Initiative (HIS) becomes an open standard and used by many companies in their ECUs like NXP MPC5646C [52] microcontroller. Some commercial ECUs have built-in IDS; the NXP TJA115x [77] series can prevent spoofing attacks and be used as an IDS. There are also commercial proprietary intrusion detection systems [78,79].

Although there have been steps taken to protect the CAN, there is still more to do. The industry does not share some of their research, and academia does not have enough resources. As such, there are

not sufficient attack data and benchmarks. Implementing attacks on real vehicles can be unfeasible for safety concerns and cost. To overcome these challenges, there should be more research on modelling CAN bus attacks like in [80] and creating attack databases like in [81,82]. Sharing datasets as an open-source (e.g., like in [66]) will help researchers; hence working on shared datasets will give a reference point to compare their research.

7. Conclusions

The CAN protocol facilitating ECUs in modern vehicles is not geared up and well-protected against the complex and evolving nature of cyberattacks. The existing security features incorporated in vehicles are not fit and adequate to resist and defy them. This is attributable to the lack of encryption and authentication mechanisms, which provide multiple opportunities for several types of attacks to materialize and as a result, jeopardize the individual data privacy and the safety of the vehicle occupants. These blemish the manufacturers' reputation and downgrade vehicle reliability, followed by substantial financial losses.

We have observed that the existing trend of attacks is mainly physical-access oriented; however, with the growing connectivity in vehicles, we have also noted a considerable increase in wireless attacks. This developing trend is indicative of wireless attacks outpacing the physical access attacks in the near future.

Moreover, an in-depth analysis of the vulnerabilities of the CAN bus to cyberattacks points to the limitations posed by the protocol. The root cause evaluation of various attacks and the critique of potential solutions has revealed the inadequacies and constraints of both the industry and academia. They are not driven toward mutual sharing of an attack database, allocation of testing and trial resources, and developing benchmarks for an open-source.

There are four main countermeasures for the CAN attacks, comprising network segmentation, encryption, authentication, and IDS. They are, however, heavy on overheads with respect to the availability of the existing resources. Further analysis has revealed IDS as the most promising option when compared to the rest of the solutions above-mentioned. It is noteworthy that the IDS may not provide complete security, but it can prevent several of the CAN vulnerabilities with acceptable overhead. We presume that future vehicles will have IDS solutions not only to secure the vehicle, but to also provide data to the manufacturer to tackle cyberattacks.

Author Contributions: Conceptualisation, M.B.; Methodology, M.B.; Writing-Original Draft Preparation, M.B. and M.S.; Writing-Review and Editing, M.B., S.A., M.S., and I.J.; Supervision, I.J. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This paper is an extended version of the conference paper presented at the 2018 International Conference on Computing, Electronics, and Communications Engineering (iCCECE). The conference paper can be reached at the DOI address 10.1109/iCCECOME.2018.8658720.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. "ECU" is a Three Letter Answer for All the Innovative Features in Your Car: Know How the Story Unfolded. Embitel. 2017. Available online: https://www.embitel.com/blog/embedded-blog/automotive-control-units-development-innovations-mechanical-to-electronics (accessed on 23 May 2018).
2. Hira, E. Automotive Electronic Control Unit (ECU) Market Size Share, 2022. Allied Market Research. 2017. Available online: https://www.alliedmarketresearch.com/automotive-electronic-control-unit-ecu-market (accessed on 22 July 2018).
3. Dürrwang, J.; Braun, J.; Rumez, M.; Kriesten, R. Security evaluation of an airbag-ECU by reusing threat modeling artefacts. In Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence, CSCI, Las Vegas, NV, USA, 14–16 December 2017; pp. 37–43.

4. Matthews, C. Jeep Cherokee hack: Fiat Recalls 1.4 Million Vehicles. Fortune. 2015. Available online: https://fortune.com/2015/07/24/jeep-cherokee-recall/ (accessed on 27 March 2020).
5. Maloney, D. Dashboard Dongle Teardown Reveals Hardware Needed to Bust Miles. Hackaday. 2019. Available online: https://hackaday.com/2019/12/16/dashboard-dongle-teardown-reveals-hardware-needed-to-bust-miles/ (accessed on 21 March 2020).
6. Vector Informatik. *Industry Trends 2019: Convergence Drives Competitiveness and Innovation*; Vector Informatik: Stuttgart, Germany, 2019.
7. Groza, B.; Murvay, S. Security solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks. *IEEE Veh. Technol. Mag.* **2018**, *13*, 40–47. [CrossRef]
8. Liu, J.; Zhang, S.; Sun, W.; Shi, Y. In-vehicle network attacks and countermeasures: Challenges and future directions. *IEEE Netw.* **2017**, *31*, 50–58. [CrossRef]
9. Greenberg, A. Hackers Remotely Kill a Jeep on the Highway. Wired.com. 2015. Available online: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/ (accessed on 10 September 2018).
10. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; et al. Experimental security analysis of a modern automobile. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 447–462.
11. Palanca, A.; Evenchick, E.; Maggi, F.; Zanero, S. A stealth, selective, link-layer denial-of-service attack against automotive networks. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Bonn, Germany, 6–7 July 2017; Volume 10327, pp. 185–206.
12. Hoppe, T.; Kiltz, S.; Dittmann, J. Security threats to automotive CAN networksPractical examples and selected short-term countermeasures. *Reliab. Eng. Syst. Saf.* **2011**, *96*, 11–25. [CrossRef]
13. Nie, S.; Liu, L.; Du, Y. Free-Fall: Hacking Tesla from wireless to CAN bus. *BlackHat USA* **2017**, *2017*, 1–16.
14. Tencent Keen Security Lab. *Experimental Security Assessment of BMW Cars: A Summary Report*; Peerlyst Inc.: Shenzhen, China, 2018.
15. Murvay, P.S.; Groza, B. Security shortcomings and countermeasures for the SAE J1939 commercial vehicle bus protocol. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4325–4339. [CrossRef]
16. Mukherjee, S.; Shirazi, H.; Ray, I.; Daily, J.; Gamble, R. *Practical DoS Attacks on Embedded Networks in Commercial Vehicles*; Springer International Publishing: Cham, Switzerland, 2016; Volume 10063.
17. Bosch, R. *CAN Specification Version 2.0*; CAN: Stuttgart, Germany, 1991.
18. CSS Electronics. *CAN Bus Explained*; CSS Electronics: Aarhus, DK, Jutland, 2019.
19. Statista. *Automotive Electronics Cost as A Percentage of Total Car Cost Worldwide from 1950 to 2030*; Statista: London, UK, 2018.
20. Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; Koscher, K.; Czeskis, A.; Roesner, F.; Kohno, T. Comprehensive experimental analyses of automotive attack surfaces. In Proceedings of the 20th USENIX conference on Security, San Diego, CA, USA, 20–22 August 2014.
21. Woo, S.; Jo, H.J.; Lee, D.H. A practical wireless attack on the connected car and security protocol for in-vehicle CAN. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 993–1006. [CrossRef]
22. Miller, C.; Valasek, C. A Survey of Remote Automotive Attack Surfaces. Black Hat USA. 2014. Available online: http://ftpcontent.worldnow.com/wbbh/documents/Remoteattacksurfaces.pdf (accessed on 21 April 2020).
23. Beek, C.; Samani, R. DEFCON—Connected Sar Security. McAfee. 2017. Available online: https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/defcon-connected-car-security/ (accessed on 18 August 2019).
24. Ishtiaq, R.A.; Rob, M.B.; Hossen, M.A.; Travis, T.A.; Sangho, O.B.; Xua, W.; Marco, G.; Ivan, S. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. *Proc. USENIX Secur. Symp.* **2010**, *39*, 11–13.
25. Hoppe, T.; Dittman, J. Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In Proceedings of the 2nd workshop on embedded systems security (WESS), Brussels, Belgium, 4 October 2007; pp. 1–6.
26. Rebecca, B. Proof-of-Concept CarShark Software Hacks Car Computers, Shutting Down Brakes, Engines, and More. Popular Science. Available online: https://www.popsci.com/cars/article/2010-05/researchers-hack-car-computers-shutting-down-brakes-engine-and-more (accessed on 29 May 2018).

27. The National Cybersecurity and Communications Integration Center (NCCIC). CAN Bus Standard Vulnerability|ICS-CERT. 2017. Available online: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-17-209-01 (accessed on 11 March 2019).
28. Murvay, P.-S.; Groza, B. DoS attacks on Controller Area Networks by fault injections from the software layer. In Proceedings of the 12th International Conference on Availability, Reliability and Security-ARES '17, Reggio Calabria, Italy, 29 August–2 September 2017; pp. 1–10.
29. SAE International. J1939: Serial Control and Communications Heavy Duty Vehicle Network. 2018. Available online: https://www.sae.org/standards/content/j1939_201808/ (accessed on 29 December 2019).
30. Federal Bureau of Investigation. Motor Vehicles Increasingly Vulnerable to Remote Exploits. 2016. Available online: https://www.ic3.gov/media/2016/160317.aspx (accessed on 5 August 2019).
31. Greenberg, A. Hackers Cut a Corvette's Brakes Via a Common Car Gadget. Wired. 2015. Available online: https://www.wired.com/2015/08/hackers-cut-corvettes-brakes-via-common-car-gadget/ (accessed on 5 August 2019).
32. Foster, I.; Prudhomme, A.; Koscher, K.; Savage, S. Fast and Vulnerable: A story of telematic failures. In Proceedings of the 9th USENIX Workshop on Offensive Technologies (WOOT'15), Washington, DC, USA, 10–11 August 2015.
33. Fowler, G. Driving Surveillance: What Does your Car Know about you? We hacked a 2017 Chevy to Find Out.-The Washington Post. Washingtonpost. 2019. Available online: https://www.washingtonpost.com/technology/2019/12/17/what-does-your-car-know-about-you-we-hacked-chevy-find-out/ (accessed on 19 March 2020).
34. Enev, M.; Takakuwa, A.; Koscher, K.; Kohno, T. Automobile driver fingerprinting. In Proceedings of the Proceedings on Privacy Enhancing Technologies, Darmstadt, Germany, 19–22 July 2016; pp. 34–51.
35. Fugiglando, U.; Santi, P.; Milardo, S.; Abida, K.; Ratti, C. Characterizing the 'driver DNA' through CAN bus data analysis. In Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services, New York, NY, USA, 23–30 October 2017; Volume 17, pp. 37–41.
36. Kammerer, R.; Frömel, B.; Wasicek, A. Enhancing security in CAN systems using a star coupling router. In Proceedings of the 7th IEEE International Symposium on Industrial Embedded Systems, SIES 2012-Conference Proceedings, Karlsruhe, Germany, 20–22 June 2012; pp. 237–246.
37. Technische Universität München. The Car Becomes Internet Hardware-TUM. 2015. Available online: https://www.tum.de/nc/en/about-tum/news/press-releases/details/32277/ (accessed on 21 November 2019).
38. Yoshida, J. CAN Bus Can Be Encrypted, Says Trillium. EEtimes. 2015. Available online: https://www.eetimes.com/document.asp?doc_id=1328081&page_number=2 (accessed on 29 May 2019).
39. CANcrypt-Home. Available online: https://www.cancrypt.eu/index.php/en/ (accessed on 29 May 2018).
40. 2015 BMW F80 M3 / F82 M4 S55 inline-6 ecu Flash Dyno Results from Jailbreak Tuning. BimmerBoost. 2014. Available online: https://www.bimmerboost.com/content.php?5101-2015-BMW-F80-M3-F82-M4-S55-inline-6-ecu-flash-dyno-results-from-Jailbreak-Tuning (accessed on 5 August 2019).
41. Jurnecka, R. Cobb Tuning Cracks Nissan GT-R's Encrypted ECU-MotorTrend. Motortrend. 2008. Available online: https://www.motortrend.com/news/cobb-tuning-cracks-nissan-gtrs-encrypted-ecu-308/ (accessed on 5 August 2019).
42. Doan, T.P.; Ganesan, S. CAN crypto FPGA chip to secure data transmitted through CAN FD bus using AES-128 and SHA-1 algorithms with a symmetric key. *WCXTM 17 SAE World Congr. Exp.* **2017**. [CrossRef]
43. Siddiqui, A.S.; Plusquellic, Y.G.J.; Saqib, F. Secure communication over CANBus. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1264–1267.
44. Harel, A.; Hezberg, A. Optimizing CAN bus security with in-place cryptography. In Proceedings of the SAE Connected and Automated Vehicle Conference Israel, Tel Aviv, Israel, 16–17 January 2019.
45. Farag, W.A. CANTrack: Enhancing automotive CAN bus security using intuitive encryption algorithms. In Proceedings of the 2017 7th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO 2017, Sharjah, United Arab Emirates, 4–6 April 2017.
46. Jukl, M.; Čupera, J. Using of tiny encryption algorithm in CAN-Bus communication. *Res. Agric. Eng.* **2016**, *61*, 50–55. [CrossRef]

47. Hanacek, A.; Sysel, M. Design and implementation of an integrated system with secure encrypted data transmission. *Adv. Intell. Syst. Comput.* **2016**, *466*, 217–224.
48. Wang, Q.; Sawhney, S. VeCure: A practical security framework to protect the CAN bus of vehicles. In Proceedings of the 2014 International Conference on the Internet of Things, IOT 2014, Cambridge, MA, USA, 6–8 October 2014; pp. 13–18.
49. Groza, B.; Murvay, S.; van Herrewege, A.; Verbauwhede, I. LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks. *Lect. Notes Comput. Sci. Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.* **2012**, *7712*, 185–200.
50. Nowdehi, N.; Lautenbach, A.; Olovsson, T. In-vehicle CAN message authentication: An evaluation based on industrial criteria. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–7.
51. NXP. 32-bit Automotive General Purpose MCUs. NXP. Available online: https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/s32-automotive-platform/32-bit-automotive-general-purpose-microcontrollers:S32K (accessed on 14 August 2019).
52. Soja, R. *Automotive Security: From Standards to Implementation*; Automotive Microcontrollers and Processors, NXP: Eindhoven, The Netherlands, 2014.
53. Scarfone, K.; Mell, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*; Technical Report; National Institute of Standards & Technology: Gaithersburg, MD, USA, 2012.
54. Müter, M.; Groll, A.; Freiling, F.C. A structured approach to anomaly detection for in-vehicle networks. In Proceedings of the 2010 6th International Conference on Information Assurance and Security, IAS 2010, Atlanta, GA, USA, 23–25 August 2010; pp. 92–98.
55. Taylor, A.; Japkowicz, N.; Leblanc, S. Frequency-based anomaly detection for the automotive CAN bus. In Proceedings of the 2015 World Congress on Industrial Control Systems Security (WCICSS), London, UK, 14–16 December 2015; pp. 45–49.
56. Hamada, Y.; Miyashita, Y.; Hata, Y.; Inoue, M.; Ueda, H. Anomaly-based intrusion detection using the density estimation of reception cycle periods for in-vehicle networks. *SAE Int. J. Transp. Cybersecur. Priv.* **2018**, *1*, 39–56. [CrossRef]
57. Lee, H.; Jeong, S.H.; Kim, H.K. OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 57–5709.
58. Cho, K.-T.; Shin, K.G. Fingerprinting electronic control units for vehicle intrusion detection. In Proceedings of the 25th USENIX Security Symposium, Austin, TX, USA, 10–16 August 2016; pp. 911–927.
59. Choi, W.; Joo, K.; Jo, H.J.; Park, M.C.; Lee, D.H. VoltageIDS: Low-level communication characteristics for automotive intrusion detection system. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 8. [CrossRef]
60. Kang, K.-D.; Baek, Y.; Lee, S.; Son, S.H. An analysis of voltage drop as a security feature in Controller Area Network. In Proceedings of the 2016 IEMEK Symposium on Embedded Technology, 216AD, Daejeon, Korea, 26–27 May 2016.
61. Mohalik, S.; Rajeev, A.C.; Dixit, M.G.; Ramesh, S.; Suman, P.V.; Pandya, P.K.; Jiang, S. Model checking based analysis of end-to-end latency in embedded, real-time systems with clock drifts. In Proceedings of the Proceedings-Design Automation Conference, Zurich, Switzerland, 26–28 March 2008; pp. 296–299.
62. Sagong, S.U.; Ying, X.; Clark, A.; Bushnell, L.; Poovendran, R. Cloaking the Clock: Emulating Clock Skew in Controller Area Networks. In Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), Porto, Portugal, 11–13 April 2018; pp. 32–42.
63. Larson, U.E.; Nilsson, D.K.; Jonsson, E. An approach to specification-based attack detection for in-vehicle networks. In Proceedings of the IEEE Intelligent Vehicles Symposium, Proceedings, Eindhoven, The Netherlands, 4–6 June 2008; pp. 220–225.
64. Si, W.; Starobinski, D.; Laifenfeld, M. Protocol-compliant DoS attacks on can: Demonstration and mitigation. In Proceedings of the IEEE Vehicular Technology Conference, Sydney, Australia, 4–7 June 2017.
65. Studnia, I.; Alata, E.; Nicomette, V.; Kaâniche, M.; Laarouchi, Y. A language-based intrusion detection approach for automotive embedded networks. *Int. J. Embed. Syst.* **2018**, *10*, 1–12. [CrossRef]
66. Seo, E.; Song, H.M.; Kim, H.K. GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust, PST 2018, Belfast, Northern Ireland, UK, 28–30 August 2018.

67. Groza, B.; Murvay, P. Efficient intrusion detection with bloom filtering in Controller Area Networks (CAN). *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1037–1051. [CrossRef]
68. Li, F.; Wang, L.; Wu, Y. Research on CAN network Security Aspects and Intrusion Detection Design. 2017. Available online: https://saemobilus.sae.org/content/2017-01-2007/ (accessed on 21 April 2020).
69. Muter, M.; Asaj, N. Entropy-based anomaly detection for in-vehicle networks. In Proceedings of the 2011 IEEE Intell. Veh. Symp. (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 1110–1115.
70. Taylor, A.; Leblanc, S.; Japkowicz, N. Anomaly detection in automobile control network data with long short-term memory networks. In Proceedings of the 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Montreal, QC, Canada, 17–19 October 2016; pp. 130–139.
71. Stabili, D.; Marchetti, M.; Colajanni, M. Detecting attacks to internal vehicle networks through Hamming distance. In Proceedings of the 2017 AEIT International Annual Conference, Cagliari, Italy, 20–22 September 2017; pp. 1–6.
72. Marchetti, M.; Stabili, D. Anomaly detection of CAN bus messages through analysis of ID sequences. In Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI, USA, 11–14 June 2017; pp. 1577–1583.
73. Vehicle Cybersecurity Systems Engineering Committee. *J3061-Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*; Technical Report; SAE International: Washington, DC, USA, 2016.
74. BSI. *PAS 1885:2018 The Fundamental Principles of Automotive Cyber Security-Specification*; BSI Standards Limited: London, UK, 2018.
75. ISO. *ISO/SAE CD 21434-Road Vehicles—Cybersecurity Engineering*; ISO: London, UK, 2019.
76. Mundhenk, P. *Security for Automotive Electrical/electronic (E/E) Architectures*; Cuvillier Verlag: Göttingen, Germany, 2017.
77. Semiconductors, N. *TJA115x Secure CAN Communication without Cryptography*; NXP: Eindhoven, The Netherlands, 2019.
78. ECUSHIELD-The Only Proven Ready for Integration Automative Cyber Security Solution. Available online: http://tower-sec.com/ecushield/ (accessed on 30 March 2018).
79. Argus Cyber Security-Automotive Cyber Security. Available online: https://argus-sec.com/ (accessed on 30 March 2018).
80. Fröschle, S.; Stühring, A. Analyzing the capabilities of the CAN Attacker. *Eur. Symp. Res. Comput. Secur.* **2017**, *10492*, 464–482.
81. Ring, M.; Dürrwang, J.; Sommer, F.; Kriesten, R. Survey on vehicular attacks-Building a vulnerability database. In Proceedings of the 2015 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2015, Yokohama, Japan, 5–7 November 2015; pp. 208–212.
82. Huang, T.; Zhou, J.; Bytes, A. ATG: An Attack Traffic Generation Tool for security Testing of in-Vehicle CAN Bus. In Proceedings of the ACM International Conference Proceeding Series 2018, Hamburg, Germany, 27–30 August 2018; pp. 1–6.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).



https://dspace.org/

Evaluation of CAN bus security challenges

Bozdal, Mehmet; Samie, Mohammad; Aslam, Sohaib; Jennions, Ian

2020-04-21
MDPI
https://doi.org/10.3390/s20082364
Bozdal M, Samie M, Aslam S and Jennions I. (2020) Evaluation of CAN bus security challenges. Sensors, Volume 20, Issue 8, April 2020, Article number 2364

Downloaded from a DSpace repository. https://dspace.org/