A SURVEY OF ATTACKS OVER CONTROLLER AREA NETWORKS AND POTENTIAL COUNTERMEASURES

Sione Talia’uli Fatai Loto’aniu

BCIS, PGDipCS, MISDF

A thesis submitted to the graduate faculty of design and creative technologies
Auckland University of Technology
in fulfilment of the
requirements for the degree of
Master of Philosophy

School of Engineering, Computer and Mathematical Sciences

Auckland, New Zealand
2020

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a University or other institution of higher learning, except where due acknowledgement is made in the acknowledgements.

Sione Talia'uli Fatai Loto'aniu

Acknowledgements

I must first give thanks to our Lord God, for his love and provides help me finished this project research. I would like to give thanks to my supervisors Prof Brian Cusack (primary) and Raymond Lutui (secondary) for their counsel and field guidance, and Bryce Coad for his technical advice and assistance.

I would very much like to thank my parents and my families for their patience and good humor throughout this research journey.

Abstract

Automotive manufacturers have a specialized internal communication network, called vehicle bus which connects the Electronic Control Units (ECU) in a motor vehicle through a single or multiple bus architecture. This avoids heavy and complicated wiring structures for communication, and unifies the system into one centralized control and functionality operation. Controlled Area Network (CAN) is one of the standards for communication in these unified vehicular networks. The CAN bus protocol is designed to be lightweight, robust, fast and secure. However, the focus on centralization and simplicity for effective communications opens new and dangerous security issues. Handling these issues efficiently and effectively is essential for safety. This exploratory research gives a comprehensive survey of attacks on CAN buses, identification of potential vulnerabilities in a CAN, a summary of existing solutions, and recommendations for new solutions.

The question that guides the research seeks to identify expectations for secure but functional requirements of an effective and efficient CAN: What is required to assure CAN vehicle security? The literature analysis suggests that there are many ways a CAN may be compromised and that many successful attempts have been made. Initially vehicles were stolen by hacking the electronic key systems that were used to secure access to a vehicle. Then hacking extended to access the full vehicle control system. The effects were malicious intervention into steering, breaking, and lighting functions. In each situation the legitimate control of the vehicle was taken by a hacker so that the vehicle could be stolen or the safe use of the vehicle compromised. Today tricking the artificial intelligence functions is a method used to compromise security. These findings suggest that more attention has to be paid to the protection of the CAN system and that counter measures have to be inbuilt in order to maintain the integrity of the system.

The exploratory research proceeds by first identifying the various vulnerabilities and current security precautions used in CAN. A CAN simulator is used for software development is then mounted and tested for functionality. Diagnostic tools are also used and the limitations noted for attack vectors. The CAN is then subjected to a range of known communication attacks, and the performance noted. From the data analysis, patterns of vulnerability emerge, and the inbuilt security measures applied to check their effectiveness. The research findings show that the inbuilt security functions are effective on a limited range of attacks. Therefore the research recommends design changes and the implementation of further precautions to prevent similar attacks. The value of this research is to alert vehicle manufacturers to security requirements, and for maintenance and repair services to assure the testing equipment has detection capability. Fault codes currently detect some vulnerabilities but into the future these capabilities require continuous updating and patching by the manufacturers. Successful CAN security assurance relies on many stakeholders and the recent legislation for car hacking compliance, privacy, and vehicle automation (for autonomous control) standardization, are all beneficial to the aims of this thesis. Figure 5.3 summarizes the learning from this research and advocates methodology for building and maintaining secure CAN technologies. (Note: Some of the testing and exploratory work was not completed (see Chapter 6) because of the Covid-19 shut downs that caused delays in the testing equipment delivery and the closure of the research laboratories in Semester 1 and early Semester 2, 2020)

Table of Contents

Table of Figures

List of Tables

List of Abbreviations

ACK	Acknowledge
AR	Augmented Reality
AV	Autonomous Vehicles
AVB	Audio Video Bridging
CAN FD	Controller Area Network Flexible Data-Rate
CAN	Controller Area Network
CAN-H	Controller Area Network-High
CAN-L	Controller Area Network-Low
CIDS	Clock base Intrusion Detection System
CRC	Cyclic Redundancy Check
DBN	Deep Belief Networks
DLC	Data Length Code
DNN	Deep Neural Networks
DOS	Denial of Service
ECU	Electronic Control Units
EOF	End of Frame
FoV	Field of View

FTDMA	Multiple Access Flexible Time Division
FW	Firewall
GW	Gateway
IDE	Identifier Extension
IDS	Intrusion Detection System
IFS	Inter-Frame Space
IoT	Internet of Thing
ISP	Image Signal Processor
LLDR	Lidar Light Detection and Range
LIN	Local Interconnect Network
MANET	Mobile Ad hoc Network
NHTSA	National Highway Traffic Safety Administration
NTSB	National Transportation Safety Board
OBD	On Board Diagnostics
OBD11	On Board Diagnostics: physical port
PAN	Personal Area Network
RRDR	Radar Radio Detection and Ranging

RTaW	Real-Time-at-Work (software or tools)
RTR	Remote Transmission Request
SAE	Society of Automotive Engineers
SBFI	Simulation Based Fault Injection
SRR	Substitute Remote Request
TDMA	Multiple Access Time Division
TSN	Time Sensitive Networking
V2I	Vehicle to Infrastructure
V2M	Vehicle to Mobile
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VANET	Vehicle Ad hoc Network
VDC	Vehicle Dynamics Control

Chapter 1
INTRODUCTION

1.0 INTRODUCTION

This exploratory research addresses the concern that the highly automated motor vehicles of today are not safe and have security vulnerabilities. There are many reports (see Chapter 2) and publications that suggest it is easy for a malicious actor to take control of a motor vehicle and to create incidents that are unintended by the legitimate operators of the vehicle. To address the problem this thesis first proceeds by identifying the various vulnerabilities and current security precautions used in the controller area network (CAN) architectures that are used for motor vehicle communications. These communications maybe between multiple CANs or within CANs to electronic control units (ECUs). An ECU is then responsible for the execution of functions that deliver effects, and receive feedback on the impact of an effect. These systems form the web of control for a modern motor vehicle and support the security and safety of the humans using them.

This exploratory research uses a CAN simulator for testing vulnerabilities in the system. Diagnostic tools are also used and the potential noted for attack vectors. The CAN is subjected to a range of known communication attacks, and the performance noted. From the data analysis, patterns of vulnerability emerge, and the inbuilt security measures are observed to check the effectiveness. The research findings show that the inbuilt security functions are effective on a limited range of attacks. Therefore, the research recommends design changes (Chapter 5) and the implementation of further precautions to prevent similar attacks. The value of this research is to alert vehicle manufacturers to security requirements, and for maintenance and repair services to assure the testing equipment has limited intervention capability. Fault codes currently detect some vulnerabilities but into the future these capabilities require continuous updating and patching for fully secure testing. Successful CAN security assurances relies on many stakeholders and the recent legislation for car hacking compliance, privacy, and vehicle automation (for autonomous control) standardization are beneficial to the aims of this thesis.

This Chapter is structured as an introduction to the overall thesis and its findings. In Section 1.1 the general background and context of study are provided. In section 1.2 expresses the motivation driving this thesis, and Section 1.3 has a summary of the methodology and expected findings. Section 1.4 lays out the structure of the thesis and what to expect and where.

1.1 BACKGROUND

The automotive manufacturers have declared a specialized internal communication network called vehicle bus which connects Electronic Control Units (ECU) in car through a single bus avoiding heavy and complicated wiring structures for communication (Guo, 2011). Controlled Area Network (CAN) is one of the standards for communication in a vehicular network that is widely adopted. The CAN bus communication protocol is designed to be lightweight, robust, fast and secure. However, many security issues have been reported in CANs that suggests further research is required to better prepare protection for the information communications (Qiang, 2018). Handling these issues efficiently is essential for safety and the assurance of safe vehicles. A comprehensive survey of vulnerabilities and attack vectors is found in the literature reviewed in Chapter 2. The aim of this research is to identify some of the problems and to give potential solutions by designing countermeasures.

Nowadays, automotive networks are complex distributed electronic networks that are interconnected with services within the vehicle and those coming from the internet and other sources outside of the vehicle. CAN acts as the network that co-ordinates all the information for the successful functioning of the vehicle. As such the management of resources is critical and the optimization of resource use cost controlled for each vehicle (Song, 2016). Many car manufacturers share common contractors and more and more applications (subsystems) built by various subcontractors are expected to interact in a modern automotive system. Advanced functionalities of the subsystem such as Vehicle Dynamics Control (VDC) require support for organized and distribution of functions. X-by-wire systems must be at least as robust as those which they replace (van Roermund, 2019). In addition, regulations requiring diagnostic emissions along with the ever-increasing in-car entertainment and feedback systems further increase the number of communications-based applications and the connectivity demands. These requirements are dynamic and one CAN system today becomes reused or docked with other CANs tomorrow for the delivery of communication services that match the resources demand.

Various automotive systems have specific networking power specifications, but generically have many similarities. The reuse of components and the supply of same or similar CAN components and configurations across different vehicle brands results in the standardization of services, and strong architectural similarities (Yan, 2016). The result is a uniformity for servicing and maintenance of vehicles. However, such consistencies also make vulnerabilities and failures more dangerous. This is seen in the recent years of airbag recalls across many brands. One or few manufacturers had produced the one airbag system that was fitted to many different vehicles but all were impacted by the failure. The failure of one vehicle engineering component impacts many different manufacturers and their different brands. The issue is relevant to security vulnerabilities that are distributed by design through technologies across many vehicles and brands.

The value of this research is to identify generic vulnerabilities in the CAN architecture, and to suggest ways for mitigating the risks. This report provides an extensive survey about previous and on-going studies concerning cyber-attacks, possible vulnerabilities, and the potential countermeasures for CAN vehicle networks. It aims to present an outline of the most common attacks that exploit the vulnerabilities of the connected vehicles. The automotive industry is advancing at a fast rate, but the security provisions are not progressing at the same pace (Cui, 2019). It is often more important to have an efficient communication system or get the system to market quickly, than to incur extra time and materials costs for security mechanisms. This can develop into a dangerous situation where the threat to vehicle safety is possible. The major objective of this research is to highlight the security issues present in the CAN automotive network with the existing solutions and to inspire other researchers to participate in designing solutions to improve the cyber security of vehicles.

1.2 MOTIVATION FOR RESEARCH

Reports of hacking as a risk to vehicle safety was a key motivating point for this research. Academic research, professional Journals, and public media all have reports of people compromising the electronic control systems for vehicles. Initially it concerned the ways electronic keys could be stolen and entry to a vehicle faked or compromised. These reports noted the strength of key communication systems (encrypted communications, etc.) so thieves were resorting to social engineering techniques to get vehicle entry. These reports reflected the traditional crime of theft but with the new technologies. However, my motivation was driven by the new and dangerous crime of hacking. First University research (e.g. Washington State with Fords) demonstrated how vehicles could be hacked remotely to gain control of the vital functions of the vehicle, such as brakes, accelerator, and so on. These stories were picked up in other media and also serialized in movie and TV crime programs. The result was that there was a general awareness of the problem but little information on countermeasures. The lack of factual information provided a fear factor motivation.

In addition a general interest in the virtual and augmented reality environments created in new vehicles of all descriptions introduced concern that the element of control was being undervalued. The general direction for new vehicles was toward full autonomy. The media showed trials for fully autonomous vehicles and in many public transport situations, such as trains, the concept was fully implemented (Stanley, 2002). These examples left questions in my mind not only for security but regarding liability and safety. Key public examples have been the numerous fatalities occurring when partly autonomous vehicles have crashed (Rosen, 2009). The Tesla example showed that the driver appeared to be lulled into a sense of security and let the vehicle have control, and also that the artificial intelligence capability that was acting from a sensor network did not detect the oncoming danger and react. These things are of serious concern if the progress toward fully autonomous vehicles is to continue at the current rate.

The concern of migrating control from a human to a machine would not be an issue if the machines were able to distinguish all patterns, and to react appropriately (Rosen, 2010). At present this is not the case and the split of controls leaves points for failure that relate to safety and security. Hackers can exploit the gap and trick either the machine or the human into giving them control. Also the human may not be fully aware of the machine limitations at this time – as in the Tesla case. These are both vulnerabilities in current systems. A hacker attack uses essential business processes in the same way a legitimate user would but for their own gain. For example, if a hacker remotely puts the breaks on in a vehicle it is done electronically in terms of the CAN system and control codes in the same way as that of the legitimate driver. Hence, there is an authentication challenge to make the security mechanisms stronger so that only legitimate users may access the system.

The balance between machine and human control of vehicles is currently an area of high risk (Skrzypczyk & Mellado 2017). When a person upgrades to a new car they may not be aware of what they control, and what the machine controls. Humans tend to transfer learning from one environment or experience to another, and yet there may be no fit or a loose fit between the system controls. The human may also have their operational perceptions shaped by non-technical information such as advertising communications, hearsay, or movie fantasies. The result is dangerous gaps in the safe control of a vehicle. All of these issues have motivated me to do this research and to gain a better understanding of the problem area.

1.3 METHODOLOGY AND OUTCOMES

This research is exploratory research that seeks to answer the research question:

What is required to assure CAN vehicle security?

Also, with the main question, it has several sub-questions that will help to provide better understandings in response to the main question. The sub research questions are described as follows:

> **Sub-question1**: What are the security attacks on automotive communication networks or Electronic Control Units (ECU)?
>
> **Sub-question2:** What are the security vulnerabilities of ECUs?
>
> **Sub-question3:** What countermeasures are available for security vulnerabilities in automotive communication networks?
>
> **Sub-question4:** What management strategies are required for prevention and countermeasures in automotive communication networks?

To answer these questions a guided investigation using CAN diagnostic tools and CAN simulation tools is structured to establish the scope of the problem. The CAN diagnostic tools are used on two motor vehicles to provide evidence to answer sub-questions 2 and

4. The CAN simulator is configured to provide evidence to answer sub-questions 1 and 3. Once all the evidence is gathered and the sub-questions have answers, the main research question is addressed by referencing these findings and the guiding literature reviewed in Chapter 2. The expected outcome is a disclosure of CAN vulnerabilities and current security measures used to protect CAN communications. From the literature reviewed it is expected to find variable strengths and weaknesses in the security testing, and hence opportunity to provide advice for system improvement. The general aim is to design a security framework that can be followed for best practices in CAN security architectures (see Figure 5.3).

1.4 STRUCTURE OF THE THESIS

This Chapter provides a brief overview of the information security problem faced by the generic production and distribution of CAN vehicle control systems. It also outlines the motivation for the research, the research questions, the methods, and the expected findings. Chapter 2, reviews information security, threats and challenges faced by CANs today, as well as proposed approaches to address them. The review provides a brief technical account of CAN and the components. It also identifies relevant attacks on CANs and the expected countermeasures. A review is also made of CAN standards and control frameworks. The Chapter concludes by identifying researchable issues and problems associated with CANs.

Chapter 3 describes the research methodology. It first reviews similar published studies to learn what methodologies worked for those researchers. The design and plan are then articulated and the data requirements. Chapter 4 reports the findings from the empirical data collection in two motor vehicles and the theoretical data from the simulator. Chapter 5 then analyses the data from Chapter 4 and answers the research questions. The findings are then used to discuss the initial literature perspectives reviewed in Chapter 2. Also personal motivations and assumptions introduced in Chapter 1 are revisited in the light of the findings. Recommendations are made for improving CAN security and a best practices framework.

Finally, Chapter 6 provides a summary of the study, and recommendations for further research. The limitations of the research are also discussed in terms of how these findings may be transferred beyond the study, and the impact of the COVID-19 University lockdowns on the completion of the thesis.

Chapter 2

LITERATURE REVIEW

2.0 INTRODUCTION

Chapter 2 provides a selected literature review that defines the context of motor vehicle electronic control technical systems. A review follows that summarizes published vulnerabilities and attacks on these systems. Then a review of literature is made that concerns how motor vehicle electronic control systems are protected and the latest ways of providing security for the information. The Chapter is structured as follows:

- Motor vehicle electronic technologies defined
- Attacks on controller area networks
- Security provisions
- Security Frameworks
- Issues and problems arising

2.1 MOTOR VEHICLE TECHNOLOGIES

Motor vehicle technologies started with mechanical engineering to provide the physical movement and control of a vehicle, and then applied the evolving electrical technologies to other features such as lights and horns. Today the field of electrical automotive technologies has been transformed by electronic technologies and the most recent artificial intelligence applications (Seung, 2014; Shalev, 2017). This thesis only considers the networks for automated features in the current motor vehicle technologies. At the centre of current design is the controller area network (CAN) technologies that provide the communication system for bus, actuator, and sensor technologies to function semi-autonomously. The modern motor vehicle has the features for an Internet of Things (IoT) functionality for connectivity, control and experience (van Roermund, 2019). In this thesis the aim is to identify security vulnerabilities in these systems and propose countermeasures to assure the safety and privacy of the people using motor vehicles equipped with these (or some of) these evolving technologies.

Controller Area Network (CAN)/Controller Area Network with Flexible Data Rate (CAN FD) is the most commonly used network in the automotive network market

(Guo, 2011; CSS Electronics, 2018). High-speed CAN bus maximum speed can exceed 1 Mbps. The length of the CAN frame in data field is 8 bytes. In addition a local Interconnect Network (LIN) is used primarily to power vehicle seats, doors, wipers, sunroof etc. LIN speed can be as much as 20 kbps. The maximum amount of a data field in a frame for LIN bus is 8 bytes. Like CAN bus, LIN bus uses the communication mode design of the master / slave pair (Qiang Hu &Feng Luo, 2018). A FlexRay bus has a higher transfer speed than CAN bus and is primarily used as the control system. The max speed of FlexRay can be up to 10 Mbps. Multiple Access Time Division (TDMA) and Multiple Access Flexible Time Division (FTDMA) are used in FlexRay to maintain real-time network communication. The FlexRay system data field length is 254 bytes. The Ethernet plays an important role in the modern E / E system of automobiles. The Ethernet is the basis for the Internet protocol, and remote configuration, remote diagnosis and remote updating between vehicle and server can be done with the connectivity. Transmission speed for the automotive Ethernet can be 100/1000 Mbps, and the data field length of the Ethernet frame more than a thousand bytes (CSS Electronics, 2018). In addition the vehicle is required to connect and communicate inside and outside of the vehicle. The common communication features are vehicle-to-X (V2X), such as, Vehicle-to-Mobile (V2M) technology that uniquely combines wireless and cellular networks to enable intelligent transport system applications. V2M technology allows vehicles to interact among themselves. V2V is also classified as an ad hoc vehicle network (VANET). It also is a mobile ad hoc network modification (MANET), and helps drivers resolve blind spots, prevent accidents and other safety factors for dangerous situations (Morales, 2008). Vehicle to infrastructure (V2I) is a term in which vehicles and road infrastructure share safety and operational information, such as smart traffic signals.

Since the first invention of the motor vehicle, security has been concern and priority because vehicles are essential for everyday life and some people want to take what is not their right (van Roermund, 2019). There have been many security features, such as the removable steering wheel (1900), locks and doors (1920), alarms (1913), immobilizers (1918), and tracking devices (1960). Although today's security systems are better equipped to prevent theft than in the past, there are still weak points that criminals can exploit within the security systems. Looking to the future, research is currently being conducted to examine the weak points of the current security systems and brainstorming methods for improvements. Although there are many anti-theft technologies that can used to obtain the upmost security, there is always room for improvement. Researchers are looking for improvement in ECU systems (Engine Control Units), and the other communication functions that control series of actuators on an internal combustion engine (Johansson, 2005). With the threat of computer hackers using the external WIFI, cellular networks, and computer programs to disrupt the ECU's firmware and hardware, researchers are developing ways to counteract these threats by securing the boot authentication firmware components during boot and a virtual technology or directed I/O to help isolate and restrict execution environments to hardware (Cui, 2019). With the future of vehicle security heading to the area of computer networks and online data, the future improvements must be focused on ways to prevent system and network breaches.

2.1.1 CANBUS

The Controller Area Network (CAN) is a communications protocol developed by Bosch in the early 1980s for vehicle control. It defines a standard for efficient and reliable communication between sensors, actuators, controllers and other nodes in real-time applications (CSS Electronics, 2018). CAN is the standard for a wide variety of networked embedded control systems (Murtra, 2010). The ISO11898-1 standard, CAN specifies the Open Systems Interconnection (OSI) model's data link and physical layer, offering a low-level networking solution for in-vehicle high-speed communications. CAN was particularly designed to minimize cable wiring so that separate electronic control units (ECUs) inside a vehicle could communicate with only one pair of wires (Figure 2.1).

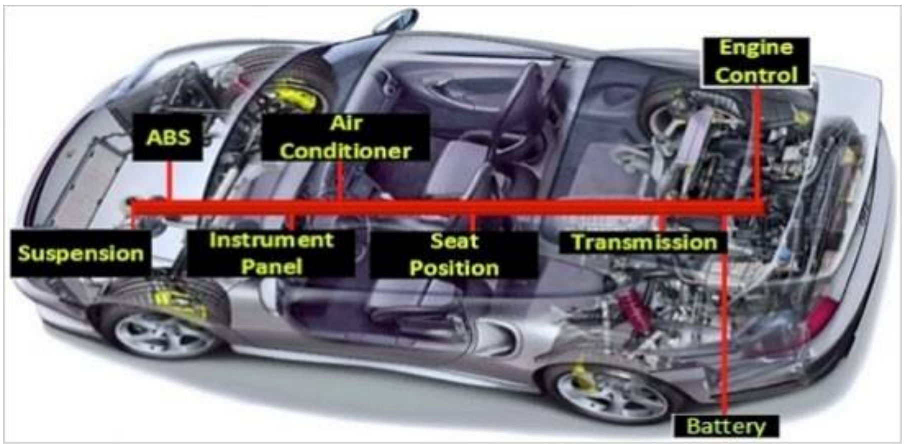


Figure 2.1: CAN BUS (CSS Electronics, 2018)

The CAN bus is mainly used in embedded systems and as its name implies, is a network technology that provides fast communication between microcontrollers for real time requirements, removing the need for a much more expensive and complex dual port RAM technologies (CANcrypt, 2017). CAN is a two-wire, half-duplex, high-speed network system that is far superior to traditional serial technologies such as RS232 in terms of functionality and performance, and the implementation of CAN is more cost-effective. Although TCP / IP is designed for the transport of large amounts of data, CAN is designed for real-time requirements and with a baud rate of 1 mbit / sec it can be easily beaten.

According to CSS Electronics (2018) standardization has changed the effectiveness and efficiency of the systems. The Pre CAN, Motor vehicle ECUs relied on complicated point-to-point wiring. In 1986, Bosch created the CAN protocol as a solution for Pre CAN. The year 1991, the same developer Bosch, published CAN 2.0 standard which has CAN 2.0A with 11 bits and CAN 2.0B with 29 bits. But in 1993, CAN was introduced as an international standard (ISO 11898) and in 2003, ISO 11898 became a standard series such as 11898-1 and 11898-2. Bosch launched a new version of CAN in 2012 known as CAN FD 1.0 with a flexible data rate. In 2015, the CAN FD protocol was standardized, which allows for more frequent sampling of bits in the data frame and CRC sectors (Guo, 2011). The CAN FD also specifies a data field of up to 64 bytes in size and a smaller CRC field in result [H+12]. The higher data rate requires shorter bits in CAN which makes the new length of 64 bytes have smaller overheads.

2.1.1.1 CAN bus Message Frames

The original ISO specification set out what is referred to as Specification CAN. Standard CAN (Figure 2.2) uses an 11-bit identifier for multiple messages, resulting in a total of 2048 different message IDs. CAN was later modified, expanding the identifier to 29 bits, giving 2 to the power 29 identifiers. This is called Extended CAN (Figure 2.3). CAN uses a multi-master bus that contains all messages. CAN uses a two-state, recessive and dominant, differential signal. Recessive means that the voltage differential is less than a minimum voltage limit. Dominant shows that the voltage difference is greater than this minimum threshold. However, the dominant state is accomplished by moving a 0 logic on the serial bus, while a 1 logic maintains the recessive condition (Zhou, 2006).

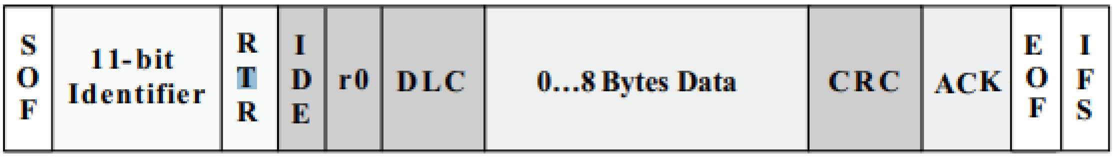


Figure 2.2: Standard CAN 11 bits

SOF is first bit is the start of the frame, this dominant bit is the start of the CAN message (Guo, 2011). The next is the 11 bits identifier, which sets the priority of the CAN message. The smaller the identifier, the higher the priority of the message. RTR (remote transmission request) bit is usually dominant, but it is recessive if one node requests information from another node. The IDE bit (Identifier Extension) is dominant when a regular CAN frame is sent, not an extended one. The r0 (reserved) bit is and is not actually used. The DLC (data length code has 4 bits) nibble is the number of bytes of information in this message. Next comes the data itself, up to 64 bits, being as many bytes as represented in the DLC bits. The Cyclic Redundancy Check (CRC) is a 16 bit checksum for the detection of errors in transmitted information. If the message is received correctly, the receiving node overwrites the Recessive Acknowledge Bit (ACK) with a dominant bit. The ACK also has a delimiter bit to keep things synchronized. EOF (End of frame) has 7 bits values that mark the end of CAN frame or message (Figure 2.3). The last frame is IFS (Inter-frame space) has 7 bits values that includes the times required by the controller to shift the correctly received frame to its correct position in a message buffer area (CSS Electronics, 2018).

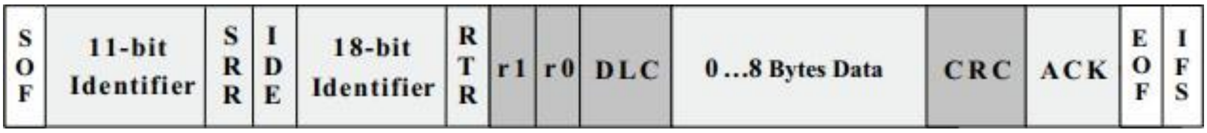


Figure 2.3: Extended CAN 29 bits

Extended CAN uses a 29-bit ID and additional bits (Guo, 2011). An extended message has a substitute remote request (SRR) bit after an 11bit identifier, which serves as a replacement to preserve the same structure as the generic CAN. Every time the Identifier Extension (IDE) should be recessive, meaning that it is preceded by the modified

Identifier. RTR bit is accompanied by an 18-bit ID and r1 is the second reserve bits. The rest of frame message remain the same as CAN standard.

2.1.1.2 Components of the CAN bus and Signals

CAN-L (Low) and CAN-H (CAN Low) are two wires and that form the CAN bus and it is connected to the network devices of a vehicle. The signal of CAN-L and CAN-H have the same data sequence, but they have opposite amplitudes (Figure 2.4). Therefore if the pulse on CAN-H line goes from 2.5V to 3.75V then the equivalent pulse on the on the CAN-L line goes from 2.5V to 1.25V as the opposed to CAN-H. By sending the data in equal and contrary ways, from this it allows for greater noise tolerance and thus less risk of corruption of the data. (CSS Electronics, 2018.)

CAN Signal

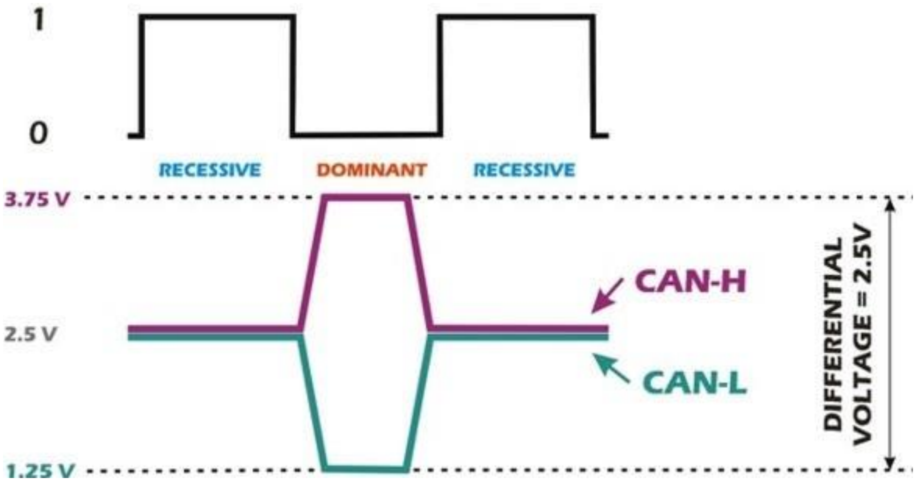


Figure 2.4: CAN-L and CAN-H Output Signal (CSS Electronics, 2018)

In the Figure 2.4 the bit status with 0 value is equal to 2.5V and it has a dominant condition and the bit status with 1 value is equal to 0V and differential voltage is the recessive state. The CAN bus components are three that include CAN Controller, CAN Transceiver and CAN Data Bus Terminal (Figure 2.5).

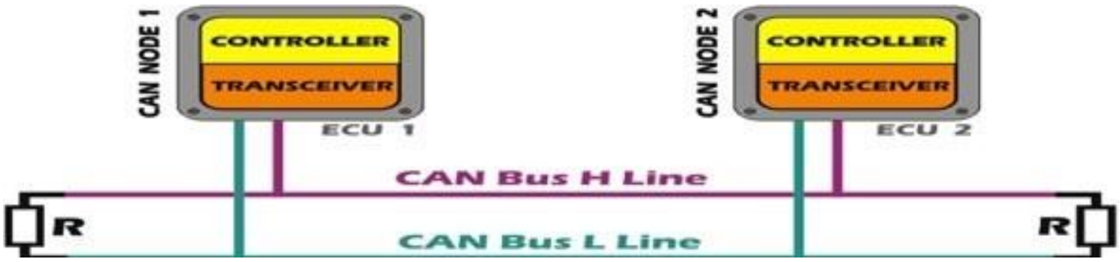


Figure 2.5: CAN Bus Components (CSS Electronics, 2018)

The CAN Controller receives data transfer from the integrated microcomputer in the control unit or CAN node. CAN controller processes and relays the data to a CAN transceiver and it also receives, processes and relays data from the CAN transceiver of the microcomputer embedded into the CAN node or Control Unit. CAN Transceiver is a single transmitter and receiver and it transforms the data supplied by the CAN controller into electrical signals and transmits those data over the bus line. The CAN Transceiver also receives data and transfers the data to the controller for CAN. The last component is CAN Data Bus Terminal is a resistor usually of 120 ohms. This helps to prevent the data sent from being replicated at the ends and returning as an echo (Guo, 2011).

2.1.1.3 The Process of CAN Data Transfer and Message Structure

The CAN data transfer involves the following stages that include data supply, sending data, receive data, checking data and data acceptance (Figure 2.6).

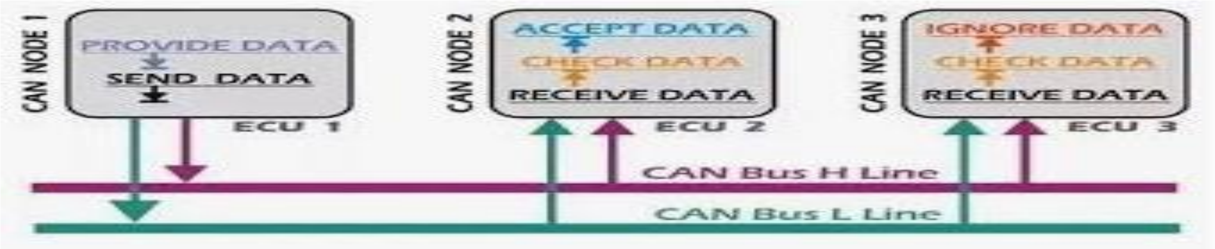


Figure 2.6: Process of CAN Bus Data Transfer (CSS Electronics, 2018)

The first stage is Supplying Data, when the CAN node supplying the data to the CAN controller is transferred. The second stage is Sending Data, The process of when the CAN transceiver gets the data from the CAN controller and transforms it into electrical signals and then transmits it back to the network. The Receiving Data stage, is when all the CAN nodes that are connected to the CAN data bus will become receivers. The fourth stage is

Checking Data, and it refers to when the CAN node checks or tests whether or not it needs the data it got for its functions. The last stage is Accept Data and it refers to accepted and processed when the data obtained is relevant. Otherwise the data obtained would be ignored (Guo. 2011).

2.1.2 Automation

Manufacturers build vehicle technology with the function of automation with the purpose that the vehicles or machine is able to control or drive itself without interacting with the human driver or is driverless (Zulueta, 2013; Silver, 2018). SAE (Society of Automobile Engineers) lays out what has become the most internationally recognised standards for driving automation which categorizes six levels of driving automation. These are dependent on the degree of human involvement in driving the motor vehicle. These levels are specified as (SAE):

- Level 0: No Automation. This is the lowest level of automation. The driver is in control of the car from the start to the end of the journey. Driver support services are limited to warnings and prompt assistance. Types include automatic emergency braking and warning of lane departure.
- Level 1: Driver Assistance. The driver is in control of the car from the start to the end of the journey. The partially automated driving support system is limited to support for brakes or acceleration, but not both. Examples of this include lane centring or adaptive cruise control.
- Level 2: Partial Automation. The driver is in control of the car from the start to the end of the journey. The partially automated driving support system is limited to both brake and acceleration support. Examples include lane centring at the same time as adaptive cruise control.
- Level 3: Conditional Automation. The driver is not responsible for driving the car when the specified automated driving features are used, but the driver must take responsibility for driving when the features allow it. Types include a traffic avoidance and positioning.
- Level 4: high automation. The driver is not responsible for driving the car when the specified automated driving features are enabled and the driver is not expected to take over the driving when the automated driving features are in use, but the

automated driving features can only work under certain conditions and will not engage unless all these conditions are met. When they are met, the level 4 vehicle is driven independently. Pedals and steering wheels are not always found in level 4 vehicles. For example, a local driverless taxi.

- Level 5: Full Automation, the driver is not responsible for driving the car when the automated driving features specified are engaged and the driver is not expected to take over the driving when the automated driving features are in use. Automated driving features can be used everywhere under all conditions. The Level 5 car is one hundred percent without a driver. It does not require a driver at any time.

2.1.2.1 Sensors

Autonomous vehicles are functional based on the sensors which make it possible for the motor vehicle to see and sense everything on the road, as well as to gather the data needed to drive safely (Shiomi, 2014; Silver, 2018). This information is processed and analysed in order to construct a path from point A to point B and to send appropriate instructions to the vehicle controls, such as steering, acceleration and braking. In addition, the information collected with the sensors, including the actual route or path, road barriers and traffic jams. The main sensors used by the autonomous vehicle are the Camera or Image sensor, Radar sensor and LIDAR sensor (Figure 2.7).

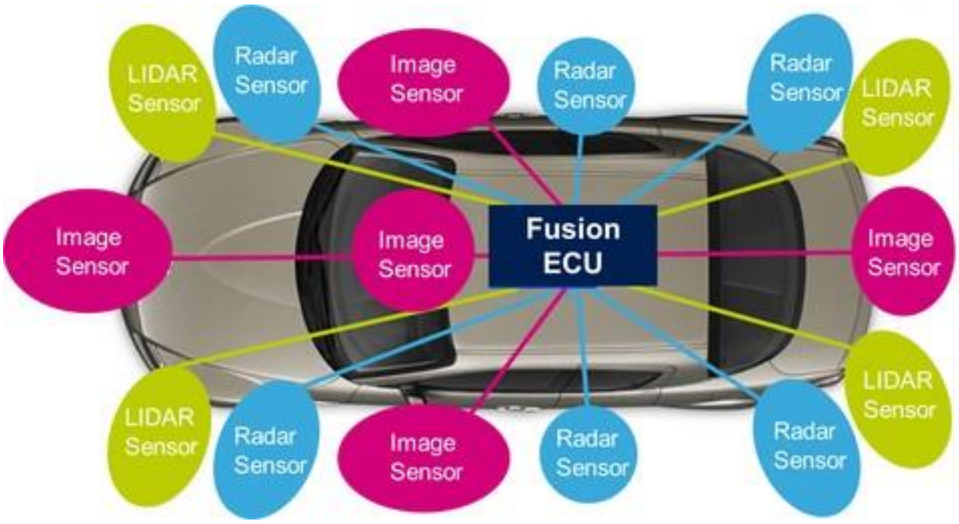


Figure 2.7: Sensors in the autonomous vehicle (CSS Electronics, 2018)

Autonomous cars (AV) have video cameras to view and interpret objects on the road just as human drivers do with their eyes (Mutra, 2010; Navarro, 2017). By equipping cars with cameras at all angles, the vehicles are able to maintain a view of the external environment and provide a broader picture of the traffic conditions around them. Today, there are different types of Camera sensors installed in Autonomous vehicles to detect the environment, for example: Rear cameras, 360° Cameras, and 3D cameras. These are available for viewing highly detailed, realistic objects. Image sensors automatically detect, classify, and measure distance to objects to recognize other vehicles, pedestrians, bikes, road signs, signals, road markings, bridges, and guardrails (Paris, 2009).

Radar (Radio Detection and Ranging) sensors make a vital contribution to the overall function of autonomous driving. Radar systems use radio waves to detect objects and measure their distance and speed in real time. Short-and long-range radar sensors are usually deployed around the car and have different functions. Although short-range (24 GHz) radar applications allow blind spot monitoring, lane support and parking aids, the role of long-range (77 GHz) radar sensors includes automatic distance control and brake assistance. Like cameras, radar systems typically have no trouble identifying objects during fog or rain (He, 2017).

Lidar (Light Detection and Range) sensors is a laser based system. It works similar to radar systems, with the only difference being that they use lasers instead of radio waves. They measure distances to various objects on the route (Tome, 2016) and produce 3D images of the objects detected and the mapping of the surroundings. A collection of Lidar, is coupled and combined with a rapidly spinning mirror to achieve a 3D simulation of the environment.In addition, Lidar can be configured to create a full 360-degree map around the vehicle rather than relying on a narrow field of view. It can recognize the objects up to 300 meters away.

Video images provide the majority of the details for the human driver but are also suitable as a highly automated driving input parameters (Wade, 2018). The rear and 360° cameras support the driver with a better image of the environment outside the vehicle. The two-dimensional cameras are widely available to display images and sometimes superimpose additional information on the display, such as the angle of the steering wheel. Most motor vehicle manufactures are beginning to install virtual cameras for three-dimensional image displays for drivers (Wang, 2014). In order to be realistic, a three-dimensional image usually requires input signals from four to six cameras and special attention must be paid to image stitching to avoid loss of image information. All 2-D and 3-D cameras need very high dynamic range image sensors greater than 130 dB.

A dynamic range is necessary for direct sunlight reflecting in the lens, to produce a clear image. The best image sensors have a dynamic range of 145 dB and a 24-bit deep interface to the ISP (image signal processor). This dynamic range is well beyond what traditional lens systems can deliver.

Automotive rear and 360° camera systems used today typically have a standardized architecture (Figure 2.8). This means that the central control unit processes the raw data from four to six cameras. Since the processing is done in software, the processor is faced with tough requirements. Additional features are necessary for the specific hardware use that can cause a high-power loss in such a system, for example modern methods of data compression often demand large processing and storage capacity (Wong, 2016).

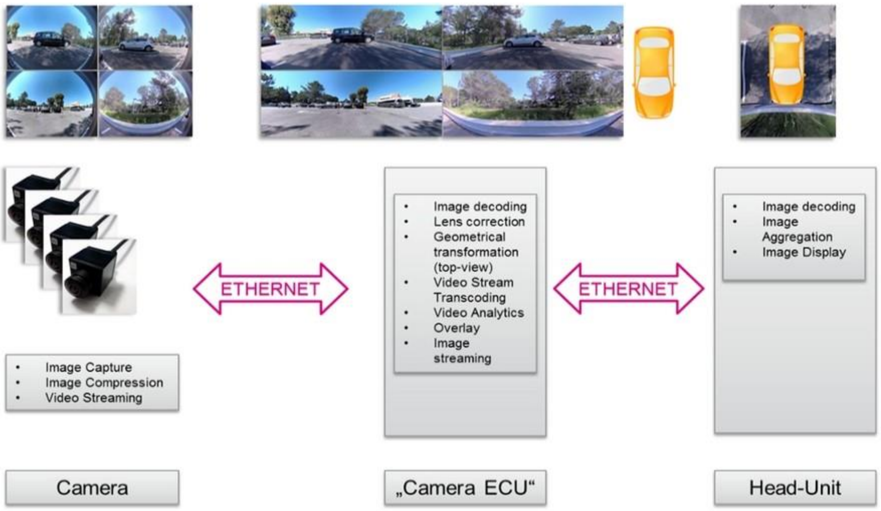


Figure 2.8: Image processing (CSS Electronics, 2018)

Camera systems receive raw data and then processed and transmitted to the image display unit. The distributed approach completely eliminates camera control units, leaving only the smart camera and head unit. It has two processing steps inside the camera as well as inside the central camera control unit for further clarification. The image is processed in the first stage (within the smart camera) and the geometric transformations such as the fish-eye EQ, over layer, and image compression, as well as the encoding and streaming of the Ethernet are completed. The second stage (within the central camera module) then takes over the encoding of the film, the intermediate storage as well as the display of the image (Figure 2.9).

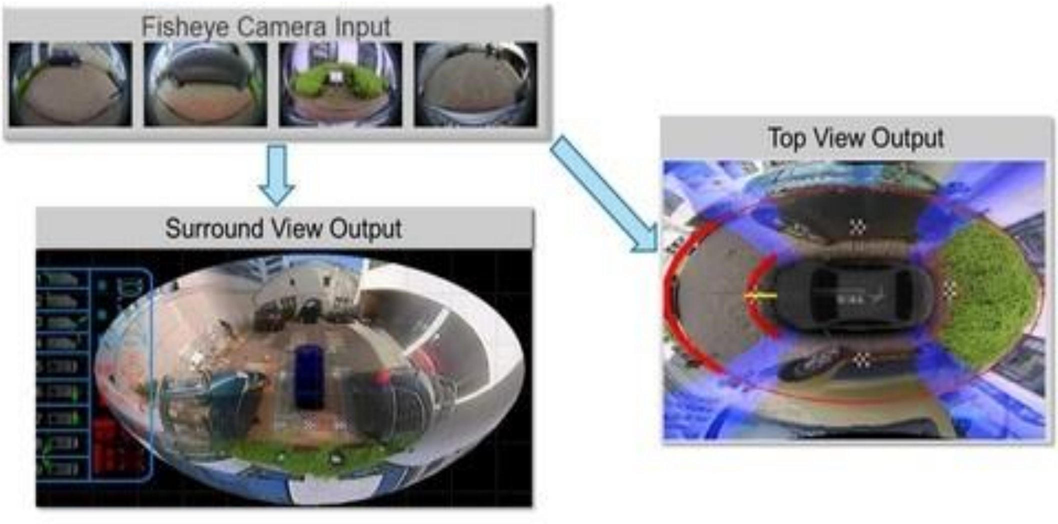
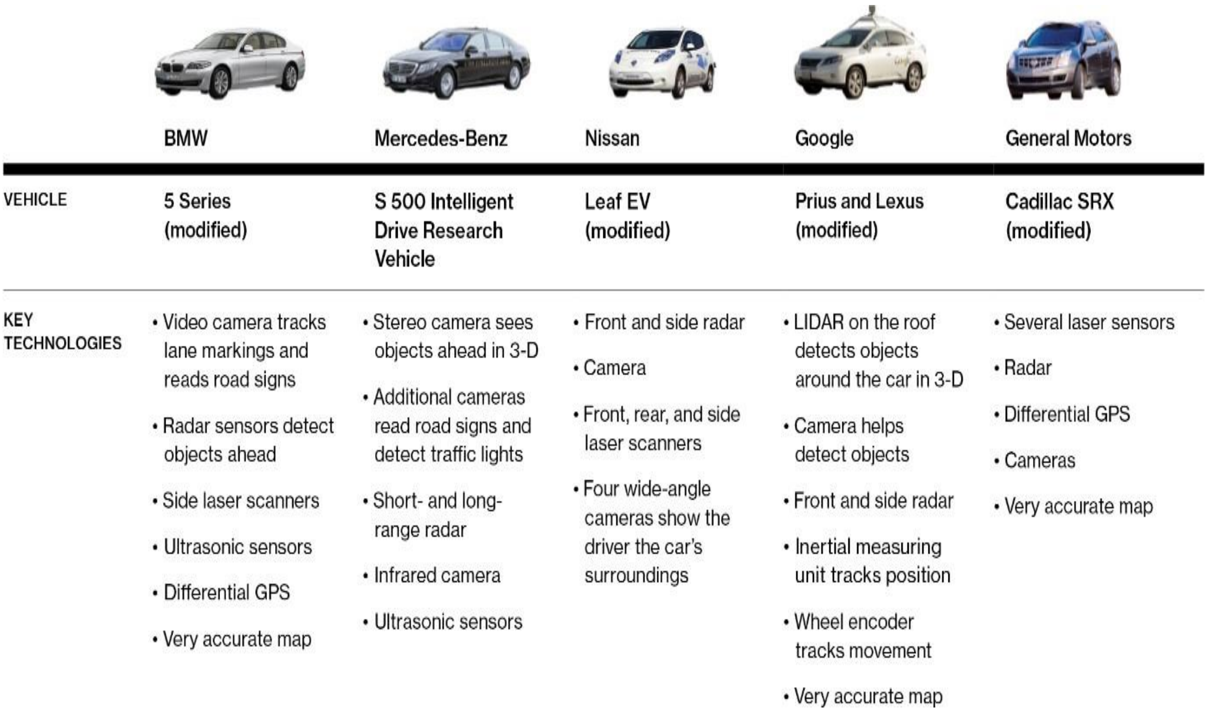


Figure 2.9: The camera views (CSS Electronics, 2018)

Forward Facing Camera Systems are medium to high range systems, such as those between 100 and 275 meters. These cameras use algorithms to automatically detect, classify and distance objects.

Variation of different autonomous vehicle Technology



	BMW	Mercedes-Benz	Nissan	Google	General Motors
VEHICLE	5 Series (modified)	S 500 Intelligent Drive Research Vehicle	Leaf EV (modified)	Prius and Lexus (modified)	Cadillac SRX (modified)
KEY TECHNOLOGIES	• Video camera tracks lane markings and reads road signs • Radar sensors detect objects ahead • Side laser scanners • Ultrasonic sensors • Differential GPS • Very accurate map	• Stereo camera sees objects ahead in 3-D • Additional cameras read road signs and detect traffic lights • Short- and long-range radar • Infrared camera • Ultrasonic sensors	• Front and side radar • Camera • Front, rear, and side laser scanners • Four wide-angle cameras show the driver the car's surroundings	• LIDAR on the roof detects objects around the car in 3-D • Camera helps detect objects • Front and side radar • Inertial measuring unit tracks position • Wheel encoder tracks movement • Very accurate map	• Several laser sensors • Radar • Differential GPS • Cameras • Very accurate map

Figure 2.10: Variation of different autonomous vehicle technologies (Auto, 2016)

The cameras can distinguish pedestrians and cyclists, motor vehicles, side lines, bridge abutments and margins of the road (Tome, 2016). The detecting of traffic signs and signals are also used the same algorithms. Medium-range cameras basically provide warning to the drivers about the cross-traffic, pedestrians, emergency braking in the car ahead, and at the same time identify lane and signal lights. High-range cameras are used for traffic sign identification, video-based distance control and road guidance. The difference between medium and high range cameras is the lens aperture angle or field of view (FoV). A horizontal view of 70° to 120° is used for medium range systems, while cameras with a wide range of apertures use horizontal angles of around 35°. Figure 2.10 shows a summary of different automotive technologies that are used in different autonomous vehicles.

2.1.2.2 Augmented Reality

Augmented Reality (AR) makes things that exist larger, such as in size, scope or quantity, by using digital information to increase human perception of actual, physical reality (Wong, 2016). Augmented Reality enhances digital images, audio recordings, and mixed realities to make reality larger, better, and more accessible to viewers. Decisions are made in such circumstances, perceptions are shaped, and actions are taken. Humans gain knowledge of real situations based on the transformation of the real into the construction and understanding of the illusion (Wang, 2014). In an autonomous vehicle, AR are navigational aids that combine representations of real world objects and relationships with digital information to make it easier for the driver to navigate the vehicle. These also allow the driver to enter real-world locations and circumstances which have never been seen or encountered before and to operate safely. These include showing driving directions over the field of view of a driver, improving protection by highlighting roads during foggy weather or calling attention to road signs (Welch, 2018). The application (overlay) of details to photographs is a long-established practice in the advertising industry (Shengli, 2018). AR is used for the autonomous vehicle technology, in way of ease and safety for transport. But at the same times AR brings the challenge of how does a human being perceive things to be? What kind of reactions do they have? How has their decision-making changed?

The benefit of AR in Autonomous Vehicles (AV) technology is for making decisions, but when it is not working properly it causes problems (Xiaoli, 2017). The National Highway Traffic Safety Administration (NHTSA) AV regulation provides a framework for service, monitoring and protection evaluation, while vendors are not required to follow the requirements of the AV legislation. The policies guidelines were designed for the Society of Automotive Engineers (SAE) level 3 to 5 of driving automation. The overarching statements in the AV regulation, however, discuss the degree of driver involvement, pointing out that there is a clear distinction between the upper three rates of the SAE classification and the lower three levels, which are focused on the automation system's human driver reliance when engaged. It ensures that there must be a human operator who is willing to take over the automation system as and when necessary, and who is actively monitoring the driving environment (NZTA, 2015). A driver may not be fully aware of the system limitations and have overreliance on the system automation capabilities.

In the Tesla crash case, the NTSB report indicates that the vehicle did not detect the truck and the trailer unit that was turning in front of the vehicle at any stage (Delhi, 2016; Cui, 2019). The sensors, or the calculation of the data received from the sensors, did not determine that the truck was either a moving hazard or a stationary object. This is confirmed by the fact that there was no effort on the part of the Autopilot to activate the Automatic Emergency Breaking feature, or to decelerate the speed before the vehicle hit the truck. The car did not detect the impact with the truck, as the airbags were not applied at this point but were deployed after the vehicle had driven about 300 m and collided with a utility pole afterwards. The Forward Collision Warning system was therefore not activated at any stage before the vehicle collided with the truck and the trailer unit and travelled at the set automatic velocity at the time of impact (Rahman, 2017).

Most applications use wireless communications in automotive systems, both within the vehicle and between the vehicle and inter-vehicle communications (Song, 2016). Vehicle communications allow more mobile devices, e.g. mobile phones, mobile GSM devices and laptop computers, to make use of the possibility of vehicle interconnection. Several new applications will also take advantage of the possibility of inter-vehicle communications, e.g. vehicle-to-vehicle communications. Bluetooth v2.0 supports up to 3 Mbps of network speed. Originally designed for the implementation of a Personal Area Network (PAN) for short-range, low power ad hoc wireless interconnection. Today’s Bluetooth technology has improved and rapidly become used in automotive vehicle communication networking technology (Guo, 2011). The Bluetooth Special Interest Group (SIG) founded the Car Working Group in December 1999. The Hands-Free model was the first of many application level requirements planned by the Car Working Group. They use the new hands Free profile products that adopt the Bluetooth requirements to facilitate the automatic direct link between the car's hands-free system as part of its audio system and the mobile phone communications.

The ZigBee 802.15.4 standard is the new low cost and low-power wireless Personal Area Network (PAN) standard designed to meet the needs of sensors and control devices (Roderick, 2016). Usually ZigBee applications are monitoring and controlling applications that do not require high bandwidth, but impose high latency and energy consumption requirements. Regardless of the number of proprietary systems with low data rates designed to meet the requirements, the use of such legacy systems presented major interoperability problems solved by ZigBee technology, offering a standardised base of sensor and control systems solutions. The network speed provided by ZigBee is up to 250 Kbps and is intended to be used as a sensor network for monitoring and control purposes in the automotive industry such as air conditioning, heating, ventilation, lighting control, (Nolte, Hansson & Bello, 2005). Wi-Fi technology has wireless fidelity and is the general term for any 802.11 standard network type. Inter-vehicle communications use Wi-Fi for communication networks, for example the Car2Car Consortium.

2.2 ATTACKS ON CAN

Hacking is a very real risk to vehicle safety as evidenced by the growing number of cyberattacks on systems and data (Petit, 2015; Petit et al., 2015). Hacking means that opponents target trusted security controls as a means of facilitating cyber-attacks. Hacking opponents (hackers) can simultaneously threaten information security on multiple levels. According to Yan (2016), hacker attacks impact essential business processes, which include compromise and data source redirection, and using websites that closely imitate institutions to obtain authentication information used in later fraud. They use strategies that are also used to hack autonomous vehicles. First is Direct Physical Access: The hacker aims at the target from an intrusion point. It is the easiest hacking attack technique, with no intermediate or third-party hosts involved except for normal traffic routing. This technique is used in cyber-attacks where the point of intrusion is of little interest to the hacker or there is very low probability of backtracking. Second is progressive access: The hacking opponent uses a series of intermediate hosts between the point of intrusion and the target, each of which is infected by the same set of exploits. The rest of this section reviews attacks on autonomous vehicles.

2.2.1 Spoofing Attack

Because of their lack of encryption, authentication, access control and message verification schema, CAN bus is vulnerable to attack. In order to spoof and exploit the vehicle, attackers can first analyse CAN messages, then master the instructions found in CAN frames and then repeatedly send messages as a flooding attack (Groll & Ruland, 2009). They can also easily connect to the CAN bus via the on-board infotainment system interface. Remote cellular communication devices also have vulnerabilities which allow access to CAN buses (Nguyen et al, 2018). Therefore quick identification of the abnormal ECUs when the in-vehicle module is under attack, is critical. The authors have found that the voltage waveform on the CAN bus is inherently difficult for adversaries to fake in malicious attacks. The messages sent by various ECUs have their own voltage waveforms, due to the hardware disparity between different modules. With this information senders from the outside can be detected.

2.2.2 Bluetooth Attack

Cardenas et al. (2011) claims Bluetooth based networks can be used to significantly increase the eavesdropping and scanning attack surface. Due to the complexity of its protocol and the underlying data, Bluetooth is considered one of the largest and most viable cyber-attack surfaces in modern vehicles. According to (Cardenas et al, 2008) Bluetooth has become omnipresent in the vehicle domain, offering cyber attackers a very secure point of intrusion to test attack scenarios. Bluetooth features integrated into the telematics systems of test vehicles were investigated. Reverse engineering gained access to the UNIX-like operating system of the telematics ECU and the software programmed responsible for handling Bluetooth functionality was established. It was verified that the ECU operating system included a duplicate of the popular built-in implementation of the Bluetooth protocol stack along with a sample hands-free application and a custom built interface. The interface provided a vulnerability that enabled any paired Bluetooth device to mount buffer overflow attacks and allowed arbitrary code to be executed on the telematics unit. By submitting carefully crafted data to an application, a cyber-attacker may trigger the application to execute arbitrary code and possibly take over the functionality of the mission critical cyber-physical system. Buffer overflow attacks are generally based on two methods, often in combination, that tricks the operating system to mishandle data types and write data to specific memory addresses.

2.2.3 Sensor and Cameras Attack

Petit & Shladover, (2015) show it can be easily achieved to attack AV sensors such as ultrasonic sensors and cameras. It is possible to attack millimetre wave radars but it is less feasible to attack the other sensors. Typically, Ultrasonic sensors are used for parking assistance. These sensors track the front and rear of the car, and alerts the driver if in the vicinity of the vehicle there are hazards that can cause collisions. Control features such as semi-automatic car parking assistance, fully automatic parking and parking space detection. (Yan et al., 2016). Jamming is one attack which can be carried out on an ultrasonic sensor systems, making measurements impossible for the sensor (Brewster, 2016). The purpose of the jamming attack is to generate ultrasonic noise and cause continuous membrane vibration on the sensor which makes the measurements impossible. Failure to detect obstacles can result in parking collisions or false manoeuvres. It is possible to place ultrasonic jammers on vehicles to make them invisible to autonomous systems, which could easily lead to a collision (Petit et al., 2015). The blinding attacks on cameras can be carried out using either lasers or LEDs. Blinding attacks cause cameras to be unable to see necessary information, such as road traffic lights and the road signs or vision-only system obstacles. Close-range laser attacks can cause permanent, irreversible damage to cameras, leading to a degraded system for random failures (Yan et al., 2016).

2.2.4 Wireless Attack

Koscher at el. (2010) showed eexperiments by remotely assaulting a car (2009 model year passenger car), without physical access beforehand. Through several elements they were able to compromise the vehicle. They included diagnostic tool, media player, Bluetooth, and cellular communications. The advantage of an attacker penetrating through the wireless components is that they can wait to launch an attack and can wait for the best time to gain maximum benefit. This is for control of radio, instrument panel cluster, body controller, engine, brakes, heating and air conditioning.

Table 2.1: Summary of Attacks in the IN-VEHICLE NETWORK

year	Vehicle type	Foothold	Hack Model	Attack Impact
2010	2009 model passenger car	OBD 11	Physical & Remote with prior physical access	Took control the radio, instrument panel cluster, body controller, engine, bakes heating and air conditioning
2015	2014Jeep Cherokee	Entertainment System Cellular (Wi-Fi)	Remote	Controls the engine, brake, steering wheel. Control the air conditioning system, traumatic driving
2015	Renault Twizy 80	OVMS	Remote	Able to move the vehicle forward and backward, control speed of the vehicle, cause damage to the engine, change motor direction
2011	2011 model moderately priced sedan	OBD 11, Media player, Bluetooth and Cellular	Remote	Take the full control of the car system, surveillance and car theft

Table 2.1 has a summary of the automotive models successfully hacked. It describes the hack. the model, access, and the outcome of the hacker control.

2.3 SECURITY CONCERNS

From a security perspective, the typical automotive network architecture can be divided into four layers (Roderick, 2016). The layers are: Individual ECU, In-vehicle Network, Gateway, and Firewall as shown in Figure 2.11.

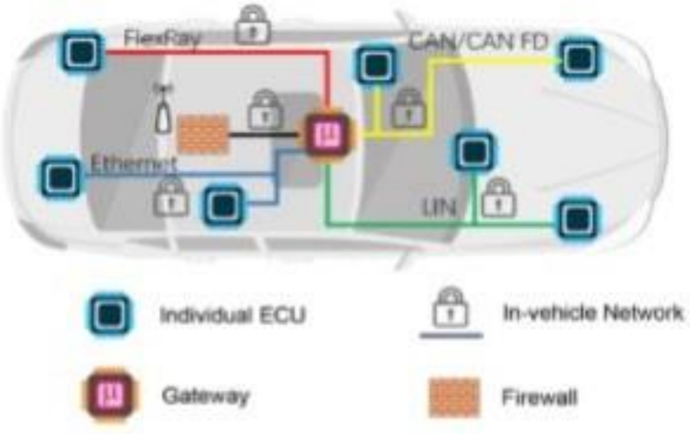


Figure 2.11: Automotive Network Architecture (CSS Electronics, 2018)

The Individual ECU Layer provides software-trusted execution and data protection, with the hardware basis for the security mechanism for the upper layers. The layers of In-vehicle Network provides mechanisms related to cryptography to be used to encrypt the data transfer within the network. The Gateway layer has the critical security functions, including access control and intrusion detection. This layer helps the data exchange in different network domains. The Firewall Layer is used to shield the main vehicle connectivity protocols, such as the OBD-II port, the V2X on-board monitor and the infotainment system (Stanley, 2002). Information security and industry standards promote automotive information security requirements to secure an automotive network. It includes the integrity of firmware, integrity of communications, data credibility, availability, and protection of intellectual property. According to Ruddle, Ward, Idree and Roudier (2009) there are eight security requirements for an automotive network. These are: Data origin authenticity, Integrity, Access control, Freshness, Non-repudiation, Privacy/anonymity, Confidentiality and Availability (Table 2.2).

Table 2.2: Security Requirement for automotive networks

Security Requirement	Description
Data origin authenticity	The basis of the data is reliability and authenticity.
Integrity	No changes are made to the data when transferring.
Access control	Authorization before information is accessed.
Freshness	Timing of related message information.
Non-repudiation	The actions of the entity are undeniable.
Privacy/anonymity	Information to an entity is confidential (secret).
Confidentiality	Only authorize entities are able to obtain the information.
Availability	Services provided shall be operational.

2.3.1 Encryption

One of the CAN protocol's most fundamental flaws is a lack of confidentiality in the post. As in the concept of "security through obscurity," automakers rely solely on a proprietary message format that is not known to the public as a means for safety. But as has been seen above, these proprietary CAN messages can be deciphered to reveal their purpose, and then CAN messages can be changed or replayed for malicious purpose (CANcrypt, 2017). To avoid CAN recognition the most effective way is to add some form of encryption to the CAN protocol. A major disadvantage facing CAN encryption is the fixed data field size of 8 bytes of the CAN protocol. According to Security Innovation (2015, p. 1), "CAN is an old technology with limited data streams, it is not possible to use any meaningful size encryption" (Yoshida, 2015). There is broad consensus that a strong encryption algorithm requires a 128 or 256 bit block size, but this has not prevented many separate researchers and security firms from offering CAN encryption solutions.

Yoshida, (2015) had an approach for encrypting CAN messages using Trillium's SecureCAN, a small Japanese application. SecureCAN is designed for 8 Bytes or less payloads, such as those found in the CAN bus, and supports variable block size and key length. The cryptography used in SecureCAN uses three different algorithms: a message is substituted first; the resulting cipher text goes through a transposition algorithm; finally, time-multiplexing is applied before transmitting the cipher text. Trillium believes that the entire encryption, transmission, and decryption process can be completed in less than a millisecond. This falls within the time limit required for CAN bus applications in real-time for automotive use. According to Yoshida, (2015) SecureCAN also uses a new key management system called' Dynamic Key-Lock Pairing. With this solution, each time a car's ignition is switched on, a new shared master key is created. Furthermore, SecureCAN uses frequency channel hopping to adjust the cipher text at random intervals, potentially multiple times per second.

2.3.2 Device Authorization

A key element in preventing an attacker from sending malicious messages on a CAN bus is to allow system authentication or authorization (Stanley, 2002). When the attacker uses devices that should have no legitimate reason to transmit data on the CAN bus, the transmitting CAN controller must be able to validate that the message originates from a legitimate source to prevent unauthorized device or rogue CAN controllers from broadcasting CAN messages. The authorization of CAN devices can be achieved by preprogramming CAN controllers for known good devices with a whitelist of CAN identifications. For example, the steering controller of the vehicle should only know how to trust controls from the controller associated with the steering wheel of the vehicle–and not from any external source. The attacker can spoof CAN identifiers, whereby messages are inserted or changed to make them appear as if they were originating from a legitimate source. Hence, the CAN identifier field must be encrypted for system authorization to work effectively. The CAN data field encryption prevents an attacker from being able to decipher a message, and with the introduction of the CAN identifier field encryption, an attacker cannot spoof an activated CAN system.

In 2011, Richards patent entitled “Secure Communications Between and Verification of Authorized CAN Devices” offers a solution for the authorization of CAN devices. This solution uses a unique encryption code stored in each of the authorized CAN bus devices in order to prevent unauthorized CAN bus nodes from communicating with the authorized nodes. Unlike in the SecureCAN encryption solution by Trillium which encrypts the data field, the solution by Richards requires encryption of the identifier field. This is problematic since any alteration of the CAN identifier field of a data frame would cause the receiver CAN controller to ignore the message because it no longer recognizes the source. CAN identifier encryption involves the use of a hardware-based encryption solution between the sending and receiving by CAN controllers. Richards ' approach calls for the use of a pair of KEELOQ peripheral devices to serve as CAN endpoints for encryption and decryption. KEELOQ is a proprietary block cypher, based on hardware, developed by Microchip Technology Incorporated. Nonetheless, there are several potential downsides to this approach as it would add additional processing time to CAN transmissions, additional cost to manufacturers, and additional vehicle weight (Roderick Currie, 2016).

2.3.3 Security Development

Automotive security is a coordinated defense strategy to identify, secure, and correct potential threats, and to defend against previously unknown or inevitable threats. Such layers provide hardware-based security in and around ECUs in next-generation vehicles. Other layers provide hardware-based protection (security) within and around the ECU, vehicle defense software (software security), network monitoring and compliance inside and outside the car, and with sufficient privacy and confidentiality. Combine the functions of the hardware security and software security it must have the ability to protect AV operations and well as data and processes to secure autonomous vehicles (McAfee, 2016). Security-in-depth protection consists of three layers: security modules for hardware, hardware services and security services for applications. Hardware security is used to protect the ECU as an enabler and enforcer of security. Its primary responsibilities are secure boot to bring the environment to an initial trusted state, secure key storage, and a trusted execution environment. Hardware security services require a build on hardware security and to provide fast cryptographic quality, immutable device identification, authentication of messages and isolation of executions. Software security services improve security functionality in addition to network compliance equipment, whitelists / blacklists, identification of anomalies, cryptographic services, biometrics, safe over - the-air updates and upgrade capabilities, all provided throughout the life of the car (Ruder, 2016).

2.3.4 Hardware Security

Physical protection systems on a car engine are such as the firewall, seat belts, and airbags. These are there to avoid intentional or accidental harm to the working components. The computer security industry has a wide range of hardware security building blocks available which help secure the ECUs and buses (Guo, 2011). These include:

- Functions Secure Boot and Application Attestation: detects exploitation of boot loaders and vital OS files by testing their digital signatures and product keys. Before they can attack or corrupt the system, invalid files are blocked from running, thereby giving an ECU its trust base while operating.
- Trusted execution technology or the processor module is trusted: uses cryptographic techniques to create a unique identifier for each authorized component, allowing an accurate statement of the elements of a startup environment against known trusted sources, and stops the launch of code that does not fit.
- Cryptographic acceleration: Offloads encryption workloads to optimized hardware, enhancing cryptographic efficiency and making it easier for applications and communications processes to implement symmetric or public key encryption.

2.3.5 Software Security

In a vehicle there are many ECUs of different capabilities. The addition of hardware security capabilities to some of them is difficult or impossible, so cooperation processors and software-based security are also required. These include engineering techniques and technological technology that can secure the vehicle (McAfee, 2016), as follows:

- Safe boot works with the hardware to ensure the parameters of the loaded program are legitimate to give the rest of the system a root of confidence
- Partitioned operating systems are a widely used mixture of software and hardware that isolates various processes or functions, such as externally facing functions from those driving the vehicle, minimizing the difficulty of consolidating multiple

systems into one ECU. Software containers, allow individual functions to be upgraded or replaced without affecting overall operations, mirror functions for redundancy and fast fail-over

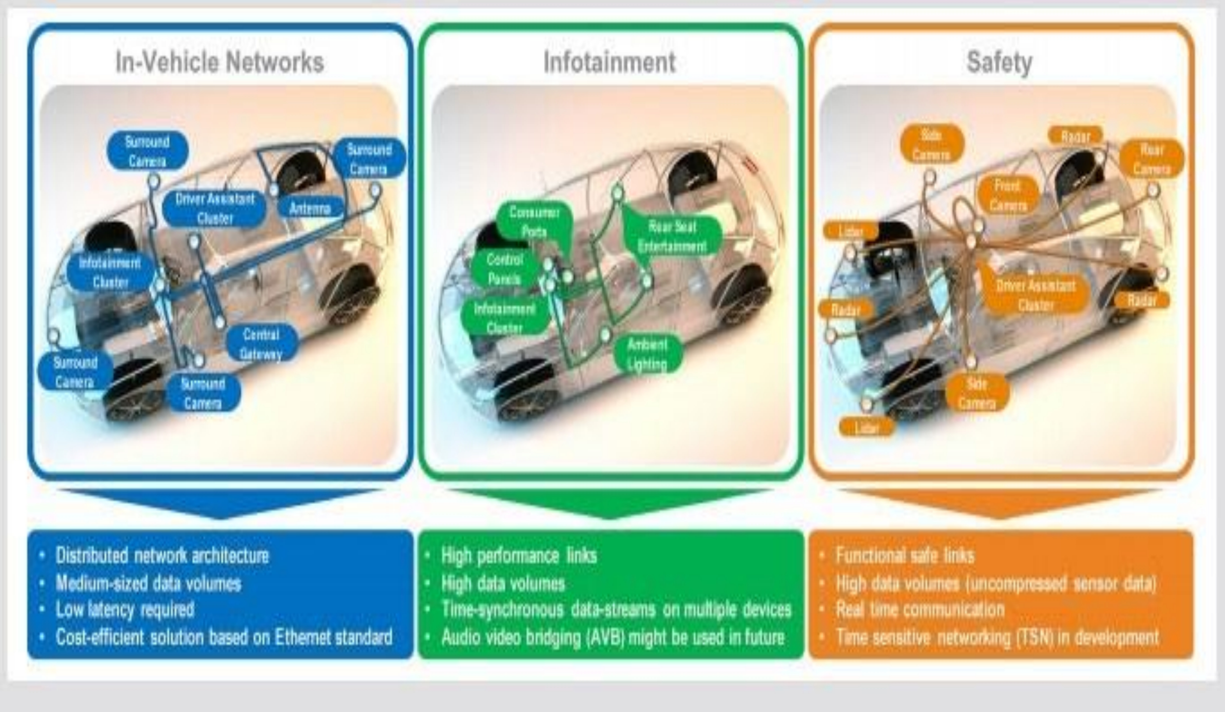
- Since cars provide customized services through multiple functions and profiles, authentication by a physical key to open doors and start the engine is no longer enough and is supplemented by software. To access personal information, such as identification, telemetry, addresses, and financial transactions, electronic keys, passwords, and biometrics must be maintained and approved. Furthermore, to prevent an intruder from faking messages or orders, the different ECUs in a vehicle need to authenticate the contact.

2.3.6 Network Security

In-vehicle networks that hold a variety of operational and personally identifiable information such as location, navigation history, call history, microphone recording, and so on; require the protecting of communication bus messages and data vital to operational security, privacy and consumer confidence. Security enhanced ECUs can communicate with security-enhanced (in-vehicle or external) networking protocols to improve the confidentiality, reliability and validity of the transmitted data. Hardware-assisted technologies that help secure networks without impeding performance, latency or real-time response include (Simonyan, 2014)

- Authentication of the message and device verifies that communications come from an approved source and protects authentications from spoofing or recording and replaying.
- Enforcement of predictably integrated behavior of all systems, limits network communications to predefined normal behavior and limits irregular types or volumes of messages so as not to affect the functions of the vehicle.
- Access controls, specifically permit communications and messages between pre-approved systems and sensors only. They block unauthorized and improper messages and warn security systems to any invalid attempts. Automakers, maintenance agencies, owners, drivers and even police and insurance companies will have different access rights to the information systems of the vehicle through licensing.

TE Connectivity Germany GmbH (2018) gives the autonomous vehicles specifications that describe tasks, standards of protection and architecture. Real-time settings, data quality, and speed specifications for each connection have to be specified at the architecture level. The link characteristics can vary depending on the functions and safety rates of the linked nodes. Given these variations, all links show characteristics and can be grouped into three groups, as shown in Figure 2.12.



- Distributed network architecture
- Medium-sized data volumes
- Low latency required
- Cost-efficient solution based on Ethernet standard

- High performance links
- High data volumes
- Time-synchronous data-streams on multiple devices
- Audio video bridging (AVB) might be used in future

- Functional safe links
- High data volumes (uncompressed sensor data)
- Real time communication
- Time sensitive networking (TSN) in development

Figure 2.12: Automotive infrastructure (CSS Electronics, 2018)

In-vehicle network connections are cost-effective solutions that can be used with medium sized data volumes and low latencies in distributed network architectures. Building these in-vehicle networks with automotive Ethernet-compliant connectors and wiring will support OTA software updates, create service advantages and repairs. The amount of data that an autonomous car is expected to generate is several terabytes a day. Infotainment connections have high performance specifications for high data rates and time-synchronous data streams on multiple devices. Accordingly, appropriate links must be engineered for optimal high frequency signal integrity properties. Infotainment links are usually used as point-to-point connectivity. Like for examples, display links for high-resolution dashboards, control panels or HUDs, and in a ring bus setup. Open protocols like Audio Video Bridging (AVB) also allow other automotive network topologies to be introduced in the future. They must ensure multiple data sources from different devices are made available on a timely basis.

Safety links implement specific requirements that are important for safety implementation and future pilot or autonomous driving applications (Paden, 2016). This type of link must ensure a high level of functional safety and real-time computing abilities. In addition, high volumes of data must be transported, as sensor data, mainly large image sequences collected by high-resolution cameras, are transmitted uncompressed at high refresh rate volumes. Safety connections are constructed using proprietary data transmission technologies and large point-to-point data pipes. Functional safety systems will be needed when self-driving vehicles become a commercial reality. One possible solution is unstable and redundant topologies based on ring bus systems. In addition, there will also be a need for real-time network technologies with low latency and high availability. Deploying open protocol time sensitive networking (TSN) is one possible solution.

2.4 SECURITY CONTROL FRAMEWORKS

There are many security Standards and control frameworks. The following sub-sections review three frameworks that are used in the CAN environment. The first is the Best Practice Framework (Auto1, 2016), the second the Auto-ISAC Security Framework (Auto2, 2016), and the third is Multi-layer Security Framework.

2.4.1 Best Practice Framework

Automotive Cyber Security Best Practices (2016), Auto Alliance and members of Global Automakers have developed a Framework of Best Practices for Automotive Cyber Security. The framework can serve as a foundation for the development of the best practices for automotive Cybersecurity. The Best Practice for automotive cyber security framework (Figure 2.13) is created to support the automotive industry's continued efforts on to address cybersecurity issues. The Best Practice frame has five guiding principles and provides methodologies for members of the industry when they refine their: recognition of, detect, prevent, protect, mitigate and respond to threats. The security framework is focused on the following guiding principles (Auto1, 2016):

- Vehicle Security by design

- Risk assessment and Management
- Threat detection and protection
- Incident response and Recovery
- Collaboration and engagement with appropriate third parties

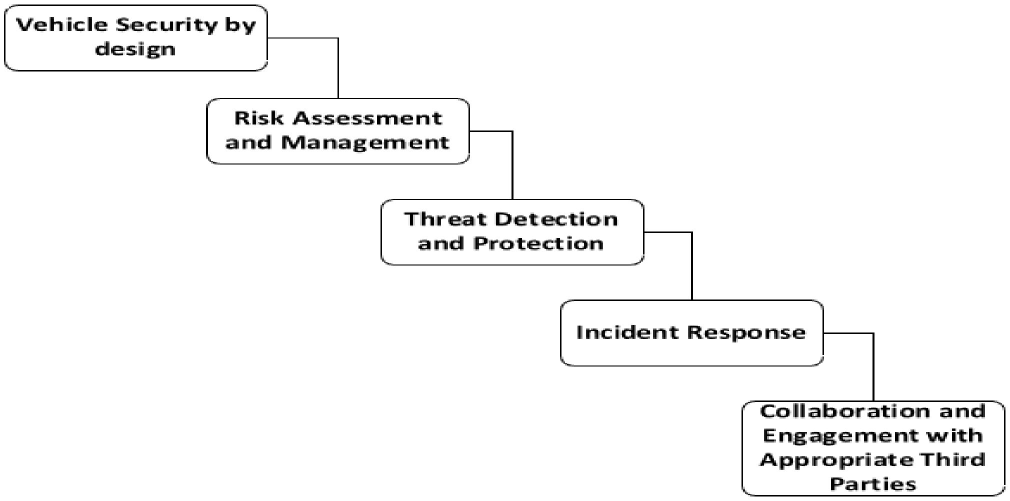

Vehicle Security by design
Risk Assessment and Management
Threat Detection and Protection
Incident Response
Collaboration and Engagement with Appropriate Third Parties

Figure 2.13: Best Practice Framework (Auto1, 2016)

2.4.1.1 Vehicle Security by design

Cybersecurity is of concern during vehicle design, development, service and use. The automotive industry also requires safety in the process of vehicle development including the design of hardware security features which protect the vehicle control system functions and communications related features, such as satellite navigation, wireless, and the telematics. Threat modelling is used for systems' design processes, security vulnerability testing, model vulnerability monitoring, and functionality tests. Design security requires a clear understanding of the threat Landscape to predict possible threats to cybersecurity and to guard against those threats. Threats may be incorporated into the software and hardware components of the vehicle (Cui, 2019).

2.4.1.2 Risk assessment and Management

Risk Assessment and Management strategies examine the impacts of identified risks. When cybersecurity risks and vulnerabilities are identified they help to develop protection strategies. The critical task of risk assessment is for finding and understanding vulnerabilities. These vulnerabilities can be exploited for personal or property implications and harm the system. Risks to cyber security are catalogued and ranked on the basis of known weaknesses, the extent to which such vulnerabilities exist, and the vulnerabilities and the severity of the possible repercussions in the real world for exploitation (Tharp, 1977; van Roermund, 2019).

2.4.1.3 Threat detection and protection

The key premise of cyber security is that the attack surface is continually evolving and sophisticated (Wong, 2016). The attacks aim to bypass even the most robust and very well designed defense system. Developing capabilities to identify attacks and to protect against cyber attacking and also alleviating the consequences of an effective cyber incident is an important goal. The principle aim of security by design is to stop cyber-attacks before they affect the system. Detection and mitigation capabilities for intrusion also extend to third parties in the automotive environment, such as manufacturers, distributors and others service partners. Those entities, although not directly controlled by car manufacturers, may represent channels through which cyber attackers can penetrate vehicles or manufacturer systems.

2.4.1.4 Incident response and recovery

The Incident Response as a complete response plan is developed to improve knowledge and abilities. It implements a contact protocol between the suppliers, automotive manufacturers, cybersecurity researchers and government agencies (Shiomi, 2014). This assists manufactures or stakeholders in synchronized efforts to resolve discovered vulnerabilities and improve the security of the product. The Best Practices security framework is focused on addressing incident response plans, which may include methodologies to activate response teams, inform an internal chain of command and response activities to assess and counter cyber-attacks. The complete incident response plan provides the tactical flexibility to manage many types of cyber incidents, taking into account internal resources and, where appropriate, external resources likely to be needed to support incident response measures (Tome, 2016). It is also important to develop protocols for recovering from cyber security incidents to ensure consistent approaches for making vehicle updates available in a reliable and efficiency manner, based on particular instances.

2.4.1.5 Collaboration and engagement with appropriate third parties

In order to defend from cyber-attacks they regularly demand the involvement of multiple stakeholders in collaboration (Wade, 2018). Creating alliances around the vehicle ecosystem offers benefits including sharing cyber threat and validated strategies with third parties to protect against cyber-attacks. Members of the Auto Alliance and Global Automakers are committed to engaging with third parties, which include organizations, suppliers, cybersecurity researchers, government agencies and the Auto-ISAC team. It will include collaborating with stakeholders to ensure the effective use of cyber security, as well as maintaining clear communication channels. Representatives of the Auto Alliance and Global Automakers will collaborate to enhance best practices relating to organized disclosures of vulnerability studies (Cui, 2019).

2.4.2 Auto-ISAC Security Framework

In July 2016, the Auto-ISAC set out to gain proactive cooperation with the industry to set safety standards through automobile cyber security (Auto2, 2016). Their method defines Best Practices for securing the ecosystem of vehicles, and provides guidance for implementing guidance. The best practices security framework has seven cybersecurity features, which are the highest level of Best Practice categorization and direct management of vehicle cyber danger. The Auto-ISAC adds Guides for new functions periodically, and to identify the evolving cyber risk landscape for vehicles. They advocate the following life-cycle steps (Auto2, 2016):

- Incidence Response
- Collaboration and involvement with the appropriate third parties
- Governance and governance
- Risk assessment and risk management
- Knowledge and Training
- Detection, monitoring and analysis of threats
- Security Development Lifecycle

2.4.2.1 Incidence Response

Incident response plan documents inform a response to cybersecurity incidents that affect the ecosystem for motor vehicles. Best Practices include guidelines for reliable and effective recovery from cybersecurity incidents, and ways to ensure continuous improvement of processes. The incident response best practices framework encompasses four areas of focus: preparing, finding, fixing and close. These terms are explained in the following paragraph.

Prepare is to assist and guide to ensure that the organization is in a position to respond efficiently and effectively. This includes documenting a plan and maintaining duties and responsibilities. These cover decision making authorities and testing the plans through training exercises. Find refers to quickly finding incidents to help to mitigate possible impacts. This includes identification, validation, classification and escalation of potential incidents using a severity matrix that is aligned to clear protocols for escalation. In the fix stage the incidents can be resolved by enabling a group or special team to quickly capture, resolve and recover from the harm. It may include carrying out technical responses such as root cause analysis, containment, and forensics. At the same time managing business risk through complementary corporate responses such as communication systems, legal, and regulatory practices. The last stage is to close every incident. This includes: briefings, evaluating the efficacy of response procedures, evaluate for appropriate protocols, policy changes, determining, enforcing and tracking any longer-term remediation actions; and updating the plan (Auto2, 2016).

2.4.2.2 Collaboration and involvement with the appropriate third parties

Defending from cyber-attacks requires cooperation between different stakeholders to increase awareness of cyber threat and the response to cyber-attacks. The industry is dedicated to interacting with third parties when confronted with cybersecurity issues. These groups include industry associations, government, academia, and researchers. The Best Practices Framework defines three structures in which activities for third party cooperation. They are sharing information, events, and programs.

Sharing information is an activity that engages and collaborates with third parties. It contribute in efforts to share cyber threats, security vulnerabilities research and best practices. Information sharing activities benefit from the identification of suitable information to be shared, the involvement of the right internal parties and the establishment of processes for receiving and acting on the information received, as well as a processes for distributing information to third parties. The events can collaborate with third parties to put together different groups of experts through focused themes. Organizations can optimize the opportunities of third party collaboration and engagement of events by recognizing and engaging in a range of event types, planning events to involve third parties or attending events conducted externally.

Lastly, identifying long-term programs to mobilize resources towards a common purpose are necessary. They organize transparency, create standards, professional exchanges or certifications. Organizations can optimize the benefits of third party collaboration and engagement events initiatives by recognizing and engaging in a range of programs styles, developing programs to involve third parties or engaging in programs run by external partners (Auto2, 2016).

2.4.2.3 Governance and governance

Better regulation aligns the cybersecurity vehicle program with the broader mission and goals of an organization. Additionally, strong governance can help promote and maintain a cybersecurity culture. Best Practices may not prescribe a specific model of vehicle cybersecurity governance but include criteria to match functional roles and responsibilities for organizational design. There are three main elements that a Best Practice framework focuses on governance. Which is design, build and operate. Although these goals are relevant for all initiatives, individual interventions are most successful when they are customized to meet each firm's particular needs.

The first element is design. A design defines and communicates the scope of the programs. Design can draw up the vision and also identify the key functions. The second element is build. This coordinates inside the program to activate leadership, establish clear decision-making authorities and to develop an employment model. The process is to start engaging across the business, then to assimilate with partner organizations across organizations, and identify and deliver on managerial communications perceptions. The last element is operate, which is to develop a process and policies that lead to operate transparent and consistent processes (Auto2, 2016).

2.4.2.4 Risk assessment and risk management

Risk assessment and management strategies reduce potential effects of cyber security problems. Best Practices focuses on the processes for detecting, categorizing, prioritizing and addressing cyber security risks that could lead to security issues. Risk management strategies can enable automakers identify and protect critical assets and help develop security precautions and support the decisions for operational risk. The Best Practices use the following tasks for Risk assessment and Management (Auto2, 2016):

- Describes the quality requirements for the implementation of the cyber risk management framework.
- Incorporate multiple kinds of safety assessment methods into appropriate vehicle or production process phases to make sure appropriate coverage.
- Outline the roles and responsibilities that will benefit for the stakeholders to understand more their roles, tasks and timing expectations.
- Decide the best timeframe for risk assessments across the risk lifecycle, as risk scores may change periodically.
- Formulate an appropriate risk profile pattern inform decision-making which is risk tolerance may vary by phase of the life cycle and is usually done by assessing risk eligibility requirements.
- Define reliable methods for assessing risk assessment results and for establishing a risk management plan.
- Discuss the risk consistently to the leaders and stakeholders, preferably using nontechnical terminology to assist them compare the cybersecurity threats of vehicles with other more traditional business risks
- Integrate procedures and principles of risk management into corporate governance and track and implement compliance

2.4.2.5 Knowledge and Training

Training and awareness programs help develop a safety culture and implement the responsibilities of cybersecurity in vehicles. Best Practices address training and awareness programs within an enterprise to increase the perception of cybersecurity threats across stakeholders. This skill usually consists of four main activities which include planning, developing, implementing, and improving.

Designing awareness and training programs through the assessment of business needs, program scope and the development of a strategies and plans, is necessary. Developing or creating the curriculum is shaped, by acquiring or creating material and items of knowledge, by acquiring or improving curricula and the learning culture. Implementation of the program is done by communication of the strategy plan, carrying out training programs, distributing products and providing training. The final step is focused on improving the program by monitoring, reporting, analyzing efficiency and identifying opportunities for improvement on a regular basis (Auto2, 2016).

2.4.2.6 Detection, monitoring and analysis of threats

Identifying threats and vulnerabilities, and proactive cybersecurity enables automotive manufacturers to minimize the associated risks and consequences. The procedures of threat detection raise awareness of malicious activity, allowing proactive repair work and recovery. The Best Practices for the identification, tracking and review of threats may include, classifying a process of threat detection and analysis by acknowledging the automotive security threats, developing a threat team structure, operational processes, and assigning responsibilities to the stakeholders. The criteria for threat intelligence will help identify sources and the collection procedures. Then create a method of tracking threats by defining goals and specific strategies and processes. Establish a framework for threat analysis that involves detection of threat incidents, confirmation and verification, and the appropriate steps to take and define the process and create or acquire the right tools to coordinate, store and exchange information to optimize its output (Auto2, 2016).

2.4.2.7 Security Development Lifecycle

The Automotive (SDL) Security Development Lifecycle concepts help make sure that suitable cybersecurity defenses are found in the early design phase such during the planning of automotive architectures. The SDL used by Best Practice has a 3 phases of: Pre-development, Design and development and Post Development.

Pre-development reflects current system designs and restricts future design choices. Lessons can be learned from previous design cycles to identify the types of cybersecurity threats acceptable and nonacceptable in the product. The Design and development phase focuses on the design and develop of the features and requirements for vehicle security. It is important to ensure consistent and testable specifications during the design phase for awareness of the threats and risks to the network. A system design is used to reduce defined threats and risks by adopting the concepts of cybersecurity protection. Security is built in the software design, code and traceability mechanisms to help make sure that the effects are no lost the requirement analysis and secure design implementation processes. Promoting testing and verifying is used to demonstrate the functioning of the implemented systems and assessing whether a system has been developed according to specifications and requirements. The last phase is Post-Development, and it focuses on the cybersecurity issues that arise after the development. It is to deliver a reaction loop on the requirements and design phases of the automotive Software development lifecycle process, and to improve security (Auto2, 2016).

2.4.3 Multilayer Security Framework

Van Roermund (2019) proposed a multilayer security framework for securing the vehicle architecture. This framework consists of four security layers that are put together to get a high level of protection. This includes securing the interface, securing gateways, and secure network and security processing. Each layer offers a dynamic type of security whilst integrating for the system's overall defenses. These layers together provide strong protection against hackers in the electronic infrastructures of automobiles (Figure 3.2, and described in the following sub sub-sections).

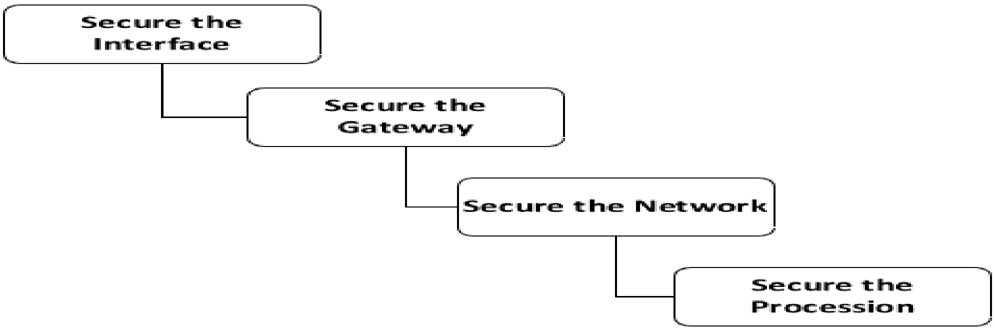


Figure 3.2 Multilayer Security Framework (Van Roermund, 2019)

2.4.3.1 Secure the Interface

Therefore, in order to protect the Connected Vehicle, these external communication networks must be secured against unauthorized access. This is through applying strong machine-to - machine authentication to stop data theft. For example, through encrypting data, and manipulation, by authenticating messages exchanged to preserve their validity and integrity.

2.4.3.2 Secure the Gateway

Gateways deal with lots of function in the In-vehicle network, such as connecting signals from the different ECUs around the vehicle and converting the vehicle communication protocols. The firewall is one of the most significant functions to secure the gateway. The firewall separates the external interfaces from the safety-critical internal vehicle network and also provides a means to manage communication between various domains and subnetworks within the vehicle network. The gateway engine is a contextually aware routing protocol feature that decides which packets are actually valid and will pass through the gateway to the target destination.

2.4.3.3 Secure the Network.

The network is divided into domains to reduce the attack surface. The architecture reduces the surface but the subdomains are still vulnerable to attack. The layers protect the subdomains with different network level protections. This includes five security safeguards. The device authentication for the device-ECU authenticity and integrity are checked before they can share information and communicate over the intranet. The message authentication makes sure messages are using cryptographic algorithms to extended and ensure an authentic sender and receiver is unchanged. The message encryption is to reduce and identity theft inside the vehicle communication exchange. Intrusion detection is used to detect anomaly attacks on the network traffic and prevent the spread of the attack to other nodes. Lastly is a rate limit and traffic monitoring to prevent (DOS) Denial of Service attacks, by getting the time frame of sending the maximum number of messages from the ECUs.

2.4.3.4 Secure the Processing

Processing is the brains of the connected vehicle. Up to one hundred independent ECUs form these brains, which together execute the control functions in the vehicle, including several advanced autonomous driving functions. ECUs constantly produce, process, share, and store substantial quantities of valuable information. Modern microcontrollers feature protected boots including real-time authentication mechanism schemes to ensure the code image is genuine, trustworthy and unaltered to protect these functions and the data. In addition, mechanisms for controlled locking of the MCU and ECU through manufacturing are used to lock out risks such as, debug access, which is valuable for hackers (Van Roermund, 2019).

2.5 SUMMARY OF ISSUES AND PROBLEMS

The biggest problem faced by CAN technologies is its open design and the emphasis on communication. Hence, the development of automotive technology that focuses on cyber security is one of the big challenges within automotive technology. The first concern is the individual electronic components that serve as tiny computers that are responsible for all kinds of vehicle functions. In additional, coordination between those individual components, represents control of the entire vehicle system. The vehicle has various interfaces with the outside world which bring data in for the CAN to process. It also processes data from within the vehicle. The transmission and retrieval of data from both inside and outside requires layers of protection and checking to assure safety.

Table 2.1 lists hacks on CANs that are reported and have been successful. They have serious real-world consequences for vehicle security vulnerabilities and this could be the impetus needed to bring about change. The problem is complex and deep-rooted, and the solution is neither straightforward nor simple. But the time has come for automakers to stop patching security flaws from the ground up and start designing safe systems with a strong protection for automotive vehicle networks. Several attempts have been made to address the issue and in the USA car hacking laws have been passed to enforce better security, and the ISO has established International Standards that include vehicle security provisions.

The outstanding issue is that hacking keeps changing and adapting to the new security provisions. Many CAN bus elements and designs have been introduced to stop previous vulnerabilities being exploited. At present physical security using encryption is securing many information exchanges. However, access to the CAN bus can be gained through GPS spoofing, where real time data is corrupted and the vehicle navigation disrupted. Also there are a variety of spoofs that corrupt the artificial intelligence applications so that incorrect signals are sent through the CAN bus. These attacks are external to the CAN bus security and are dramatized scenarios that the sensor systems correctly detect but the nodes and controllers incorrectly interpret for a response. One example is that a thief wanting to break into a recent Tesla or Mercedes-Benz cannot hack the encrypted electronic controls to open a door, but if the front bumper or grill is hit hard enough (eg. with a hammer or a rock) then the safety features are hijacked, and the airbags will go off and the doors spring open. Hence, scenario and heuristic script attacks are still not mitigated.

2.6 CONCLUSION

The literature review in this Chapter has covered:

- Motor vehicle electronic technologies defined
- Attacks on controller area networks
- Security provisions
- Security Frameworks
- Issues and problems arising

In Chapter 3 the problem of CAN security is explored to define a research methodology that can answer questions around vehicle protection and safety.

Chapter 3
METHODOLOGY

3.0 INTRODUCTION

Chapter 2 has reviewed selected literature that defines the CAN technology, and information security risks. Three risk management strategies have also been reviewed in the form of frameworks. Chapter 3 is to formulate the research question for the problem area of CAN security, and the sub-questions, and a research methodology. It focusses on framing the research questions and generating a research design for mapping the research experimental tasks. The Chapter begins by reviewing previous similar studies to identify how others have approached security research in the CAN area. The problems in Sections 2.5 and 2.3 are then reviewed to justify the selection of questions relevant to the research problem. Section 3.3 lays out a practical research design and Section 3.4 the data requirements. Section 3.5 discusses the limitations for transferring the findings from this study.

3.1 REVIEW OF SIMILAR STUDIES

Three studies are relevant to this project topic have been analysed and reviewed to frame the research and determine definitions. The analysis will help to get an idea of how to do the research, and build the selection of relevant methods to be used. The focus is on the methods and techniques they used for detecting the CAN packets or the Intrusion Detection System (IDS) methods for CAN bus communication. The purpose of this section is identify the learning points that can come from other published works in the general security area. These points can guide the practical testing activity and place limits on the scope of what may be attempted. The first study shows the frequency of CAN packets can be used to detect suspicious behaviour in the CAN. The second identified anomalies by statistical methods which are used to construct a natural baseline, and then to recognise deviations. The third study reviewed used a neural anomaly detector to detect CAN bus attacks. Each of these three studies provide insight into CAN security research methods.

3.1.1 Frequency Analysis for IDS

The frequency of CAN packets can be used to detect suspicious behaviour in the CAN bus, because of the limited computing capacity in vehicles. Song et al. (2016) suggested a lightweight intrusion detection algorithm based on the assumption that each message ID has a normal frequency and that the message frequency shifts suddenly when attackers insert messages into the CAN bus. The paper first reviews similar studies and recent research projects on vehicle security, and defines the standard of vehicle security and the tools used to hack with. Then they review and analyse the current IDS for motor vehicle networks. They note that the previous work on intrusion detection based on message rate for CAN bus needs to collect enough CAN bus messages to determine a message's distribution. Its detection methods therefore requires some time to detect anomalous messages. Therefore they propose a new IDS system known as Lightweight Intrusion Detection System to overcome the issues. According to Song et al, (2016) their experiment with Lightweight IDS, captured CAN messages from the K-cars made by a popular manufacturer and carried out three kinds of injection attacks with messages. As a consequence, the time interval is a significant function for detecting attacks in the CAN traffic. The intrusion detection system also detects all injection attacks from the messages without creating false positive errors.

Cho and Shin, (2016) focused on the fact that most network messages in vehicles are periodic and transmitted over CAN and proposed that the time intervals of these periodic messages be used as ECU fingerprints. These techniques are most successful for regular messages so an attacker who periodically injects messages can go undetected. When the ECU itself is the source of the malicious packet IDs, attacks will go undetected. The research reviewed similar studies on defending against vehicle cyber-attacks. There have been two key sources of security solutions to protect against various forms of cyber threats in vehicles which is message authentication and intrusion detection. The authors used an Adversary model as an attack model for their study. The attacks can compromise more than one in-vehicle ECU physically or remotely via various attack surfaces. They find an opponent who attempts to control functions within the vehicle can do this either by inserting arbitrary messages into the in-vehicle network with a spoofed ID, attack messages, or by preventing the compromised ECU's message transmissions. From this study, the authors used three types of attack scenarios: the Fabrication attack, the

Suspension attack and the Masquerade attack. The defence against these attacks was the IDS. They did experiments on a Honda Accord 2013, a Toyota Camry 2010, and a Dodge Ram Pickup 2010.

The authors observed that the IDS can deal with some threats but cannot protect against other vital security incidents like the masquerade attack. They have proposed a new IDS called Clock base IDS (CIDS) by improving the algorithm to remedy this issue. It detects clock movement that skew from expected message intervals, fingerprints the pattern, and models their clock behaviours using RLS. Based on their experimental tests on a CAN bus prototype and on actual vehicles, CIDS has been shown to be capable of detecting different types of network intrusions in vehicles. CIDS can answer attacks that current IDSs can and cannot handle and also promote the study for the root causes.

3.1.2 Statistical Analysis

The Muter and Asaj (2011) & Marchetti et al. (2016) approach to identifying anomalies is to use statistical methods to construct a natural baseline, and then to recognise deviations from the standard. One concept is a model based on Entropy, and they suggested a technique for identifying entropy-based anomalies. Muter and Asaj, (2011), used the in-vehicle network entropy as the basis of normal behaviour. The basic intuition is that the entropy is relatively small due to the simple and restrictive specification of in-vehicle traffic, and thus attacks like changing packet payload, and packet injection will cause the entropy to increase.

The first step that Mutter and Asaj did, was to review the previous research on vehicle network security issues. They located IDS characteristics and the challenges of achieving a low rate of false positives. They choose to apply the principles of entropy-based anomaly detection into the area of in-vehicle networks, to extend the known present works. The method recognises a reactive definition and acts as an additional security shield, when usual preventive steps have failed. They thus demonstrate how a self-adaptive existence enables a simple adaptation to the automotive domain and a convenient extension to new vehicles. They further examined the key parameters that are critical for the realization of an in-vehicle domain definition for information-theoretical intrusion detection. The researchers illustrate the applicability of security design by testing it on a real vehicle's CAN network with various attack scenarios that include frequency attack scenarios, message flooding attack scenarios, and plausibility of interrelated attack scenarios.

Marchetti et al, (2016) performed comprehensive experimental tests using hours of real CAN data. Many drawbacks were noted about entropy-based approaches. In particular, to detect low volume attacks, an anomaly detector must be built for every message ID. The researchers propose and assess an entropy-based algorithm to detect anomalies in CAN messages created by a licenced unmodified vehicle. Firstly they review a similar study based on Anomaly detection and vehicle information security. They realised many reports have already suggested the use of anomaly detection algorithms to analyse CAN messages and to search for signs of attacks and other illegal activities, although with restricted evaluation for real in-vehicle network traffic. But in order to address this issue the researchers declares an experimental evaluation of the efficacy of an anomaly detection algorithm based on entropy calculations and applied to CAN messages exchanged in the in-vehicle network. They consider two separate scenarios of attacks, which imitate activities that an attacker can conduct to reverse engineer an unknown CAN bus and to assess vulnerabilities of several ECUs connected to the CAN bus.

In all cases, it is presumed the attacker is able to read and write arbitrary messages to the CAN bus, either by physical access or by a remote vulnerability. The experiment analysed data obtained from a 2011 Ford Fiesta CAN bus. The vehicle was fitted with a custom CAN bus logger that was built with a Genuino UNO prototyping board, a CAN bus shield, and a data logger shield that writes CAN message to an SD card. The sniffer can be connected directly to the CAN bus or the OBD-II diagnostic port (mandatory for all European licenced vehicles).

3.1.3 Deep Learning Detection

Taylor and Nitschke (2017) proposed a Long-Short Term Memory neural anomaly detector to detect CAN bus attacks. The detection system works by learning to predict the next word of data coming from each transmitter in the bus. Extremely surprising bits in the next word are flagged as anomalies. The detector is evaluated by synthesising anomalies with the changed CAN bus data. The synthesised anomalies are planned to imitate recorded attacks in the literature. They prove the detector is able to detect anomalies that are synthesised with low false alarm levels. Furthermore, the granularity of the bit predictions may provide clues to the investigators about the existence of marked anomalies. They do this through reviewing anomaly detection in related research, and doing anomaly detecting CAN bus attacks. They saw a problem with the previous anomaly detecting techniques which were unable to identify attacks where the compromised ECU is the usual emitter of malicious packet IDs. They propose a technique that can identify packet data irregularities that are unique only in the sense of variation from the rest of the series (Zeiler, 2012; Punyawiwat, 2018).

Their research focuses on data collected from a high speed bus from a Subaru Impreza 2012. Using LSTM neural networks, they demonstrated the identification of sequence data anomalies on the CAN bus. The LSTM solution has the benefits of not needing awareness of the particular protocol and has shown promising results in detecting a number of anomalies relating to identified attacks. Kang and Kang suggested, (2016) using an intrusion detect system based on Deep Neural Networks (DNN) for in-vehicle networks. Unsupervised Deep Belief Networks (DBN) were used as a pre-processing method to initialize the DNN parameters. The data set was generated using a packet generator and anomalies were injected with packet manipulation and with Gaussian noise added (Szegedy, 2015; Taylor, 2017).

The research was conducted by reviewing a similar study on CAN and architectures of the IDS used for vehicle security such as Intrusion Detection with machine learning. The IDS provides different modules to store and process vast amounts of data packets. The controlling module usually detects a type of incoming packet and the functionality. The profiling module contains the trained off-line features. Where the monitoring module detects a new type of attack, the profiling module may update the outlining module database for potential packets. The experiment was based on a data set generator and an attack scenario which had malicious data packets inserted into a CAN bus in the vehicle. The proposed techniques train CAN packet data patterns to evaluate the essential statistical properties of common attacks and extract the correct features for protection, and to classify the threat (Pedregosa, 2011; Perez, 2017).

3.2 THE RESEARCH QUESTION

The review of previous research in Section 3.1 shows that there are security risks in CAN architecture. The studies took data from motor vehicle CANs and designed IDS and countermeasures. In Chapter 2, Sections 2.1 and 2.2 define the CAN technologies and security risks. In section 2.5 the outstanding issues and problems are listed. The concern of CAN security is an outstanding feature and recent tricking of the artificial intelligence control features for CAN are noted. Hence, the research question is: *What is required to assure CAN vehicle security?* Also, with the main question, it has several sub-questions that will help to provide better understandings in response to the main question. The sub research questions are described as follows:

Sub question1: What are the security attacks on automotive communication networks or Electronic Control Units (ECL)?

Sub question2: What are the security vulnerabilities of ECUs?

Sub question3: What countermeasures are available for security vulnerabilities in automotive communication networks?

Sub question4: What management strategies are required for prevention and countermeasures in automotive communication networks?

3.3 THE RESEARCH DESIGN

The research is designed to investigate a problem that is reported in the literature and speculated in general media reports. The problem is the security of CANs and the implications for motor vehicle safety. The exploratory examination is intended to begin by investigating and reviewing the present literature found in the body of knowledge, and then to follow a logical progression for experimental inquiry. Figure 3.1 provides an overview of the proposed sequence of research activities. It will cumulate in the recommendations for security improvement and recommendations for further research. The design also provides a plan to follow and a forecast for what can be achieved. In practice the plan can change according to pragmatic problem solving to make it work in practice.

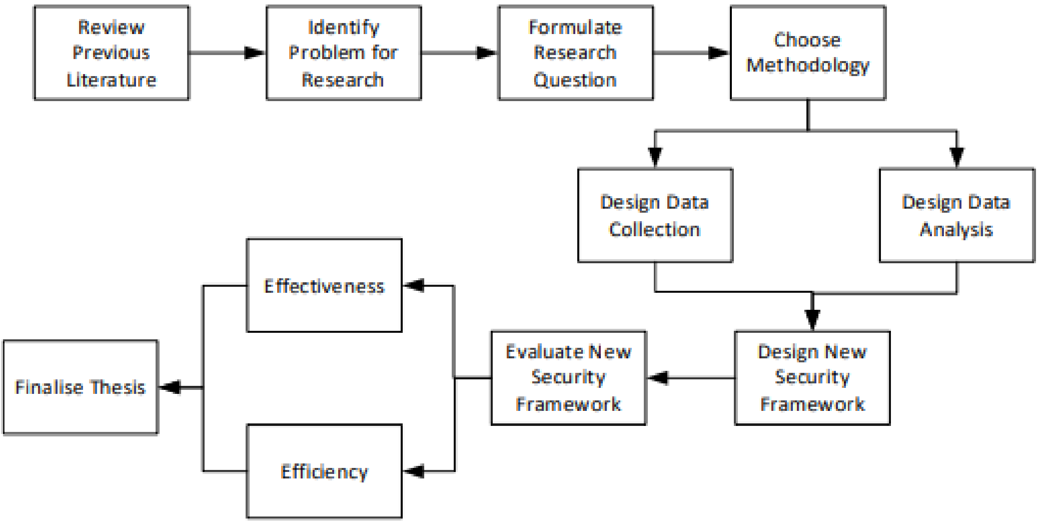

Review Previous Literature
Identify Problem for Research
Formulate Research Question
Choose Methodology
Design Data Collection
Design Data Analysis
Design New Security Framework
Evaluate New Security Framework
Effectiveness
Efficiency
Finalise Thesis

Figure 3.1: Design of the study

3.4 THE RESEARCH METHODODOLGY

In order to get the best outcome for this research, a methodology is selected to guide the study. Since this research will investigate cyber security in motor vehicle communication networks as an exploratory study, two potential sources of data are possible. One is the use of diagnostic tools to interrogate actual CAN in motor vehicles, and the other is to use CAN simulation software to gain data on the theoretical functionality of a CAN. As a consequence two different sets of data can be acquired that reflect different aspects of a CAN functionality. First the use of diagnostics tools discloses the degree of common protection given to the CAN and also the authorisation to change elements within the CAN. A simulator, however, allows the collecting of theoretical clock shifts and response surges when attacked. Both of these methods are to be used to collect data. The following sub-sections define the tool specifications, the data to be collected, and the ways to analyse the data.

3.4.1 Data Collection Tools

There are two types of data to be collected: theoretical and empirical. The following two sub sub-sections describe the collection of simulator theoretical data, and the collection of diagnostic tool data.

3.4.1.1 Theoretical Data

The RTaW-Sim package for CAN simulation version 14.4.13 Professional version was downloaded and used to monitor rate shifts when the CAN was attacked. This simulator was selected because it was not only free but it had a full range of the capabilities required for this exploratory research. RTaW-Sim is a timing accurate simulator that provides frame response time distributions and statistics about the frame buffer usage at the microcontroller and communication controller level. RTaW-Sim is able to simulate and predict the performances of CAN 2.0A, CAN2.0B, ARINC825 and CAN FD networks, with modelling of the communication stack and communication controller. It helps a designer compare the impact of different design alternatives, choose the right communication stacks (e.g., waiting queue policy) and communication controllers (e.g., number of buffers), and configure them. Also it enables the designer to perform Simulation Based Fault Injection (SBFI), for testing the impact of transmission errors on transmission latencies. These features are helpful for testing the four attacks laid out in Chapter 2.1-2.4.

The simulator has statistical and network visualisation tools to communicate the different designs and control relationships (Figure 3.2).

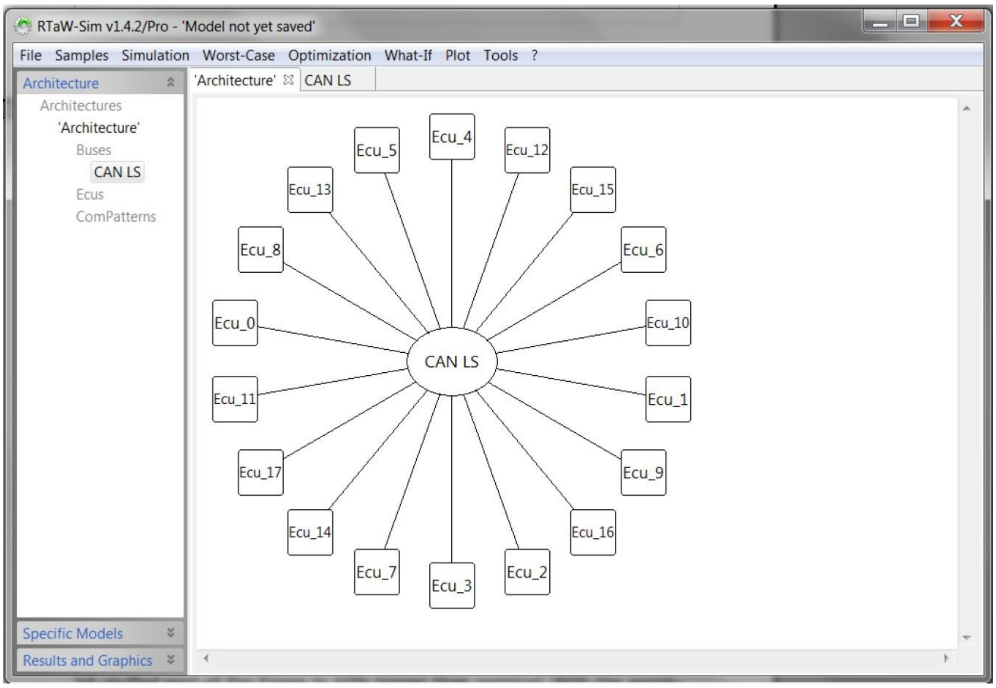


Figure 3.2: Simulator CAN Architecture

3.4.1.2 Empirical Data

The USB ELM327 V1.5 OBD2 Code Scanner (Figure 3.3) is an entry level CAN scanner that gives access to a wide range of motor vehicle CANs. It is functional by plugging the socket into a motor vehicle dock and the USB into a work surface, such as: a laptop, a mobile phone or the like. Some versions also have wireless connection to the work surface. The work surface then has a GUI of critical metrics and the capacity to save data.

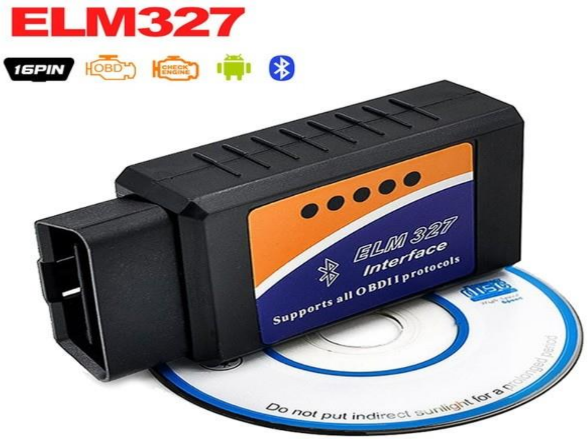

ELM327
16PIN
OBD
CHECK ENGINE
ELM 327
Interface
Supports all OBDII protocols
Do not put indirect sunlight for a prolonged period

Figure 3.3: USB ELM327 V1.5 OBD2 Code Scanner

The Code Scanner can provide diagnostics data on the CAN. A more expensive handheld diagnostic scanner will provide data bases of fault correction solutions and increase the resolution of ECU fault detection. However, for this study we are interested in the scope of intervention obtainable, and the information disclosure available. Hence, a simple and cheap scanner delivers to the research requirement. It also has the capacity to do real-time performance checking of the vehicle which will show the effect of any intervention on performance. An ELM327 scanner operates based on three sub-parts. The first is the ELM327 chip. This is an integrated circuit that’s rated at 4MHz. It is the brain behind the whole device on-board its purpose is to convert data from the ECU to a format that can be understood by modern computing devices. The second sub-part is the set of electric voltage adaptors. Their primary function is to identify the various on-board systems of a car using their unique voltage properties. For example, an ELM327 scanner is able to distinguish the EVAP system from the fuel system using its electric voltage adaptors. They are usually given as protocols like CAN, K/L or PWM/VPN. Finally, there is a set of voltage adaptors for PC. These ones make it possible for an ELM327 adapter to adapt its electrical levels to those of a computer device (PC, smartphone or tablet). By doing so the adapter and the computer can successfully communicate.

To use the scanner the first step is a Data Collection Phase, where is the CAN bus data stream is obtained for analysis from the actual vehicles. Then this data has to be interpreted in terms of the vehicle code IDs and the sequences (See Appendix A and B). A higher level of abstraction is then obtained by the use of data models to lift patterns from the time series and sequences. The final step is dependent on being able to intervene in the CAN. The phase of analyses and detection can only occur when the detection of anomalous behaviours occurs. At a simple level this is a fault code but at a complex level this is the successful execution of the attacks.

3.4.2 Data Requirements

The data requirements are divided into theoretical and empirical requirements. The theoretical data is to come from the simulator tests and the empirical data from the vehicle diagnostics testing. The theoretical data is able to answer questions about the potential to hack a CAN system by showing different packet loadings for each of the four defined attacks. The empirical data is able to answer questions around the security of a system and the physical limitations for system intervention. Challenges are expected for the co-ordination of hardware and software setups and the potential to actually achieve the experimental interventions. However, the following templates are prepared to collect data (Tables 3.1 and 3.2) for the attacks defined in Chapter 2.

Table 3.1: Theoretical Data

Attack	Evidence	Comments
Spoofing	Scanning	
	DoS	
	On/Off	
	Scanning	

Bluetooth	Overflow	
	Execution	
Sensor Jamming	Ultrasonic	
	Blinding	
	Fakenews	
Wireless Attacks	Instruments	
	Entertainment	
	Controls	

Table 3.2: Empirical Data

Attack	Evidence	Comments
Spoofing	Frame Rate	
	Work Loads	
	Actions	
Bluetooth	Packet Rates	
	Bottlenecks	
	Insertions	
	Frame Rates	

Sensor Jamming	Response Rate	
	ECU Rates	
Wireless Attacks	ECU Activity	
	Bottlenecks	
	Functionality	

3.4.3 Data Analysis

The data is to be collected in accordance with the tables and tools outlined in Sections 3.4.1 and 3.4.2. The data is to be analyzed based on the security requirements for the CAN vehicle security. Hence, it is divided into three different types of data. The first is the security requirement for a vehicle when connected and its focus on CAN traffic or internal Network traffic. The second security requirement is the vehicle communication such as Wi-Fi and wireless LAN connectivity of the vehicle or External network traffic. Lastly is the security requirement of the vehicle data security. The Tables 3.1 and 3.2 will present summative data that has been analyzed from more primitive forms of data such as packets, rates, check sheets, and tabulated status lists. The analysis of the primitive forms of data will be left to the software to aggregated and report, for example packet rates and frame rates. Descriptive interpretative analysis will be made of the Diagnostic tool output and reported.

Hence, data analysis is performed by aggregation, tabulation, and comparison. The data presentation is in the forms of text, numerical outputs, and visual forms such as charts and maps. These formats are used to communicate the states of security features, and the variations resulting from the four clusters of attack.

3.5 LIMITATIONS OF THE STUDY

The limitations for this study concern the ability to transfer the learnings to other contexts. This research was proposed to be exploratory because many of the previous studies used equipment that would not be available to the researcher. Hence, the data range has been limited to clusters of theoretical and empirical possibilities that the researcher expects to be able to obtain. For example two ODBII compliant motor vehicles are available for empirical testing but the tests will be static and up until the limits of the diagnostic tool capability. It is not expected that malicious codes will be injected into the motor vehicles because the selected tool does not appear to have the capability. However, the tool is representative of entry level diagnostic tools and it can disclose the limitations on intervention in to the CAN systems. These expected findings will be useful to moderate the often over stated claims regarding risks and ease of malicious access to CAN systems through diagnostic tools.

Similarly the theoretical expected findings are limited to one simulator when there are many with different capabilities for purchase or for free on the internet. Also the time limitations for the study prevent a full exploration of possible CAN architectures in the simulator, and only one standard configuration is used. As such the results are indicative of what may be found in CAN security risks but not comprehensive. This is exploratory research and it is expected many other tools, techniques and avenues for investigation will open up as the research proceeds. A summary of these incomplete explorations will be made in Chapter 6 as recommendations for further research.

3.6 CONCLUSION

Chapter 3 has outlined the research methodology including the research design and research questions. Chapter 3 started with a review of similar studies that are relevant and related to the area of CAN security. They confirm that risks exist and that further research is required. The tools and the data collection methods have been defined. In Chapter 4 the results of applying these plans is reported. Chapter 4 will present the findings of this exploratory research, and the collection of the results, that are based on the research plan in Chapter 3.

Chapter 4
RESEARCH FINDINGS

4.0 INTRODUCTION

The research methods and methodology as outlined in Chapter 3 was followed to get the planned results. However, the COVID-19 lock downs closed the Laboratories on two separate and extended periods, disrupting the data collection phase of this research. As a result not everything that was planned and shown in Figure 3.1 was completed. The result is two independent sets of data that represent the empirical and theoretical (simulator) results. In sub-section 4.1 the theoretical results are reported, and in sub-section 4.2 the empirical findings. These results are evaluated and discussed in Chapter 5.

4.1 THEORETICAL FINDINGS

The RTaW-Sim simulator was chosen as it had a number of completed research reports supporting its use in research, and the vendors appeared proactively involved in CAN bus research and development. It was also free, multi-platform, and only required an email sign up to their database. The only difficulty encountered was the AUT image supplied on the test equipment. Full root privileges (administrator rights) are required to configure and run the simulator. Once the problem was identified other equipment had to be borrowed and used off campus where free access to the internet was available. The test machine was an IBM PC with an i7vPro processor, 16M of RAM and a 500Mb SSD drive. Initially the RTaW-Sim Starter edition functioned correctly. This provided simulator capability, the CAN2.0 protocols, and Gateway tests. However, attempts to get the RTaW-Sim Professional version working within the limited time frame were not successful. The consequence was that all the basic tests could be performed but that the full range of security tests could not be done, and this has to be a future project.

The Standard CAN template was set up for testing. The minimum requirements are:

- The definition of the CAN features
- The frames for the bus
- The bus interface formation

- The setting of offsets to zero
- Definition of the ECUs and the relationships

The architecture of the simulator is based on the concept of discrete event simulation where discrete occurrences happen at a moment in time. Although the duration of the event may extend beyond one representative moment in time the metric system unitizes and represents the occurrence in discrete intervals. This means the simulation of a CAN bus does not cover all possibilities but reflects the underlying model and the ability for manageable investigations. Similarly, simultaneous events require a deterministic order to program which will be executed first. The scope is limited with these constraints but it is sufficient to test designs and innovation plans for CAN systems. The occurrence of an event modifies the test system state and hence testing proceeds by successive occurrences of the events, and the system will report these outcomes in data and visual statistics.

The simulator constrains determine reducing the complexity of events and distributing these into manageable quantities. Time is a critical management function as the simulator has to keep the transmission duration and processing of frames as close to the real world times as possible. An assumption would be that communications have a real world transmission rate close to zero but these measures are variable on account of real load volumes and distribution. Hence, in a simulator these variables have to be set and controlled to emulate communications. In the simulator each ECU has a local clock to create and synchronize the sending of frames. Successive instances are managed by a periodic frame that sends the message (packets) to the COM stack. Time is measured from the release of a packet from the ECU in units of the offset frame number (can be set by default or by the researcher). In the COM stack a frame is either stored in a hardware buffer or placed in the software queue. The information management problem and the need to maintain realism are multiplied by the number of ECUs in each test design. Again for practical purposes the complexity in design has to be minimized and the number of ECUs limited. Hence, in this study the selection of the RTaW-Sim Starter version is sufficient to allow testing in a feasible manner.

Data inputs can be made in limited file types (eg, CSV, XML, etc.) once the CAN is configured. Figure 4.1 shows the standard CAN configuration downloaded for testing. In addition the NETCAR-Analyzer file was downloaded to configure the frame offsets, measure response times and to report changes when the attacks occurred. The

NETCARBENCH file was also downloaded to use the automated automotive message sets for the baseline.

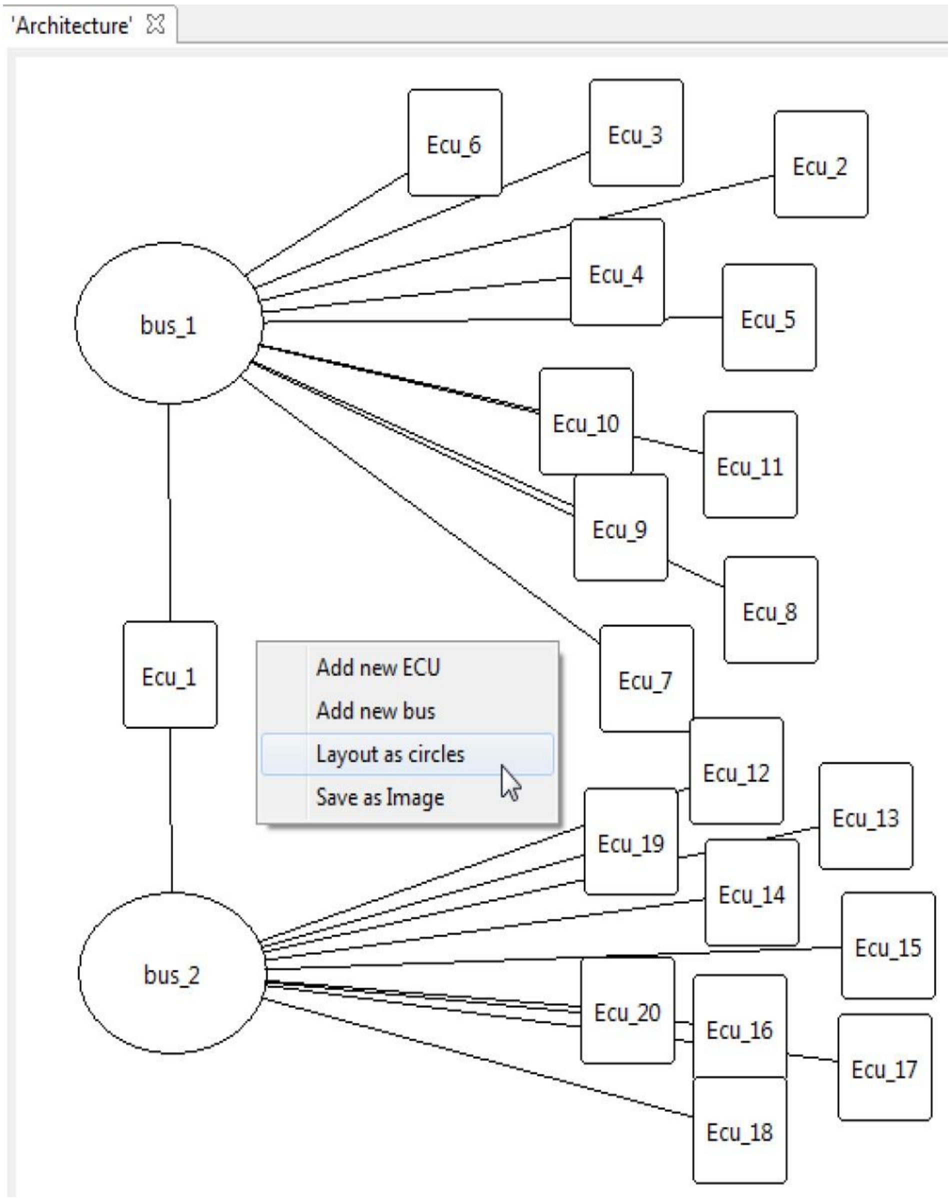


Figure 4.1: Standard CAN Model

To make the testing work the Standard templates and data set were loaded to set the baseline for normal CAN functionality. Then data introduced by CSV file to simulate each of the planned attacks. The data was then collected and compared with the baseline. Unfortunately this is where the time line for completion and the limits of the configuration set ups were encountered. The setup of a non-Standard and customized CAN for this research (for example a replication of the Nissan Van or the Ford Car) would have been feasible is the COVID-19 lockdowns had not occurred. Similarly there would have been time to set up the Professional version with the added features. However, the plan for testing is consistent with the requirements for answering the research questions and the data summarized here.

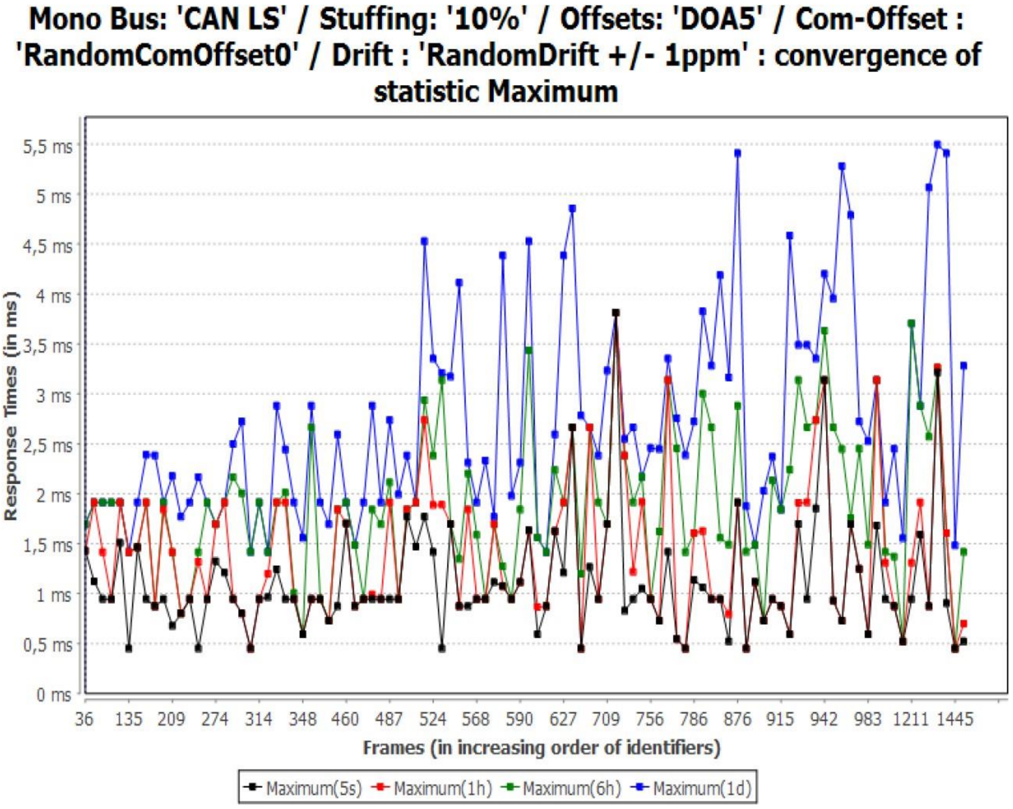


Figure 4.2: Standard Functionality

Figure 4.2 shows the time based reporting from the Standard CAN design. Natural variations are built in to simulate real CAN events and the expected time elapses for each frame. The simulator once set up and loaded with a test CAN configuration produces data and these visual statistical reports. Each visual report can be further analyzed by zooming into the Baseline and isolating particular instances or relationships of interest. This means that the baseline communication activity can be compared with intervention data (see Figure 4.3). The challenge for the researcher was to fit the attack data into the CSV file format for processing. This proved time consuming as the format constrained the extent to which an attack could be described. The results show the capability of the tool but further development and use is required to perfect the techniques. Figure 4.3 shows the visual presentation of intervention data for a single attack as compared with the baseline data for the Standard design communication activity. All metrics reflect time-based differentials which correspond to the maximum capacity of a CAN. If the intervention data set contains excessive information then the queuing mechanisms priorities and orders the input to sustain functionality. In a real situation the volume (eg. a DoS attack) would overwhelm and jam the CAN communications, or if the data input was too complex (eg. scenario attacks) then only some information would be processed, leading to inappropriate responses. In Figure 4.3, the green line is the baseline (no events) and the red line intervention data (infrequent events).

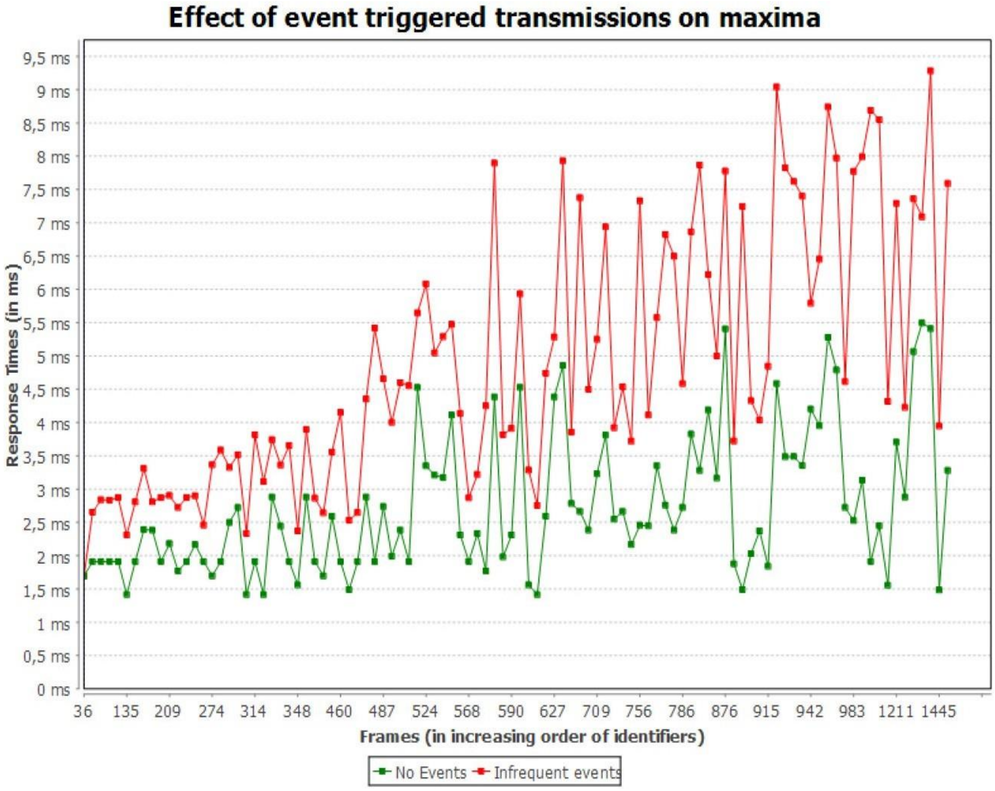


Figure 4.3: Triggering Events

In Figure 4.4 different types of attacks are compared with the green baseline of CAN communication activity. There are infrequent communications (eg. scenario attacks) and frequent attacks (eg. DoS attacks). Each is shown and distinguishes itself from normal activity. As was suggested by other researchers in Chapter 3, these discrimination activities allow the identification of attacks and the deployment of countermeasures. My hope is that these experiments can be completed in the future and features developed for intrusion detection systems.

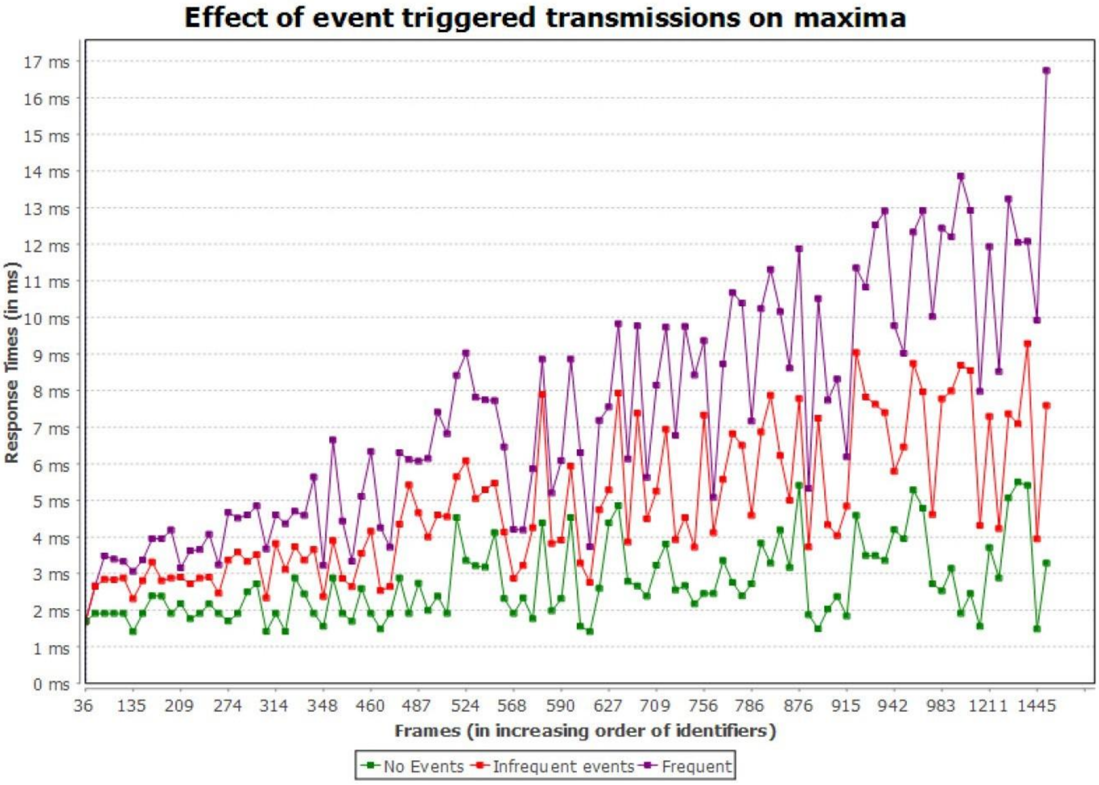


Figure 4.4: Attack Variations

At this point in time the theoretical modelling and testing of the CAN systems is incomplete. The RTaW-Sim simulator has shown itself to be a powerful tool for investigating CAN architectures. Further research and development are required to explore the full scope for security research.

Table 4.1: Theoretical Findings Summary

Attack	**Evidence**	**Comments**
Spoofing	Scanning	The simulator had weak authentication and the attacks were successful.
	DoS	The CAN and ECU messaging could be disrupted /slowed.

	On/Off	No the response was proportional.
Bluetooth	Scanning	n/a = not available
	Overflow	n/a = not available
	Execution	n/a = not available
Sensor Jamming	Ultrasonic	n/a = not available
	Blinding	n/a = not available
	Fake news	n/a = not available. This is where the sensors receive false or trick information.
Wireless Attacks	Instruments	n/a = not available
	Entertainment	n/a = not available. The ECU was available but no GW attached.
	Controls	n/a = not available. The controls could be subverted but no wireless capacity was available to do this.

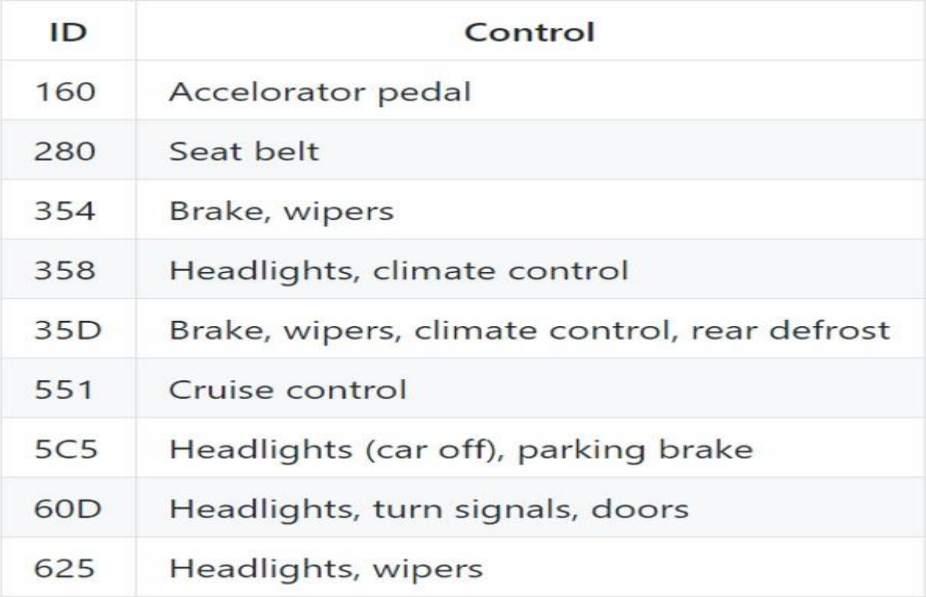
4.2 EMPIRICAL FINDINGS

The ELM 327 USB interface was connected to two ODB11 compliant vehicles: a 2010 Nissan Van, and a 2005 Ford car. These findings are reported in the following sub-sections.

4.2.1 Nissan Van Data

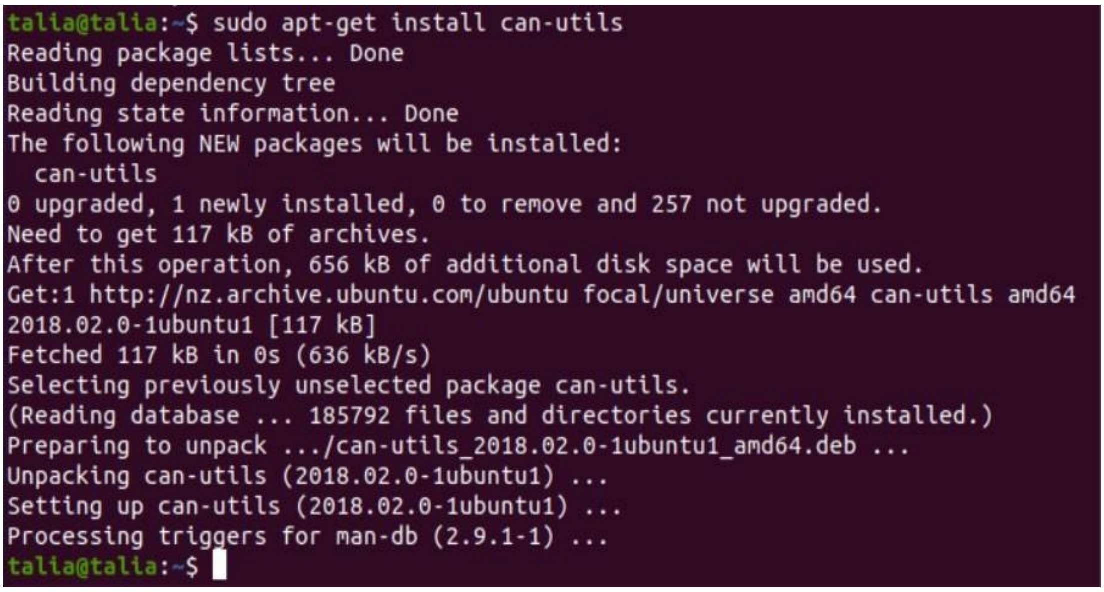
Figure 4.1 has the standardized CAN Nissan 2010 Van IDs and controls (see Appendix A for full document). These controls are the critical components for safe vehicle operation. Each control has a unique identification code that is transacted by the CAN to maintain correct and appropriate actions (Table 4.2). in a 2010 Nissan van no functions are fully automated and the final responsibility for compliant driver behavior rests with the human controller.

Table 4.2: Nissan ID Codes



ID	Control
160	Accelorator pedal
280	Seat belt
354	Brake, wipers
358	Headlights, climate control
35D	Brake, wipers, climate control, rear defrost
551	Cruise control
5C5	Headlights (car off), parking brake
60D	Headlights, turn signals, doors
625	Headlights, wipers

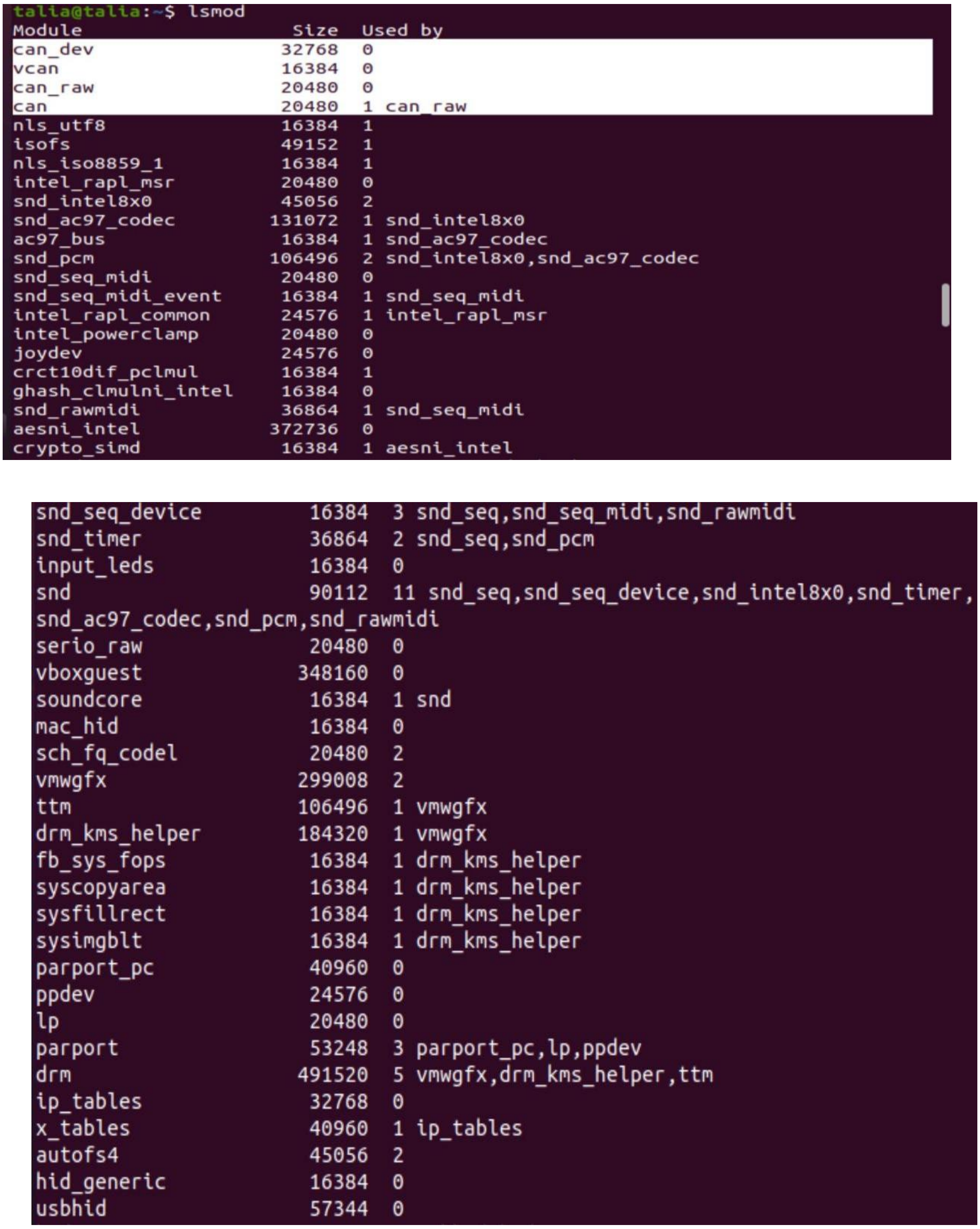
To prepare the physical connection with ELM 327 usb interface OBD11 connector, the Socket CAN is to build up a link to communicate with the vehicle network or CAN bus, by installing the can-utils packages and the relevant kernels (Figure 4.2).



```
talia@talia:~$ sudo apt-get install can-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  can-utils
0 upgraded, 1 newly installed, 0 to remove and 257 not upgraded.
Need to get 117 kB of archives.
After this operation, 656 kB of additional disk space will be used.
Get:1 http://nz.archive.ubuntu.com/ubuntu focal/universe amd64 can-utils amd64
2018.02.0-1ubuntu1 [117 kB]
Fetched 117 kB in 0s (636 kB/s)
Selecting previously unselected package can-utils.
(Reading database ... 185792 files and directories currently installed.)
Preparing to unpack .../can-utils_2018.02.0-1ubuntu1_amd64.deb ...
Unpacking can-utils (2018.02.0-1ubuntu1) ...
Setting up can-utils (2018.02.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
talia@talia:~$
```

Figure 4.5: CAN-Utils

To get the relevant kernel modules, first install the canprobe modules which include can modprobe, can_dev modprobe. For physical connection with the devices that connect to the CAN bus, the vcan or virtual connection is used (Figure 4.3).

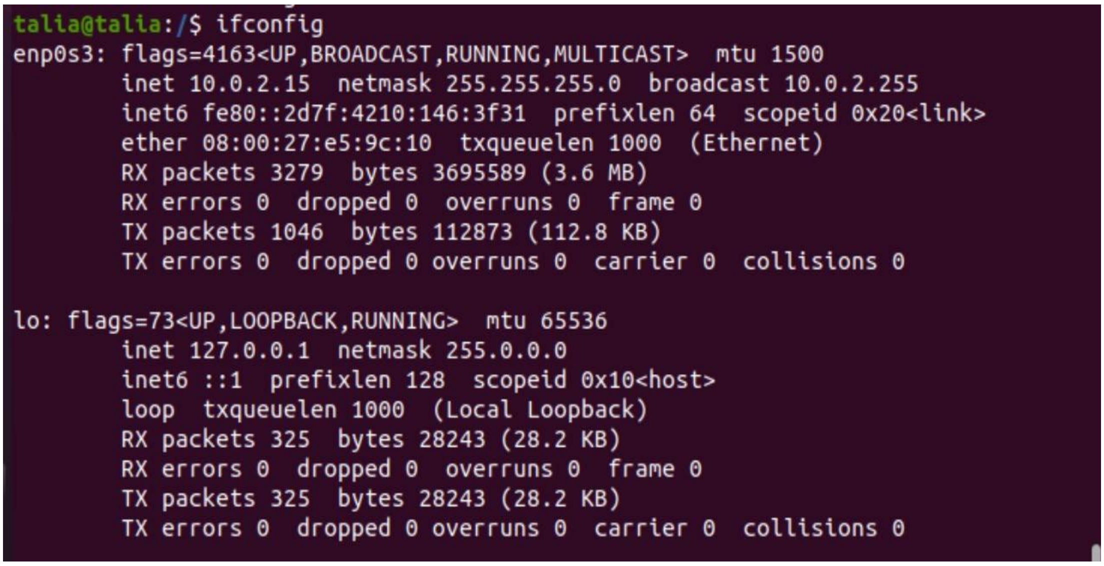


```
talia@talia:~$ lsmod
Module                  Size  Used by
can_dev                32768  0
vcan                   16384  0
can_raw                20480  0
can                    20480  1 can_raw
nls_utf8               16384  1
isofs                  49152  1
nls_iso8859_1          16384  1
intel_rapl_msr         20480  0
snd_intel8x0           45056  2
snd_ac97_codec        131072  1 snd_intel8x0
ac97_bus               16384  1 snd_ac97_codec
snd_pcm               106496  2 snd_intel8x0,snd_ac97_codec
snd_seq_midi           20480  0
snd_seq_midi_event     16384  1 snd_seq_midi
intel_rapl_common      24576  1 intel_rapl_msr
intel_powerclamp       20480  0
joydev                 24576  0
crct10dif_pclmul       16384  1
ghash_clmulni_intel    16384  0
snd_rawmidi            36864  1 snd_seq_midi
aesni_intel           372736  0
crypto_simd            16384  1 aesni_intel
```

```
snd_seq_device         16384  3 snd_seq,snd_seq_midi,snd_rawmidi
snd_timer              36864  2 snd_seq,snd_pcm
input_leds             16384  0
snd                    90112  11 snd_seq,snd_seq_device,snd_intel8x0,snd_timer,
snd_ac97_codec,snd_pcm,snd_rawmidi
serio_raw              20480  0
vboxguest             348160  0
soundcore              16384  1 snd
mac_hid                16384  0
sch_fq_codel           20480  2
vmwgfx                299008  2
ttm                   106496  1 vmwgfx
drm_kms_helper        184320  1 vmwgfx
fb_sys_fops            16384  1 drm_kms_helper
syscopyarea            16384  1 drm_kms_helper
sysfillrect            16384  1 drm_kms_helper
sysimgblt              16384  1 drm_kms_helper
parport_pc             40960  0
ppdev                  24576  0
lp                     20480  0
parport                53248  3 parport_pc,lp,ppdev
drm                   491520  5 vmwgfx,drm_kms_helper,ttm
ip_tables              32768  0
x_tables               40960  1 ip_tables
autofs4                45056  2
hid_generic            16384  0
usbhid                 57344  0
```

Figure 4.6 CAN-Module

Then the command for setting up the interface starts by using the ifconfig command to assign the usb interface: ifconfig can0. The example given is of the CAN bus message, Ifconfig van0 for the virtual interface (Figure 4.4).

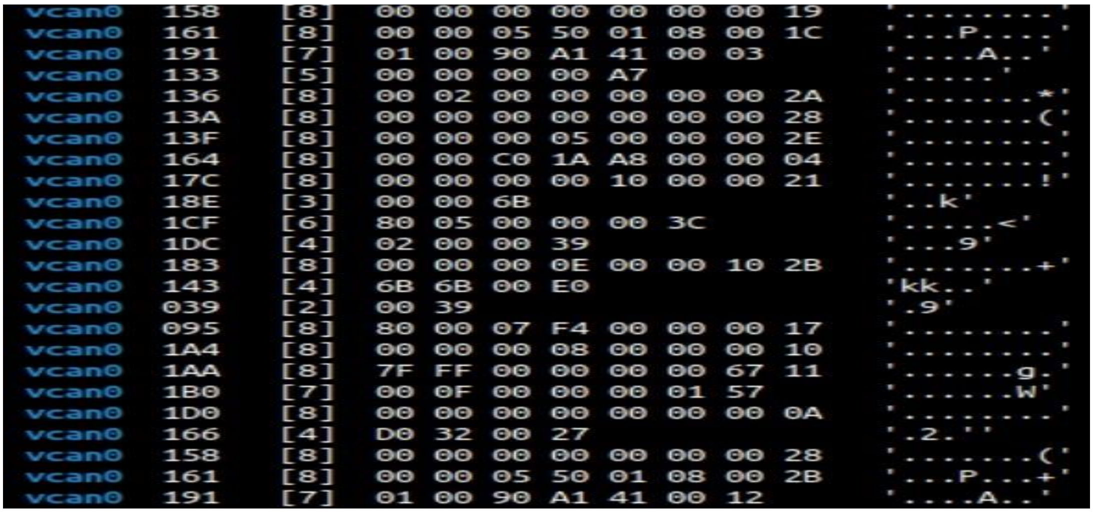


```
talia@talia:/$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::2d7f:4210:146:3f31  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:e5:9c:10  txqueuelen 1000  (Ethernet)
        RX packets 3279  bytes 3695589 (3.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1046  bytes 112873 (112.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 325  bytes 28243 (28.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 325  bytes 28243 (28.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 4.7: Physical Interface

The command set up is for the interface for bus speed, rates, bitrate, and so on.



```
  vcan0  158   [8]  00 00 00 00 00 00 00 19   '........'
  vcan0  161   [8]  00 00 05 50 01 08 00 1C   '...P....'
  vcan0  191   [7]  01 00 90 A1 41 00 03      '....A..'
  vcan0  133   [5]  00 00 00 00 A7            '.....'
  vcan0  136   [8]  00 02 00 00 00 00 00 2A   '.......*'
  vcan0  13A   [8]  00 00 00 00 00 00 00 28   '.......('
  vcan0  13F   [8]  00 00 00 05 00 00 00 2E   '........'
  vcan0  164   [8]  00 00 C0 1A A8 00 00 04   '........'
  vcan0  17C   [8]  00 00 00 00 10 00 00 21   '.......!'
  vcan0  18E   [3]  00 00 6B                  '..k'
  vcan0  1CF   [6]  80 05 00 00 00 3C         '.....<'
  vcan0  1DC   [4]  02 00 00 39               '...9'
  vcan0  183   [8]  00 00 00 0E 00 00 10 2B   '.......+'
  vcan0  143   [4]  6B 6B 00 E0               'kk..'
  vcan0  039   [2]  00 39                     '.9'
  vcan0  095   [8]  80 00 07 F4 00 00 00 17   '........'
  vcan0  1A4   [8]  00 00 00 08 00 00 00 10   '........'
  vcan0  1AA   [8]  7F FF 00 00 00 00 67 11   '......g.'
  vcan0  1B0   [7]  00 0F 00 00 00 01 57      '......W'
  vcan0  1D0   [8]  00 00 00 00 00 00 00 0A   '........'
  vcan0  166   [4]  D0 32 00 27               '.2.''
  vcan0  158   [8]  00 00 00 00 00 00 00 28   '.......('
  vcan0  161   [8]  00 00 05 50 01 08 00 2B   '...P...+'
  vcan0  191   [7]  01 00 90 A1 41 00 12      '....A..'
```

Figure4.10 Examples of CAN Messages

T command is: #sudo ip link set can0 type can bitrate 500000 can; and, #Sudo ip link set can0 or ifconfig can0 up. The CAN message, candump is the command tool used to view the CAN message. The content of the CAN message has message ID and the

Decimal value of the message contents, using: Command: #candump vcan0 (Figure 4.5). The ELM 327 usb interface OBD11 device is used to scan and read the fault codes of the vehicle but cannot write or edit into the vehicle system. Live data was captured and analyzed using Table 4.2. When an identified fault was identified and rectified then the fault code could be turned off. If the code function was simply toggled to on then the fault code reverted to off as a correction cycle. The van CAN retained no histories and the ELM327 only displayed a print out until the next one was requested. This feature over wrote several of the live print outs.

As the research progressed it was clear the tools used could not intervene in the CAN to change codes or to inject malicious codes. Other researchers have demonstrated a tool called "CAN Sniffer" that allows the researcher to capture the CAN codes in Hex and to alter them. In this work I could capture the codes and create a substitute but it had two limitations: (1) the tool could not inject code; and, (2) the CAN would not accept any codes that were not authenticated by the hash system. Obviously security mechanisms were at play that my tool could not crack. The hacking concept is appealing but it appears only to work under specific circumstances and with different tools.

4.2.2 Ford Car Data

The following data was collected from static analysis, and a separate data set was collected from mobile data. It recorded as the researcher drove without any distraction to the driver. The data analysis then happened after the vehicle was returned to parking and on the Laptop in the laboratory. The data showed a full search of the CAN and the identified error codes. These could only be reset after the fault was rectified. It correctly identified a fault in the ABS breaking sensors. The sensors were replaced and the code reset to off. There was limited history in the CAN but the real time data could be saved on the computer hard drive when the ELM327 was plugged in and docked with the software. The following readout shows the examination of each control module, and the status. The ABS module is found to be at fault and hence intervention is required and is permitted (Appendix B has the ID codes and explanation for ABS faults).

===PCM DTC None===

Successful DTC reading, no error codes found

Module: Powertrain Control Module

```
===END PCM DTC None===

===ACM DTC None===

Successful DTC reading, no error codes found

Module: Audio Control Module

===END ACM DTC None===

===IC DTC None===

Successful DTC reading, no error codes found

Module: Instrument Cluster

===END IC DTC None===

===RCM DTC None===

Successful DTC reading, no error codes found

Module: Restraint Control Module

===END RCM DTC None===

===ABS DTC C1236===

Code: C1236 - Left Rear Wheel Speed Sensor Input Signal Missing

Module: Antilock braking system

Diagnostic Trouble Code details

Left Rear Wheel Speed Sensor Input Signal Missing

This DTC may be caused by :

Open circuit

Sensor

Sensor Previously disconnected.

Short circuit

Damaged or contaminated connector

===END ABS DTC C1236===

===ABS DTC C1234===
```

Code: C1234 - Right Front Wheel Speed Sensor Input Signal Missing

Module: Antilock braking system

Diagnostic Trouble Code details

Right Front Wheel Speed Sensor Input Signal Missing

Open circuit

This DTC may be caused by:

Sensor

Sensor previously disconnected.

Short circuit

Damaged or contaminated connector

===END ABS DTC C1234===

===ABS DTC C1233===

Code: C1233 - Left Front Wheel Speed Sensor Input Signal Missing

Module: Antilock braking system

Diagnostic Trouble Code details

Left Front Wheel Speed Sensor Input Signal Missing

Short circuit

This DTC may be caused by:

Open circuit

Sensor

Sensor Previously disconnected.

Damaged or contaminated connector

--- Possible Causes --------

C1233 can be generated by:

-Missing LF Wheel Speed Sensor

- Dislocated or misaligned LF Wheel Speed Sensor

- Missing LF Wheel Speed Ring.

The code is set normally when the speed from one sensor differs from the speed on the other three sensors. The vehicle speed must exceed 20 kph (13 mph) for this code to be set.

===END ABS DTC C1233===

Table 4.2 reports the data analysis from the tests.

Table 4.3: Empirical Findings Summary

Attack	**Evidence**	**Comments**
Spoofing	Frame Rate	n/a = not visible
	Work Loads	Insertion of code prevented in both vehicles.
	Actions	Off/Only All else rejected.
Bluetooth	Packet Rates	n/a = Not visible
	Bottlenecks	n/a = physical connection slow. No Bluetooth.
	Insertions	n/a = Not available in these vehicles.
Sensor Jamming	Frame Rates	N/a = not visible
	Response Rate	n/a = only on/off which appear immediately a real failure.
	ECU Rates	n/a = not effected. Physical jamming is too slow to jam CAN.

Wireless Attacks	ECU Activity	n/a = wireless not available.
	Bottlenecks	n/a = wireless not available.
	Functionality	n/a = wireless not available.

4.3 CONCLUSION

In this Chapter 4 the results of the empirical and theoretical testing are reported. It was found that the two motor vehicles tested with the diagnostic tool had limited access to the CAN features. Everything was represented as a fault code and this could be altered as on or off. Other more advanced hacking equipment was not available to this research but it was concluded limited disruption to vehicle functionality could occur from the basic tools. Also, the vehicles available for testing had none of the wireless and Bluetooth features of more expensive models which precluded full testing. In addition the results of the theoretical testing showed that some attacks could not be performed but those that succeed delivered the expected change to packet and frame flows, and the expected information disruption, such as denial of service features. Chapter 5 now takes these findings and answers the research questions.

Chapter 5

DISCUSSION

5.0 INTRODUCTION

The idea of Chapter 5 is to discuss the findings from Chapter 4, and develop arguments based on analyzing and critiquing the theoretical thinking that was raised in the literature review in Chapter 2 and the previous similar studies outlined in Chapter 3. The Section **5.1** analyses the data of Chapter 4 to answer the sub-questions. Section 5.2 then answers the main research question. Section 5.3 provides a discussion of the findings against the back drop of Chapter 2 literature and the limitations of the research. Section 5.4 makes recommendations for improving CAN security.

5.1 THE RESEARCH SUB-QUESTIONS

This research is exploratory research that seeks to answer the research question:
What is required to assure CAN vehicle security? The several sub-questions help to provide better understandings that can assist answering the main question. The sub research question answers are described as follows:

Sub-question 1: What are the security attacks on automotive communication networks or Electronic Control Units (ECU)?

Sub-question 1	
Location	**Evidence**
Chapter 2.2	Bluetooth eavesdropping; buffer overflows; Code execution
Chapter 2.3	Jamming; Ultra sonic; blinding; trick stimulation

Chapter 2.4	Wireless attack; CAN access; ECU control; sensor over load
Chapter 2.1	Scanning; DoS; Toggling
Table 4.1	Scanning; DoS
Table 4.2	Spoofing
Answer: The security attacks on automotive communication networks or Electronic Control Units are defined by protocol, effect and technique. The evidence suggests that wired, wireless and Bluetooth connections are used to access a CAN. The attack vector can use information that directly effects code (hex) level communications, or it can indirectly influence the CAN by injecting trick or spoof data.	

Sub-question 2: What are the security vulnerabilities of ECUs?

Sub-question 2	
Location	**Evidence**
Section 3.1.1	Injection attacks, spoofed IDs, disrupting legitimate messages
Section 3.1.2	Reverse engineering and scanning
Section 3.1.3	Injection and AI packet generation

<table>
<tr><td>Table 4.1</td><td>DoS attack on ECU; Entertainment module attack vector vis ECU</td></tr>
<tr><td>Table 4.2</td><td>Physical jamming too slow to influence ECU</td></tr>
<tr><td colspan="2"><b>Answer:</b>
The security vulnerabilities of ECUs relate to the lack of protection from IDSs and encryption technologies. The ECU has little intelligence or protection unless these features are provided. An open ECU allows the compromise of the CAN and also the sensors and actuators. False actions can be executed and the CAN functionality disrupted when the ECU is compromised by any of the attacks listed here.</td></tr>
</table>

Sub-question 3: What countermeasures are available for security vulnerabilities in automotive communication networks?

<table>
<tr><th colspan="2">Sub-question 3</th></tr>
<tr><th>Location</th><th>Evidence</th></tr>
<tr><td>Figure 5.1</td><td>Including IDS and not only load balancing functions.</td></tr>
<tr><td>Figure 5.2</td><td>Including IDS at each ECU; plus firewalls on gateways.</td></tr>
<tr><td>Section 3.1.3</td><td>Deep learning detection systems.</td></tr>
<tr><td>Section 5.4</td><td>PKI Encryption technologies on each CAN system component.</td></tr>
</table>

Answer:

The countermeasures are available for security vulnerabilities in automotive communication networks are design issues that require implementation. Security countermeasures for CANs come at a cost that has an initial cost of design, implementation, and resources; and, an ongoing cost of slower information transactions and processing overheads.

Sub-question4: What management strategies are required for prevention and countermeasures in automotive communication networks?

Sub-question 4	
Location	**Evidence**
Figure 5.3	Full security design life-cycle.
Chapter 2	User training; manufacturer awareness; standards; forensic capability; user oversight.
Section 4.1	Knowledgeable operators who can intervene if anomalies appear and investigate.
Section 4.2	Regular hardware and software updates.

Answer:

The management strategies required for prevention and countermeasures in automotive communication networks are lifecycle maintenance designs. It starts with the manufacture of security by design (Figure 5.3) and then extends to a security management lifecycle (see ISO/IEC 27001) to protect the information use. Awareness of the vulnerabilities and security issues around CAN has to be a priority for all stakeholders.

5.2 THE RESEARCH QUESTION

This research is exploratory research that seeks to answer the research question: *What is required to assure CAN vehicle security?* Several sub-questions have also been answered in Section 5.1 and provide further evidence for answering the main question.

What is required to assure CAN vehicle security?	
Location	**Evidence**
SQ1	Security attacks on automotive communication networks are defined by protocol, effect and technique. The attack surface has increased radically as each new technology is added for CAN control.
SQ2	ECUs require protection from IDSs and encryption technologies. The use of deep learning technologies and innovative designs are required.
SQ3	Countermeasures are available for security vulnerabilities in CANs but responsibility has to be shared for paying the costs.

SQ4	Management strategies are critical to safety. They start with the manufacturer and extend to the user.
Answer: To assure CAN vehicle security each of the sub-question learnings delivers an overview of vulnerability, innovative solutions, the cost, and management issues. These four dimensions are critical for assurance and designing safety into autonomous and semi-autonomous vehicles.	

5.3 DISCUSSION OF FINDINGS

The outstanding issue with these findings is that there is a substantial gap between the theory presented in Chapter 2, and the practical investigation findings. The most pronounced difference is between the empirical findings and the theoretical attacks of Chapter 2. This is on account of the availability of motor vehicles to test, the tools used, and the time delays created by lockdowns. The 2005 and 2010 vehicles tested complied with the OBDII standard for CAN but they lacked many of the features required to test the theory reports of Bluetooth and Wireless hacking. The result is that Table 4 has many not appropriate (n/a) entries in the cells. The choice of vehicle is critical for testing all the theoretical attacks and this would require a Tesla or late model European vehicles. These vehicles were not available to the researcher. However, the data points out several critical understandings. The first is the rate of change in the motor industry, and the second is the scope of security challenges the latest vehicles have introduced.

The rate of change in the motor industry can been seen by the general adoption of advanced semi-autonomous features in lower priced vehicles today. For example, a medium priced SUV or car has self-parking, blind spot alert, navigation, and many other intelligent features that rely on sensor networks and CAN information co-ordination. The older vehicles used in this study, all had electronically assisted control of mechanical functions that were mediated by the CAN unit(s) but none had peripheral sensor networks or cloud connectivity. Similarly a 25 year old Holden excluded from the study had many driver assist functions such as cruise control, digital reporting and so on but no external connectivity other than a radio. These examples suggest that semi-autonomous controls have been in use in mid-range vehicles for at least a 15 year period before external connectivity and much higher levels of automation have been introduced in the last 10 years. It is these new and intelligent options that mediate human intervention and potentially fully automate a vehicle control that are of concern for security studies.

The rate of change is measured against the weighting of control given to the machine, and that allowed for the human. Fully automated train and tram systems have been in operation on dedicated lines, at airports for example, but the focus of this research is on the generalization of the core CAN control units into open systems where for example, fully automated vehicles use public highways. This is a different context and problem area for security studies. In open systems more variables impact decision-making and have potential to overwhelm the computational capacity and learning an artificial intelligence control unit may acquire. There are many highly publicized cases where human life has been lost because the machine has failed to learn and interpret a situation. The freeway Tesla case with a turning truck trailer, and a lady with a pushchair appeared to be unlearned phenomena. The theft of vehicle contents has also been achieved by false triggers of safety systems. The concern of this research is the protecting from unintended machine and human behavior that occurs when the intended control features go out of control.

The scope of security challenges faced by the new semi-autonomous vehicles require mitigation before fully autonomous vehicles are generally available. The issues raised in Chapter 2 specifically relate to the newly connected vehicles that have enhanced attack surfaces both internally and externally to the vehicle. Many of the mechanisms used for connectivity have gone through security upgrades to reach the current level of performance, but none is perfect. For example Bluetooth and Wireless networks have many cases of violation and the implementation of countermeasures. Navigation systems generally connect to GPS communications that also have many demonstrated hacks to disrupt the intended communication of information. Most mid-price range vehicles and above, connect directly or via a user mobile phone to the cloud. The information sent to the cloud is generally comprehensive of all the communications occurring in the vehicle. The cloud offers storage and information services that reduce costs for onboard services. It also allows manufacturers to monitor and to regulate whole fleets of their product for servicing and other alerts. The navigation map updates and hands free voice communication services are particularly useful for currency of knowledge. However, with all the clear communication features comes risks. These risks may range from privacy disclosures, to coercion for brand retention, to incorrect information transfers.

If used as intended the advancement in autonomous features and mediated human experience have great advantages for safe and secure vehicle use. This research has focused on the unintended outcomes of automating vehicle control systems. Much more use has to be made of CAN and IoT simulators to scenario test old and new designs for possible vulnerabilities before implementation. Many simulators lack a current security module or neglect updating for current attack libraries. Many current manufacturers are still focused on fast and efficient communication systems for rapid deployment and use, at the expense of effective systems that prioritize information security and intentional effects. The cost of failure is high. For example, the many global recalls for vehicle safety defects. The cost of failures can be mitigate by building stronger defenses and testing for failures before implementation and product release. This requires a balancing of the race to market to make money objective with the building of quality by design. The testing and correcting of security defects before use reduces the risk of future financial failures.

5.4 RECOMMENDATIONS FOR SECURITY FRAMEWORKS

Currently a CAN has a security architecture as shown in Figure 5.1. The CAN bus interfaces with the electronic control units (ECUs) and a mediation gateway (GW) for connected system communications. The ECUs then provide the specialized information services for the network of sensors and actuators. The system also has an on-board diagnostics (OBD) unit for a self-diagnostic and reporting capability. This is where the research ELM 327 tool connected. However, only recently some CAN architectures have an intrusion detection module that constantly scans the system for abnormalities and patterns of unplanned behaviours or intervention. In this section improvements to these architectures are suggested based on the research completed in Chapter 4, and the Chapter 2 readings.

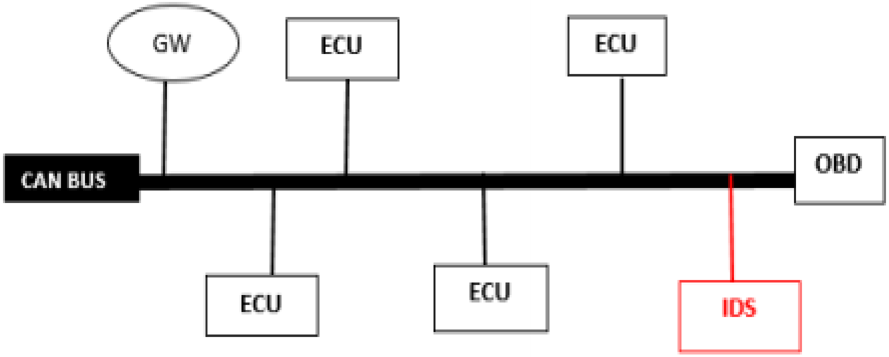


Figure 5.1: Current State of CAN Protection

In a CAN framework all the ECU's reside on the CAN bus and the OBD port has full access to the CAN bus. The IDS (if implemented) reports to the CAN bus any unauthorized CAN packets. The GW is the gateway also connects to the infotainment systems, such as Bluetooth, wireless, Wi-Fi, and so on. The first proposed security improvement measures are shown in Figure 5.2. Here the IDS taps directly into each ECU communication channel so that if a denial of service attack is being initiated through one sensor or actuator system then it is immediately detected, isolated and prevented from harming the whole system. In the most advanced security architectures shown in Figure 5.1 the IDS that monitors the whole system may not discriminate between ECUs, it may respond too slowly to stop an attack, and result in the whole system shutting down before error correction. The GW is also a vector for attack and should have its own IDS and error correction mechanisms. It should use a firewall (FW) to filter the traffic from infotainment system. This recommendation acknowledges that there are numerous costs in implementing more secure systems and they will have to be factored into the improved systems.

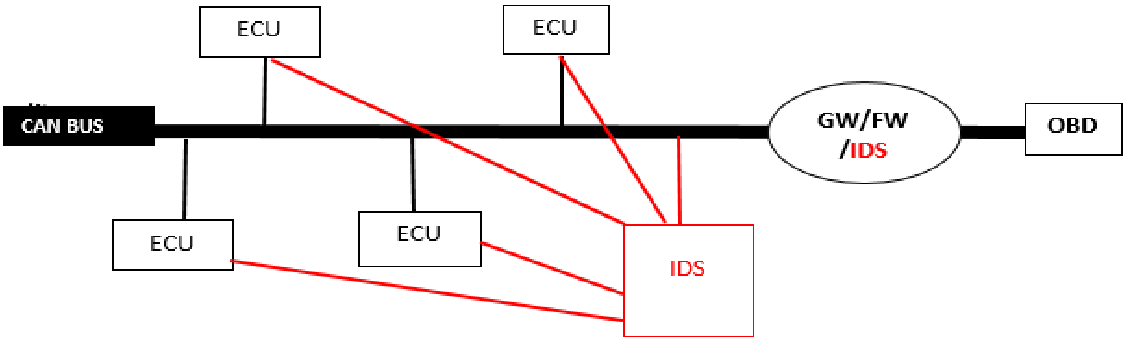


Figure 5.2: Suggestion for CAN Protection

In the bigger picture an overall plan for a Cyber Security Framework for Automotive systems can increase protection and assurance of safety. The proposed framework in Figure 5.3 was develop based on the analysis of literature in Chapter 2 and Chapter 3. The practical findings and experience in the CAN technologies also contributed skills, knowledge, and ideas. The attempt is to systematize actions engineers are to take in the design and build of CAN systems. By moving sequentially from phase 1 to phase 6 a full spectrum of risks are treated. Mitigation comes from the comprehensive scope of the framework and the specific dimensions of each. Each phase is itemized and explained below Figure 5.3.

A further extension to these recommendations is that all communications are encrypted within the CAN. This would mean the distribution of private keys to the ECU units and where possible to the sensors and actuators. The innovation would require increasing to processing power at different locations within the CAN and a public key exchange mechanism overhead. The addition would add a layer of protection not often present in current CAN architectures. Chapter 2 discussed these general concepts and the design for key distributions, refreshing, and so on, but the recommendation is for the extension and refinement of these technologies to all components, including sensors (ie. Each with a private key and secure refresh sequences.)

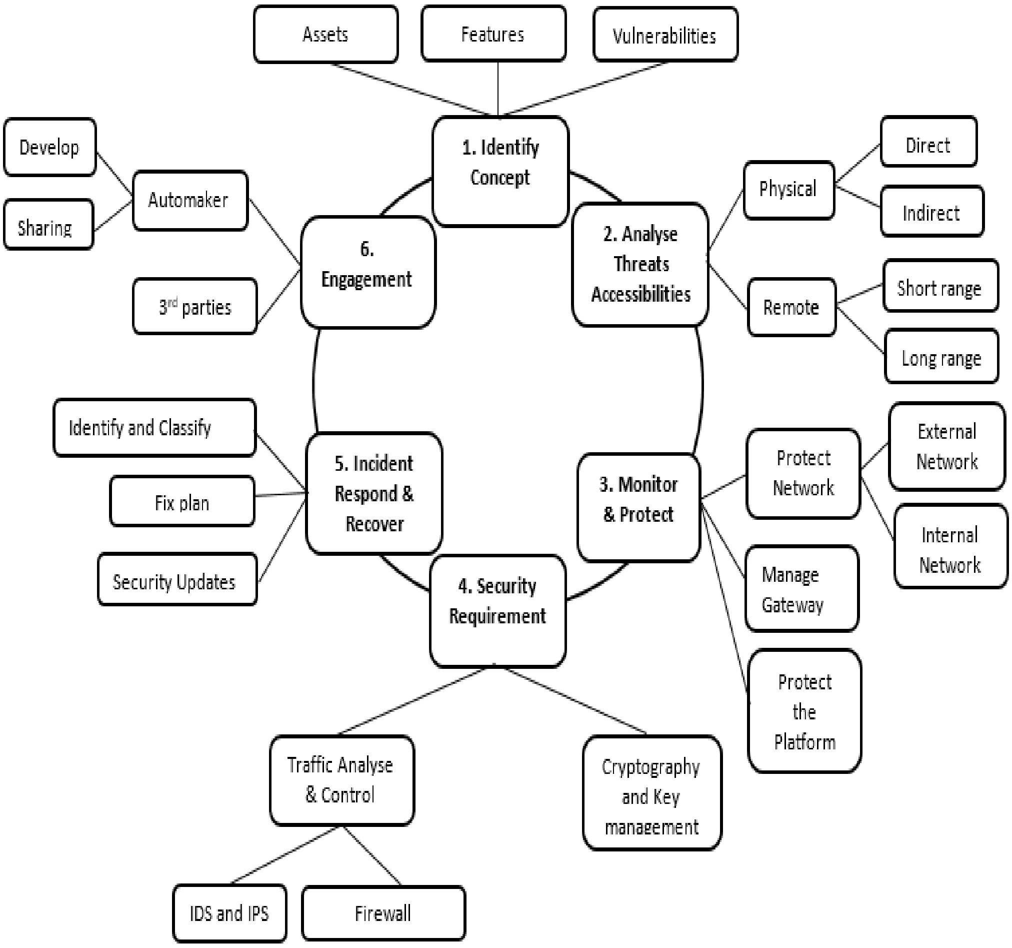


Figure 5.3: Proposed Security framework for CAN

Identify Concept: The first phase is to identify the concept of the vehicle. In terms of vehicles information security this is not just for a mechanic equipment but also the automakers and developers use design of the vehicles. The automakers attach equipment into the vehicle with a purpose to communicate with other devices. Some of the equipment communicates with the whole vehicle and all services in the vehicle, others have limited reach. The framework for Identify concept is divided into three parts: vehicle assets, features and vulnerabilities.

- Vehicle Assets

The first step is to identity the vehicle assets. In this step it helps to know and be aware of what are the vehicle assets and automotive systems are. There a lots of physical components in vehicles, such as the engine, OBD11, body control, network communication control, etc. The vehicle assets are based on the vehicle model, where different models have different assets.

- Features

In the vehicle there is equipment or devices attach into and from the automotive system in order for the car to connect to the peripheral services, such as telematics services, infotainment, navigation etc. In this phase the design has to identifying the features of the attached equipment in to and from the vehicle system, such as ECU, sensor, and CAN.

- Vulnerabilities

The last phase of Identify Concept is identify the vulnerabilities of the vehicle assets and the vulnerabilities of add on equipment features. Look at the vulnerability of physical components such as metal, hardware and software used by the car and the protocol vulnerabilities used by add on devices or features for vehicles.

Analyses Threats Accessibilities: In the previous phase the vehicle assets and it features and the vulnerabilities were identified. In this phase threats of access into vehicle system is considered. These are the threats for access being: physical access and remote access.

- Physical

Physical threat access has a two types: direct access and physical indirect access. Indirect access is access by using the physical connection interface to access to vehicle system such as CD or USP port to connect to the Infotainment functions and the interface port for mobile phone to connect to telematics functions. The direct access is when the attackers plug hardware straight into the OBD11 port that connects straight into in-vehicle LAN. All the communication through vehicle is done through in-vehicle LAN. The paths to in-vehicle LAN include the On Board Diagnostics (OBD11) functions which have physical ports for an attack via in-vehicle LAN.

- Remote

Attackers can remotely access through the wireless interface of the vehicle. Analyzing the wireless interface technology vulnerability of a vehicle will warn of remote access threats and access vulnerabilities for the attackers. Remote access has two types: short range and long range. Short range is approximately between 5 to 300 meters, and long range depends on the technology channel. For example, Bluetooth, Wi-Fi, RFID, and

Remote Keyless access, and so on. The Long range is usually by wireless technology that is able to cover a long distance of 1 km or more. It includes the broadcast channels and addressable functions.

Protect & Monitor: This phase focuses on protecting and monitoring the vehicles network for threats. In the vehicle architecture there are three areas that need protection and monitoring in order to protect the vehicle network. First is to secure the two main networks which are the external network and the internal network. The second is to manage or secure the gateways, and the third is to secure the platform.

- Protect the network

Securing the vehicle is to secure the external network and the internal network. The external network is controlled by the communication from the external network. It has to make sure the extranet network that communicates with the vehicle is secure and is trusted. The Internal Network has to protect and control the internal functions that include the delivery systems such as the Control Area Network (CAN). Confidentiality and Integrity of the message, and key management, are critical.

- Manage the Gateway

Protect and monitor the gateway is to control all the traffic follow coming toward the vehicle gateway. By controlling the gateway only access to the authorized traffic is given. Therefore it is able to detect malicious traffic and unauthorized access to vehicle system including data security and privacy.

- Secure the platform

Secure the platform requires paying attention to the software and hardware security for the vehicle. This includes electronic equipment such as the ECU, the program or code that run or write on the ECU. Also update the booting to verify the firmware and Operating system.

Security Requirement: The phase of security requirement has the list of requirements that are need to be used in order to protect vehicle from threats. These include traffic analysis and control, and Encryption and key management security.

- Traffic Analyze & Control

This step analyses and controls the traffic coming toward the vehicle through the external network and internal network, and the traffic coming through the gateways. The security requirement is for an advanced firewall and IDS and IPS system. The firewall and IDS and IPS system will control the access to the vehicle network and detect and prevent the unknown access and malicious traffic coming to the vehicle network. It also limits access to the end users.

- Encryption and key management security

Encryption and key management is focused on the confidentiality of the data transmission in the vehicle network. By using advanced and strong encryption to secure the vehicle network communication protection can be gained. These include the AES128 and CAN crypt algorithms.

Incident Respond and Recover: A plan to respond and recover from a cyber-security incidents is required. This phase handles cyber security incidences. The proposed framework is divided in to three parts: identify and classify, fix plan, and update.

- Identify and classify

In the event of incidence, the first step is to identify the incidence and classify it as what kind of cyber security incidence it fits. Identification of the incidence and what kind of impacts are critical to assess.

- Fix plan

Ensure there is an effective plan to respond to all cyber incidences. This includes special incidence teams with specific tasks, documentation, time specifications the incidence teams to provide a solution, and testing and verification of the solution.

- Security Updates

The final step, is to check the solution has been verify from the incidence teams and then to update the Security. Always make reports or documents from each security update.

Engagement: Building a relationship with the automakers and third parties for awareness of cyber threats and incidences. Also positively manage and control the issues of vehicle cyber security threats. Create events to collaborate with automakers and third parties to share information about the security threats. This proposed framework points out the necessity of engagement, sharing and training.

- Sharing

Engage with automakers security experts to share information about cyber security threats in vehicle systems. The sharing helps to identify more threats and more vulnerabilities, and to promote more interesting research projects between automakers and third parties for vehicle cyber security.

- Training

Providing training for the new vehicle technology. Hence, the users are up to date and aware of threats. Automakers need to design a program and share with third parties and users for awareness of cyber security in the new vehicles.

5.5 CONCLUSION

Chapter 5 has answered the research question and sub questions. These outcomes have been discussed in the light of the original objectives for the thesis and the limitations imposed by the lockdowns and tools. The findings also leave unfinished work and further leads for research. These will be discussed in Chapter 6.

Chapter 6
CONCLUSION

6.0 INTRODUCTION

Chapter 6 reviews what has been achieved in this research against what was intended when the research started. The first section lists the research contributions and also the limitations to transferring the findings to other contexts or generalizing outcomes. Some of the obstacles encountered in the practical work are noted, and the impact of the COVID-19 lockdown. This project was undertaken as exploratory research so the benefits are also for the researcher. Personal learning is reported in the second section. Finally the Thesis is concluded in the third section by listing potential starting points for further research. These starting points begin with what has been learned while doing this research and suggest what can be further explored to gain a fuller understanding of CAN security.

6.1 RESEARCH CONTRIBUTION AND LIMITATIONS

The limitations for this research are noted. Some were beyond the researcher control (such as the University shutting down for 8 weeks and then again for 4 weeks. No access to laboratories during these times.), and others were on account of the equipment used for testing. The equipment used for testing demonstrated the limits of each choice but eliminated many assertions that are found in the literature. The delays to getting equipment was aggravated by finally getting equipment that was defective. The researcher had to improvise and borrow from other researchers enough gear to get the testing done. The scope of each tool limited the range of data choices and delivered a set of results within the tool performance range. These limitations leave further work for future research and starting points for other researchers. The key contributions of this thesis are:

- Risk assessment for CAN systems
- CAN security issues
- Potential secure designs
- Identification of legal and manufacture increased securitymeasures (progressive)
- The Hacker challenge of keeping up with the security changes

- The challenge for continuous improvement of security designs and performances
- Remaining vulnerabilities and dramatic attacks on virtual environments
- Responsive assessment of changes in CAN design and security performances
- Gaps for further research

The limitations imposed on the research were both time constraints and tool availability. The time limitations came because the COVID-19 pandemic shut the University on two occasions for extended periods of time. This meant that there was no access to the laboratory and that the lab technicians were not available to purchase and supply the necessary equipment. When it finally became available some of it did not work and other alternatives had to be improvised. These delays were helped by a three month extension but it did not compensate for the lost momentum and focus on the testing environment build. The incorrect tools and the loss of time impacted on what could be achieved. This is seen in the incomplete part of Figure 3.1 the research design. The new and proposed security framework (Figure 5.3) has not been tested or evaluated for effectiveness or efficiency as originally planned. The thesis had to be completed after the "Design New Security Framework" phase. Hence, the security framework evaluation is proposed for future research.

6.2 PERSONAL LEARNING

I started this exploratory research with a limited knowledge of CAN bus or how the electronics of a vehicle work. I had been exposed to media and University security papers that suggested the modern cars, which are moving towards autonomy, have serious defects with negative impacts for safety. Hence, I had high motivation (Section 1.3) to learn and to discover both the technical and the associated social knowledge. As the research progressed I learned most of the theoretical knowledge from the literature review (Chapter 2) and the similar study analysis (Section 3.1). The practical knowledge came quickly as I obtained a diagnostic tool and started using it on motor vehicles. This was a combination of software mastery and physical connectivity challenges. The software simulator also gave challenges for learning. First I had to select a platform from the many on GitHub for Raspberry Pies or the free commercial grade simulators. Each require significant mounting and customization to gain functionality. While using both the hardware and software there were always things that did not function as expected and I had to work out solutions.

By learning and understanding of the components of the vehicle network communication such CAN bus and how CAN bus communication works, I was able to apply security learning and analysis. This gave detail of identifying the different types of attack in the vehicle network and how the attack access into the vehicle networks became possible. The study is incomplete but it has given significant insight into the original problem area and the potential countermeasures. The following is a list if key personal learning points gained from this research:

- The history of CAN technology developments
- Research methodology planning and application
- Risk Management Framework
- How to write a research report
- The use of tools
- The limitations of tools
- The software integrations and configurations
- The reading of methodology (initially Design Science)
- Understanding the requirement for a strong methodology so findings can be justified
- The strength of and rapidly changing manufacturer security provisions
- Techniques for doing the research
- Learning differ types of vehicle technology
- Learning different motor vehicle industry security procedures
- Finding some of the extreme hacking claims lacked technical details
- Learning the different types of motor vehicle attacks
- Showing the ability to reproduce findings confirms claims
- The importance of building quality improvement techniques into research designs
- Practical skills and techniques for the motor industry
- Learning the trouble code or fault code meanings
- Learning how to critique others research findings

6.3 FUTURE WORK

Future research has to consider the speed with which the security world moves. It is always a cat and mouse game between adversaries so what holds today can be changed tomorrow. In this research it was found hacks and intervention methods for CAN access have changed from those published at earlier dates. For example, the capability of diagnostic tools to access and manipulate the CAN environment is now limited in scope. Today diagnostic tools prevent the changing of fault codes when a vehicle is in motion but still provide static and dynamic code reporting. Also more advanced CAN sniffing tools are required to store and manipulate fault codes at hex level, and most CAN permissions now prevent injection or spoofing attacks.

But, on the hand the competition between the attackers and the security defense in the field of technology security for vehicle cybersecurity is still going on. It is because, when the security is more complex and stronger that also helps and makes the attackers get smarter in order to break through the security. Like for example, modern vehicles have lots of security features that are designed to help protect the vehicle assets such alarm, camera and automatic lock etc. But the attackers are still able to break through those security features and steal the vehicle by using new tricks. At the same time as the vehicle security features were not able to handle their jobs in order to protect the vehicle, a new challenge is given to the manufacturer security engineers. In this study the cat and mouse competition was clear and it will continue until people change their selfish behaviours. The following recommendations for further research are based on what has been achieved in this research and reflection on where it should go in the future:

- Join online forums for the development and use of CAN tools
- Look at device complexity and performance issues
- Performing a remote hacks on attack via Bluetooth or Wi-Fi
- Man in the middle attack in the vehicle network
- Get a sniffer tool and rerun the tests in this research
- Explore histories and privacy issues
- Look at ECU analysis level and communication protocol
- Analysis the vulnerabilities of external interface of vehicle
- Management of the public and private key infra-structures

- Possible cryptographic improvements
- Regular vulnerability and countermeasure assessment for new CAN architectures
- Tool education and certification opportunities
- Connection with some of the top international industry that work on the vehicle security for more relative information such as:
    - SAE International Vehicle Electrical System Security Committee
    - Automotive Security Guidelines
    - Risk Development Task Force
- Standardized testing plans
- Research on modern security threats in modern motor vehicle
- Trying to perform a hack project on a new modern vehicle
- Evaluation of the proposed security framework
- Greater consideration of changing human criminal behaviours

One of the major dreams of this study is hoping that this research will inspired other researchers. There are other car hacking projects to be done and to advance the cause of motor vehicle cybersecurity. Traditionally, when researchers identified motor vehicle cyber security vulnerabilities, the automakers were forced to overcome that problem by improving the vehicle security. The vulnerabilities in automotive security continue to change, and researchers or the research community can make a positive impact on this vehicle cybersecurity environment. Therefore, it is hoped that this research will prompt the current and the future cybersecurity researchers to undertake their own vehicle hacking projects to improve safety and the automaker industry performance in general. Secure and safe vehicle systems require the motivation of all the stakeholders.

REFERENCES

Auto1: Automotive Cyber Security Best Practices, (2016). Auto Tech Review, 5(8), 20.

Auto2: Automotive Information Sharing and Analysis Center (AUTO - ISAC), (2016) Retrieved from: https://www.automotiveisac.com/best-practices/

CANcrypt (2017). Retrieved from: https://www.cancrypt.eu/index.php/en/

CSS Electronics, (2018). "*CAN Bus Explained—A Simple Intro*." Retrieved from https://www.csselectronics.com/ screen/page/simple-intro-to-can-bus/language/en

Cui, J., Sabaliauskaite, G., Liew, L. S., Zhou, F., & Zhang, B. (2019). Collaborative Analysis Framework of Safety and Security for Autonomous Vehicles. *IEEE Access*, *7*, 148672-148683

Delhi, S. I. N. (2016). Automotive Cyber Security Best Practices. *Auto Tech Review*, *5*(8), 20-25.

Guo S., (2011). "*The Application of CAN-bus Technology in the Vehicle*". International Conference on Mechatronic Science, Electric Engineering and Computer, pp. 755- 758.

He, M., Luo, H., Chang, Z., & Hui, B. (2017). Pedestrian detection with semantic regions of interest. *Sensors*, *17*(11), 2699. Doi: 10.3390/s17112699

Johansson, K. H., Törngren, M., & Nielsen, L. (2005). "*Vehicle applications of controller area network"*. In Handbook of networked and embedded control systems (pp. 741-765). Birkhäuser Boston.

Morales, N., Arnay, R., Toledo, J., Morell, A., & Acosta, L. (2016*). "Safe and reliable navigation in crowded unstructured pedestrian areas".* Engineering Applications of Artificial Intelligence, 49, 74-87. doi:10.1016/j.engappai.2015.11.008

Morales, Y., Takeuchi, E., Carballo, A., Tokunaga, W., Kuniyoshi, H., Aburadani, A., Tsubouchi, T. (2008). *"1Km autonomous robot navigation on outdoor pedestrian paths running the Tsukuba challenge 2007*". In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 219-225). IEEE

Murtra, A. C., Trulls, E., Sandoval, O., Pérez-Ibarz, J., Vasquez, D., Mirats-Tur, J. M., & Sanfeliu, A. (2010). *"Autonomous navigation for urban service mobile robots".*

IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4141-4146). IEEE.

Murtra, A. C., Trulls, E., Tur, J. M. M., & Sanfeliu, A. (2010). Efficient use of 3D environment models for mobile robot simulation and localization. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots* (pp. 461-472). Springer, Berlin, Heidelberg.

Navarro, P. J., Fernandez, C., Borraz, R., & Alonso, D. (2017). *"A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data"*. Sensors, *17*(1), 18. Doi: 10.3390/s17010018

NZ Transport Agency. (n.d.). Testing autonomous vehicles in New Zealand. Retrieved from https://www.nzta.govt.nz/vehicles/vehicle-types/automated-and-autonomousvehicles/testing-autonomous-vehicles-in-new-zealand/

Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016*). "A survey of motion planning and control techniques for self-driving urban vehicles"*. IEEE Transactions on intelligent vehicles, 1(1), 33-55.

París, D. L., & Brazalez, A. (2009). *"A new autonomous agent approach for the simulation of pedestrians in urban environments"*. Integrated Computer-Aided Engineering, 16(4), 283-297. Doi: 10.3233/ICA-2009-0320

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, E. (2011). *"Scikit-learn: Machine learning in python"*. Journal of Machine Learning Research, 12, 2825-2830.

Perez, L., & Wang, J. (2017). *"The effectiveness of data augmentation in image classification using deep learning*". arXiv preprint arXiv: 1712.04621.

Petit, J., & Shladover, S. E. (2015*). "Potential cyberattacks on automated vehicles"*. IEEE Transactions on Intelligent Transportation Systems, 16(2), 546-556

Petit, J., Stottelaar, B., Feiri, M., & Kargl, F. (2015). *"Remote attacks on automated vehicles sensors: Experiments on camera and lidar"*. Black Hat Europe, 11, 2015.

Punyawiwat, P. (2018). *"Interns explain basic neural network"*. Retrieved from https://blog.datawow.io/interns-explain-basic-neural-network-ebc555708c9

Qiang Hu, Feng Luo (2018), "*Review of Secure Communication Approaches for In-Vehicle Network* "International Journal of Automotive Technology, Volume 19, Number 5, Page 879

Rahman, A. H. A., Ariffin, K. A. Z., Sani, N. S., & Zamzuri, H. (2017*). "Pedestrian detection using triple laser range finders"*. International Journal of Electrical & Computer Engineering (2088-8708), 7(6), 3037-3045.

Roderick (2016). *"Developments in Car Hacking."* SANS Institute

Rosén, E., & Sander, U. (2009). *"Pedestrian fatality risk as a function of car impact speed".* Accident Analysis & Prevention, 41(3), 536-542.

Rosén, E., Källhammer, J.-E., Eriksson, D., Nentwich, M., Fredriksson, R., & Smith, K. (2010). *"Pedestrian injury mitigation by autonomous braking".* Accident Analysis & Prevention, 42(6), 1949-1957.

Ruder, S. (2016). "*An overview of gradient descent optimization algorithms"*. Retrieved from https://ui.adsabs.harvard.edu/abs/2016arXiv160904747R

Schlüter, J., & Grill, T. (2015). *"Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks".* In *ISMIR* (pp. 121-126).

Seung-Jun, H., & Jeongdan, C. (2014). "*Real-time precision vehicle localization using numerical maps"*. ETRI Journal, 36(6), 968-978. doi:10.4218/etrij.14.0114.0040

Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2017). *On a formal model of safe and scalable self-driving cars*. ArXiv preprint arXiv: 1708.06374.

Shiomi, M., Zanlungo, F., Hayashi, K., & Kanda, T. (2014). *Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model*. International Journal of Social Robotics, 6(3), 443-455.

Silver, D. (2018). *How localization works for self-driving cars*. Retrieved from https://www.linkedin.com/pulse/how-localization-works-self-driving-cars-david-silver/

Simonyan, K., & Zisserman, A. (2014). *"Very deep convolutional networks for large-scale image recognition"*. ArXiv preprint arXiv: 1409.1556.

Skrzypczyk, K., & Mellado, M. (2017). "Vehicle navigation in populated areas using predictive control with environmental uncertainty handling". Archives of Control Sciences, 27(2), 351-359.

Song, D., Li, J., Ma, Z., Li, Y., Zhao, J., and Liu, W., (2016). *'Application of CAN in Vehicle Traction Control System*'. In IEEE International Conference on Vehicular Electronics and Safety, pp.188-192

Stanley, K. O., & Miikkulainen, R. (2002). "*Evolving neural networks through augmenting topologie*s". Evolutionary computation, 10(2), 99-127.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Taylor, L., & Nitschke, G. (2017*). “Improving deep learning using generic data augmentation”*. ArXiv preprint arXiv: 1708.06020.

Tharp, K. J., & Tsongos, N. G. (1977). “*Injury severity factors-traffic pedestrian collisions”* (No. 770093). SAE Technical Paper. doi.org/10.4271/770093

Tomè, D., Monti, F., Baroffio, L., Bondi, L., Tagliasacchi, M., & Tubaro, S. (2016). “*Deep convolutional neural networks for pedestrian detection”.* Signal Processing: Image Communication, 47, 482-489. doi:10.1016/j.image.2016.05.007

Van Roermund, T. (2019) “*In-Vehicle Networks and Security*” In: Dajsuren Y., van den Brand M. (Eds) Automotive Systems and Software Engineering. Springer, Cham. Retrieved from: https://doi.org/10.1007/978-3-030-12157-0_12

Van Roermund, T. (2019). *“In-Vehicle Networks and Security”*. In Automotive Systems and Software Engineering (pp. 265-282). Springer, Cham.

Wade, M. (2018). Silicon Valley is winning the race to build the first driverless car. *International Institute for Management Development (IMD), Lozana, Švajcarska, Februar*. Retrieved from https://theconversation.com/silicon-valley-is-winning-the-race-to-build-the-first-driverlesscars-91949

Wang, W., Song, Y., Zhang, J., & Deng, H. (2014). “*Automatic parking of vehicles.”* International Journal of Automotive Technology, 15(6), 967-978. doi: 10.1007/s12239-014-0102-y

Welch, D., & Behrmann, E. (2018). “*Who’s winning the Self-Driving Car Race?”* Bloomberg.com. Retrieved from: https://www.bloomberg.com/news/features/2018-05-07/who-s-winning-the-self-driving-carrace

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). *Understanding data augmentation for classification: when to warp?* International conference on digital image computing: techniques and applications (DICTA), (pp. 1-6). IEEE

Xiaoli, M., Heng, W., & Bingbing, L. (2017). A robust vehicle localization approach based on GNSS/IMU/DMI/LiDAR sensor fusion for autonomous vehicles. Sensors (14248220), 17(9), 2140. Doi: 10.3390/s17092140

Yan, C., Xu, W., & Liu, J. (2016). Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON*, *24*(8), 109.

Yoshida, J. (2015). *CAN Bus Can Be Encrypted*, Says Trillium. Retrieved from http://www.eetimes.com/document.asp?doc_id=1328081

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. ArXiv e-prints. Retrieved from https://ui.adsabs.harvard.edu/abs/2012arXiv1212.5701Z

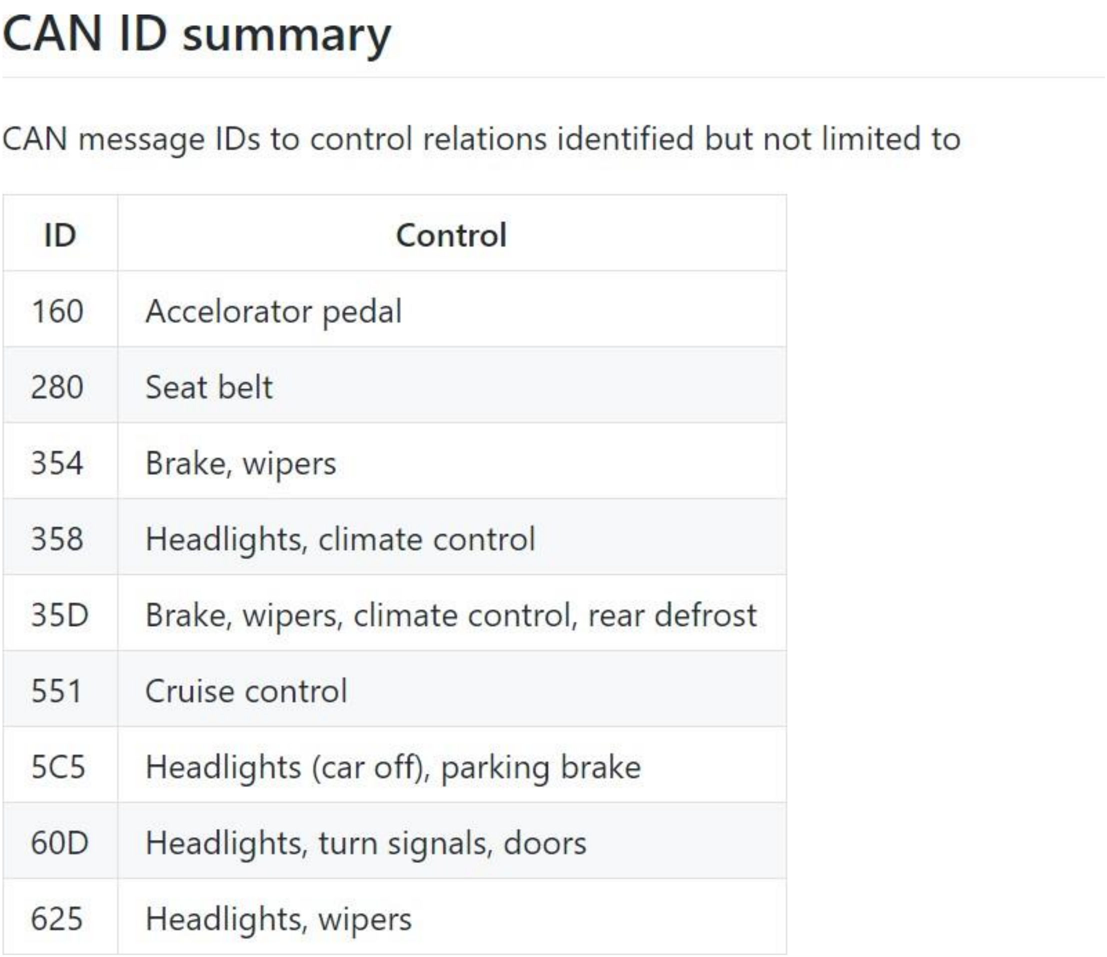
Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020). Random Erasing Data Augmentation. In *AAAI* (pp. 13001-13008).

Zhou Y., Wang, X. and Zhou, M. (2006). *"The Research and Realization for Passenger Car CAN Bus".* In the 1st International Forum on Strategic Technology. 18-20 Oct. 2006, pp. 244-247.

Zulueta, A. G. (2013). *"Cooperative social robots: Accompanying, guiding and interacting with people"*. PhD thesis. Universitat Politecnica de Catalunya,

APPENDIX A

Nissan Sentra 2010 CAN bus data



CAN ID summary

CAN message IDs to control relations identified but not limited to

ID	Control
160	Accelorator pedal
280	Seat belt
354	Brake, wipers
358	Headlights, climate control
35D	Brake, wipers, climate control, rear defrost
551	Cruise control
5C5	Headlights (car off), parking brake
60D	Headlights, turn signals, doors
625	Headlights, wipers

Key inserted, ACC off, engine off

CAN bus load 7% when key inserted or removed, messages from the following IDs only last a few seconds then stop sending

- **358**
- **35D**
- **625**

When key moved to first position (before ACC), messages from the following IDs are seen periodically and constantly

- **35D**
- 60D
- **625**

When vehicle is off (no key inserted at all) and a control is activated (one that is allowed to be used when the vehicle is off; eg. door open, hazard & head lights) messages with the following IDs are seen in addition to the usual ones associated with said control

- **35D**
    - B.1-B.2 **=> 0b00=doors closed, 0b11=driver door opened**
- **625**
    - **DLC=6**
    - D.1-D.2 **=> 0b10=doors closed, 0b01=any door opened**

ACC on, engine off

CAN bus load 30% Messages from the following IDs always actively changing

- **174**
    - E **=> cycles through values 0x01 0x04 0x05 0x09 0x0C 0x0D**
- **176**
    - **DLC=7**
    - G **=> cycles through values 0x01 0x05 0x06 0x09 0x0D**
- **180**
- **182**
- **280**
- **284**
- **285**
- **551**
    - A.1-A.2 **=> cycles through values 0b01 0b10**
- **560**
    - **DLC=3**
- **6E2**
    - **DLC=3**
    - C **=> cycles through values 0x78 0x7A and 0x7B 0x79**

Engine running

CAN bus load 30% Messages from the following IDs always actively changing. Bolded IDs are newly seen relative to when ACC on.

- 160
- **174**
- **176**
- **180**
- **182**
- 1F9
- **280**
- **284**
- **285**
- **551**
- **560**
- **6E2**

Vehicle controls

11 Accelerator pedal

- **160**
    - E–H **=> seem to correspond to pedal position**
    - A–B **=> 0x3D64 when pedal not pressed, increases to 0x41D4 when fully pressed**
    - C **=> always 0xA1**

21 Brake pedal

- **354**
    - G.5 **=> 0=not pressed, 1=pressed**
- **35D**
    - E.5 **=> 0=not pressed, 1=pressed**

31 Steering wheel angle

No change in messages

Only tried with ACC on, engine off

41 Vehicle lights

Forward facing lights

Possible states are: off, headlights on, fog lights on, high beams on

- **358**
  - B.8 **=> 0=off, 1=any light on**
- **60D**
  - A.2-A.3 **=> 0b00=off, 0b10=headlights on, 0b11=fog lights on**
- **60D**
  - B.4 **=> 0=high beams off, 1=high beams on**
- **625**
  - **DLC=6**
  - B.5-B.7 **=> 0b000=off, 0b100=headlights on, 0b110=fog lights on, 0b001=high beams on**

When car is off (no key inserted)

- **5C5**
  - A.7-A.8 **=> 0b10=off, 0b01=any light on**

51 Windshield wipers

Wiper control stick

Possible states are: off, oneshot ("MIST"), intermittent ("INT"), slow ("LO"), fast ("HI")

- **35D**
  - C.6-C.8 **=> 0b000=off, 0b110=oneshot, 0b010=intermittent, 0b110=slow, 0b111=fast**
  - **Bits for oneshot and slow settings are the same**
- **354**
  - **appears during intermittent wiper setting**
  - E.2 **=> 0=off, 1=intermittent, ...**
- **625**
  - A.2-A.3 **=> 0b01=off, 0b10=intermittent, ...**

61 Turn signals

Turn signal control stick

Data bytes change each time turn signal ticks/lights up

- **60D**
  - B.6-B.7 **=> 0b00=off tick, 0b01=left turn tick, 0b10=right turn tick**

71 Hazard lights

Data bytes change each time lights are flashed on and off

- **60D**
  - B.6-B.7 **=> 0b00=off, 0b11=blink on**

This is really just a combination of both the left and right turn signal lights. Same bits seen when using turn signal control stick.

81 Cruise control

Cruise control on/off button on steering wheel

- **551**
  - F.5-F.7 **=> 0b000=off, 0b101=on**

Did not test setting cruise control speed or increasing/reducing speed setting, this would require the vehicle to be moving (minimum 40 km/h before cruise control can be active)

91 Horn

No change in messages

101 Climate control fan

- **358**
  - B.7 **=> 0=off, 1=fan on (any speed)**
- **35D**
  - A.1 **=> 0=off, 1=fan on (any speed)**

No other climate control setting showed any change in messages

111 Rear defrost

Rear window defrost on/off button

- **35D**
    - A.2-A.3 **=> 0b00=off, 0b11=on**

121 Door ajar

- **60D**
    - A.4 **driver side front door => 0=closed, 1=open**
    - A.5 **passenger side front door => 0=closed, 1=open**
    - A.6 **driver side rear door => 0=closed, 1=open**
    - A.7 **passenger side rear door => 0=closed, 1=open**

131 Seat belt

When seat belt is clicked in for the driver

- **280**
    - A.2 **=> 0=clicked in, 1=not clicked in**

Most likely for enabling the seat belt indicator light on the dash

Also tried passenger seat belt but saw no change in messages

141 Parking brake

- **5C5**
    - A.3 **=> 0=off, 1=parking brake engaged**

151 Radio buttons

No change in messages

Unlikely to be on the same bus as the rest of the vehicle controls

161 Power windows

No change in messages

Tried lowering and raising all four windows

171 Power door locks

No change in messages

Tried locking and unlocking all four doors

181 Power side mirrors

No change in messages

Tried adjusting both left and right side mirrors

191 Cabin lights

No change in messages

Tried turning on front cabin light and back seat cabin light

201 Gas cap release

No change in messages

Likely just a mechanical release

211 Trunk release

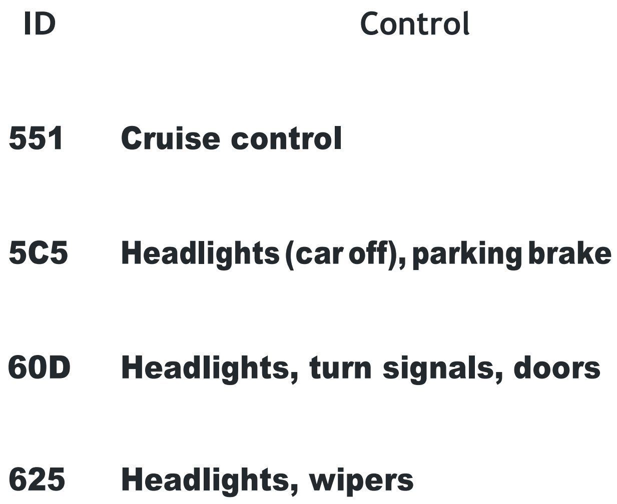
No change in messages

Likely just a mechanical release

1.4 CAN ID summary

CAN message IDs to control relations identified but not limited to

ID	Control
160	**Accelerator pedal**
280	**Seat belt**
354	**Brake, wipers**
358	**Headlights, climate control**
35D	**Brake, wipers, climate control, rear defrost**



ID	Control
551	Cruise control
5C5	Headlights (car off), parking brake
60D	Headlights, turn signals, doors
625	Headlights, wipers

Appendix B: Ford ID Data

ABS Service Codes

Ford refers to ABS service codes as on-demand codes and continuous codes. On-demand codes are “hard” codes that occur during a key-on, engine-off (KOEO) self-test. Continuous codes are memory codes from the ABS control module. These indicate intermittent problems that have occurred in the past, during normal vehicle operation. Memory codes cannot be set while running a self-test. During the KOEO self-test, the ABS control system transmits hard (on-demand) codes. Continuous memory codes are gathered by selecting MEMORY CODES from the SERVICE CODE MENU for ABS systems.

Service Code Menu

When you select SERVICE CODES from the ABS MAIN MENU, the SCANNER displays another menu similar to this:

```
SERVICE CODE MENU >KOEO SELF-TEST MEMORY CODES CLEAR CODES
```

The ABS Service Code Menu selections are: Key-on, engine-off (KOEO) test This test displays on-demand hard codes present with the ignition on, but the engine not running. These are usually electrical open and short circuits and must be serviced first, before any memory codes.

MEMORY CODES – This selection displays continuous memory codes of intermittent faults from ABS controller memory. Memory codes should be serviced last, after any other hard codes generated during the KOEO self-test.

CLEAR CODES – This selection allows you to clear continuous memory codes of intermittent faults from ABS system memory. Main Menu – Bosch 67ABS Service Codes

When you select KOEO SELF-TEST for a Ford ABS system, the SCANNER displays:`*ON-DEMAND SELF-TESTTURN KEY ON. DO NOT START ENGINE.WITH KEY ON PRESS Y TO CONTINUE.` Follow the instructions on the screen. The SCANNER displays the following message for a few seconds as the ABS controller performs the system self-test: `SELF TEST INITIATED...WAIT FOR CODES......................................IF NO RESPONSE IN 60 SECONDS, SEE FORDREFERENCE MANUAL --APPENDIX B.The message: SELF TEST INITIATED...WAIT FOR CODES This` means that the SCANNER has attempted to start the test. It does not mean that the vehicle has responded. If the message stays on the screen for more than 2 to 3 minutes, the test probably did not start. Refer to Appendix B to diagnose the cause of a vehicle self-test failure. When the ABS controller finishes the test, the display will change and be similar to this: `SERVICE CODES:** ON-DEMAND SELF-TEST CODES-FIX FIRST **C1210 OPEN/SHORT RF DUMP VALVE SOLC1096 HYDRAULIC PUMP MOTOR OPEN` If codes are not present, the SCANNER displays:`P0000 NO FAULTS PRESENT.` The top line of the display – REVIEW CODES: – remains fixed. You must press the thumb pad up to view additional lines of the display on lines 2, 3, and 4. When you reach the end of the list, the last line will say:

`*** END OF LIST ***`Press N to return to the SERVICE CODE MENU from the KOEO self-test.

Memory Codes This selection displays continuous memory codes of intermittent faults from ABS controller memory. Memory codes should be serviced last, after any other hard codes found during the KOEO self-test. Select MEMORY CODES from the ABS SERVICE CODE MENU and the SCANNER displays:`*CONTINUOUS MEMORY CODES* KEY MUST BE ON WITH ENGINE OFF OR RUNNING.PRESS Y TO CONTINUE.` This test can be run with the key-on, engine-off, or with the key-on, engine-running. Make sure the key is on and press Y to continue. The SCANNER displays: `SELF TEST INITIATED...WAIT FOR CODES.....................................IF NO RESPONSE IN 60 SECONDS, SEE FORDREFERENCE MANUAL -- APPENDIX B.` When the ABS controller finishes the test, the display will change and be similar to this: `SERVICE CODES:** CONTINUOUS MEMORY CODES-FIX LAST **P0000 NO CODES PRESENT*** END OF LIST ***`The top line of the display – SERVICE CODES: – remains fixed. You must press the thumb pad up to view additional lines of the display on lines 2, 3, and 4. When you reach the end of the list, the last line will say: `*** END OF LIST ***`Press N to return to the SERVICE CODE MENU.ABS Service Codes

Clear Codes – lets you clear the ABS controller code memory. When you press Y from the SERVICE CODE MENU, the SCANNER displays:`*CLEAR CODES* KEY MUST BE ON WITH ENGINE OFF OR RUNNING. PRESS N TO EXIT WITHOUTCLEARING. PRESS Y TO CLEAR CODES.` This test can be run with the key-on, engine-off, or with the key-on, engine-running. Make sure the key is on and press Y to continue. The SCANNER displays: `SELF TEST INITIATED WAIT FOR CODE CLEARING.IF NO RESPONSE IN 60 SECONDS, SEE FORDREFERENCE MANUAL -- APPENDIX B.` When the ABS controller finishes the test, the display will change and be similar to this: `CODES CLEAREDPRESS N TO EXIT. If the CODES CLEARED` message does not appear after about 1 minute, refer to Appendix B to diagnose a vehicle self-test failure. Press N to return to the SERVICE CODES MENU. Remember that only continuous memory codes can be cleared. If a code reappears when you clear codes and repeat the KOEO self-test, it is a hard (on-demand) code that must be serviced.

Clearing SCANNER Code Memory The SCANNER retains codes in its memory that it receives from the vehicle. SCANNER (not vehicle) memory is cleared in three ways: • By repeating the test, which overwrites the previous code list• By selecting a different system for testing • By entering a new vehicle ID.

Review Codes When you return to the ABS SERVICE CODE MENU from the engine-off self-test or the memory code test, the display appears like this: `SERVICE CODE MENU >KOEO SELF-TEST MEMORY CODES CLEAR CODES REVIEW CODES PRINT CODES`

The SCANNER has now recorded codes in its memory from either, or both, the engine-off and the memory codes tests. You can review the code list. If you select REVIEW CODES, the SCANNER displays a code list similar to the lists displayed at the end of the self-tests: REVIEW CODES** KEY ON, ENGINE OFF CODES-FIX FIRST **XX - - code description - - XX - - code description - -** CONTINUOUS MEMORY CODES-FIX LAST **XX - - code description - -XX - - code description - -*** END OF LIST ***Fix the problems in the order listed. Also, remember these important points about the REVIEW CODES list: 1. To review all codes, you must press the thumb pad up until *** END OF LIST *** appears.2. The SCANNER saves the codes from the most recent engine-off test for display under REVIEW CODES. If you the test, previous codes from that test will be replaced with a new list.3. Always record continuous memory codes after any test.4. If you have read memory codes, the SCANNER saves them in its memory, but you must use the CLEAR CODES selection to clear them from the ABS controller. Press N to return to the SERVICE CODE MENU from REVIEW CODES.ABS Service Codes

Bendix LC5 ABS – Main Menu Selections

The MAIN MENU for ABS testing provides the following general functions. Selection titles on the menu will be:• Codes & Data selections let you read fault codes and view information from the controller on vehicles with ABS control systems.• Actuator Test selections let you perform specific actuator operating tests on vehicles with ABS control systems.• Movie recording capability to record and save ABS system operating data for review and printing.• Custom Setup programs to set your SCANNER for your specific needs.• Other Systems lets you exit the Main Menu and return to the System Selection screen to select a different control system and enter a new vehicle identification. For more information on menus also refer to the Scanner Plug-in User's Manual.

Bendix LC5 ABS Codes & Data

The ABS CODES & DATA display available for vehicles with Bendix LC5 ABS systems will appear like this: PUMP____OFF PUMP (V)____0.1 SOL (V)__12.1** CODES & DATA. OK TO DRIVE. **NO CODES PRESENTWARNING LAMP______OFF STOP LAMP_________OFFLF WHEEL___________66 RF WHEEL_____________66LR WHEEL__________66 RR WHEEL__________66SYSTEM RELAY_____OFF MTR PUMP RELAY_____OFF In the ABS CODES & DATA diagnostic mode, the antilock functions of the Bendix system are fully functional, and new trouble codes can be set. The car can be driven safely for testing. CAUTION: Before driving a vehicle with an ABS complaint, especially if the red BRAKE warning lamp is on, test the brakes at low speed to make sure that the car will stop normally. An illuminated red BRAKE warning lamp can indicate reduced braking ability. The top line of the display remains fixed. The second and all other lines can be scrolled backward and forward through the ABS CODES & DATA list.