THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Bad Actors in Vehicular Environment

A look into FMVSS NO. 150 proposed
vehicular communication for road safety-application
and BSM content verification

Galton Yapdi Saputra
B. Information Technology

A thesis submitted for the degree of Master of Computer Science at
The University of Queensland in 2019
School of Information Technology and Electrical Engineering

A little magic can take you a long way.

~ R. Dahl ~

Abstract

Starting with the 75 MHz spectrum allocation by the FCC for DSRC-based communication, to the standardization of IEEE 802.11p (WAVE), IEEE 1609.0-4 as well as SAE J2735, the drive for a standardize C-ITS is warranted. V2V not only improve traffic management and mobility, but most importantly, it increases road safety through enabling vehicle communication. Recognising the potential safety applications of V2V, the NHTSA submitted a proposal standardizing V2V's communication security, message structure, encoding and transmission. Conversely, it blindly trusts message content and fails to propose security measures to check the content of the transmitted message.

Presented below is a demonstration evaluating NHTSA's proposed V2V security framework by simulating two vehicular actors connected to a verification framework. Car one is tagged as a good actor, where the broadcasted BSM Core Data are actual readings of the car. Car two on the other hand, contains manipulated speed readings and broadcasting falsified BSM messages. This demonstrator shows the current gaps in NHTSA’s proposed V2V’s communication security, and if mandated stipulates that inherited trust hierarchy from message authentication could translate to trusting message content.

Galton Saputra
galton.saputra@gmail.com

10 June 2019

Prof Michael Brünig
Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Brünig,

In accordance with the requirements of the Degree of Master of Computer Science in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled

"Bad Actors in Vehicular Environment"

This thesis was performed under the supervision of Dr. Konstanty Bialkowski. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely,

_____
Galton Yapdi Saputra

Acknowledgement

I wish to express my sincerest gratitude and acknowledgement to those who have provided their various supports and encouragements throughout the development of this thesis. Although there are only a few here that have been acknowledged, I am forever grateful to the unspoken, silent heroes.

I am greatly thankful for the supervision and guidance od Dr. Konstanty Bialkowski. His in-depth technical explanation and guidance throughout this thesis has made it a memorable and enjoyable experience.

I would like to say thank you to all my family members for their unconditional support and patience throughout this educational journey.

Lastly, thank you Mr. Robert Rolls for the patience towards my understanding of *vires acquirit eundo*.

Table of Contents

List of Figures

List of Table

List of Abbreviations

BSM	Basic Safety Messages
DSRC	Dedicated Short Range Communication
ICS	International Classifications for Standards
ISO	International Organization for Standardization
ITS	Intelligent Transport System
NHTSA	National Highway Traffic Safety Administration
OBE	Onboard Equipment
RSU	Road Side Units
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
VANET	Vehicular ad-hoc network
WAVE	Wireless Access in Vehicular Environment

1.0 Introduction

Intelligent Transportation System provides an automated solution to increase public road safety, traffic efficiency, reduction of emission and energy consumption. Scoping further, ITS has a subset branch of Cooperative Intelligent Transportation System with information sharing as the currency driving safety application. Through C-ITS enabling advancement in vehicular communication, governments are looking to adopt and utilise new vehicular communication technology in order to tackle classical traffic management problem and increase road safety.

Vehicle-to-Everything is a vehicular communication concept taken from VANET, where vehicles are independent nodes and able to interact and communicate with vehicles within its surroundings (V2V), communicating with road side infrastructure (V2I), communicating to pedestrian (V2P) and even to an electricity grid (V2G). This novel concept of vehicular communication interconnecting and integrated to our daily lives is slowly becoming a reality rather than a possibility. The realization of this idea was further pushed when V2V starts garnering governmental attention until the recent proposed rulemaking by the NHTSA mandating DSRC-based communication technology to be equipped to all new light-vehicles if the proposed legislation has been finalised. Vehicular technology not only opens multiple avenue and development in safety application, but an integration and adoption of a technology this scale requires robust and reliable vehicular communication security framework.

NHTSA's proposed rulemaking under FMVSS No. 150 highlights the security architecture framework to govern message authentication and security. Through the use of a Security Credential Management System, it establishes the management of digital certificates and it a central Certificate Authority to verify all registered vehicular certificates. This proposal of having a centralised central authority not only invites unwanted attack, but also highlights an inherited trust hierarchy that blindly trust message content through message authentication.

In a heterogenous network environment, where information sharing is the currency driving vehicular safety application; accuracy and reliability of message content results in the different between engaging a preventative or mitigative protocol. The proposed framework below investigates and proposes a verification framework to mitigate blind trust hierarchy by verifying message content.

Thesis Objective

This project aims to design a verification framework that allows RSU to inspect message content from broadcasted Basic Safety Message (BSM). To accomplish this aim, several objectives have been defined as follows:

1. Evaluate and analyse NHTSA proposed security framework for V2V

2. Investigating the applicability and feasibility of utilising a Public Key Infrastructure for V2V's security framework.

3. To develop a framework inspecting Basic Safety Message content rather than trusting message content based on validated message certificate proving message authentication.

4. To develop RSU acting as an independent verification sensor to validate broadcasted Basic Safety Message content.

2.0 Literature Review

2.1 Intelligent Transportation System (ITS)

With the continuous increase in population growth, unstoppable migration pattern to urban areas, transportation systems are stretched and stressed to handle urban traffic flows. Classical traffic managements are becoming less efficient to handle the fluctuating demands of urban cities. An Intelligent Transportation System (ITS) is seen as a solution integrating technology with transportation infrastructure and road users. Through automated management and monitoring of traffic, with real-time notifications, ITS provides an automated solution to increase public road safety, traffic efficiency, reduction of emission and energy consumption. ITS is further classified into three categories: Safety (improving public road safety), Efficiency (traffic monitoring and management) and Infotainment (internet access and video streaming).

The drive for Intelligent Transportation System can be traced as early as October 1999, when the FCC allocated spectrum in the 5.9GHz range to increase highway safety and efficiency. Operating within the radio spectrum of the class: ultra-high frequency (UHF) to super-high frequency (SHF); the FCC allocated 75 MHz to the 5.850 – 5.925GHz band for Dedicated Short Range Communications (DSRC) [1]. DSRC reserves[1] and provides a standardize communication infrastructure within the radio spectrum for Intelligent Transportation System (DSRC spectrum range differs between region). This communication architecture led to the development of traffic light monitoring and control, automatic toll collection and traffic congestion detection.

As the need to utilise ITS applications in transportation system became evident, a standardize framework was warranted to facilitate ITS implementation internationally. The International Organization for Standardization (ISO) published a standardized framework for the development in Intelligent Transport System under ISO/TC 204.

2.2 ITS/ISO Standards

Formed in 1946, The International Organisation for Standardization (ISO) is an independent, non-governmental international organisation, comprising of 164[2] national standards bodies. Standards Australia (SA) represents Australia's membership within ISO, the International Electrotechnical Commission (IEC) and the International Council of Societies of Industrial Design (ICSID) [2].

Derived from the Greek word '*isos'* – equal, ISO aims '*to facilitate the international coordination and unification of industrial standards*' [3]. To date, ISO has published 22,572 International Standards ranging from food safety, agriculture, technology, healthcare and other industries. These publications are the results of ISO's technical committees (TC) - working to publish International Standards (IS). Publications of IS are catalogued by the International Classification for Standards (ICS). Indexed and stored at ISO's Standards catalogue (SC), it has two browsing categories:

Browse by ICS

ICS not only provides a structure for cataloguing publications of international standards; classifications of industry fields (e.g. ICS29 - Electrical engineering vs ICS25 – Manufacturing engineering), but also provides a systematic approach in cataloguing these publications – below is an example that

[1] Different nations have differing reservations or allocations of frequencies utilised for DSRC. Please see Appendix A.

[2] Members to date (2019)

decomposes an ICS code to its root category. An ISO document may have two ICS reference classification code. Take for example:

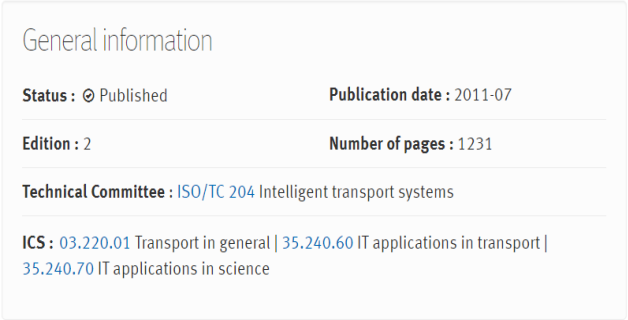
ISO 14825:2011 -- Intelligent transport systems -- Geographic Data Files (GDF) -- GDF5.0

Within the Standards Catalogue, ISO 14825:2011 has been classified into three classification categories, as shown in Figure 1 [4].

Figure 1 - ISO 14825:2011 General information

ICS 03.220.01 -- Transport in general
ICS 35.240.60 -- **IT applications in transport**
ICS 35.240.70 -- IT applications in science

Decomposing ICS 35.240.60:
35.0 Information Technology >
35.240 Applications of information technology >
35.240.60 IT applications in transport



Browse by TC

Technical Committees are key bodies comprising of technical experts within its field, sourced from the national committees, handling ISO's development and publication of standards.

ISO is not the only organisation which develops ITS Standards. Figure 2 shows other international organisations, collaborating to the development and publication of ITS Standards. They include:

CEN: Comité Européen de Normalisation or European Committee for Standardization

SAE: Society of Automotive Engineering

IEEE: Institute of Electrical and Electronics Engineers

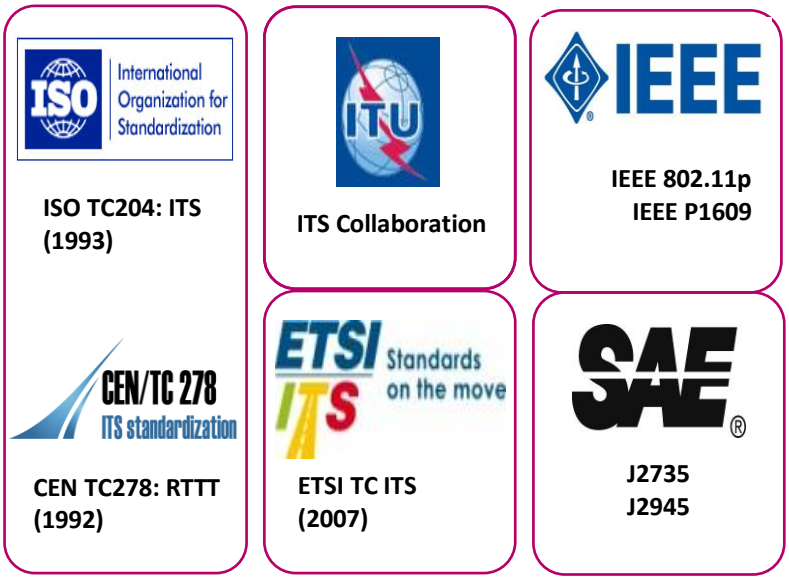


Figure 2 - ITS Standards Developing Organizations

2.3 ISO/TC 204 – Intelligent Transport Systems

Classified under Technical Committee 204 or ICS 03.220.01|35.240.60, TC 204 is responsible for the overall system aspects and infrastructure aspects of intelligent transport systems (ITS). Including the coordination of the overall ISO work programme within this field, as well as scheduling for standards development [5].'

To date, there are 262 published ISO standards under the direct responsibility of ISO/TC 204; **excluding** ISO/TC22 – 'In-Vehicle Transport Information and Control Systems' [5]. It is also directly

responsible for the two standards of Cooperative ITS (CITS)[3], ITS station security services - ISO 17427 and ISO 21177 respectively.

2.3.1 ISO 17427 – Intelligent transport systems -- Cooperative ITS

ISO 17427 has been segmented into nine parts [6]. Table 1 below shows the nine ISO published standards for the different fields in C-ITS.

Standard and/or project	Stage[4]	TC
ISO 17427-1:2018 Intelligent transport systems -- Cooperative ITS -- Part 1: Roles and responsibilities in the context of co-operative ITS architecture(s)	60.60	ISO/TC 204
ISO/TR 17427-2:2015 Intelligent transport systems -- Cooperative ITS -- Part 2: Framework overview	60.60	ISO/TC 204
ISO/TR 17427-3:2015 Intelligent transport systems -- Cooperative ITS -- Part 3: Concept of operations (ConOps) for 'core' systems	60.60	ISO/TC 204
ISO/TR 17427-4:2015 Intelligent transport systems -- Cooperative ITS -- Part 4: Minimum system requirements and behaviour for core systems	60.60	ISO/TC 204
ISO/TR 17427-6:2015 Intelligent transport systems -- Cooperative ITS -- Part 6: 'Core system' risk assessment methodology	60.60	ISO/TC 204
ISO/TR 17427-7:2015 Intelligent transport systems -- Cooperative ITS -- Part 7: Privacy aspects	60.60	ISO/TC 204
ISO/TR 17427-8:2015 Intelligent transport systems -- Cooperative ITS -- Part 8: Liability aspects	60.60	ISO/TC 204
ISO/TR 17427-9:2015 Intelligent transport systems -- Cooperative ITS -- Part 9: Compliance and enforcement aspects	60.60	ISO/TC 204
ISO/TR 17427-10:2015 Intelligent transport systems -- Cooperative ITS -- Part 10: Driver distraction and information display	60.60	ISO/TC 204

Table 1 - ISO 17427:1 – 9 [6]

ISO 17427 highlights that for a C-ITS to be a complete solution, it requires the implementation of a framework, core system, compliance/standards and what information can be presented to the user. TC204 represents the overall umbrella of ITS, covering all transport mode of land, sea and air. ISO 17427 on the other hands is a standard for C-ITS – a subset of ITS.

2.3.2 International Standardization and Interoperability

As ISO's standards in CITS matured, extending and covering data communication standards for V2X communication (including V2G). It provides an international assurance of quality in standardized implementation of CITS as a solution to tackle classical traffic management. Publication and implementation of ISO standards promotes confidence in interoperability and facilitates cross-region C-ITS development. With ISO's standards enforcing international standardize ITS implementation, a

[3] CITS or C-ITS may be used interchangeably

[4] Represents where a standard is in its developmental track. E.g. 60.60 represents that an International Standard has been published.

joint research task group for cooperative ITS was formed in 2012. The EU-U.S. Research Cooperation in Cooperative Systems allows further C-ITS development in Cooperative Systems (V2V and V2I) [9].

Figure 3 - EU-U.S. Research Cooperation in Cooperative Systems [10]

With ISO and its technical committee publication, it not only sets an international standard but promotes confidence in cross-platform and even region for vehicular communication. Their joint collboration enables further development of C-ITS across two continents, furthering integration of C-ITS worldwide. Allowing vehicle manufacturer in the US to expand its market reach and able to produce vehicle complying to legislation set by the EU.

2.4 Cooperative Intelligent Transportation System

Cooperative Intelligent Transportation System (C-ITS) is a system able to sense, analyse, control and exchange of information with other vehicles and/or infrastructure. An 'ITS system' is not dependent like C-ITS with regards to the exchange of data, through the interaction with other vehicles and or infrastructure [11]. CEN/TC 278[5] defines C-ITS as:

*"C-ITS is a subset of the overall ITS that **communicates** and **shares information** between ITS Stations* to **give advice** or **take actions** with the objective of improving safety, sustainability, efficiency and comfort beyond the scope of stand-alone ITS."* [12]

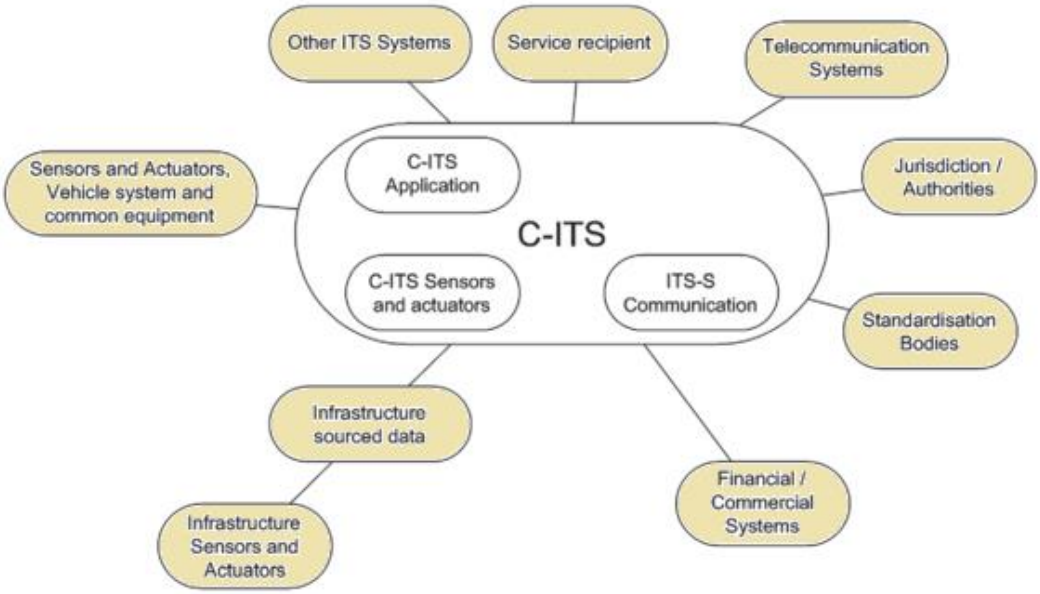
It is important to note that C-ITS is not a single deployable entity. But a combination of protocols, techniques, system and sub-systems working in unison to enable a 'cooperative' service. Figure 2.1 below gives a representation of the different entities involve (protocol, system, sub-system), working together in providing a collaborative service.

2.4.1 C-ITS Architecture

Referenced from ISO 17427-1[6], figure 2.0 below shows the architecture of Cooperative ITS; providing a platform enabling heterogenous interaction, exchange of information, in an interoperable manner.

[5] CEN/TC 278 Intelligent transport systems from ITS Standards EU

[6] ISO 17427-1:2018 Intelligent transport systems -- Cooperative ITS -- Part 1: Roles and responsibilities in the context of co-operative ITS architecture(s)



C-ITS Architecture Model [13]

Actors-Model

In the context of C-ITS architecture, an actor[7] is an abstraction concept representing various entities (highlighted in yellow) interacting and exchanging information with CITS (model). These actors could include: the telecommunication systems (e.g. network tower), Standardization Bodies (e.g. ISO, SAE, IEEE), Vehicle system (e.g. V2V) and common equipment (e.g. OBE). For the research purpose in security and verification of basic safety messages, the C-ITS Application component is further analysed.

2.4.2 C-ITS Application

There are abundant applications of C-ITS, but one C-ITS application of relevance in solving traffic management, is the application for Cooperative, connected and automated mobility (CCAM). CCAM is the application, deployment and integration of C-ITS directly with road infrastructure. CCAM not only increases road safety, but also provides real-time, accurate input to ADAS – Advanced Driving-Assistance Systems [14].

2.4.2.1 Advanced Driving-Assistance Systems

ADAS are systems to assist driver with their decision making and ultimately their driving process. Sharing a similar goal of ITS to increase road safety, ADAS is a solution achieved through various sensors built within an automobile. Different types of ADAS sensors are classified with respect to its technological application (i.e. vision, lidar or radar), Figure 4 below shows an ADAS taxonomy based on the type of sensors utilised. ADAS technology comes in many features or forms, (e.g. anti-lock brakes, lane departure warning or adaptive cruise control), aiming to improve driver's decision making and reduce human errors. Advanced ADAS solution further utilise and fuse radar-based sensor with advanced vehicular communication systems [15].

[7] Derived from The Actor-Model framework

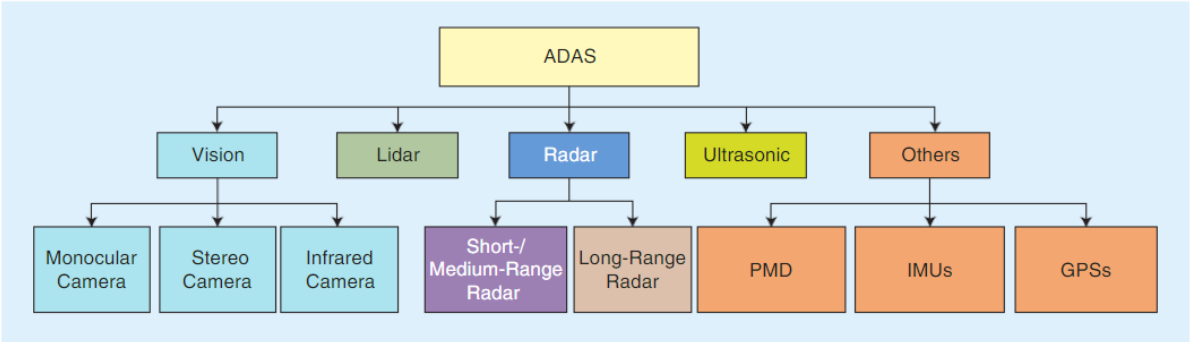


Figure 4 - Taxonomy of ADAS based on sensor types

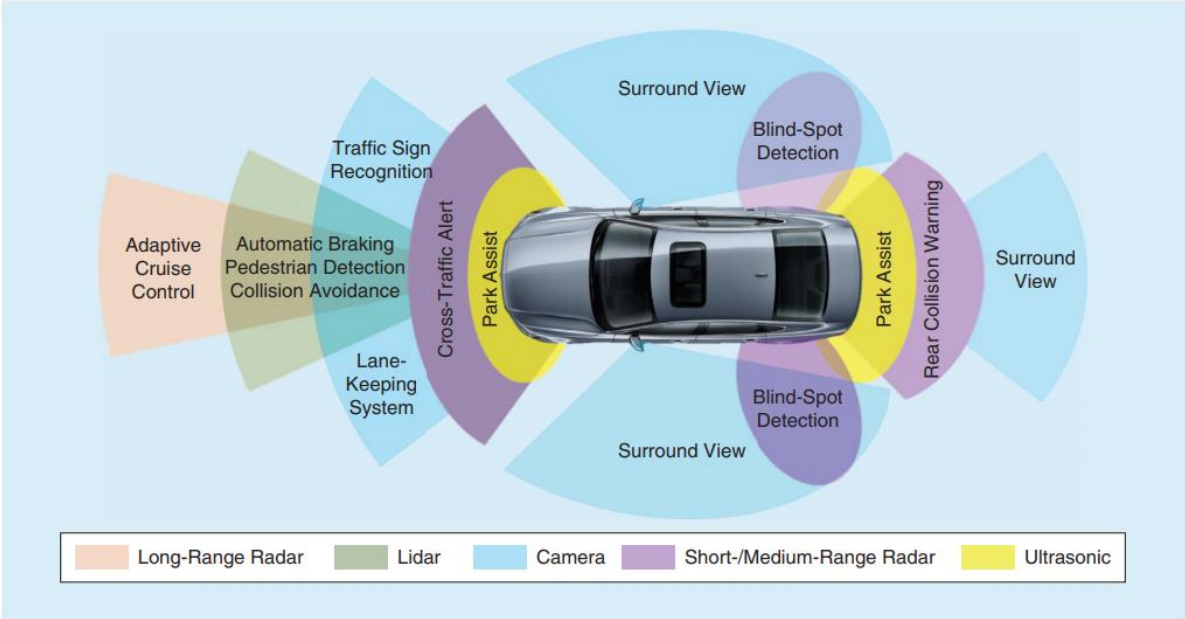


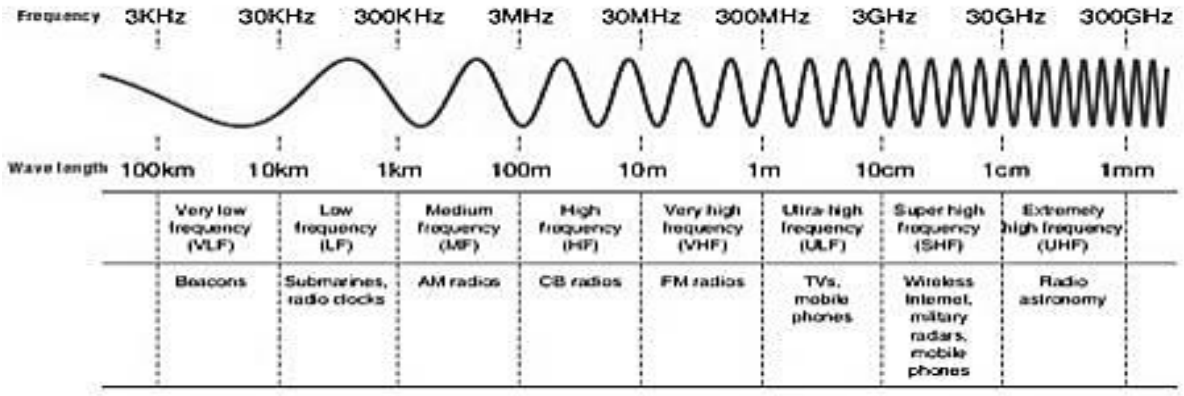
Figure 5 - State-of-the-art ADAS sensors [15]

2.4.2.2 ADAS – Short-/Medium Range Radar

The ADAS taxonomy above, radar[8] and its subgroup 'short-/medium range radar' – has a safety application use-case such as Blind-Spot Detection or Cross-Traffic Alert (Figure 3.0). Driver as well as its vehicle can detect the presence of other road participants. Radar-based sensors and advanced sensor-fusion further equips driver with relevant decision-making information. This information can be extended by cooperatively sharing its information to its surrounding. Information sharing is the currency of C-ITS, and by utilising radio wave's advantageous properties of having long wavelengths[9], its proven application in wireless technology (e.g. Wi-Fi), this can be further extended and utilised with vehicular communication [15].

[8] Radio Detecting and Ranging (RADAR)
[9] See Appendix B for electromagnetic spectrum

Figure 6 - Radio frequency classification [16]



Wi-Fi – a form of radio wave we interact and use daily, operates at SHF – super high frequency. ADAS Short-/Medium Range radar operates and utilises the same wireless technology found in home routers. Piggybacking this proven and prevalent technology, vehicular communication became possible by utilising the same wireless technology and properties we interact and use daily.

2.4.3 Vehicular Communication System

Vehicular communication system is a concept where vehicles (e.g. cars, bicycle, etc) and roadside unit becomes a wireless node, sharing information and communicating autonomously – turning a moving vehicle (i.e. car) into a wireless node. With heterogenous and mobile actors participating within an environment, governance of information sharing requires a network architecture to support interoperable, moving vehicle networks.

2.4.4 Vehicular Ad-hoc Network (VANET)

Vehicular Ad Hoc Network (VANET) was taken from a network architectural concept of Mobile Ad Hoc Network (MANET) - note that VANET is a sub-category of MANET. In MANET, devices in the network are independent entity and have the ability to roam freely, continuously self-configuring to optimise traffic routing through an **infrastructure-less network of mobile-devices connected wirelessly**. Piggy-backing this unique architectural concept, VANET on the other hand is not a network of mobile-devices, but a network of vehicles connected wirelessly.

Within VANET, there are 3 interacting actors:

ITS Services (ITS): ITS services represent actors from public services (e.g. ambulance, firetruck).

Road Side Units (RSU): RSU is a deployable roadside instrument/device (e.g. Traffic lights) providing wireless communications from roadside infrastructure to participants within WAVE.

On-Board Equipment (OBE): OBE's are retrofitted device within each vehicle enabling participation of broadcasted message within VANET.

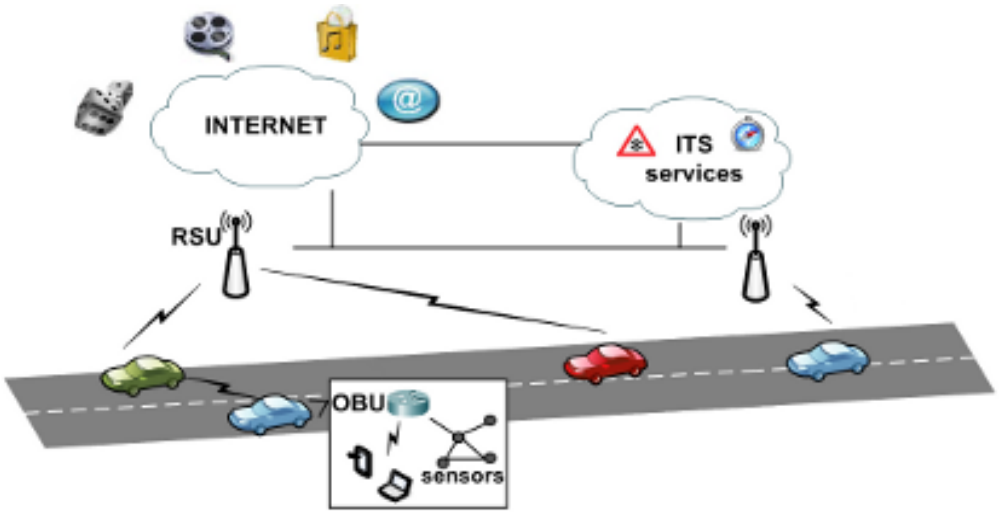


Figure 7 - Interacting Actors Within VANET

Figure 7 above, shows interacting actors within VANET can utilise the network it is connected, to access internet and media sharing. This network architectural concept of infrastructure-less and self-organising, enables intervehicle (V2V) and vehicle-to-infrastructure (V2I) communication.

2.4.5 Dedicated Short-Range Communication (DSRC)

Operating within the 5.850 GHz – 5.925 GHz block of spectrum, DSRC is a two-way, short-to-medium-range wireless standard, providing wireless link of information transfer between vehicles and roadside systems (RSU). Established in October 1999 by the FCC, this medium range communication service was selected due to its effectiveness over short to medium distance transmission. It also can handle/transmit large volumes of data reliably and with accuracy due to its low latency. Its scope of accuracy and proximity reach made it suitable for various vehicular communication applications – especially for safety use case application [1].

2.4.5.1 DSRC Applications

Below are several applications of DSRC:

- Traffic light control
- Traffic monitoring
- Automatic/Electronic toll collection (ETC)
- Traffic congestion detection,
- Emergency vehicle signal pre-emption of traffic lights,
- Safety applications (e.g. collision prevention)

Safety application – collision prevention

In vehicular communication, real-time accuracy, reliability and availability are vital factors in enabling vehicles to 'talk' to each other, also known as Vehicle-to-Vehicle (V2V) [17]. DSRC provides the communication standard in facilitating V2V. One primary motivation and application of DSRC is collision prevention. By giving vehicles the ability to sense and 'talk' to its surrounding, it increases a driver's field of vision and its vehicle to an omnidirectional (360) vision. United States Department of Transport (DOT) has estimated that V2V communication utilising DSRC can address up to 82% of all crashes in the United States [18]. This led to the Vehicle Safety Communications – Applications project

completed in 2009. This project evaluated and demonstrated the feasibility of various V2V safety applications such as:

- Forward collision warning (stopped vehicle ahead);
- Emergency electronic brake lights (hard-braking vehicle ahead);
- Blind spot warning;
- intersection movement assist;
  do not pass warning;
- control loss warning

These safety applications broadcasting vehicle safety messages, communicate on a reserved channel within the DSRC spectrum. Channel 178 is a reserved channel for the communication of safety application.

2.4.5.2 DSRC Spectrum & Channels

DSRC is licensed by the FCC at 5.9 GHz with a 75-Mhz spectrum. It is further divided into seven segments of 10 MHz channels with a 5MHz guard band [19]. Figure 8 below shows the seven segmented channels within the DSRC spectrum.

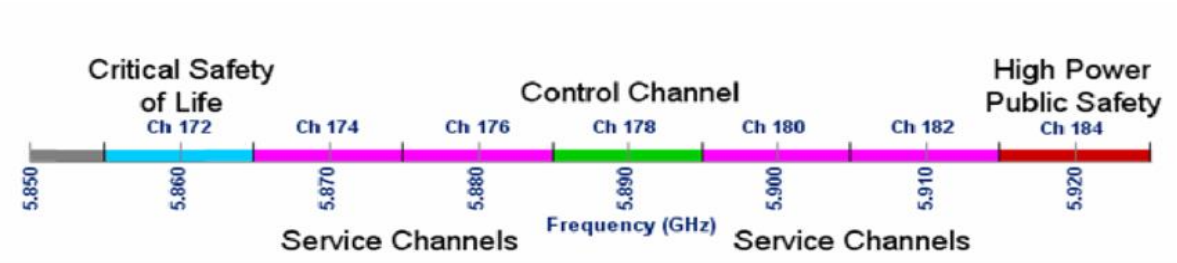


Figure 8 - U.S. segmentation of DSRC spectrum band & channels

Within DSRC spectrum, there are six Service Channels (SCHs) and one Control Channel (CCH). SCHs are used for infotainment and other commercial applications (e.g. Electronic toll collection, priority-vehicles such as ambulances). CCH on the other hand, is used to carry high-priority short messages or management data. Table 2 below further drills into each of the seven channels.

Table 2 - DSRC Channel Type & Purpose

Channel/ Type	Frequency Band	Purpose
	5850-5855 MHz	Guard band
Ch 172 **Service**, 10 MHz	5855-5865 MHz	Critical Safety of Life – these are the channels for public service vehicles (i.e. ambulances).
Ch 174 **Service**, 10 MHz	5865-5875 MHz	SCH Channels 174 and 176 can be combined to provide a 20 MHz service channel.
Ch 176 **Service**, 10 MHz	5875-5885 MHz	

Ch 178 **Control**, 10 MHz	5885-5895 MHz	Control Channel – strictly use for safety communication application. For verifying broadcasted vehicle basic safety messages, channel 178 - the control channel (CCH) is further analysed and discussed in the below section.
Ch 180 **Service**, 10 MHz	5895-5905 MHz	SCH Channels 180 and 182 can be combined to provide a 20 MHz service channel.
Ch 182 **Service**, 10 MHz	5905-5915 MHz	
Ch 184 **Service**, 10 MHz	5915-5925 MHz	High Power Public Safety - to be used for high-power, longer distance communications public safety applications involving road intersection collision mitigation.

Safety Channel

Each channel plays a dedicated part and purpose. First channel, with 5GHz spacing at 5.850 is used as a *guard band* (coloured grey) – it plays a protective role against collision with other channels. Channel CH178 [5.860GHz] and CH184 [5.920GHz] are both used as *safety **dedicated** channel*. Both channels are located at the front and back of the 5.9 band respectively. Using 5.860 GHz, CH178 provides a security solution for reserving a dedicated channel for critical public service vehicle – i.e. ambulance or police. Where CH184 (5.920GHz), is used as a protective barrier against congestion with other channels [21].

Service Channel

There are four service channels allotted for bidirectional communication between different types of unit. Each channel is spaced at 10MHz apart, however, through combining CH174, CH176 and CH180, CH182, each pair can form a single 20MHz channel - CH 175 and CH 181 respectively.

Control Channel

Only the control channel (CCH) is responsible for the control of transmission broadcast and link establishment. Operating on CH178, using 5.855-5.895, CCH is strictly use for safety communication application (e.g. a V2V communication containing Basic Safety Messages).

In the United States, the DSRC spectrum band is a free but licensed spectrum. The FCC does not charge a fee for spectrum usage. However, different regions may have differing DSRC regulation, frequency-band and channel allocation. Appendix A shows a comparison of three countries (US, Japan and EU) DSRC governance. Regardless of the different frequency allocation for DSRC, it adheres to the international standardized body governing each standard's publication (e.g. IEEE, SAE or ISO).

2.4.5.3 DSRC Protocol Stack

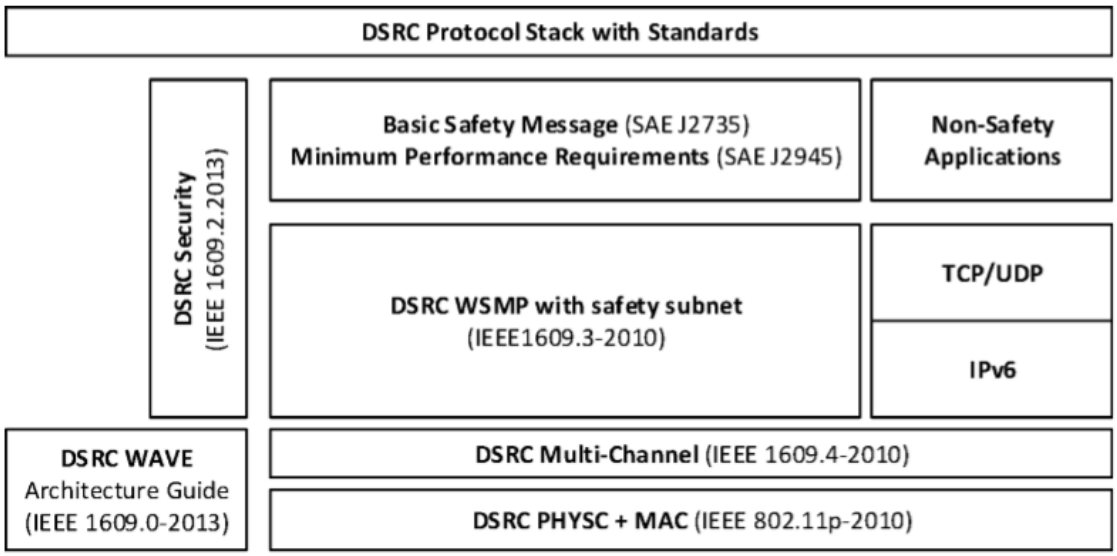


Figure 9 - NHTSA DSRC Protocol Stack

DSRC communication relies fundamentally on interoperability among devices from different manufacturers. Figure 11 above, captures the Physical, Data Link, Network and Transport (Layer 1-4) within the OSI model. Starting from the bottom layer, 'DSRC PHYSC + MAC' - IEEE 802.11p is an approved amendment to the IEEE 802.11 family.

IEEE 802.11p

IEEE 802.11 is a collection of specification for the physical and media access control layers to implement Wireless Local Area Networks (WLANs). Maintained by the 802 LAN Standards Committee, 802.11p is an additional amendment to the 802.11 WLAN standards family.

IEEE Std. 802.11p-2010
IEEE Standard for Information technology—
Telecommunications and information exchange between systems—
Local and metropolitan area networks —
Specific requirements for

Part 11: **Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**

Amendment 6: **Wireless Access in Vehicular Environments**

Published in 2010, 802.11p is an amendment laying down requirements to allow the use of the 5.9GHz band (5.850 - 5.925) GHz for Wireless Access in Vehicular Environments (WAVE). Utilising the mechanism provided initially from the 802.11a protocol, operating in 5.8 GHz frequency range. This amendment offers data exchange among vehicles (V2V) and roadside infrastructure (V2I). Unlike other protocols within the 802.11 family which utilises TCP and checksum to ensure successful payload delivery, 802.11p will not send any acknowledgement for its broadcasted packets – similar to UDP [20].

WAVE aims to establish a common physical platform in facilitating communication between ITS application and Vehicular Infrastructure (VI) - allowing real-time messages to be relayed to drivers/passengers [21] [22].

IEEE 802.11p standard's purpose is to:

- ➔ *Describe the functions and services required by WAVE-conformant stations to operate in a rapidly varying environment and exchange messages without having to join a Basic Service Set (BSS).* [21]

- ➔ Define WAVE signalling technique and interface functions that are controlled by IEEE 802.11 MAC.

This amendment is of key reference to IEEE1609 as it specifies the extensions to IEEE802.11 which provides wireless communications in a vehicular environment.

IEEE 1609 Family of Standards for WAVE

IEEE 1609 is a Family of Standards for Wireless Access in Vehicular Environments (WAVE). It provides an interface to allow homogenous communication between different automotive manufacturers. These standards defines the '*architecture, communications model, management structure, security mechanisms and physical access for high speed (up to 27 Mb/s) short range (up to 1000m) low latency wireless communications in the vehicular environment* [23].' The main architectural component targeted by these standards applies to On Board Unit, Road Side Unit participating in WAVE.

Collectively, the IEEE1609 Family of Standards for WAVE describes the wireless exchange of data, security, service advertisement between vehicles and roadside devices and defining layers to allow applicable protocol enabling IntelliDrive applications to access when communicating with vehicles. Furthermore, it describes the physical mechanism of communication, command and management of services, and provides two options for V2V and V2I communication – these options are WAVE short message and IPv6. Providing the basic standards to allow heterogenous manufacturers or actors to have a seamless integration and interface regardless of manufacturers.

IEEE 1609 Family of Standards

P1609.0	Draft Standard for Wireless Access in Vehicular Environments (WAVE) – Architecture **describes the WAVE architecture and services necessary for multi-channel DSRC/WAVE devices to communicate in a mobile vehicular environment.**
1609.1-2006	**Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Resource Manager** specifies the services and interfaces of the WAVE Resource Manager application.It describes the data and management services offered within the WAVE architecture . It defines command message formats and the appropriate responses to those messages, data storage formats that must be used by applications to communicate between architecture components, and status and request message formats.
1609.2-2006	**Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Security Services for Applications and Management Messages** defines secure message formats and processing. This standard also defines the circumstances for using secure message exchanges and how those messages should be processed based upon the purpose of the exchange.

IEEE 1609.3 -2007	**Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services** defines network and transport layer services, including addressing and routing, in support of secure WAVE data exchange. It also defines Wave Short Messages, providing an efficient WAVE-specific alternative to IPv6 (Internet Protocol version 6) that can be directly supported by applications. Further, this standard defines the Management Information Base (MIB) for the WAVE protocol stack.
IEEE 1609.4 -2006	**Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) - Multi-Channel Operations** provides enhancements to the IEEE 802.11 Media Access Control (MAC) to support WAVE operations.
IEEE P1609.11	**Over-the-Air Data Exchange Protocol for Intelligent Transportation Systems (ITS)** will define the services and secure message formats necessary to support secure electronic payments.

[23]

These family of standards displayed above are utilised by automotive and traffic engineer, transportation authority that are involved with the specification, design, implementation and testing of WAVE devices. Engineers that are involve with implementing the communication for DSRC-based V2V and V2I interactions, should conform to 1609 standards to address low-latency interface, transmission and communication of On-Board and Roadside devices. The framework above, provides an interface definition between system component, and provides a structured framework for application architecture. For the purpose of this research, IEEE 1609.2 – Security protocol will be evaluated further.

IEEE 1609.02 – Security

IEEE 1609.02 is a subset of the IEEE WAVE Family Standard description. Within the 1609 family, 1609.2 – Security Services for Applications and Management Messages standardises a set of services and methods to secure WAVE management messages as well as application messages. They address the issue of privacy with a core design philosophy for message security to be user-privacy centric [5]. Their approach to privacy is that owners of the vehicle have an expectation of privacy, knowing that the data their device/vehicle broadcast does not disclose personal identifiable information, or linkable data to unauthorised third parties. [9]

These standards ranging from publication from IEEE, ISO to the DSRC protocol stacks all strive to enable interoperable vehicular communication. Automobile manufacturer must conform to these standards set by publishing standards if they would like to produce vehicle capable of participating in a wireless vehicular environment. In the United States, the NHTSA is the federal transportation body to standardize vehicular access for vehicles to participate in vehicular communication.

3.0 Wireless Access in Vehicular Environments

Wireless Access in Vehicular Environment or WAVE in short, is a radio communication system formed through VANET, to provide interoperable vehicular communication services to road participants (actor). Utilising DSRC, 802.11p and IEEE 1609 family of standards, it allows information sharing between road participants (vehicles) within WAVE.

3.1 Vehicle-to-Everything (V2X)

Collectively known as Vehicle-to-Everything (V2X) – V2X is a **vehicular communication** concept allowing information sharing between a vehicle to its connected wireless environment (WAVE). Through information sharing, V2X allows wireless participation and information sharing to different actors travelling on public road and connected to WAVE. Diagram below describes the different application possible through vehicular communication - V2X.

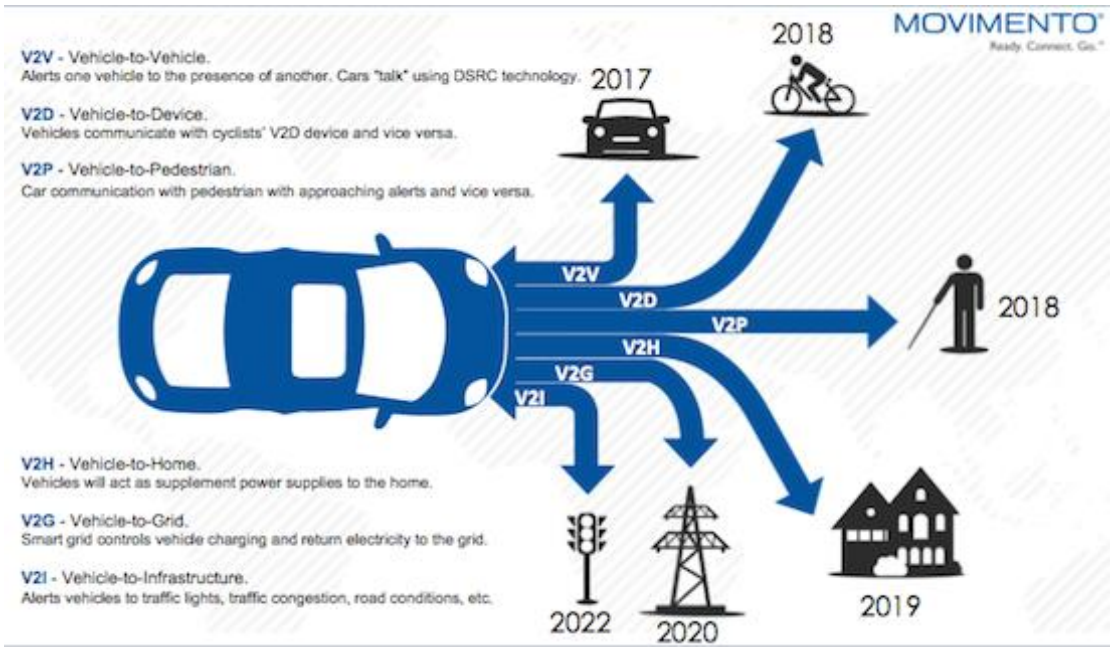


Figure 10 - V2X concepts and its participating actors [24]

Within the scope and purpose of this research in evaluating V2V message security, V2V and V2I will be discussed further.

3.1.1 Vehicle-to-Vehicle (V2V)

V2V is the concept where an automobile has the ability to communicate or "talk" with other cars as well as to its surrounding. In V2V scope, a vehicle is defined and classified to be as of **light vehicle**. This research evaluates two vehicle classification types from two countries: United States and Australia.

Each concept within V2X has a different scope and definition of a vehicle. Within V2V, a vehicle is defined to be of a light-vehicle type based on their gross vehicle mass (GVM). Section 3.1.1.1 below shows Australia and United States' definition of a light-vehicle as well as its vehicle classification assigned by governmental regulation/federated bodies.

3.1.1.1 Vehicle Definition

Australia	United States
ATAP - Australian Transport Assessment and Planning ATAP utilises vehicle classification from Austroads as *'… vehicle classification in Austroads (2005a) provides a sufficiently broad range of vehicle types… including the Austroads 12 bin classification outlined in Austroads (2013).'* [25] Austroads is a private entity which their publication of vehicle classification standards has been adopted by the Australian Transport Assessment and Planning as well as state implementation for transport regulation (e.g. WA, QLD, etc). A light vehicle with a gross vehicle mass less than 4,500 kg (or 4.5 tonnes) is represented as a Class C within AusRoads vehicle category. Class C denotes a vehicle category type of a Car – for further information please refer to Appendix B. This classification of vehicle type is adopted and implemented by different Australian states and its governing transport bodies. **Main Roads,** Western Australia (WA) *'A motor vehicle with a gross vehicle mass (GVM) not greater than 4,500 kg and constructed or equipped to seat no more than 12 adults (including the driver).'* [26] **Transport and Mains Road (TMR),** Queensland (QLD) *Car [not more than 4.5 tonnes (t) Gross Vehicle Mass (GVM) built to carry not more than 12 adults including the driver].'* [27]	DEPARTMENT OF TRANSPORTATION, NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION 82 FR 3854 - Federal Motor Vehicle Safety Standards; V2V Communications [Docket No. NHTSA–2016–0126] The NHTSA defines light vehicles as: *'Light vehicles include passenger cars, vans, minivans, sport utility vehicles, crossover utility vehicles and light pickup trucks with a gross vehicle weight rating (GVWR) less than or equal to 10,000 pounds.'* [28] This definition and classification of light vehicle is governed by Section 523, Title 49 of the Code of Federal Regulation. 49 CFR 523 - VEHICLE CLASSIFICATION Code of Federal Regulation Title 49 - Transportation Part 523 - VEHICLE CLASSIFICATION § 523.3 Automobile. *(a) An automobile is any 4-wheeled vehicle that is propelled by fuel, or by alternative fuel, manufactured primarily for use on public streets, roads, and highways and rated at less than 10,000 pounds gross vehicle weight…* Vehicle classification and regulation enacted by NHTSA is driven from legislation through the Code of Federal Regulation Title 49. [29]

Both Australia and United states have similar classification of a light-vehicle: being less than 4.5 tonnes (as 10,000 pounds equates to ±4500kg). Through defining a vehicle to be of light-vehicle class, it sets a precedence to implement a testbed in defining the type of actors participating in V2V communication. Within this thesis and NHTSA, a light-vehicle is defined to be: 4 wheeled vehicle, less than 4.5 tonnes and for use on public streets, roads and highways.

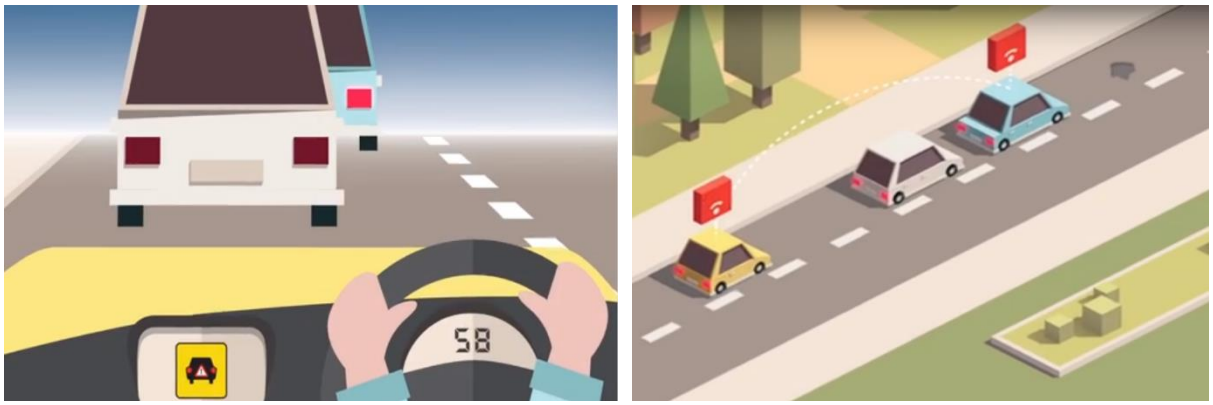
3.1.1.2 CAVI Project

Information sharing between light-vehicle allows for numerous safety use case application to be developed which are applicable to traffic management, drivers as well as increasing of public road safety for its user. The realisation for vehicular communication is not far off in Australia, as a testbed pilot project for vehicular communication is in progress. The CAVI project plans to develop and prototype different safety use case application to be tested and piloted by 2019 in Ipswich.

Under the Queensland Government, the Cooperative and Automated Vehicle Initiative (CAVI) project aims to test cooperative and automated vehicle technologies with an overall *'…vision of zero road deaths and serious injuries on state's roads'* [30]. CAVI consists of 4 components:

1. **Cooperative Intelligent Transport Systems (C-ITS) Pilot**—the largest on-road testing trial in Australia of cooperative vehicles and infrastructure.
2. **Cooperative and Highly Automated Driving (CHAD) Pilot**—testing a small number of vehicles with cooperative and automated technologies.
3. **Vulnerable road user pilot**—a project looking at how new technology applications can benefit vulnerable road user safety including pedestrians, motorcycle riders and bicycle riders.
4. **Change management**—a process for the Department of Transport and Main Roads to consider the change of current business and practices.

Figure 11 - Emergency braking warning [30]



Emergency braking warning (in-car view) [30]

From the four components above, the largest component of CAVI is C-ITS Pilot. This pilot will take place on public road and tested in the city of Ipswich from 2019 where around 500 public and fleet vehicles are retrofitted with C-ITS technologies, and roadside C-ITS devices are installed on arterial roads and motorways [30]. These installed devices allow vehicles and infrastructure to communicate and share real-time information about the road, generating safety-related warning messages to equip driver with better decision-making information. Two examples of C-ITS use case applications in V2V are:

3.1.1.3 V2V Application

<u>Emergency braking warning (V2V)</u>

This warning alert drivers if a cooperating vehicle (within WAVE) is braking hard some distance ahead.

Scenario:

There are three cooperative cars traveling along a road. The first car suddenly brakes hard, which

causes the in-vehicle C-ITS device of the first car to send a warning message to its vehicular environment advising of a sudden braking action. The third car behind the second car, receives this warning message via its in-vehicle C-ITS system – or sometime referred to as an OnBoard Unit (OBU) gave the third car's driver enough information to allow for him to react and brake smoothly.

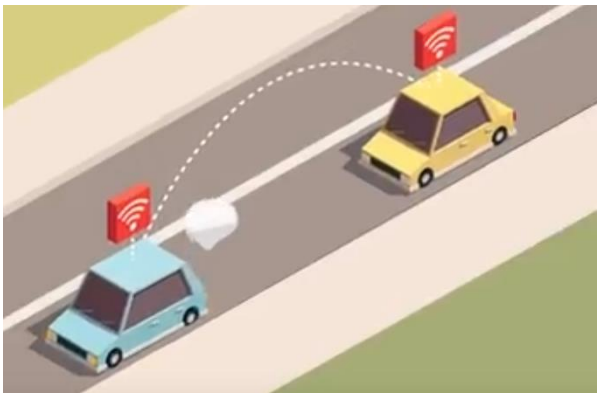
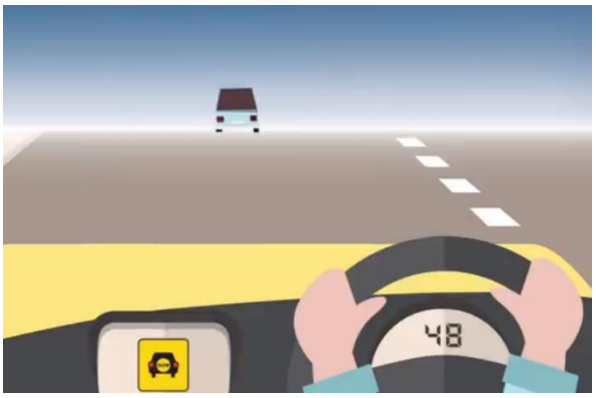


Figure 12 - Stopped or slow vehicle warning [30]



Stopped or slow vehicle warning (in-car view) [30]

Stopped or slow vehicle warning (V2V)

A warning to alert drivers regarding an impending rear-end collision with another cooperative vehicle that is ahead of them.

Scenario:

A car is travelling along a road where it receives an advisory message that a vehicle up front has significantly slowed down (i.e. braking hard). The driver receives information regarding stopped/slowed vehicles ahead of them, displayed through in-vehicle monitors and equips them with better driving decision making. This in turn allows the driver to slow down, brake smoothly, decreasing the probability of an impending rear-end collision.

Figure 13 - In-vehicle speed warning (V2I) [30]

With the two examples above, it shows how these safety use-case applications equips driver with better information for decision masking and aiming to decrease the probability of an impending rear-end collision.

3.1.2 Vehicle-to-Infrastructure (V2I)

V2I also known as Vehicle-to-Infrastructure is a roadside unit deployed within a public-road infrastructure allowing actors to participate within WAVE; receiving and sharing infrastructure communication – for example a message from a traffic light communicated to a car.

Road Side Units (RSU): is a deployable roadside instrument/device providing wireless communications from roadside infrastructure to participants within WAVE.

3.1.2.1 Vehicle Definition

Here in V2I, a vehicle is defined to be any heterogenous, public road mode-of-transport including public transportation (e.g. Bus or Trams). Regardless private or public, a vehicle in V2I can extend to

anyone/anything who traverses on the public road: cars, bicycle, pedestrian, buses, tram, etc. Two examples of C-ITS use case applications in V2V are:

3.1.2.2 V2I Application

In-vehicle speed warning (V2I)

Provides driver with information about the current road's speed limits; notifying them if there is a change in speed limits due to different roads and warns if they are exceeding that limit.

In-vehicle speed warning (in-car view) [30]

Scenario:

Vehicle A is travelling along a road at 50km/h. The car is traversing on a road with a speed limit of 60km/h. It passes a school zone with a speed limit of 40km/h. The car receives a warning informing that speed limit has changed (40 ahead) and that they are exceeding the current road's speed limit of 40km/h and alerting them to slow down. The interaction between V2V's On-board Unit (OBU) and V2I's Road-side unit (RSU) communication flow is depicted in the diagram below.

3.1.3 V2V & V2I Communication Architecture

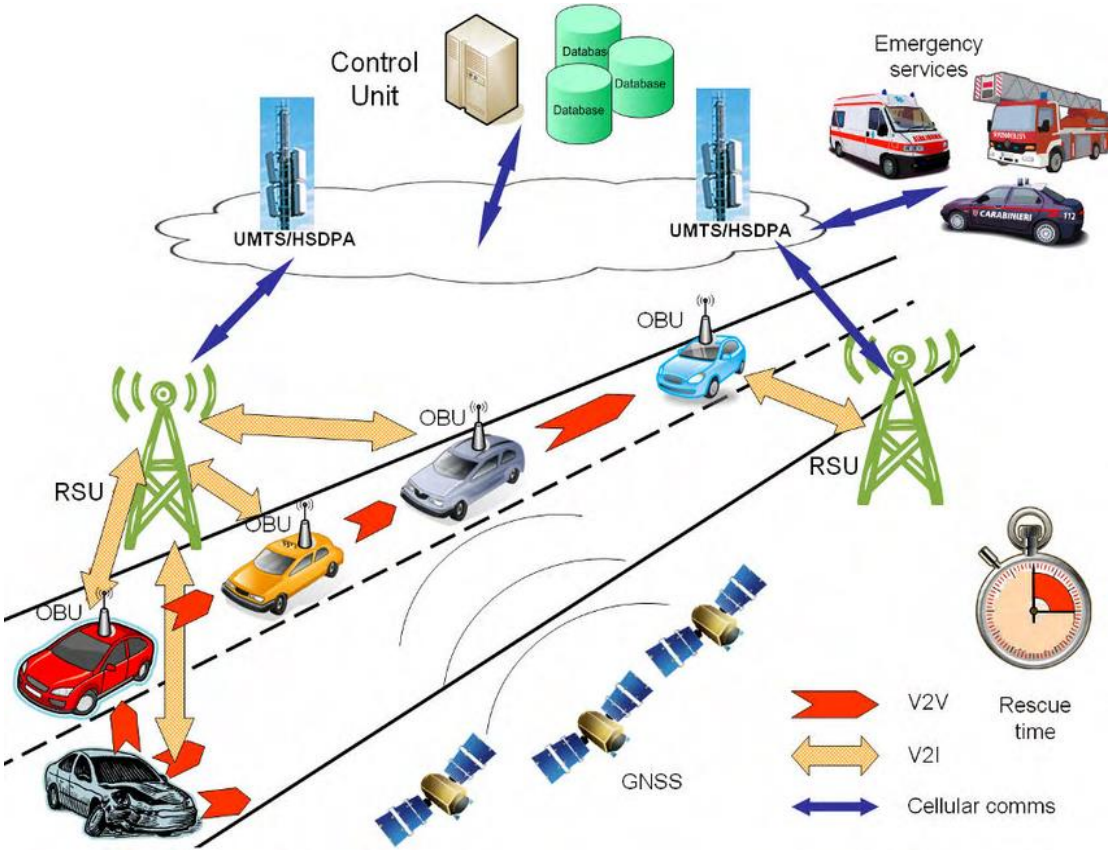
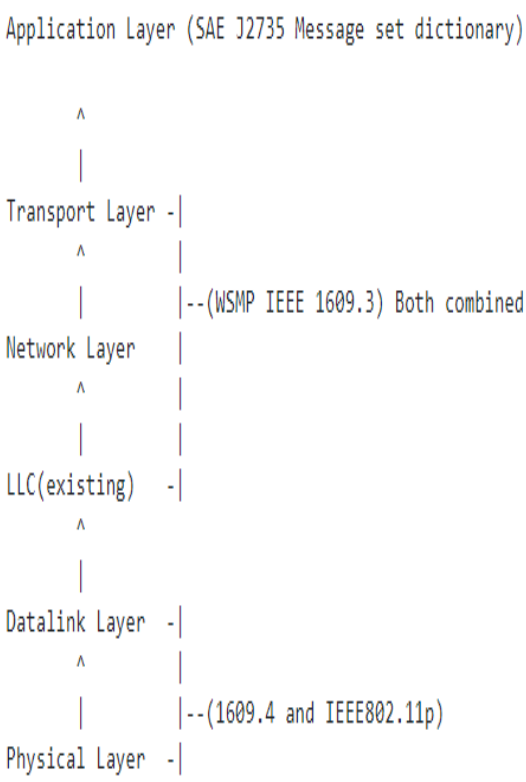


Figure 14 - V2V & V2I Interaction [21]

Figure 15 - DSRC/WAVE stack [28]

Through IEEE802.11p and 1609 family of standards, it allows heterogenous and cross-platform communication regardless of device or vehicle manufacturer participating in WAVE. It created a framework for DSRC communication and standardized surface to vehicle communication.



Vehicles need a standard communication framework to allow cross-platform consumption of V2V messages. Referring to the picture above, imagine that there has been an accident on the road. Each coloured car represents a different make and model. The various actors on the current road represents different private car manufacturers, road side unit, and emergency services. They would all require a common messaging structure to govern interoperable communication. With the discussed PHY + MAC standards above, we proceed further up to the Application Layer. It is here that an interface description (IDL) language be introduced to allow abstracted, cross-platform communication of messages communicated through DSRC.

3.2 Surface to Vehicle Standard

Drilling further into the standards of communication within the Application Layer of DSRC/WAVE stack, V2V communication is governed by two types of SAE standards for performance and transmission requirements.

SAE J2735	Dedicated Short Range Communications (DSRC) Message Set Dictionary
Publication Status:	Work in Progress (WIP)
Issuing Committee:	V2X Core Technical Committee
SAE J2945/1	On-Board System Requirements for V2V Safety Communications
Publication Status:	Work in Progress (WIP)
Issuing Committee:	DSRC Technical Committee

Referring to Figure 11 of the DSRC protocol stacks, SAE J2735 and J23945 are two vital standards in implementing V2V communication. However, for the purpose of this research, J2735 will be further evaluated.

3.2.1 SAE J2735 – DSRC Message Set Dictionary

J2735 is a message set dictionary defining data element, frame definitions and encoding. Using ASN.1 syntax and ASN.1 encoding, it creates a standardized message structure for the exchange of messages through DSRC within a Wireless Access in Vehicular Environment (WAVE).

Known as Abstract Syntax Notation One (ASN.1), is a standard in describing abstract data types and values for interoperability. It is an interface description language (IDL) which defines data structure, allowing serialization and deserialization in a cross-platform manner. It is a widely used standard in telecommunication, computer networking and cryptography. Note that this is a **type declaration notation** which declares and defines data elements and type, but it does not define how to consume/manipulate data type variable – this is defined in Specification and Description Language (SDL).

3.2.1.1 Message Design

The DSRC channels (segmented into 7 channels) over which J2735 messages are communicated through uses a finite space, where each message consumption should be designed to achieve good performance in a realistic traffic scenario. Messages broadcasted over DSRC are required to conform to IEEE 1609.03 – defining standards for WAVE Short Message Protocol (WSMP). A single message protocol to govern both the Transport and Network layer; with messages frequently broadcasted in short packets in an unacknowledged delivery mode. Below outlines the design philosophy of a J2735 Message Standard, its dense encoding (data structure) of information for transmission, adhering to IEEE1609.03.

The three-way dense encoding below is in descending order. This can also be thought of as the different data concept it has declared.

1. **Messages:** Top level of complexity of the data structure to be standardised
2. **Data Frames:** Medium level complexity of data structures to be standardised
3. **Data Elements:** Bottom level and smallest divisions of information content to be standardized.

[35]

Data concept 2 – 3 are all encapsulated to a MessageFrame envelope, ready to be transmitted to its intended recipients. Further breakdown of the above data structure and concepts relating to Basic Safety Messages (BSM) -> BSMCoreData Part I is given below.

3.2.1.2 Message

A message is an envelope containing data frames, inheriting along its children (data elements). It contains standardised type declaration for a specific message; in the context of this standard, the type declared is a data frame containing data elements. Below are two different messages in relation to the scope of this research.

The root and parent envelop of any transmitted J2735 message is wrapped inside a MessageFrame. As a MessageFrame represents the top entry of the data dictionary, all data frame and element are

Figure 16 - MSG_MSGFrame

enveloped within a MessageFrame. This can be analogised as the physical mail envelop which protects and provides a vessel for the mail's content to reach its intended recipient.

ASN.1 Representation:

```
MessageFrame ::= SEQUENCE {
   messageId   MESSAGE-ID-AND-TYPE.&id({MessageTypes}),
   value       MESSAGE-ID-AND-TYPE.&Type({MessageTypes}{@.messageId}),
   ...
   }

MESSAGE-ID-AND-TYPE ::= CLASS {
   &id    DSRCmsgID UNIQUE,
   &Type
   } WITH SYNTAX {&Type IDENTIFIED BY &id}

MessageTypes MESSAGE-ID-AND-TYPE ::= {
   { BasicSafetyMessage    IDENTIFIED BY basicSafetyMessage    } |
   { MapData               IDENTIFIED BY mapData               } |
   { SPAT                  IDENTIFIED BY signalPhaseAndTimingMessage } |
   { CommonSafetyRequest   IDENTIFIED BY commonSafetyRequest   } |
   { EmergencyVehicleAlert IDENTIFIED BY emergencyVehicleAlert } |
   { IntersectionCollision IDENTIFIED BY intersectionCollision } |
   { NMEAcorrections       IDENTIFIED BY nmeaCorrections       } |
   { ProbeDataManagement   IDENTIFIED BY probeDataManagement   } |
   { ProbeVehicleData      IDENTIFIED BY probeVehicleData      } |
   { RoadSideAlert         IDENTIFIED BY roadSideAlert         } |
   { RTCMcorrections       IDENTIFIED BY rtcmCorrections       } |
   { SignalRequestMessage  IDENTIFIED BY signalRequestMessage  } |
   { SignalStatusMessage   IDENTIFIED BY signalStatusMessage   } |
   { TravelerInformation   IDENTIFIED BY travelerInformation   } |
   { PersonalSafetyMessage IDENTIFIED BY personalSafetyMessage } |
   { TestMessage00         IDENTIFIED BY testMessage00 } |
```

MSG_MessageFrame (FRAME)

Use: "*The **MessageFrame** message is used to hold all the defined messages of this standard. Each of the defined messages in this standard has one or more selected locations where additional "regional information" can be inserted into data frames in the message.* [citation]" Figure 6 on the right shows the data structure of a MessageFrame. This structure house and stores a variety of message set, one of which is MSG_BasicSafetyMessage.

MSG_BasicSafetyMessage (BSM)

Use:

*"The **basic safety message (BSM)** is used in a variety of applications to exchange safety data regarding vehicle state. This message is broadcast frequently to surrounding vehicles with data content as required by safety and other applications. Transmission rates are beyond the scope of this standard, but a rate 10 times per second is typical when congestion control algorithms do not prescribe a reduced rate. Part I data shall be included in every BSM. Part II data items are optional for a given BSM and are included as needed according to policies that are beyond the scope of this standard. A BSM without Part II optional content is a valid message.* [35]*"*.

A basic safety message has 2-parts:

1. BSM Part I: BSMcoreData
2. BSM Part II: Content

BSM Part I

Part 1 contains the data frame BSMCoreData. This contains a collection of data element relating to a vehicle's sensor reading. Figure 21 below shows the different elements within DF_BSMCoreData. This is **not optional** and requires Part 1 to be sent '*at all times with each message* [35]'. BSM CoreData Part 1 an important part of vehicular communication, as it contains key information properties relating to vehicle state (e.g. speed, heading, brakes) for consumption by safety applications. The properties BSM CoreData Part I is further discussed below, with DE_Speed field utilised for the development of a model car.

BSM Part II

Part two on the other hand is optional. This may contain optional data relating to specific road region policies. For example, in Australia, this section may content data relating to 'hook turns' that are applicable to the state of Victoria, but not applicable Queensland. The different road regulation/policies within each state can be enforced and applied through BSMpartII and partIIExtension.

DF_BSM is also referenced and used by the following data structures:

- MSG MSG_BasicSafetyMessage (BSM)
- MSG MSG_IntersectionCollisionAvoidance (ICA)

3.2.1.3 Data Frame

A data frame declares the set of data elements requirement for a specific data structure type. In the MESSAGE of BasicSafetyMessage, it contains a data frame of BSMcoreData. Where within this data frame, it defines the data structure to house 12 different data elements.

Figure 17 - DF_BSMCoreData

ASN.1 Representation:

```
BSMcoreData ::= SEQUENCE {
   msgCnt        MsgCount,
   id            TemporaryID,
   secMark       DSecond,
   lat           Latitude,
   long          Longitude,
   elev          Elevation,
   accuracy      PositionalAccuracy,
   transmission  TransmissionState,
   speed         Speed,
   heading       Heading,
   angle         SteeringWheelAngle,
   accelSet      AccelerationSet4Way,
   brakes        BrakeSystemStatus,
   size          VehicleSize
   }
```

DF_BSMcoreData

Use: "*The DF_BSMcoreData data frame contains the critical core data elements deemed to be needed with every BSM issued. This data frame's contents are often referred to as the "BSM Part One", although it is reused in other places as well.* [citation]"

This frame not only need to be sent with each transmission of a message, but also needs to be transmitted at a rate of 10times/second. However, this transmission requirement is outside the scope of J2735.

3.2.1.4 Data Element

Data elements are the smallest unit of information within J2735. An element represents an atomic, unique piece of information entry – allowing them to be shared to other data structure. In the context of BSMCoreData and scope of this research, the following elements are discussed further: *Speed and TemporaryID.*

DE_Speed

ASN.1 Representation:

```
Speed ::= INTEGER (0..8191) -- Units of 0.02 m/s
          -- The value 8191 indicates that
          -- speed is unavailable
```

Figure 18 - DE_Speed

Use: *'This data element represents the vehicle speed expressed in unsigned units of 0.02 meters per second. A value of 8191 shall be used when the speed is unavailable.* [citation]'

DE_Speed is used by 2 data structure within J2735:

- DF_BSMcoreData
- DF_PathHistoryPoint

Please note that this… '*element has been maintained for use by the BSM message. For all new work, the entry DE_Velocity shall be used*' [citation].

DE_TemporaryID

ASN.1 Representation:

```
TemporaryID ::= OCTET STRING (SIZE(4))
```

Figure 19 - DE_TemporaryID

Use: '*This is the 4 octet random device identifier, called the TemporaryID. When used for a mobile OBU device, this value will change periodically to ensure the overall anonymity of the vehicle, unlike a typical wireless or wired 802 device ID. Because this value is used as a means to identify the local vehicles that are interacting during an encounter, it is used in the message set. Other devices, such as infrastructure (RSUs), may have a fixed value for the temporary ID value* [citation]'.

DE_TemporaryID is used by 6 other data structure within J2735 standard:

- DF_BSMcoreData
- DF_VehicleID
- MSG_CommonSafetyRequest (CSR)
- MSG_EmergencyVehicleAlert (EVA)
- MSG_IntersectionCollisionAvoidance (ICA)
- MSG_PersonalSafetyMessage (PSM)

With J2735 defining the message dictionary set of a surface-to-vehicle communication, it does not standardise nor define requirement for a message transmission. J2945/1 on the other hand provides the physical requirement to achieve uniform conformity. With these messaging standards, promoting cross-platform interoperable vehicular communication, it increases accuracy in safety collision prediction.

3.2.2 SAE J2945/1 – On-Board System Requirements for V2V Safety Communications

One challenge within V2V's safety communication is in its communication scalability. As the number of communications between vehicle increases, it needs to ensure that safety applications consuming safety messages are still effective, reliable and non-congested. The need for a standard transmission performance requirement and communication scalability are addressed in SAE J2945.

SAE J2945 standard [1] not only provides the minimum performance requirements for V2V safety communication, but also, it includes a congestion algorithm to support communication scalability. As the number of vehicles communicating increases, communicating channels utilised (CH 172) may reach a saturation point - negatively affecting the performance of safety application, relied on real-time data being consumed. Thus, J2945/1 includes a congestion control algorithm which runs **locally** on each OBE, via adaptation of the transmission power and Inter-Transmission Time (ITT) of the generated BSM. ITT is the interval between two consecutive BSM messages transmitted by a single OBE through the safety channel. Keeping channel utilization remains below saturation level, while maintaining performance requirements for V2V safety communications to be used for safety applications [31].

It is important to highlight that the feasibility of V2V communication for safety application is **reliant** on the adoption of this technology by the public mass. For V2V to begin providing significant safety benefits, it needs to achieve critical mass: where it requires a significant fraction of light-vehicles having the ability to transmit and receive information in an interoperable manner. The **improvement**

in motor-vehicle safety is also dependent on this factor, as the appeal/value to potential buyer of purchasing a vehicle equipped with V2V communication technology is directly related to the numbers of vehicle-owners who have purchased vehicles having similar V2V communication capability.

3.3 V2V's Integration, Adoption & Feasibility in the United States

In the United States, V2V's feasibility not only faces challenges in consumer's integration and adoption, but also conformity from private car manufacturers. With various light-vehicles participating within WAVE, each vehicle assembled from different manufacturer having differing standards, it begs the question on regulation and enforcements of standardized vehicular safety communication? A manufacturer's incentive to install DSRC system to all its light-vehicles, regardless of the product's financial viability? How to ensure coordinated commitment across all vehicle manufacturers? NHTSA realised the lack of motivation or incentive for vehicle manufacturers to conform to the proposed standards. Their market research into the characteristic of the automobile market shows that *'… manufacturers will inevitably face changing economic conditions and perhaps imperfect signals from vehicle buyers and owners, and these signals may not be based on complete information about the effectiveness of V2V technology, or incorporate the necessary foresight to value the* ***potential life-saving benefits of V2V technology*** *during the crucial phase of its diffusion* [28]*'*. Thus, NHTSA believes that without government intervention and support, the resulting uncertainty could undermine vehicle manufactures' plan or weaken its incentive to develop V2V technology to reach its full potential.

A conductor to orchestrate, oversee and regulate participation of various entities involve in WAVE is warranted. Governmental actions are necessary to promote the wide-spread adoption, promotion and integration of this technology. It is imperative that a federal regulation be established to achieve conformity resulting in an interoperable, uniform communications across all light-vehicles.

NHTSA believes that the fusion of V2V with vehicle-resident technologies will promote the further development of vehicle automation systems, striving for a connected vehicular environment and true self-driving vehicles. Acknowledging the potential benefits of V2V, not only in saving lives; but realise that by equipping vehicles with V2V technology, it can be further integrated with other connected applications such as connectivity with RSU (V2I) and pedestrians (V2P) – collectively under the V2X concept. Realising a change to their road infrastructure was warranted, NHTSA proposed a new safety standard (FMVSS No. 150) to create an information environment that would promote, support and be a developmental stage in achieving widespread deployment and adoption of vehicular communication technology.

In establishing a regulatory standard for V2V, it is the first step in propelling and integrating the deployment of other wireless connected hardware. Furthermore, it provides a platform for communication-based technology to allow future integration of automated vehicles, independent of its manufacturer.

4.0 National Highway Traffic Safety Administration

The NHTSA is a branch of the Department of Transportation with a core mission to '*save lives, prevent injuries and reduce economic costs due to road traffic crashes, through education, research, safety standards and enforcement activity* [28].' As part of its responsibilities, it is task with writing and enforcing Federal Motor Vehicle Safety Standards, or FMVSS in short. With the advancement in wireless technology, promising better safety application and reducing lives, the NHTSA is proposing in issuing a new FMVSS No.150 to achieve its core mission.

4.1 FMVSS No. 150 – Vehicle-to-Vehicle Communications

In January 2017, the NHTSA issued a noticed of proposed rulemaking (NPRM) to establish a new FMVSS No. 150 which mandating that all new light vehicles manufacturer be capable of V2V communication and the ability to send and receive Basic Safety Messages to and from other vehicles within its surrounding.

Federal Motor Vehicle Safety Standards, No. 150

AGENCY:	National Highway Traffic Safety Administration (NHTSA), Department of Transportation (DOT)
ACTION:	Notice of Proposed Rule Making (NPRM)
Docket No:	NHTSA-2016-0126

FMVSS No. 150 proposes to mandate V2V communication standardization for new light vehicles in the United States. This proposal proposes to mandate:

- V2V communications for new light vehicles and to standardize the message and format of message transmissions.
- V2V communication performance for a vehicle's OBU (onboard unit) to transmit Basic Safety Messages Part I to surrounding vehicles.
- Provides a framework for other technologies to adhere to interoperability requirements and importantly its interoperability with DSRC.

Upon its publication, it requested for comments against its proposed rulemaking. Within a 90-day comment period, the NHTSA received 450 comments relating to the topics below:

- Technology Strategy;
- Implementation Timing;
- Detailed Technical Information;
- Cost Estimates;
- Potential Health Effects; and
- Privacy and Security.
  [32]

The NHTSA firmly believes that by equipping drivers with timely warnings, relevant information of impending crash situations, V2V has the potential to revolutionize motor vehicle safety standards.

4.1.1 FMVSS No. 150 – Security Framework

This section below is an extract from NHTSA's proposed rulemaking. Please note that the proposal topics below are scoped to V2V communication, security and privacy.

Topic	Proposal
Communication Technology	*NHTSA proposes to mandate DSRC technology—A DSRC unit in a vehicle sends out and receives ''basic safety messages'' (BSMs). DSRC communications within the 5.850 to 5.925 MHz band are governed by FCC 47*[10]*.* *In reference to the OSI model, the physical and data link layers (layers 1and 2) are addressed primarily by IEEE 802.11p as well as P1609.4;* ▪ *network, transport, and session layers (3,4 and 5) are addressed primarily by P1609.3;* ▪ *security communications are addressed by P1609.2; and* ▪ *additional session and prioritization related protocols are addressed by P1609.12.* *This mandate could also be satisfied using non-DSRC technologies that meet certain performance and interoperability standards.*
Message Format and Information	*NHTSA proposes to standardize the content, initialization time, and transmission characteristics of the Basic Safety Message (BSM) regardless of the V2V communication technology potentially used.* *The agency's proposed content requirements for BSMs are largely consistent with voluntary consensus standards* ***SAE 2735 and SAE 2945*** *which contains data elements such as speed, heading, trajectory, and other information, although NHTSA purposely does not require some elements to alleviate potential privacy concerns. Standardizing the message will facilitate V2V devices ''speaking the same language,'' to ensure interoperability. Vehicles will not be able to ''understand'' the basic safety message content hindering the ability to inform drivers of potential crashes.*
Message Authentication	*NHTSA proposes V2V devices sign and verify their basic safety messages using a* ***Public Key Infrastructure (PKI) digital signature algorithm*** *in accordance with performance requirements and test procedures for BSM transmission and the signing of BSMs.* *The agency believes this will establish a level of confidence in the messages exchanged between vehicles and ensure that basic safety message information is being received from devices that have been certified to operate properly, are enrolled in the security network, and are in good working condition. It is also important that safety applications be able to distinguish these from messages originated by ''bad actors,'' or defective devices, as well as from messages that have been modified or changed while in transit.* *<u>Alternative Approach— Performance-based Only</u>* *This first alternative for message authentication is less prescriptive and defines a performance-based approach but not a specific architecture or technical requirement for message authentication. This performance only approach simply states that a receiver of a BSM message must be able to validate the contents of a message such that it can reasonably confirm that the message originated from a single valid V2V device, and the message was not altered during transmission. The agency seeks comment on this potential alternative.* *<u>Alternative Approach—No Message Authentication</u>* *This second alternative stays silent on a specific message authentication requirement. BSM messages would still be validated with a checksum, or other integrity check, and be passed through a misbehavior detection system to attempt to filter malicious or misconfigured messages. Implementers would be free to include message authentication as an optional function. The agency seeks comment on this potential alternative.*

[10] FCC 47 CFR parts 0, 1, 2 and 95 for onboard equipment and part 90 for road side units.

Misbehavior Detection and Reporting	*Primary Misbehavior Detection and Reporting Proposal* *NHTSA proposes to mandate requirements that would establish procedures for communicating with a Security Credential Management System to report misbehavior; and learn of misbehavior by other participants. This includes detection methods for a device hardware and software to ensure that the device has not been altered or tampered with from intended behavior. This approach enhances the ability of V2V devices to identify and block messages from other misbehaving or malfunctioning V2V devices.* *Misbehavior Detection Alternative Approach* *An alternative for misbehavior detection imposes no requirement to report misbehavior or implement device blocking based to an authority. However, implementers would need to identify methods that check a devices' functionality, including hardware and software, to ensure that the device has not been altered or tampered with from intended behavior. Implementers would be free to include misbehavior detection and reporting and as optional functions. The agency seeks comment on this alternative.*
Hardware Security	*NHTSA proposes that V2V equipment be ''hardened'' against intrusion (FIPS– 140 Level 3) by entities attempting to steal its security credentials.*
Effective Date	*The agency is proposing that the effective date for manufacturers to begin implementing these new requirements would be two model years after the final rule is adopted, with a three-year phase- in period to accommodate vehicle manufacturers' product cycles. Assuming a final rule is issued in 2019, this would mean that the phase-in period would begin in 2021, and all vehicles subject to that final rule would be required to comply in 2023.*
Authority	*Under the Vehicle Safety Act, 49 U.S.C. 30101 et seq., the agency has the legal authority to require new vehicles to be equipped with V2V technology and to use it, as discussed in Section VI below. NHTSA has broad statutory authority to regulate motor vehicles and items of motor vehicle equipment, and to establish FMVSSs to address vehicle safety needs.*
Privacy and Security	*V2V systems would be required to be designed from the outset to minimize risks to consumer privacy. The NPRM proposes to exclude from V2V transmitting information that directly identifies a specific vehicle or individual regularly associated with a vehicle, such as owner's or driver's name, address, or vehicle identification numbers, as well as data ''reasonably linkable'' to an individual.* *NHTSA intends for the term "reasonably linkable," as used in this NPRM, to have the same meaning as the term "as a practical matter linkable" as used in the definition of "personal data" in Section 4 of the White House Consumer Privacy Bill of Rights: "data that are under the control of a covered entity, not otherwise generally available to the public through lawful means, and are linked, or as a practical matter linkable by the covered entity, to a specific individual, or linked to a device that is associated with or routinely used by an individual."* *Additionally, the proposal contains specific privacy and security requirements with which manufacturers would be required to comply. The Draft Privacy Impact Assessment that accompanies this proposal contains detailed information on the potential privacy risks posed by the V2V communications system, as well as the controls designed into that system to minimize risks to consumer privacy.*

Section 4.1.1 above are referenced from NHTSA FMVSS No. 150 extracting relevant key topics relating to the proposed security framework for V2V. The below sections evaluate and breaks down the working of the proposed V2V security framework enabling vehicular communication for the transmission of BSM.

4.1.2 FMVSS No. 150 – Basic Safety Message

With BSM collecting key information for safety application to help mitigate crashes, FMVSS No. 150 proposes that vehicular communication system is built into the vehicle at manufacturing. OnBoard Unit (OBU) device being able to conform to the security standards of 1609 and encoding/decoding format to J2735. With J2735 declaring data structure and type declaration, the proposed vehicular communication of BSM warrants a security certificated issued by a certificate authority to digitally sign each basic safety message.

Certificate	Message content	Signature	Timestamp
Pseudonym Certificate • *Public Key* ..... • Signature of the Pseudonym Certificate Authority. Validity Period ..... • Says when certificate effective and when expires.	(*i.e.*, the speed, heading, location, etc. information that supports the safety applications).	Produced from the following steps: • Compute hash of the Message Content and Timestamp. • Use your *private key* to create an encoded string of numbers. • The encoded string of numbers is your *signature*.	(*i.e.*, when the information is transmitted.]).

Figure 20 - Components within a BSM wrapper [32]

NHTSA proposed BSM transmission is broken into four components (read from left to right), with the security certificate composed with the following elements:

1) Certificate validity period - a date range describing the validity period of a specific issued certificate
2) Public Key corresponding to a Private Key
3) Digital signature given from the Pseudonym Certificate Authority (PCA) – contains the signature of the PCA issued from the SCMS allowing message receivers

Using the message content and timestamp component as inputs it constructs a signature with the following process:

- Creates a hash using the 2 input of message content and timestamp.
- Produced hash content is then inputted through an Elliptical Curve Digital Signature – an equation to create a string of encoded number
- This resulting encoded string of numbers is the 'digital signature'

SCMS and its distribution of public keys are discussed further in the below section. But before drilling into security, FMVSS No. 150 outlines the hardware requirement and workings of a BSM transmission.

4.1.3 FMVSS No. 150 – BSM Transmission Requirement

J2945 spawned from a research showing that a separate standard describing specific requirements for data elements for BSM messages is warranted. Table 3 below is an extract which specifies the performance requirements for a BSM transmission.

It specifies that a BSM message would need to be transmitted at a frequency of **10 times** per second under non-congested conditions; having a minimum longitudinal range of 300m and lateral range of 360 degrees around the vehicle; transmitting at a 6 Mbps data rate on CH 172. Moreover, CH 172 is responsible for all BSM transmission and reception as this is a 'safety-critical communication' channel. With BSM messages being a key driver for V2V safety awareness communication and application, standardized BSM transmission is integral to achieve interoperability. Regardless of the different manufacturing entity of a vehicle and its on-board equipment (OBE), each vehicle broadcasting any

BSM message would need to conform to the requirement of Range (longitudinal, lateral and elevation), Reliability, Data Rate, Transmission Frequency and Staggering Transmission Time.

Table 3 - BSM Transmission Requirements [32]

TABLE III–1—SUMMARY OF BSM TRANSMISSION REQUIREMENTS

Requirement	Proposal	Basis	Relationship to standards	Reason
Range (longitudinal & lateral) ..	Minimum 300m; 360 degrees around vehicle.	CAMP—application tested in SPMD also calculation of range needed for DNPW.	SAE J2945/1	The setting is based on the need to provide accurate and timely safety alerts. The setting was obtained by extensively testing commercially available equipment and automotive sensors in a wide variety of driving environments.
Range (Elevation)	At elevation angle of +10 degrees and −6 degrees.	CAMP and BAH research and testing capabilities.	SAE J2945/1	Same as above.
Reliability	Packet Error Rate <10%	CAMP and BAH	SAE J2945/1	Same as above.
BSM Radio Channel	All BSM transmissions and receptions on 172 (safety-critical communications).	FCC rules	SAE J2945/1	Same as above.
Data Rate	6 Mbps	CAMP and BAH research—CAMP research shows PER degradation using 12 Mbps. BAH research indicates problems after 500m, also BAH test done under "open field" conditions.	SAE J2945/1 (one of the bitrates included in 802.11).	Same as above—Also Current developers support a 6 Mbps data rate. More data and testing is needed to change the data rate and determine if a changing rate can be used and support crash avoidance.
Transmission Frequency	10 times per second under non-congested conditions.	CAMP—trade-off between long inter-packet delays experienced by V2V safety applications and heavy wireless channel utilization.	SAE J2945/1	Accepted among experts to support V2V crash avoidance.
Staggering Transmission Time	Random transmission of BSMs every 100 +/−ms between 0 and 5 ms.	Mitigate channel congestion if all devices transmitted at same time—CAMP and BAH research.	SAE J2945/1	Due to accuracy of devices need to mimic the stagger experienced during SPMD to avoid message collisions to facilitate efficient channel usage.

4.1.4 FMVSS No. 150 – Message Authentication

For message authentication, NHTSA is proposing to establish a Security Credential Management System (SCMS) to sign and verify BSM through utilising Vehicular Public Key Infrastructure (V-PKI) to prohibit unauthorized message modification, safeguard data and uphold privacy.

4.1.4.1 Security Credential Management System (SCMS)

Also known as SCMS, it is the leading candidate design implementing V-PKI to facilitate trusted communications through management of security certificates for authorised devices while protecting the privacy of its vehicles (users).

These are the primary use cases of a SCMS:

- Bootstrapping – establishing devices on the network
- Provides PCA (Pseudo Certificate)
- Sending misbehaviour reports
- Distribution of Certificate Revocation List (CRL)
- Managing root certificate Authority (CA)

Technically, the proposed SCMS will:

1) Issue 20 public key certificates per week for each vehicle
2) During the transmission of a BSM, a certificate out of the 20 is chosen to sign and populate the first row of figure 4.1 – BSM message wrapper.
3) To ensure privacy, the 20 certificates issued for a single vehicle is rotated every five minutes to ensure that no BSM will be linked to more than 5 minutes of other safety messages at a time.
4) NHTSA further proposes that each device to completely discard their twenty certificates each week, and to replace them with 20 new certificates issued by SCMS.

4.1.4.2 Certificate Revocation List

A certificate revocation list is utilised to maintain and police valid actors participating within WAVE. CRL contains a list of digital certificates that have been revoked and is generated by the Misbehaviour Authority. A part of the SCMS, the MA ensures that V2V participants are aware of bad actors, and if bad messages are received from actors placed in CRL, its system (e.g. OBU) will warn participant to ignore the message. Bad actors are placed within CRL through misbehaviour reports received from vehicles sent from their OBU.

4.1.4.3 Misbehaviour Authority/Reports

Through FMVSS 150, each OBU will have its own algorithm baked within its Hardware Security Module (HSM) to detect local anomaly and report a misbehaving device. Algorithms of misbehaviour have not been explicitly defined by the NHTSA nor published. If an anomaly is detected and confirmed by a series of secondary plausible checks through vehicles communicating with its surrounding vehicle, a misbehaviour report is sent to the Misbehaviour Authority (MA). MA lies within SCMS and broadcast CRL frequency have not been explicitly defined and RFC.

MA is intended to gather misbehaviour reports from OBU devices participating in the network. These anomaly report would be analysed to assist the governance of bad actor and publication of CRL. It is expected that an OBU device would be able to detect, generate and create '*within 2 seconds after the misbehavior is detected, and signed* [32]'. Each misbehaviour reports are signed with the same transmitted credentials as enclosed within a BSM wrapper. Its timestamp is signed by the reporting device at the time of report creation. Reporting and updating CRL in real time require robust and active connection both within VANET and cellular tower. This active connection is impeded if participating vehicles are traveling through a tunnel or an area with limited network connection.

4.2 Vehicular Public Key Infrastructure (V-PKI)

<u>Cumbersome Key Management</u>

With the proposed generation of 20 new public certificates per vehicle every week, this creates a cumbersome and large key management. Not only does the SCMS being the CA, have to maintain a very large certificate database to verify identity, it also has to maintain a dynamic and organic revocation list to propagate to participating vehicles within WAVE.

281.3 million [33] registered light vehicles in the United States, with each vehicle having 20 public certificates per week, the SCMS is looking to maintain and store at minimum 5,626,000,000 certificates. Maintaining this large amount of database is not only inefficient, but also becomes a single source of failure. The integrity of vehicular communication lies on one centralised SCMS, if compromised it could jeopardize the lives of millions and cause millions of dollars.

Inefficient CRL Propagation Method

The proposed security framework is not mature to cover prevention and efficient propagation of CRL. A vehicle is deemed a bad actor **after** it has committed the transgression, however this goes against the concept of V2V of reducing road accident through prevention.

NHTSA's proposed propagation of pushing out certificate revocation lists to all vehicles participating within WAVE will be too late to prevent the initial accident. The addition of the vehicle to the list is a mitigating action to prevent future accident caused by that specific vehicle, now deemed as a bad actor. Furthermore, the transmission of CLR to each light-vehicle participating in WAVE would be traffic intensive and inefficient. Having each individual car to regularly download gigabytes of data concurrently and synchronised to SCMS would be far larger than the largest CRLs used in WebPKI today.

Assurance of identity does not mean assurance of content

Public Key Infrastructure Framework uses digital keys & certificate to verify and assure the identity of its subscribers for external identity authentication. However, the verification and assurance of a digital signature through a CA proving an identity could not translate to assurance of message content.

With the proposed message wrapper (Figure 21) and the utilization of PKI, NHTSA is *'confidence in the information contained in a BSM message because it knows that the SCMS previously confirmed the sender is an approved device and issued these credentials* [28]'. Its proposal of a Security Credential Management System utilising a PKI, in NHTSA's perspective is sufficient enough in order to uphold vehicular communication security as well as guaranteeing its message content through having confidence in its ability to authenticate message identity.

NHTSA proposed rulemaking of FMVSS No. 150 does not fully interrogate a BSM message when it is being ingested. It's belief and inherited assurance of if a message has been certified through its signature and endorsed by a Certificate Authority through SCMS, that the message content within the message is authenticated and proven. However, this is not the only security vulnerability it opens, if the proposed credential management system got developed and realising vehicular public key infrastructure.

4.2.1 SCMS Threat Analysis in Vehicular Communications

Research conducted by Dartmouth University analysed NHTSA's proposed SCMS to identify and evaluate threats using Microsoft Thread Modeling Toolkit (MTMT). This tool identifies threats into six categories represented through the acronym STRIDE:

1. **S**poofing Identity
2. **T**ampering with Data,
3. **R**epudiation
4. **I**nformation Disclosure
5. **D**enial of Service
6. **E**levation of Privilege

For a full reference of the different threats evaluated by STRIDE, please refer to Appendix D. Through the analyses conducted, its vulnerability assessment highlighted category 1 (S) and 5(D) to be at most risk. These analysis shares the same threat insight as to this research in evaluating PKI's feasibility when implemented at a scale this large.

4.2.2 Spoofing of Identity

Spoofing of Identity looks at assessing the impact when an attacker could potentially act as another user/component of the system, spoofing/falsifying authentication identity. Within a vehicular network (VANET), authentication is the first line of defence against any attacker. However, implementing an authentication system through V-PKI becomes tricky, due to its core design philosophy of privacy and anonymity – to protect the identity of the owners/passenger of their respective vehicle. The rotating public keys and issuance of new certificates each week highlights NHTSA's importance of privacy and anonymity, as a vehicle within VANET could be track through its travel habit and eventually deduce the behaviour path and presence of that vehicle at any given time.

4.2.3 Denial of Service

DOS is one of the largest threats highlighted through STRIDE. It's high impact ability to affect the availabilities of participating actors (e.g. OBU or RSU) within WAVE is major vulnerability to SCMS. As SCMS cannot guarantee the direct and constant communication availability to all participating actors, albeit RSU or an OBU. For example, a SCMS cannot guarantee direct access and notification to receive the updated distribute certification list. As most vehicles will not have direct access to SCMS all the time; i.e. a car traversing through a tunnel could not have direct access to SCMS, as such they would require to communicate with a pass-through interface such as an RSU to retrieve the latest copy of the CRL.

Unfortunately, if an attacker has already commenced its attack, by the time surrounding vehicles are aware of misbehaving vehicles, this is already in the act of mitigating, not preventing the initial attack. Clashing against the core objective of V2V to prevent and reduce road accident. Moreover, an attacker could delay and slow down the distribution of the latest CRL by flooding RSU with falsified transmission, potentially giving the attacker more time to launch attacks prior to being detected and its certificate revoked.

Direct and constant access for a vehicle to SCMS cannot be considered to be highly available and constant. Traversing through a tunnel or a closed area would disrupt access and causes a delay for cars within the vicinity to receive the new distributed CRL. NHTSA's proposed method of tackling this problem would be to communicate with an RSU within its surrounding to receive the updated CRL. The CAVI project discussed in section 4 also did not consider active connections when a car traverses through a tunnel. However, the RSU does not check message content and shares the same philosophical belief of SCMS – where a message content is guaranteed and assured if a message's identity has been certified through its signature and endorsed by a Certificate Authority. The proposed system does not ensure immediate tagging and revocation of bad actors/vehicle. Having a limitation of network access, inherited blind trust from message identity transcending down to message content does not deter and tag bad actors fast enough especially in a scenario where a car is travelling down a tunnel.

This research developed and proposed a verification system to address these points. An RSU does not need to wait for an updated distributed CRL as it has its own independent BSM verification framework to discern whether a car is broadcasting untampered or tampered messages. Allowing it to broadcast updated CRL through evaluating in real-time BSM messages transmitted by vehicles in its current vicinity. This system, not only bolster security confidence at locations with primitive network access, but most importantly, it is managed independently to SCMS and its core function is to verify that the BSM message content is untampered and true to what the RSU is reading. As BSM Part I is a mandatory

transmission for all vehicles participating, it is imperative that the data encapsulated within a BSM message is genuine, as tampered messages increases road accident, reduces security confidence of SCMS and deter the development of V2V safety applications.

4.3 Direct Misbehaviour Reporting

Accuracy and warning notifications from safety applications are heavily reliant on frequent updates of CRL being distributed and consumed. The generation and distribution of CRL by Misbehaviour Authority (MA) is dependent on incoming reports from vehicle participants and confirmation from MA to place a vehicle into CRL. The process of reporting, confirmation leading to the generation and frequency of CRL distribution for safety application predictions, relies on surrounding network infrastructure.

Imagine a scenario where a car participating within WAVE travels through a tunnel. Through VANET and V-PKI, the car travels further through a 5km tunnel for 15 minutes with having CRL v.1 stored within its OBU. Travelling through the tunnel, a drunk driver has been spotted and its vehicle is swerving left, right and speeding. This in turn causes multiple misbehaviour reports to be generated by vehicles within its surroundings. However, the ability for this report to be generated and reach MA under 2 seconds within an enclosed space, under a tunnel is highly unlikely. Begging the question, how would a vehicle report an anomaly to MA fast enough under these scenarios to allow generated CRL v.2 update? Noting further, NHTSA's vague proposal for 'periodical' updates and distribution of CRL does not increase the probability of preventing a bad actor; but more towards a mitigation factor to reduce the chance of it happening again. Instead of relying on vehicles to generate a report to the misbehaviour authority, reusing sensors installed on tunnels (e.g. Figure 21 using the speed sensors and signs already installed), it could be paired with a road side unit with direct connection and report

Figure 21 - Speed cameras & signs within a tunnel

escalation to MA for faster process and distribution. RSU can be used as a wireless extender, allowing further propagation reach and spread in distributing CRL.

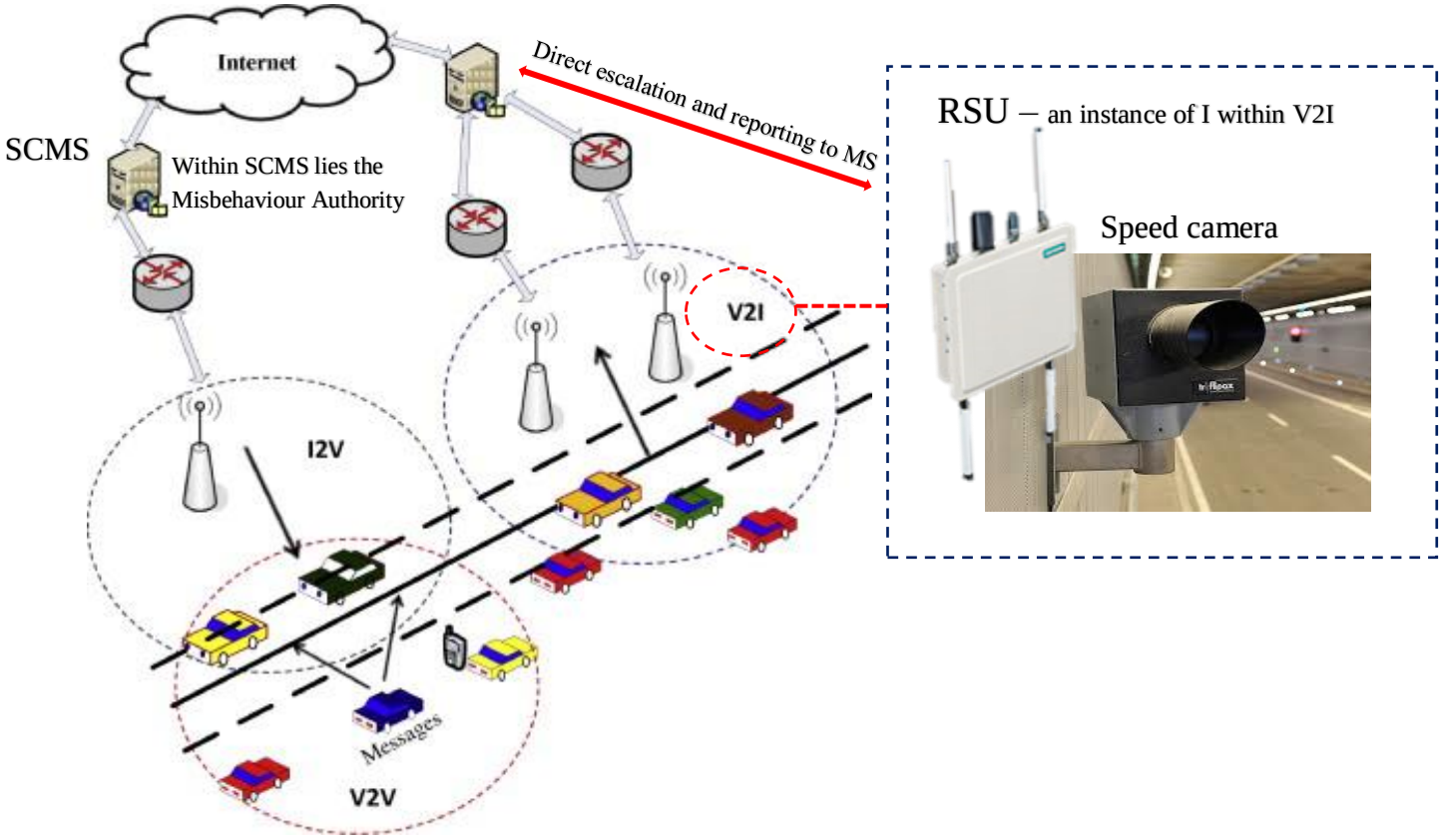
Through preinstalled sensors within existing road infrastructure, paired with a road side unit to provide faster report escalation and CRL distribution point; and able to analyse reports from VANET regarding a bad vehicular actor in its presence, produces a solution which gives a real time situational awareness.

Direct RSU Reporting

Pivoting perspective from the proposed misbehaviour reports being generated only from OBU, what if road side units (RSU) are able to detect and generate misbehaviour reports to update and report misbehaviour to MA directly? Under the assumption that RSUs (e.g. traffic lights, speed cameras) are govern and maintain by public/federal enforcement, it is able to grant direct misbehaviour reports from RSU to MA for a real time situational awareness of anomalies.

RSU can be seen as an extension to a speed sensor/camera. For example, with the current installation of speed sensor in tunnels, RSU can be mounted together with a speed camera in order to read raw speed sensor input from the sensor and generate a report directly to MA; resulting in an easier integration and adoption. Figure 30 above shows a tunnel and its highlighted red circles depicts different speed signs and sensors (e.g. camera) installed within the tunnel. A road side unit can be extended to be able to have its own algorithm and HSM baked inside to generate a misbehaviour report with hardware level encryption transmitted directly to MA using data read directly from previously installed speed sensor. Extending the RSU's ability to verify BSM message through an independent verification sensor; reusing sensors already installed within road infrastructure (e.g. speed cameras, induction plates, RFID gantry readers) and provide a faster misbehaviour reporting and escalation point to MA.

Figure 22 - Extended RSU in V2I



Reusing current sensors within road infrastructure

The RSU can be further extended and utilised as a V2V message inspector. With light-vehicles already communicating through V2V, the communication between a vehicle to an **infrastructure** (V2**I**) can be further extended by allowing the RSU to inspect and verify message content. Broadcasted basic safety messages will be inspected and integrating with previously installed sensors (e.g. speed, induction, gantry reader) it is able to associate and verify transmitted BSM fields contained within the message with readings from paired sensor. Conceptually like Figure 22 above, a road side unit device capable of receiving and transmitting J2735 messages as well as adhering to ISO 17427 standards it is able to provide direct MA report access to allow faster and more accurate CRL distribution.

In combination with the thread assessment model STRIDE, a verification framework can be developed within the RSU to provide its own misbehaviour report. Not only providing a faster escalation point to MA (Misbehaviour Authority), as it confirms bad actor using its independent sensor rather than waiting for confirmation and agreement from vehicles in its surroundings. This patches the trust hierarchy of PKI where through message authentication certifying sender's identity, to blindly consume message content. The verification framework within the RSU is part of V2I, and is able to inspect message content against readings from paired sensor. For example, an independent sensor providing atomic reading such as car velocity from a speed camera paired with an RSU, extends V2I capability to be able to verify broadcasted BSM.

Extending RSU with sensors

Pairing a road side unit like the figure above not only increases the chance of preventing an accident as it does not wait for vehicles to confirm the presence of a bad actor to generate a report; but also gives the ability to verify broadcasted BSM, as it inspects message content detecting bad actor and able to generate report directly to the MA. Using existing sensors installed within road infrastructure, a verification framework within RSU is able to compliment anomaly detection and reporting. For example: a speed camera – providing km/h independent reading from speed camera to improve detection of message spoofing of BSM content of 'speed' field; or a RFID Gantry Reader – providing vehicle and registration detail to improve law authority detection and enforcement of stolen and/or unlicensed vehicle registration.

An independent verification sensor provides further confidence for the reports RSU generate directly to Misbehaviour Authority. As the RSU is a connected device within SCMS network, it is able to generate an updated CRL confidently using information it receives from RSU device sitting and maintain by SCMS network. A verification system inspecting BSM content is proposed and have been developed below using speed as the variable to compare and verify from an independent sensor reading. Note that the proposed prototype and system below represents a smaller simulated scale and does not cover all security aspect of IEEE1609.02. The proposed system below shows the extended capability if pre-existing sensors are reused and paired with a road side unit. Enabling BSM content verification, allowing faster detection of bad actors and providing a faster escalation point to report misbehaviours.

5.0 System Design, Development and Procedures

5.1 Overall Design

With NHTSA's proposed FMVSS No. 150 and its pitfall of inherently trusting message content from message identification, this section discusses the proposed system design, development and procedures of developing a BSM verification framework using two model cars.

The developed verification framework contains two deliverable components, outline below are the in-scope and out-of-scope requirements of each components.

5.1.1 Model Car Requirements

This model car prototype simulates good or bad actors participating within WAVE.

Prototype's Physical Requirement

- Develop 2 remote controlled car representing good and bad.
- Remote controlled (RC) car with a 4WD motors.
- Manoeuvring controls includes Forward, Backward, Left and Right
- Conversion of speed from PulseWidthModulation (PWM) to linear speed.

Message Structure Requirement

- RC car wirelessly broadcast message to RSU server using 802.11ac protocol
- RC car connects and are assigned a static IP address by RSU server.
- Each car representing an actor will be assigned a hardcoded Temporary_ID to discern the identity of the car.
- Broadcasted message is in ASN.1 format conforming to SAE J2735
- Each broadcasted MessageFrame must contain the correct data structure defined in J2735 including all message type.
- Broadcast of MessageFrame will contain **real-time update** to the below message type, frame and field.

  MessageID: 20
  MessageType: Basic Safety Message
  DataFrame
  BSMCoreData: DF_BSMcoreData //Part 1
  Field: Speed

Message Transmission Requirement

- Each BSM message is transmitted 10 times/second, with no specified channel.

GUI Requirement[11]

- Provides an interface to allow user to interact with the car
- Provides an interface to display warning messages from RSU

Due to time constraints during the development of this research, the following aspects listed below are not implemented as part of the design, however they are discussed:

- Certificates within each message wrapper

[11] Graphical User Interface

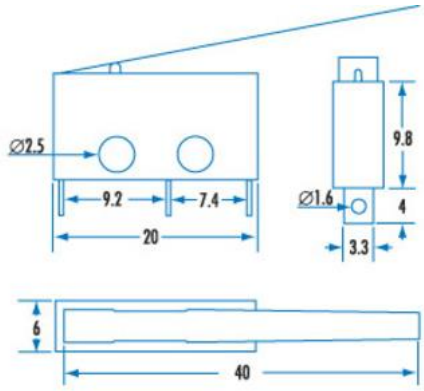
- Other DataFrame within the MessageWrapper that is not BSM.
- TemporaryID is not randomly assigned and does change periodically to discern a good and bad car's identity.
- The implementation of Security Credential Management System. Each car does not receive 20 public key certificate.

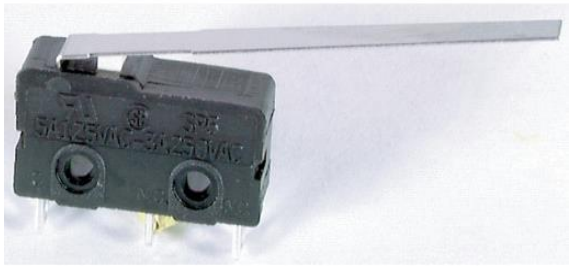
5.1.2 RSU Requirements

The developed prototype simulates a Road side Unit (RSU) of the following capabilities:

Hardware Requirements

- Receive and transmit BSM using J2735 protocol and 802.11ac
- SPDT[12] 3A micro switch with lever to trigger speed verification
- Independent wiring (i.e. resistors) for independent speed verification with Pi's pin





Software Requirements

- Decode ASN.1 message to verify its speed reading
- Maintain multiple concurrent client connections and readings
- Calculates separately two sets of speed lever readings using DTS Formulae:

$$speed = \frac{\text{Distance}}{\text{time}}$$

- Connects an independent speed verification sensor to compare and verify car's broadcasted speed in J2735 format.
- To tag bad actors when speed verification fails and from participating at the road.
- To be able to notify connected vehicles regarding bad actors within the road.

This ends the section of requirement and pro ceeds to the development of system.

[12] Single pole, double throw

5.2 Model Car Development

A model toy car was developed with the above requirements and utilising the below components, interfaced and processed by a RaspberryPi.

Invented by Eben Upton, RaspberryPi is a single-board computer created to promote and educate basic interaction of electrical circuitry and software. A program was created and ran within Pi 3 to wirelessly control the car while broadcasting J2735 messages.

[Raspberry Pi 3 Model B+]

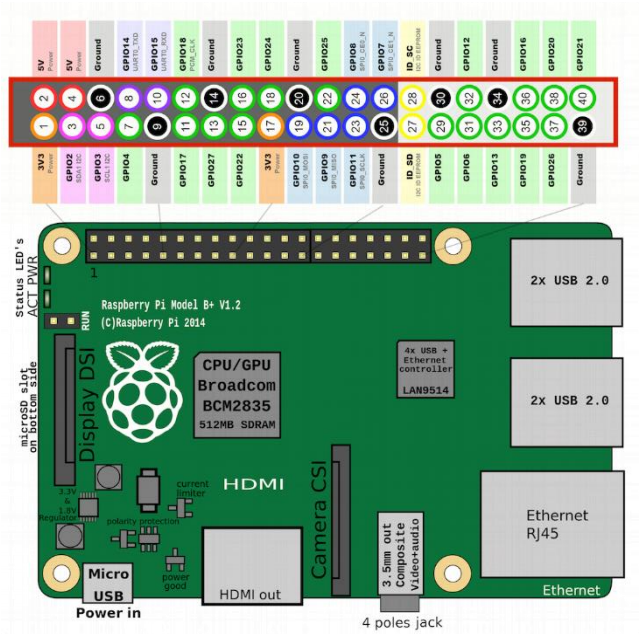
- Broadcom BCM2837B0
- 64bit
- ARMv8
- Quad Core Cortex A53
- 1.4 GHz
- IEEE 802.11.b/g/n/ac compatible

OS Installed:
Raspbian Stretch with desktop and recommended software

Description:
Image with desktop and recommended software based on Debian Stretch

Version:
Release date:2019-04-08
Kernel version:4.14

Figure 23 - PI 3 Model B+ with Pin Layout



Raspian Stretch desktop was then burnt using Etcher [13]onto a microSD card. This provides a standard of operating system to boot for the RaspberryPi (RPi).

5.2.1 Software Component

A car controller program using C++ and Visual Studio 2017 was developed to utilise the PI's physical and wireless capability. Through utilising its 40-pin GPIO header, this program combines readings from connected sensors (listed below); wirelessly broadcasting J2735 message and controlled through a console.

[13] Etcher is a software which flashes OS images to SD cards & USB drives

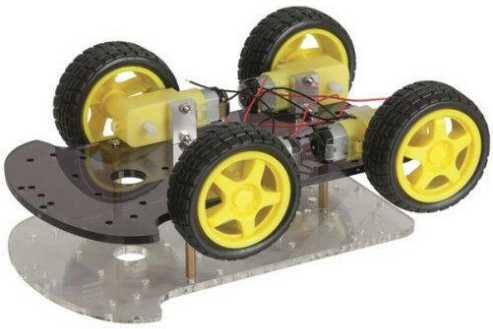
5.2.2 Hardware Components

5.2.2.1 Model Toy Car Chassis

A model toy car kit was purchased with one motor and gearbox per wheel.

Motor voltage: 5-10VDC

Dimensions: 240(L) x 160(W) x 100(H)mm



5.2.2.2 Sensors

Below are the sensors installed onto the car chassis displayed above and wiring diagram of the components wired to the PI.

Figure 24 - Car wiring diagram

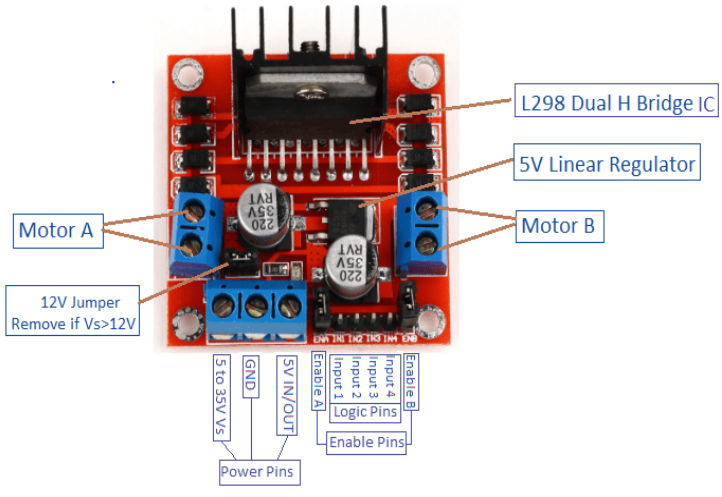
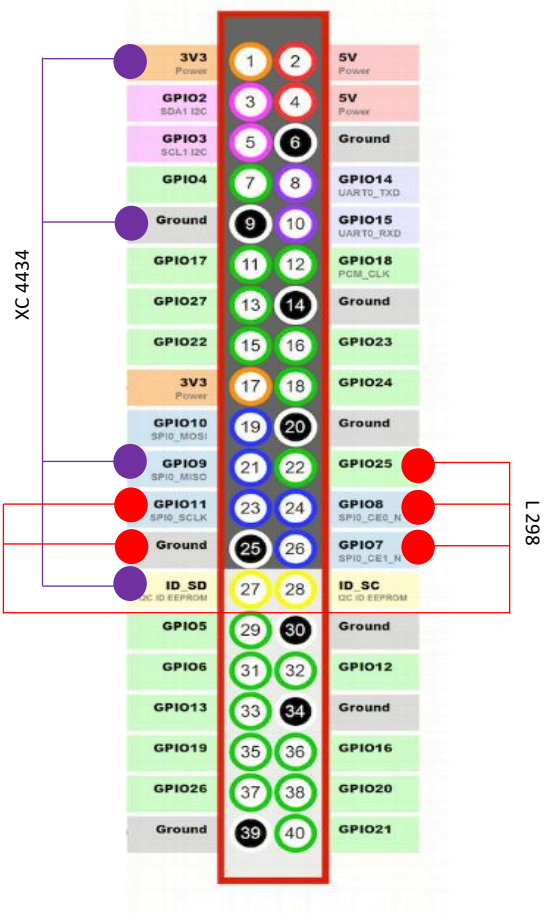


Figure 25 - L298N Motor Controller



L298N – Motor Controller

L298N is powered by 2 x 9V batteries, for each Motor A and B. Following a H-bridge configuration, it allows control over the direction of current within a DC motor. By reversing the direction of the current, it gives the ability for forward and reverse controls.

XC 4434 – Hall Sensor

The XC4434 hall sensor was utilised to detect and count a rotation of the wheel. From a single rotation, we are able to calculate rotation per minute (RPM), which gives us linear speed reading of the car.

Figure 26 - XC 4434 Hall Sensor

5.2.3 WiringPi

WiringPi (WP) is an open source GPIO access library written in C to allow easy access to Broadcom chips. It is used extensively throughout CarController program as it allows for better pin management and GPIO configuration access. Using its GPIO access library, providing a mapping to Broadcom chip (e.g. BCM7), WiringPi allows easy access for software developer to manipulate RPi's 40 GPIO pins. The above sensors are wired and coded into specific pins within the RPi using WP library and numberings. For a complete wiring diagram, please see Figure 24.

5.2.4 Miniaturised car prototype

Below outlines the integration stages of hardware and software components:

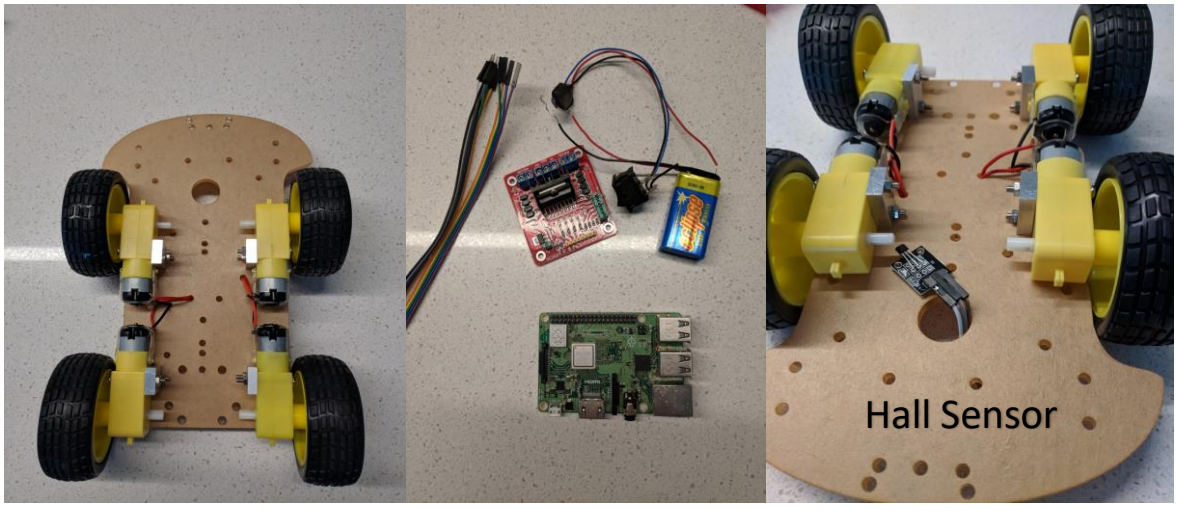
5.2.4.1 Power supply

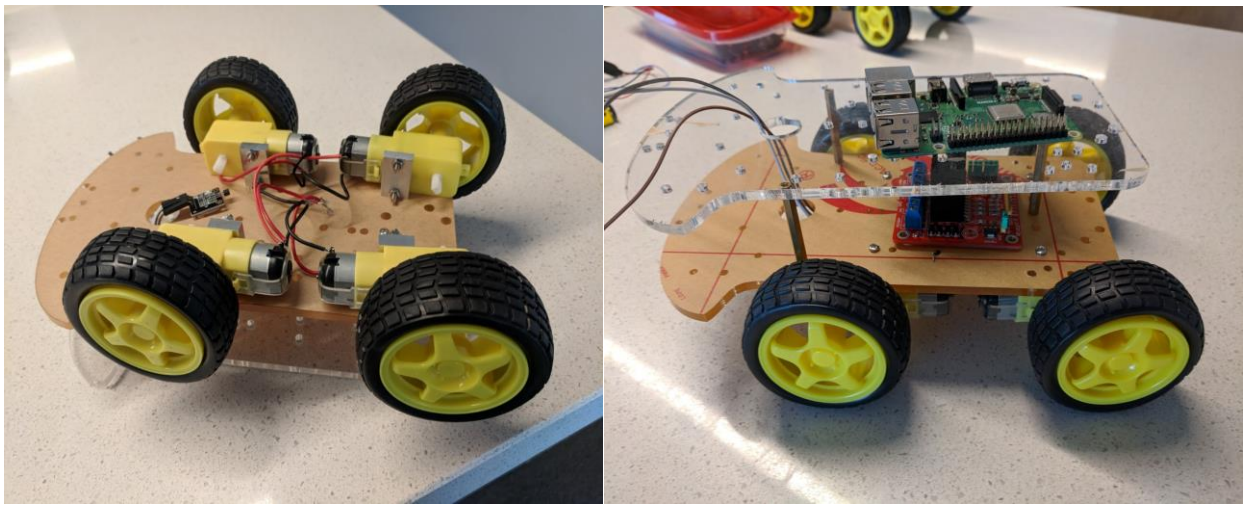
To provide mobility, each car is equipped with the following items for power:

Raspberry Pi Model 3B+:	Power Bank with the following minimum capability:	
	Power source:	DC 5V
	Input voltage:	DC 5.1V
	Input current:	2.0A(TYP)
	Output current:	2.1A(TYP)
	This provides the required +5.1V (2.5Amp) to power the RPi.	
L298 Motor Controller:	2 x 6LR61-9V batteries	
	Each 9V battery powers one side of the car.	

5.2.4.2 Sensor to chassis integration

Sensors described above are installed on the 4WD Motor Chassis. Using a power bank with the above specification, a Raspberry Pi is mounted on top and secured with double-sided tape.





5.2.4.3 Car Control Program

Utilising C++ as the language to integrate hardware and software, this section describes the software architecture and related source code to operate and drive the car using the developed CarControl program.

Software Architecture

Main.cpp is the program's first entry, it will then execute the following:

1. Attempts to connect with the RSU Server. If successful, prints:

```
std::cout << "Server connected... \n";
```

2. Initialise car's wiring setup for sensor and pin control using WiringPi library[14], if **fails** it exits the program and prints:

```
 std::cout << "WiringPi Lever init failed. Exiting now... \n";
```

3. Enters a loop and waits for user input command (Figure 27).

4. Alerts and send BSM message, whenever a car's speed changes from accelerating/decelerating, as well as going forward, it sends a J2735 message using SendBSMSpeedMessage(). This function then calls the SendMessage() to populate BSM Core Data field of speed, DER encodes the MessageFrame to wirelessly send the message.

```
//Population and sending of BSM
static void SendBSMSpeedMessage()
{
        connection.carSpeedReading = s.car_speed_reading;
        connection.SendMessage(connection.PopulateBSM(0));
}
```

5. Awaits for user input 0 to exit loop and close the program.

```
 case 0: //Stop, Break and exit(0)
        car_control_L298::stopCar();
        connection.ClientCloseConnection();
        runningStatus = false;
```

[14] Allows for software integration for manoeuvre controls

802.11AC wireless communication/configuration

Utilising Broadcom's chip and wireless capability, each PI representing a vehicle is connected to the server using a static IP address. To discern the identity of the vehicle, each vehicle is assigned a static IP by the RSU server.

J2735 MSG Encoding/Communication

Utilising an ASN.1 struct and a reference library to compile and generate J2735 libraries. This was used to encapsulate the speed readings, wrapping into BSM, then into a message frame to be transmitted wirelessly using 802.11ac to the RSU server.

External Software Libraries

Using public repositories such as GitHub and PI reference libraries such as WiringPi, these libraries assisted with the development of the toy car.

Figure 27 - Code snippet CarControl main loop

```
while (runningStatus)
{
        int command;

        std::cout << "Enter command: \n";
        std::cin >> command;

        switch (command)
        {
                case 0: //Stop, Break and exit(0)
                        car_control_L298::stopCar();
                        connection.ClientCloseConnection();
                        runningStatus = false;
                        break;
                case 8: //Forward
                        car_control_L298::forwardCar();
                        car_control_L298::carActive = true;
                        break;
                case 5: //Stop
                        car_control_L298::stopCar();
                        car_control_L298::carActive = false;
                        break;
                case 7: //Accelerate
                        car_control_L298::accelerate();
                        break;
                case 9: //Deccelerate
                        car_control_L298::decelerate();
                        break;
                case 4: //Left
                        if (car_control_L298::carActive == true)
                        {
                                car_control_L298::turn_left();
                        }
                        break;
                case 6: //Right
                        if (car_control_L298::carActive == true)
                        {
                                car_control_L298::turn_right();
                        }
                        break;
                case 2: //Reverse
                        car_control_L298::carActive = true;
                        car_control_L298::reverseCar();
                        break;
        }
}
```

Software development of each sensor component is organised and coded with separation from its interface(.h) and implementation(.cpp).

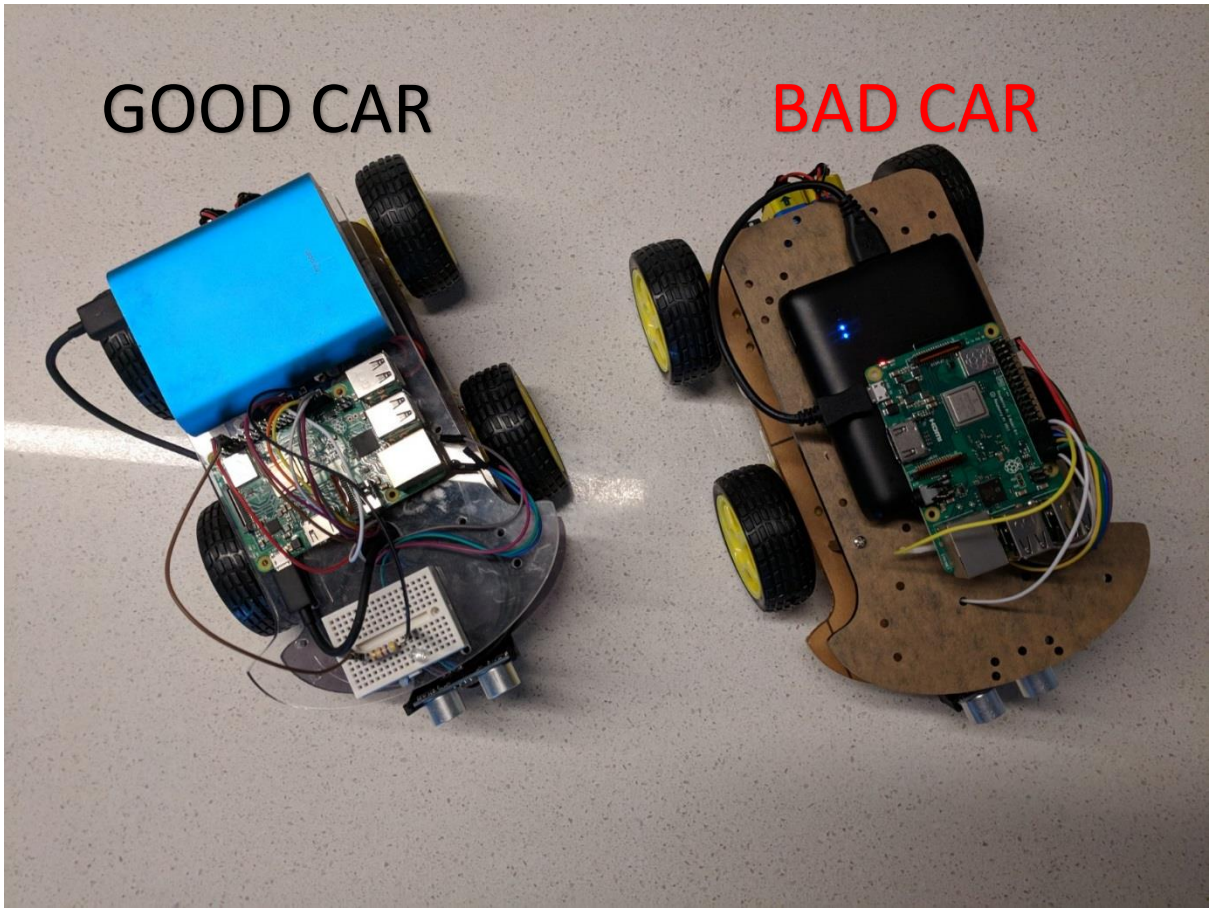
Sensor	Software Implementation
L298N Motor Controller	Description: Manages and controls the car's electrical at the physical layer. Code files: car_control_L298.h car_control_L298.cpp
XC4434 Hall Sensor	Description: Manages and gives power[15] to the led and hall sensor. Code files: Utilities.h Utilities.cpp

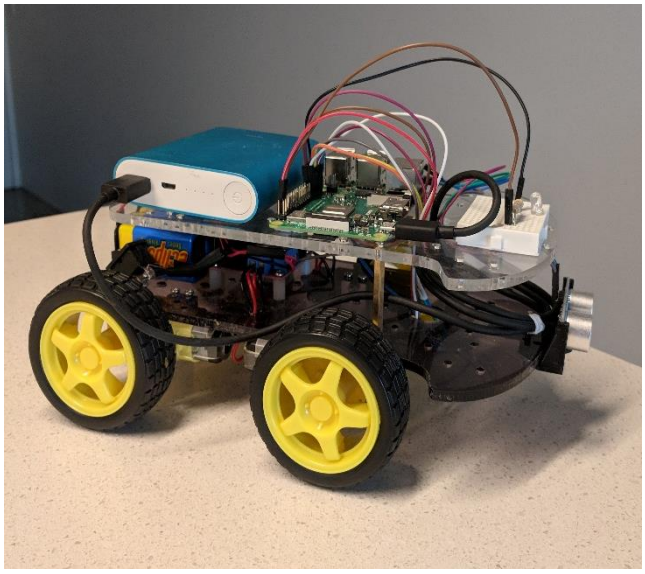
Wireless Connection	Description:
	Manages wireless connection to the RSU server, functions to encode to J2735 message and populate the speed field within BSM Core Data - Part 1.
	
	Code files:
	car_control_connection.h
	car_control_connection.cpp
	
	Key function within this wireless class:
	`int CarConnection::ClientConnect(const char *serverIp)`
	`BasicSafetyMessage_t* CarConnection::PopulateBSM(int messageType)`
	`void CarConnection::SendMessage(BasicSafetyMessage_t *bsm)`

[15] Using one of Raspberry Pi's designated 5V pins

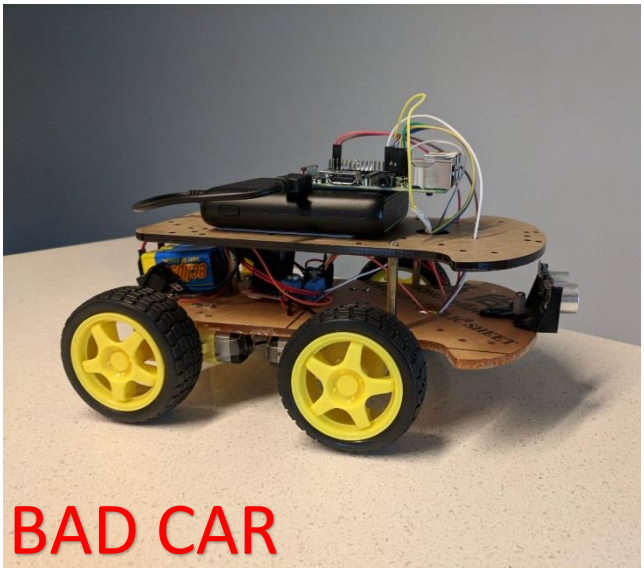
5.2.4 Assembled prototypes

With the wiring diagram in Appendix C, assembly of the sensors to the toy car's chassis, controlled by a CarControl program, two miniaturised car prototypes have been developed.





The transparent car represents a good actor, which its broadcasted speed is read untampered from L298 motor control.



The brown car represents a bad actor, which its broadcasted speed is tampered and manipulating readings from the L298 motor control.

5.2.5 Model Car Testing

Speed readings from L298 is received as integers. Car is controlled using CarControl software. Using pulse-width modulation, to modulate current going through the PI's pin to command acceleration, deceleration and stop. The input on which is fed to control the speed of the car is in the form of an integer. For example, the car initiates and starts at a speed written from software to L298 hardware of x = 500. Raspberry Pi's PWM pin has a maximum integer range of 0-1024. However, this unit of reading does not translate to linear distance.

5.2.6 PWM and Speed conversion

Utilising WiringPi's library to easily access Pi's pins, PWM is used to adjust the average value of the current running through the specified coded pin. This integer and unit of measurement does not translate to linear distance on which the car is traveling and speed field on which will be broadcasted. A conversion table is created to convert speed values written to the L298 car controller in increments of ±100, starting at 500 to be converted to linear distance of cm/sec.

5.2.6.1 Speed Table Conversion Testing

Gear 1: 300 - 399

Distance	Elapsed time	Speed
100 cm	5.0903 seconds	NA, current drawn too small as vehicle was not moving.
100 cm		
100 cm		
100 cm		
100 cm		
	NA	

Gear 2: 400 - 499

Distance	Elapsed time	Speed
100 cm	2.25	44.44
100 cm	2.64	46.88
100 cm	2.50	43.00

100 cm	2.48	42.32
100 cm	2.34	42.74
	Average speed	44.8 cm/seconds

Gear 3: 500 - 599

Distance	Elapsed time	Speed
100 cm	1.43	69.93
100 cm	1.41	70.92
100 cm	1.47	68.03
100 cm	1.41	70.92
100 cm	1.42	70.42
	Average speed	70.04 cm/seconds

Gear 4: 600 - 699

Distance	Elapsed time	Speed
100 cm	1.32	75.76
100 cm	1.29	77.52
100 cm	1.32	75.76
100 cm	1.37	72.99
100 cm	1.31	76.34
	Average speed	75.67 cm/seconds

Gear 5: 700 -799

Distance	Elapsed time	Speed
100 cm	1.33	75.19
100 cm	1.32	75.76
100 cm	1.35	78.07
100 cm	1.38	76.46
100 cm	1.35	78.07
	Average speed	76.71 cm/seconds

5.2.6.2 Good/Bad Actor Vehicle Configuration

Using the table above, speed being passed through start, accelerate and decelerate speed values are converted into the above category for BSM transmission populating speed field. As Gear 4 and 5 does not contain a lot of different in its average speed, speed range from 600-800 is considered to be Gear 4.

To mark whether a vehicle is a bad or good actor, a boolean flag separates whether the car manipulates speed reading before message encapsulation into J2735 message frame. The flag of badActor if set to True marks the vehicle as Bad and manipulates speed reading. For example, if the vehicle is marked as a bad actor, when broadcasting Gear 3 it does not translate its actual reading from the conversion table above of 70, but actually 81. Only Gear 3 and 4 is manipulated as a bad vehicle does not need to manipulate lower gear at lower speed to fool speed cameras.

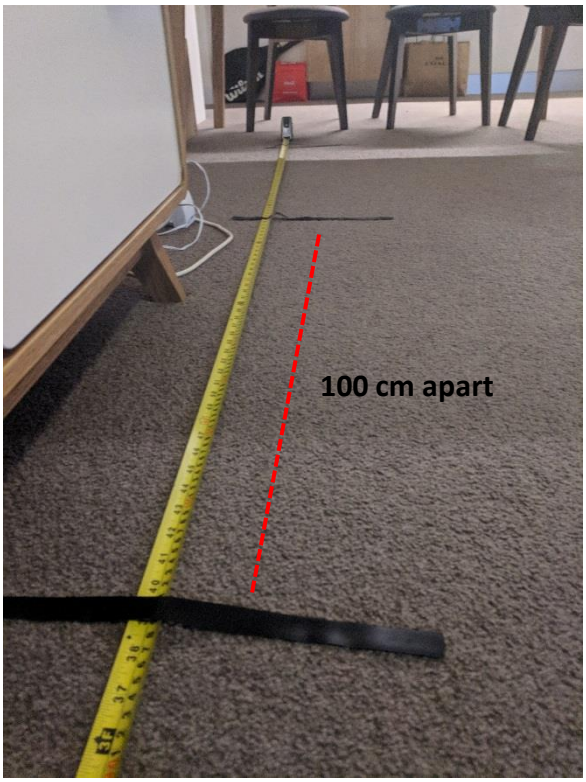
```
BasicSafetyMessage_t* CarConnection::PopulateBSM(int messageType)
{
        //Creates a new Octet String to parse IP.4 as TempID
        OCTET_STRING_t* t = OCTET_STRING_new_fromBuf(&asn_DEF_OCTET_STRING,
                                                addrAssigned, strlen(addrAssigned));

        TemporaryID_t* tempId = t;

        BasicSafetyMessage_t* bsm;
        bsm = (BasicSafetyMessage_t*)calloc(1, sizeof(BasicSafetyMessage_t));
        bsm->coreData.id = *tempId;
        bsm->coreData.msgCnt = 1;

        //Actor profile switch: T:Bad or F:Good
        bool badActor = false;
        int speedTranslationToCM = 0;

        //Below represents the table to map PWM speed to tested linear distance in cm/seconds.
        switch(messageType)
        {
                case 0:
                        //GEAR 1
                        if (verifyCarSpeed.inRange(300, 399, s.car_speed_reading))
                        {
                                speedTranslationToCM = 0;
                        }
                        //GEAR 2
                        else if (verifyCarSpeed.inRange(400, 499, s.car_speed_reading))
                        {
                                speedTranslationToCM = 44;
                        }
                        //GEAR 3
                        else if (verifyCarSpeed.inRange(500, 599, s.car_speed_reading))
                        {
                                if (badActor)
                                {
                                        speedTranslationToCM = 60;
                                }
                                else
                                {
                                        speedTranslationToCM = 70;
                                }
                        }
                        //GEAR 4
                        else if (verifyCarSpeed.inRange(600, 800, s.car_speed_reading))
                        {
                                if(badActor)
                                {
                                        speedTranslationToCM = 60;
                                }
                                else
                                {
                                        speedTranslationToCM = 75;
                                }
                        }

                        bsm->coreData.speed = speedTranslationToCM;
                        break;
                case 1:
                        bsm->coreData.heading = 100;
                        break;
                case 2: //GPS Coordinates of UQ
                        bsm->coreData.lat = 27.4975;
                case 3:
                        bsm->coreData.Long = 153.0137;
        }

        return bsm;
}
```

5.2.7 Testing

The above section provides a speed conversion table from an integer value modulating current to PI's pin to control car's velocity. Each value in the form of an integer is then converted into linear distance in centimetre per seconds using the table above. The table above (Section 5.2.6) has been populated using manual testing with a stopwatch and measured distance. Five readings is simulated per each gear level and an average is taken to be the converted linear distance.

Manual Testing

Using a measurement taped, and a black electrical tape to mark distance of 100cm ~ 1m, different speeds starting from 300 going up to 700 was tested. Each speed category, correlating to gear levels was tested with 5 runs each, the readings are then averaged to represent the average speed and distance travelled under that gear.



Manual testing is done to confidently cross compare readings taken from Hall Sensor. The hall sensor component (Section 5.2.2) gives rotation per minutes which is then used to convert into linear distance. Through simulating Gear 3, running at 500 for one minute, the hall sensor counts how many rotations it detects through a magnet attached within its axel. However, the table above provides more confident readings than the hall sensor and attached magnet.

Findings

Through the test conducted, what was found was that Gear 1 (300) in a carpeted surface was too slow for the car to move. At Gear 1, the car is not moving and testing showed that there was not much different in the linear distance speed for Gear4 and Gear 5. A difference of 1cm/sec was too small to be considered and tested.

Through testing, it showed that only three gears representing 3 levels of speed need to be considered.

Gear 2: 400-499 – broadcasting 41.8 cm/sec

Gear 3: 500-599 – broadcasting 70.04 cm/sec

Gear 4: 600-799 – broadcasting 75.67 cm/sec

This translated speed conversion table is translated into code and into the CarControl program. Every time a user accelerates or decelerates, it cross-references its current speed being feed into L298 and convert an integer into the table's reading for BSM transmission containing speed readings from the car.

5.3 RSU Server Development

This section outlines the development for Road side Unit that connects the 4 SPDT micro switch with lever to a verification program, calculating to compare independent speed reading with received J2735 broadcasted BSM containing speed reading. The development of the RSU server is categorised into two sections: Hardware and Software.

5.3.1 Hardware Component

Physical Road

4 medium-density fibreboards (MDF) with the following dimensions:

W: 45cm
L: 150cm

The length of the board is further divided to the following requirements:

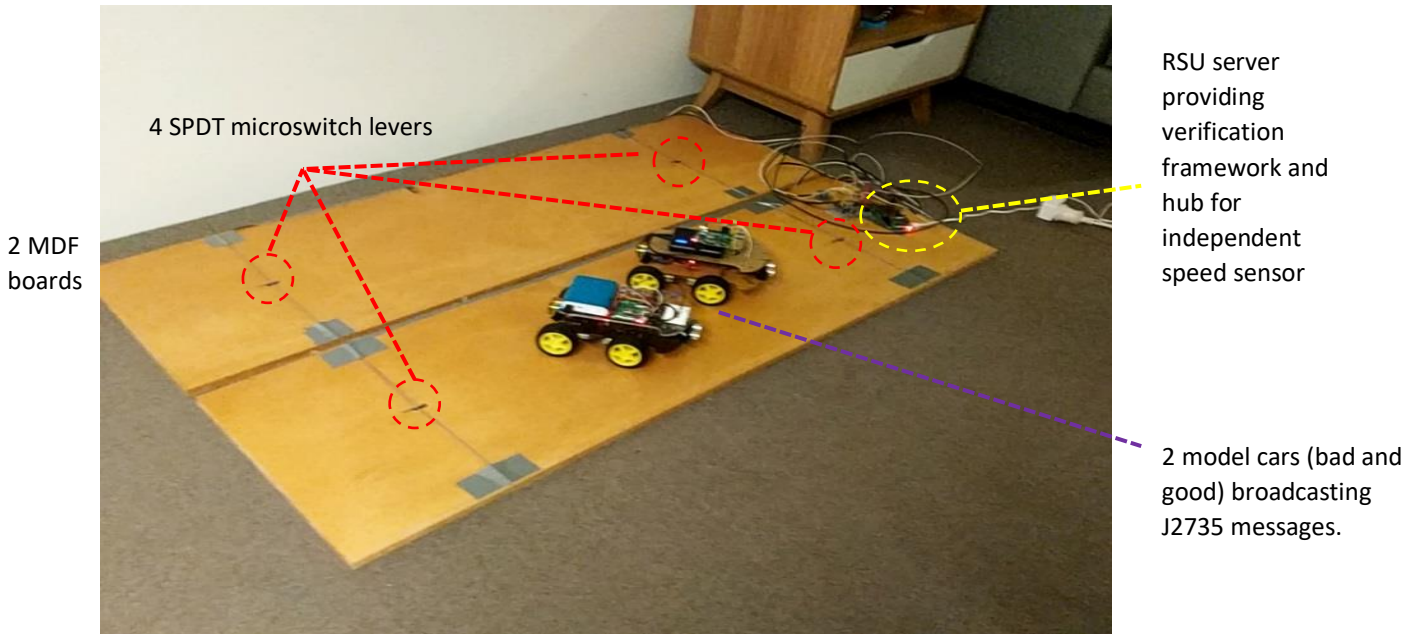
- 100 cm for fixed distance of car to travel
- 25 cm for car start position
- 25 cm for car stop position
- Each start and stop requires a SPDT micro switch lever sensor to be installed
- A trip wire to assist with lever triggering

This yields 2 independent roads with 2 sets of speed readings from its installed lever. The lever returns a boolean upon an event click, triggering a stopwatch to start and wait for the next interrupt from its corresponding stop lever click. These behaviours and procedures are further described in the below software components.

Raspberry PI

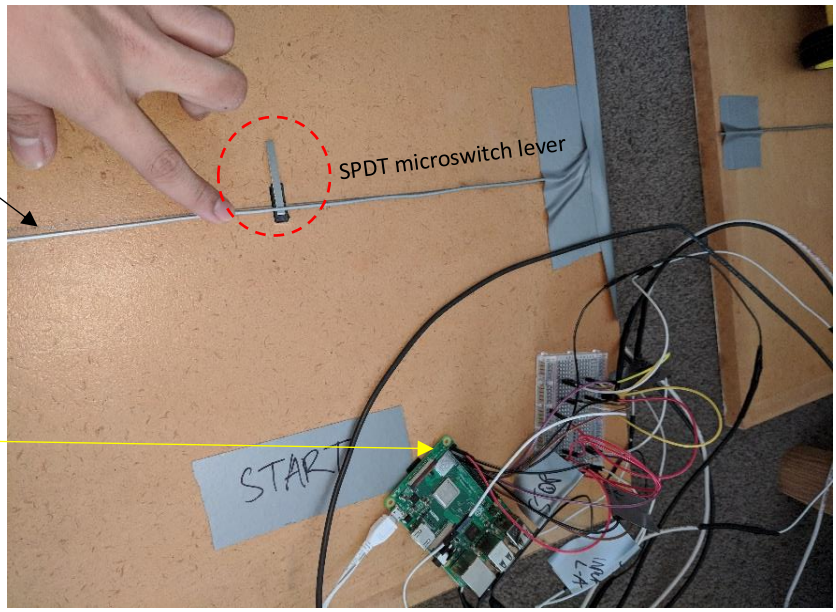
Utilising the PI and the OS installed as described in Section 5.2, a software developed in C++ manages the 4-lever sensor, receive/transmit J2735 messages as well as a verification framework.

5.3.1.1 Road development



5.3.1.2 Lever sensor wiring

Trip wire
A trip wire placed on top of the lever. This allow easy triggering of the lever and in whichever direction



RSU Server
4 microswitch levers are connected to the pins of the RaspberryPi

With the above four levers installed, it is connected to single PI to connect to its verification framework. This translate to two independent roads, capable of verifying to different car velocities. Within the PI is the developed verification framework allowing to cross compare sensor readings with BSM speed content.

5.3.2 RSU Software Architecture

All of the below components are wrapped and packaged into a single execution file (.out) and ran inside the RaspberryPi. Main.cpp is the orchestrator of the solution, this will be described in further details below as it contains the verification framework.

RSU solution contains three main files and corresponding header file:

- Main.cpp
- RSU_Server_Lever.cpp/.h
- SocketConnection.cpp/.h

The solution compiles into a single execution file (.out). Program enters in main.cpp and within it spawns the main process and 3 child threads from its execution:

```
std::thread openMultipleConnection;
std::thread processQueueMessage;
std::thread ActivateSensor;
```

Main.cpp execution sequence:

1) Open Socket Connection
2) Process Received J2735 Message
3) Initialise Lever Sensor
4) Speed Verification Framework
5) Decode J2735 Message from Pipe

5.3.2.1 Open Socket Connection

The purpose of this thread is to open multiple client socket to connect and send for the decode of J2735 messages.

```
openMultipleConnection = std::thread{ SocketConnection::StartServer };
```

- Starts a thread to host 30 connections coming in.
- Clients are not spawned into new child-thread, using fd_select.
- Closing of server thread is invoked here, after detach().

Client & Host Connection

In this software the RSU awaits connection on the designated port (8888). During which, the two vehicles (representing good and bad actors) connect and start transmitting J2735 messages. An important requirement is the ability to separate multiple established connection and maintaining independent communication streams.

Using the static void function StartServer(), it binds an IP and a port, giving a socket, and listen on the publicly broadcasted socket. Server is able to handle up to 30 client connections and if bind or initialisation setup fails, it throws the following:

```
//bind the socket to localhost port 8888
if (bind(master_socket, (struct sockaddr *)&address, sizeof(address)) < 0)
{
        perror("bind failed");
        exit(EXIT_FAILURE);
}
printf("RSUServer is listening on port %d \n", PORT);
```

Upon successful setup, server will enter a thread executing a while loop waiting for client connections.

5.3.2.2 Process Received J2735 Message

With each car tagged as a bad or good actor, the RSU server knows indefinitely from which vehicle is the speed reading being decoded and which message type received. For the purpose of this demo, the RSU server is listening for MessageID: 20 – Basic Safety Message type.

```
void ProcessQueueMessages()
{
        char msg[100];
        char msgReceivedFrom[100];

        while (true)
        {
                // Blocking call to wait for message
                auto item = MsgQueue.pop();
                memset(msg, 0x32, 100);

                // Process message from queue -> prints
                switch (item->messageId)
                {
                        case 20:
                        {
                                BasicSafetyMessage_t* bsm = &item->value.choice.BasicSafetyMessage;

                                //TemporaryID containing IP conversion to string
                                std::string keyIP(bsm->coreData.id.buf, bsm->coreData.id.buf +
                                                                        bsm->coreData.id.size);

                                storeReceivedMessage[keyIP] = bsm->coreData.speed;

                                //Message print: Speed received from IP
```

```
                                sprintf(msgReceivedFrom, "Message received from: %s ...\n",
                                                                        bsm->coreData.id);
                                printf(msgReceivedFrom);
                                sprintf(msg, "Transmitted speed reading of: %ld\n",
                                                                        bsm->coreData.speed);
                                printf(msg);




                                //A - CarOne GOOD
                                if (keyIP == "192.168.43.212")
                                {
                                        speedCar_A = bsm->coreData.speed;
                                }

                                //B - CarTwo BAD
                                else if (keyIP == "192.168.43.52")
                                {
                                        speedCar_B = bsm->coreData.speed;
                                }

                                break;
                        }
                }

                // Free the MessageFrame once we've processed it.
                // to think how to free when you compare and plug in to equation.
                ASN_STRUCT_FREE(asn_DEF_MessageFrame, item);
                memset(msg, 0, 100);
        }
        return;
}
```

5.3.2.3 Initialise Lever Sensor

This section both initialise the electrical wiring setup from lever to the pins assigned, and assigns lever trigger to corresponding function.

- Initialise lever sensor and setting pins to INPUT
- Utilising WiringPi's library, to configure each lever sensor to be an INT_EDGE_FALLING interrupt signal to trigger a corresponding stopwatch function.

```
/* Initialise lever sensor and setting pins to INPUT */
if (Lever_Switch::InitWiringPi() == -1)
{
        std::cout << "WiringPi Lever init failed. \n";
        exit(-1);
}

else
{
        Lever_Switch::SetLeverPin();

        //LEVER A - CarOne_Bad
        wiringPiISR(LeverAStart, INT_EDGE_FALLING, &Lever_Switch::StartStopWatch);
        wiringPiISR(LeverAStop, INT_EDGE_FALLING, &Lever_Switch::StopStopWatch);

        //LEVER B - CarTwo_Good
        wiringPiISR(LeverBStart, INT_EDGE_FALLING, &Lever_Switch::StartStopWatch_LeverB);
        wiringPiISR(LeverBStop, INT_EDGE_FALLING, &Lever_Switch::StopStopWatch_LeverB);
}
```

Lever switch

Through assigning pins and function to represent a starting watch or stop, the below shows an example of Road A's lever functions of start and stop:

```
//LEVER A
void Lever_Switch::StartStopWatch()
{
        printf("Lever A – START stopwatch \n");
        printf("Lever A start triggered by: 192.168.43.56 \n");
        auto startWatch = std::chrono::high_resolution_clock::now();
        startLever_A_Watch = startWatch;

        //Straight system call to set pin's edge to none -> disable
        std::string strCommand = "gpio edge 2 none";
        const char *command_LeverA_Start = strCommand.c_str();
        std::system(command_LeverA_Start);
        return;
}

void Lever_Switch::StopStopWatch()
{
        printf("Lever A – STOP stopwatch \n");
        printf("Lever A stop triggered by: 192.168.43.52 \n");
        auto stopWatch = std::chrono::high_resolution_clock::now();

        //Calculate elapsed in seconds
        std::chrono::duration<double> elapsedTime = stopWatch - startLever_A_Watch;
        //Prints fixed-notation to 2 decimal place
        std::cout.precision(2);
        std::cout << "Lever A stop: " << elapsedTime.count() << " s\n" << std::fixed;

        //Straight system call to set pin's edge to none -> disable
        std::string strCommand = "gpio edge 0 none";
        const char *command_LeverA_Stop = strCommand.c_str();
        std::system(command_LeverA_Stop);

        //To store elapsed time count for speed calculation
        lever_A_STOP_trigged = true;
        elapsedTime_LeverA = elapsedTime.count();
}
```

Floating triggers

Floating triggers represent the off state on which the device takes the time back to reset. The red box highlighted above shows how this is circumvented. Through making a system call straight into Linux kernel, it is able to reset the state of the pin's edge to be disabled, resetting to default.

5.3.2.4 Speed Verification Framework

This framework first starts a thread with a function ActivateSpeedSensor() to

- Listen & wait for sensors lever click
- Start/Stop stopwatch
- Calculate AverageSpeed in cm/s
- Enters a loop to wait for an flag symbolizing which stop lever has been triggered

```
        ActivateSensor = std::thread{ ActivateSpeedSensor };
```

ActivateSpeedSensor is a function executing on an independent thread. Within its loop, it continuously runs to enter two IF statements to symbolize which lever has been triggered. For example, if lever A has its stop lever triggered, it will set a global flag of lever_A_STOP_trigged to be true. This will allow the program to enter the if statement and execute the verification and comparing speed readings within a BSM.

```
void ActivateSpeedSensor()
{
        VerifySpeed vs;
        while (!flag)
        {
                //A is good
                if (lever_A_STOP_trigged)
                {
                        double avgSpeedReadingLeverA = vs.CalculateSpeed(elapsedTime_LeverA);
                        std::cout << "Average speed A: " << avgSpeedReadingLeverA << " cm/s" << endl;

                        //Lever speed 20 (16-24). car speed 12
                        double min = vs.Calculate_LeverReading_DevMin(avgSpeedReadingLeverA);
                        double max = vs.Calculate_LeverReading_DevMax(avgSpeedReadingLeverA);

                        std::cout.precision(2);
                        std::cout << "Range (-10%) Lev_A minimum: " << min << "\n";
                        std::cout << "Range (+10%) Lev_A maximum: " << max << "\n";

                        //Verification Check that it falls within this range
                        if (isSpeedinRange(min, max, speedCar_A))
                        {
                                printf("Speed of CarA: %d cm/sec is within normal range. \n",
                                                                                speedCar_A);
                                printf("CarOne speed VERIFIFED! \n");
                        }
                        else if (speedCar_A == 0)
                        {
                                printf("No speed reading received from CarOne... \n");
                        }
                        else
                        {
                                printf("CarOne speed is MISMATCH with speed sensor. \n");
                                printf("Where broadcasted BSM-Speed of vehicle: %dcm/s \n",
                                                                                speedCar_A);
                                printf("Tagging and initiating bad actor protocol... \n");
                        }

                        lever_A_STOP_trigged = false;
                }
                //B is BAD
                else if (lever_B_STOP_trigged)
                {
                        double avgSpeedReadingLeverB = vs.CalculateSpeed(elapsedTime_LeverB);
                        std::cout << "Average speed B: " << avgSpeedReadingLeverB << " cm/s" << endl;

                        double min = vs.Calculate_LeverReading_DevMin(avgSpeedReadingLeverB);
                        double max = vs.Calculate_LeverReading_DevMax(avgSpeedReadingLeverB);

                        std::cout.precision(2);
                        std::cout << "Range (-10%) Lev_B minimum: " << min << "\n";
                        std::cout << "Range (+10%) Lev_B maximum: " << max << "\n";

                        //Verification Check that it falls within this range
                        if (isSpeedinRange(min, max, speedCar_B))
                        {
                                printf("Speed of CarB: %d cm/sec is within normal range. \n",
                                                                                speedCar_B);
                                printf("CarTwo speed VERIFIFED! \n");
                        }
                        else if (speedCar_B == 0)
                        {
                                printf("No speed reading received from CarTwo... \n");
                        }
                        else
                        {
                                printf("CarB speed is MISMATCH with speed sensor. \n");
                                printf("Broadcasted BSM-Speed of vehicle: %dcm/s \n",
                                                                                speedCar_B);
                                printf("Tagging and initiating bad actor protocol... \n");
                        }

                        lever_B_STOP_trigged = false;
```

```
                }
        }
}
```

Reset Flag

After every speed calculation and verification of speedreadings from received J2735 messages to its calculated speed from sensor reading, it resets each lever's trigger flag to false:

```
lever_A_STOP_trigged = false;
```

Stopwatch

After wiring and initial testing of the lever, an internal stopwatch function is created in the RSU program to pair 2 levers to mark start and stop. Four lever sensors yield two independent stopwatch levers installed at 2 MDF boards, representing two separate roads.

DTS Formula

Utilising the same formula in Section 5.3.2.4, the roads depicted above have a fixed distance between the levers (overlayed by a tripwire), to be 1.5 meter. With a fixed distance and speed given through the levers installed on each road, an average cm/second speed reading can be given from the above simulation.

```
//Speed formulae where Speed = Distance / Time
double VerifySpeed::CalculateSpeed(double time)
{
        double calculatedSpeed = vs.roadDistance / time;
        return calculatedSpeed;
}
```

Speed readings

There are two speed readings which are taken into account:

- Speed readings received from transmitted BSM message from vehicle (J2735)
- Speed readings from installed lever sensor, representing an independent speed verification sensor from the RSU

Speed readings taken from each vehicle (1), the last message received readings is stored to be compared with readings taken from independent lever (2).

Verify & Compare

Whenever a speed reading from its independent lever sensor has been triggered, it calculates and store that speed reading. The following procedure verifies a vehicle's transmitted BSM speed reading and cross compare speed with readings from its independent sensor:

1. Start lever is interrupted and triggers StartStopWatch() as a vehicle passes and trips the wire

2. Stop lever is interrupted to trigger StopStopWatch() as vehicle passes the stop line marking the end of 100cm.

3. Upon StopStopWatch gets called, it calculates the elapsed time by subtracting the current with the time taken in StartStopWatch. The section 'Lever switch' above contains the code snippet referencing the below.

```
        //Calculate elapsed in seconds
        std::chrono::duration<double> elapsedTime = stopWatch - startLever_A_Watch;
```

It stores elapseedTime_LeverA as a global variable for Main.cpp to access

4. Sets global flag of lever_A_STOP_trigged = TRUE. Publishing an event notifiying ActivateSPeedSensor function that an time has been received and to cross reference readings.

5. Highlighted in the blue box above, the setting of the global flag allows the program to enter the if statement of lever_A to calculate and verify received speed with lever A's calculated speed.

6. The outputted average speed reading from lever is then used to calculate the range (min,max) of the lever's speed reading. This is done to account for 10% standard deviation for lever readings.

```
double VerifySpeed::Calculate_LeverReading_DevMin(double AvgSpeedLeverReading)
{
        double calculateDevFromReading = AvgSpeedLeverReading * (stdDev);
        return AvgSpeedLeverReading - calculateDevFromReading;
}

double VerifySpeed::Calculate_LeverReading_DevMax(double AvgSpeedLeverReading)
{
        double calculateDevFromReading = AvgSpeedLeverReading * (stdDev);
        return AvgSpeedLeverReading + calculateDevFromReading;
}

//Lever speed 20 (16-24)
double min = vs.Calculate_LeverReading_DevMin(avgSpeedReadingLeverA);
double max = vs.Calculate_LeverReading_DevMax(avgSpeedReadingLeverA);

std::cout.precision(2);
std::cout << "Range (-10%) Lev_A minimum: " << min << "\n";
std::cout << "Range (+10%) Lev_A maximum: " << max << "\n";

}
```

7. The outputted average speed reading is then used to calculate the range (min, max) of the lever's speed reading. This is done to account for 10% standard deviation upon readings.

8. At this stage we have the following parameters stored within or memory

   Elapsed time from lever A's stopwatch sensor:
   - Calculated speed for Road A from its own lever sensor
   - Accepted range of (min,max) speed calculated from elapsed time from lever
   - Last broadcasted speed message of the vehicle travelling on Road A

   With the above information, this is fed into the isSpeedinRange function to output a boolean:

```
bool isSpeedinRange(unsigned rangeLow, unsigned rangeHigh, unsigned variable)
{
        return ((variable - rangeLow) <= (rangeHigh - rangeLow));
}
```

   This function checks whether the last broadcasted speed from the vehicle is within the minimum or maximum boundary. If it is not, then the vehicle is broadcasting false speed.

5.3.2.5 Decode J2735 Message from Pipe

Pipe processed message

This pipe allows message transfer and decapsulation from TCP/IP, decoding J2735 message frame to push the message onto a queue for further processing.

```
/*
        Read the pipe so we can process in-coming messages from the server
        the data is a collection of encoded MessageFrames as bytes.
*/

stack<MessageFrame_t*> mystack;

void SocketConnection::ReadPipeToProcessMessage(int pipe)
{
        uint8_t* rcvPipeBuffer = (uint8_t*)calloc(1, 1024);

        //Blocking call to wait and read pipe's message
        int bytesRead = read(pipe, rcvPipeBuffer, 256);
        if (bytesRead == -1) {
                return;
        }
        MessageFrame_t* msgFrame = 0;
        asn_dec_rval_t rslt = asn_decode(0, ATS_DER, &asn_DEF_MessageFrame, (void**)&msgFrame,
                                                                rcvPipeBuffer, bytesRead);
        if (rslt.code == RC_OK) {
                MsgQueue.push(msgFrame);
                mystack.push(msgFrame);
        }

        memset(rcvPipeBuffer, 0, 256);
}
```

This void function is the only function that is run within the main while loop of Main.cpp. This architecture is designed this way to allow for memory management from the number of threads being invoked, easier debug and exit clean-up for the program. Essentially, within int Main() only contains 1 main while loop and its task is to wait and listen for messages to be put on a pipe, for it to push onto a queue for J2735 decode into message frame.

5.4 Results and Discussions

There are numerous implementation SCMS simulating a full BSM wrapper and issuance of public certificate from a certificate authority. However, the above simulation highlights that inheritance of trust from digital certificate should not transfer into blindly consuming message content. This research highlights the need to consider trust beyond only the scope of a digital identity, but to surpass digital signature and add a layer of message verifier.

Upon further testing for speed sensor, it was evidently clear that there is a gap with the current verification framework which considers only 1 parameter. Mismatching speed readings taken from a bad sensor reading does not equal bad actor. There would be further verification needed in order to determine a vehicle is a bad actor, not solely relying on one field such as speed.

6.0 Conclusions & Recommendations

6.1 Conclusion

In a heterogenous network environment, where information sharing is the currency driving vehicular safety application; interoperable vehicular communication is not the only key in the realisation and adoption of V2X, but also a robust, extended security framework to verify message content is necessary. This thesis evaluates the security architecture and applicability of V-PKI proposed by the NHTSA through FMVSS No. 150. Through its evaluation and investigation, it was found that security is heavily focused on securing message authentication through digital signature to prove message identity. Inherited trust hierarchy promotes blind consumption of message content due to confidence from digital signature certified by the CA. A verification framework has been developed to propose mitigate and pivot the perspective that the authenticity of a message's identity should not inherit trust to blindly consume message content.

Realising the numerous safety applications and problems V2X tackle for traffic management, Section 2 discusses how NHTSA saw the need to standardize and proposed a rulemaking to mandate the production of light vehicles to be equipped with DSRC-based communication technology. In establishing FMVSS No. 150, NHTSA realised that government intervention is necessary to create an information environment enforcing compliance and provide a developmental stage in achieving widespread development and adoption of vehicular communication technology. NHTSA estimate phase-in schedule to begin by 2021, with 50% of new light-vehicles within public road meeting proposed standard and phase-in to have be completed by 2023 with all cars equipped with vehicular technology. FMVSS 150 is currently now advancing to the Federal Register to seek public review and comments (RFC).

In section 3, NHTSA realised the complexity of VANET having to secure message authentication through a Security Credential Management System (SCMS) – utilising V-PKI. Its proposal attempted to uphold V2V's core philosophy of anonymity through rotating one of the twenty assigned vehicle certificates every five minutes. Unfortunately, the proposed message wrapper structure for a BSM highlights that message authentication can confidently secure and guarantee message content. Its belief that a digital signature, issued from a CA, certifies and inherently trusting BSM content. Inherited trust hierarchy from digital signature should not translate down to message content, and misbehaviour reporting should be extended to other participants within WAVE, such as RSU.

Section 4 evaluates further the applicability and feasibility of the proposed Security Credential Management System utilising Vehicular Public Key Infrastructure (V-PKI) for V2V. It was found that the Misbehaviour Authority (MA) within SCMS is the component responsible for the generation and broadcast of CRL from misbehaving reports it receives from vehicles. Reports to MA needs to be extended to not only source reports from vehicles, but also from RSU. FMVSS 150 further states that the generation and confirmation of misbehaviour reports should be generated within two seconds by the OBU. This is not feasible, as it fails to address the former: frequency of CRL distribution have not been addressed by this proposal and maintenance of certificate (generating/discarding 20 certificates every week for 1 vehicle) is a costly and hardware intensive process. Not to mention the distribution of Certificate Revocation List (CRL) is a network traffic intensive task forcing vehicles to download gigabytes of data frequently.

The proposed verification solution highlights and attempts pivot the trust perspective within vehicular communication. To not rely on inherited trust hierarchy assured through message authentication, blindly trusting and consuming message content. Message authentication is important, but digital signature cannot be mistaken to guarantee untampered message content. A few points to mention in

relation to the developed verification solution, it does not address the gaps of V-PKI as well as proposed a replacement system. It still utilises the same architecture of a PKI where it is still centralised but proposes the idea that a road side unit can be paired with an existing road side sensor (such as a speed camera). In turn allows further anomaly detection algorithm to be developed together with the various existing road sensors within current road infrastructure.

6.2 Recommendations for future works

In vehicular communication, accuracy of information not only is imperative, but update frequency and consumption of a Basic Safety Message determines the different between engaging a prevention or mitigation protocol. With the investigation and proposed development above, there are several developmental and informational gaps that can be continued for future works. Recommendations for these future works are as follows:

(1) Misbehaviour Authority can receive report from RSU

Reporting ability to misbehaviour authority should be extended to RSU. Not only does waiting for report to be generated and confirmed by surrounding vehicle actor to be sent to MA, it takes too long for it to promote preventative measure for safety application. The ability to report to a MA is dependent on the connectivity of the vehicle to the internet (e.g. vehicle entering a tunnel has limited connectivity to the outside world).

(2) Improved RSU anomaly detection algorithm and network

One key factor discovered through developing the prototype is that bad sensor readings does not equate to a bad vehicle. Like the consensus network in blockchain, each RSU and anomaly algorithm would need to be able to reach a consensus to report misbehaviour of a vehicle. It requires a wide consensus agreement that readings from independent sensor is mismatching from transmitted BSM content and through RSU's internal mesh network, connected and controlled under SCMS network, unanimously generate a misbehaving report directly to MA. A road side unit can be further extended to be a silent listener, sieving through millions of transmitted vehicular communication messages, inspecting BSM and verifying content against its paired sensor, connected to a network of RSU continuously validating and giving real time rod traffic situational awareness.

(3) Pivoting from a centralised SCMS

Security Credential Management System based on Vehicular Public Key Infrastructure is too risky and prone to attack, making it a single source of failure. Being a centralised repository for certificate authority, authenticating and verification of digital identity, the management and distribution of certificates at this scale is far too costly and inefficient. With 20 certificates issue per week for each vehicle, having 270.4 million registered light-vehicle, the SCMS proposed by the NHTSA is looking at a scale of around 5.6 billion certificates to manage and distribute weekly.

By allowing RSU to be paired together with existing road side sensor within current road infrastructure, it could function as a further extender for better CRL propagation and reach. It could further extend the SCMS reach and also be a faster report escalation point. Utilising its paired independent sensor, it is able to utilise its own anomaly detection algorithm to inspect BSM content and police vehicular environment for bad actors. This is a future recommendation that verification framework inspecting BSM content be implemented in a real-world system.

7.0 Appendix

Appendix A – C-ITS in Japan, US, EU

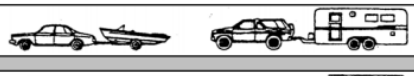
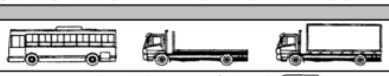
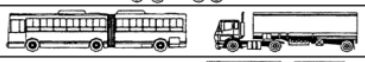
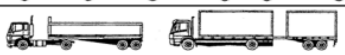
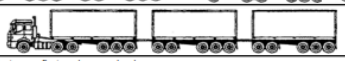
[34]

	JAPAN	US	EU
STANDARD / COMMITTEE	ITS-FORUM	IEEE802.11P/1609.X	CEN/ETSI EN 302 663
FREQUENCY RANGE	755 ~ 765 MHZ	5850 ~ 5925 MHZ	5855 ~ 5925 MHZ
NO. OF CHANNELS	ONE 10 MHZ	SEVEN 10 MHZ (TWO 20 MHZ BY COMBINING 10 MHZ)	SEVEN 10 MHZ
MODULATION	OFDM		
OFDM SUBCARRIERS	3 ~ 18 MBIT/S	3 ~ 27 MBIT/S	3 ~ 27 MBIT/S
OUTPUT POWER	20 DBM (ANTENNA INPUT)	23 ~ 33 DBM (EIRP)	23 ~ 33 DBM (EIRP)
COMMUNICATION	ONE DIRECTION MULTICASTING SERVICE (BROADCAST W/O ACK)	ONE DIRECTION MULTICASTING SERVICE, ONE TO MULTI COMMUNICATION, SIMPLEX COMMUNICATION (BROADCAST W/O ACK, MULTICAST, UNICAST W/ ACK)	
UPPER POROTOCOL	ARIB STD-T109	WAVE (IEEE 1609) / TCP/IP	ETSI EN 302 665 (INCL. GEONETWORKING, …) TCP/UDP/IP

SOURCE - INTELLIGENT TRANSPORTATION SYSTEMS USING IEEE 802.11P, ROHDE & SCHWARZ

Appendix B – AUSTROADS Vehicle Classification System

AUSTROADS Vehicle Classification System

Level 1 Length (indicative) Type	Level 2 Axles and Axle Groups Axles	Groups	Level 3 Vehicle Type Typical Description	AUSTROADS Classification Class	Parameters	Typical Configuration
			LIGHT VEHICLES			
Short up to 5.5m		1 or 2	**Short** Sedan, Wagon, 4WD, Utility, Light Van, Bicycle, Motorcycle, etc	1	d(1) ≤ 3.2m and axles = 2	
	3, 4 or 5	3	**Short - Towing** Trailer, Caravan, Boat, etc	2	groups = 3 d(1) ≥ 2.1m, d(1) ≤ 3.2m, d(2) ≥ 2.1m and axles = 3, 4 or 5	
			HEAVY VEHICLES			
Medium 5.5m to 14.5m	2	2	**Two Axle Truck or Bus**	3	d(1) > 3.2m and axles = 2	
	3	2	**Three Axle Truck or Bus**	4	axles = 3 and groups = 2	
	> 3	2	**Four Axle Truck**	5	axles > 3 and groups = 2	
Long 11.5m to 19.0m	3	3	**Three Axle Articulated** Three axle articulated vehicle, or Rigid vehicle and trailer	6	d(1) > 3.2m, axles = 3 and groups = 3	
	4	> 2	**Four Axle Articulated** Four axle articulated vehicle, or Rigid vehicle and trailer	7	d(2) < 2.1m or d(1) < 2.1m or d(1) > 3.2m axles = 4 and groups > 2	
	5	> 2	**Five Axle Articulated** Five axle articulated vehicle, or Rigid vehicle and trailer	8	d(2) < 2.1m or d(1) < 2.1m or d(1) > 3.2m axles = 5 and groups > 2	
	≥ 6	> 2	**Six Axle Articulated** Six axle articulated vehicle, or Rigid vehicle and trailer	9	axles = 6 and groups > 2 or axles > 6 and groups = 3	
Medium Combination 17.5m to 36.5m	> 6	4	**B Double** B Double, or Heavy truck and trailer	10	groups = 4 and axles > 6	
	> 6	5 or 6	**Double Road Train** Double road train, or Medium articulated vehicle and one dog trailer (M.A.D.)	11	groups = 5 or 6 and axles > 6	
Large Combination Over 33.0m	> 6	> 6	**Triple Road Train** Triple road train, or Heavy truck and three trailers	12	groups > 6 and axles > 6	

Definitions: Group: Axle group, where adjacent axles are less than 2.1m apart
Groups: Number of axle groups
Axles: Number of axles (maximum axle spacing of 10.0m)

d(1): Distance between first and second axle
d(2): Distance between second and third axle

[26]

Appendix C – ATAP Vehicle Classification in Australia

Table 40 Vehicle classifications in Australia

From Austroads vehicle classification scheme (circa 1990's) (Austroads 1994, 2002 & 2013b)	From ARRB report RC2062 (Thoresen & Ronald 2002) for Austroads. Consistent vehicle types adopted for use in HDM-4 in Australia	From Austroads report AP-R264-05 (Austroads 2005a)				HDM-4 vehicles (8) adopted for 2014 update to AP-R264-05 Tables 6 & 7 (Austroads 2005a)
		From Appendix A – Vehicle nomenclature			NIMPAC vehicle types (8) from Austroads AP-R264-05 Tables 6 & 7 (Austroads 2005a)	
Vehicle class (a)	Vehicle name (b)	Vehicle category (also Austroads 2012a) (c)	Code (d)	Designation (e)	Vehicle category (f)	Vehicle name (g)
1	01. Small Car	Small Car	C	C1.1	Cars	
	02. Medium Car	Medium Car	C	C1.2		Medium car
	03. Large Car	Large Car	C	C1.3		
	04. Courier Van-Utility	Light commercial (2 axle 4 tyre)	V1.1	V1.1	Light commercial (2 axle 4 tyre)	Light commercial (2 axle 4 tyre)
	05. 4WD Mid-Size Petrol	4WD Mid-Size SUV Petrol				
3	06. Light Rigid	Light truck (2 axle 6 tyre) petrol	11	L11p	Light truck (2 axle 6 tyre)	Light diesel truck (2 axle 6 tyre)
		Light truck (2 axle 6 tyre) diesel	11	L11d		
	07. Medium Rigid	Medium truck (2 axle 6 tyre)	11	11	Medium truck (2 axle 6 tyre)	Medium truck (2 axle 6 tyre)
		Small bus	Bu	Bu1		
		Route bus (includes school bus)	Bu	Bu2		
4	09. Heavy Bus	Large bus (coach)	Bu	Bu3	Large bus	Large bus (3 axle)
	08. Heavy Rigid	Large truck (3 axle)	12	12	Heavy truck (3 axle)	Heavy truck (3 axle)
6		Articulated truck (3 axle)	A	11S1	Articulated trucks (3,4,5 & 6 axle)	
7	10. Artic 4 Axle	Articulated truck (4 axle)	A	11S2		
8	11. Artic 5 Axle	Articulated truck (5 axle)	A	12S2		
9	12. Artic 6 Axle	Articulated truck (6 axle)	A	12S3		Articulated truck (6 axle)

Appendix D – SCMS STRIDE Threats

STRIDE	Threat Description
S	Spoofing of Source Data Store Linkage Authority 2
S	Spoofing of Destination Data Store Linkage Authority 1
S	Spoofing of Source Data Store End Entity Devices (OBE / RSE)
S	Spoofing of Destination Data Store Linkage Authority 2
S	Spoofing of Source Data Store CRL Store
S	Spoofing of Destination Data Store CRL Store
S	Spoofing of Destination Data Store End Entity Devices (OBE / RSE)
S	Spoofing of Source Data Store Linkage Authority 1
T	Risks from Logging
T	The End Entity Devices (OBE / RSE) Data Store Could Be Corrupted
R	Data Logs from an Unknown Source
R	Lower Trusted Subject Updates Logs
R	External Entity Location Obscurer Proxy Potentially Denies Receiving Data
R	External Entity Device Configuration Manager Potentially Denies Receiving Data
R	Data Store Denies End Entity Devices (OBE / RSE) Potentially Writing Data
R	Potential Weak Protections for Audit Data
R	Insufficient Auditing
I	Weak Access Control for a Resource
I	Weak Credential Storage
D	Data Store Inaccessible
D	Data Flow Is Potentially Interrupted
D	Potential Excessive Resource Consumption for Misbehavior Authority or CRL Store
D	Potential Excessive Resource Consumption for Misbehavior Authority or Linkage Authority 1
D	Potential Excessive Resource Consumption for Misbehavior Authority or Linkage Authority 2
E	Elevation Using Impersonation
E	Weakness in SSO Authorization

SCMS MITIGATION STRATEGIES

STRIDE	Threat Description	Mitigation Description
S	Authenticity of Messages	Message Digital Signatures (Using Private Keys + Timestamp)
S	Spoofing of Messages / Entities	Hardware Security Modules
T	Replay of Messages	Message Digital Signatures (Using Private Keys + Timestamp)
T	Integrity of Private Keys	Hardware Security Modules
T	Integrity of Messages	Message Digital Signatures (Using Private Keys + Timestamp)
R	Origin of Messages	Message Digital Signatures (Using Private Keys + Timestamp)
R	Linkage of Messages	Multiple Linkage Authorities
I	Confidentiality of Messages	Encryption of Sensitive Messages
I	Privacy of Vehicle Identities	Location Obscurer Proxy Use of Pseudonyms and Frequent Changes
D	Availability of Services	Digital Signature Based Authentication
E	Privilege of Both Linkage Authorities	Verification of Misbehavior Reports
E	Privilege of Enrollment	Secure Area Bootstrapping
E	Privilege of Root Authorization	Verification of Certificate Chain Root Management Electors

[36]

Appendix E – Intelligent Transportation System

[5]

Bibliography

[1] "FCC Allocates Spectrum 5.9 GHz Range for Intelligent Transportation Systems Uses." Office of Engineering and Technology, 21-Oct-1999.

[2] "SA - Standards Australia." [Online]. Available: https://www.iso.org/member/1524.html. [Accessed: 03-Jun-2019].

[3] "ISO - About us." [Online]. Available: https://www.iso.org/about-us.html. [Accessed: 03-Jun-2019].

[4] "ISO 14825:2011 - Intelligent transport systems -- Geographic Data Files (GDF) -- GDF5.0." [Online]. Available: https://www.iso.org/standard/54610.html. [Accessed: 03-Jun-2019].

[5] "ISO/TC 204 - Intelligent transport systems." [Online]. Available: https://www.iso.org/committee/54706.html. [Accessed: 03-Jun-2019].

[6] "ICS 03.220.01 - Transport in general." [Online]. Available: https://www.iso.org/ics/03.220.01/x/. [Accessed: 03-Jun-2019].

[7] "ISO/PRF TS 21177 - Intelligent transport systems -- ITS station security services for secure session establishment and authentication between trusted devices." [Online]. Available: https://www.iso.org/standard/70056.html. [Accessed: 03-Jun-2019].

[8] "ISO/TC 22/SC 31 - Data communication." [Online]. Available: https://www.iso.org/committee/5383568.html. [Accessed: 03-Jun-2019].

[9] "EU/US joint declaration of intent on Research Cooperation in Cooperative Systems. Joint Showcase at the ITS World Congress 2012 in Vienna. | Digital Single Market," EU-US, Vienna.

[10] "Progress and Findings in the Harmonisation of EU-US Security and Communications Standards in the field of Cooperative Systems: EU-US Task Force - Reports from HTG1 and HTG3 | Digital Single Market," EU-US, Vienna.

[11] L. SUN, Y. LI, and J. GAO, "Architecture and Application Research of Cooperative Intelligent Transport Systems - ScienceDirect," *Procedia Engineering*, vol. 137, pp. 747–753, 2016.

[12] "CITS - CEN/TC 278 Intelligent Transport Systems (ITS)." [Online]. Available: https://www.itsstandards.eu/cits. [Accessed: 03-Jun-2019].

[13] "ISO 17427-1:2018 - Intelligent transport systems -- Cooperative ITS -- Part 1: Roles and responsibilities in the context of co-operative ITS architecture(s)." [Online]. Available: https://www.iso.org/standard/66924.html. [Accessed: 03-Jun-2019].

[14] "Cooperative, connected and automated mobility (CCAM) | Mobility and Transport." [Online]. Available: https://ec.europa.eu/transport/themes/its/c-its_en. [Accessed: 03-Jun-2019].

[15] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles," *IEEE Consumer Electronics Magazine*, vol. 7, pp. 18–25, 2018.

[16] "Visible/infrared spectroscopy (VIRS) as a research tool in economic geology; background and pilot studies from New Foundland and Labrador." [Online]. Available: https://www.researchgate.net/publication/288899855_Visibleinfrared_spectroscopy_VIRS_as_a_research_tool_in_economic_geology_background_and_pilot_studies_from_New_Foundland_and_Labrador. [Accessed: 03-Jun-2019].

[17] R. Bera, J. Bera, S. Sil, S. Dogra, N. B. Sinha, and D. Mondal, "Dedicated short range communications (DSRC) for intelligent transport system," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, 2006, pp. 5 pp. – 5.

[18] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 07, pp. 1162–1182, Jul. 2011.

[19] K. A. Hafeez, L. Zhao, B. Ma, and J. W. Mark, "Performance Analysis and Enhancement of the DSRC for VANET's Safety Applications," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3069–3083, Sep. 2013.

[20] "IEEE 802.11p-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments."

[Online]. Available: https://standards.ieee.org/standard/802_11p-2010.html. [Accessed: 03-Jun-2019].
[21] S. Hamato, S. H. Syed Ariffin, and N. Fisal, *Overview of Wireless Access in Vehicular Environment (WAVE) Protocols and Standards*, vol. 7. 2013.
[22] A. Abdelgader and L. Wu, *The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges*, vol. 2. 2014.
[23] "ITS Standards Program | Fact Sheets | ITS Standards Fact Sheets." [Online]. Available: https://www.standards.its.dot.gov/factsheets/factsheet/80. [Accessed: 03-Jun-2019].
[24] "V2X Application and Diagram - Movimento." [Online]. Available: http://eecatalog.com/automotive/files/2016/05/Movimento_new.png. [Accessed: 03-Jun-2019].
[25] "Australian Transport Assessment and Planning (ATAP) Guidelines," TRANSPORT AND INFRASTRUCTURE COUNCIL, PV2 Road Parameter Values, 2016.
[26] "WA Vehicle Classification." [Online]. Available: https://www.mainroads.wa.gov.au/Documents/Austroads%201994%20Vehicle%20Classification%20System.RCN-D15%5E23121105.PDF. [Accessed: 03-Jun-2019].
[27] "Vehicle standards (Department of Transport and Main Roads)." [Online]. Available: https://www.tmr.qld.gov.au/-/media/Licensing/Gettingalicence/Licence-classes-and-codes/Pdf_internet_vehicle_types_1_july_09.pdf?la=en. [Accessed: 03-Jun-2019].
[28] National Highway Traffic Safety Administration (NHTSA) and Department of Transportation (DOT), *Federal Motor Vehicle Safety Standards; V2V Communications; Proposed Rule*. p. 3853.
[29] "CFR 49 CFR 523 - VEHICLE CLASSIFICATION - Content Details - CFR-2011-title49-vol6-part523." [Online]. Available: https://www.govinfo.gov/app/details/CFR-2011-title49-vol6/CFR-2011-title49-vol6-part523/summary%20https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/fmvss-quickrefguide-hs811439.pdf. [Accessed: 03-Jun-2019].
[30] "The CAVI project | Transport and motoring | Queensland Government." [Online]. Available: https://www.qld.gov.au/transport/projects/cavi/cavi-project. [Accessed: 03-Jun-2019].
[31] A. Rostami, H. Krishnan, and M. Gruteser, "V2V Safety Communication Scalability Based on the SAE J2945/1 Standard," p. 10.
[32] "Vehicle-to-Vehicle Communication | NHTSA." [Online]. Available: https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication. [Accessed: 03-Jun-2019].
[33] "US VIO Vehicle Registration Data 2018, Fast Quote on Car Data." [Online]. Available: https://hedgescompany.com/automotive-market-research-statistics/auto-mailing-lists-and-marketing/. [Accessed: 04-Jun-2019].
[34] "About us - CEN/TC 278 Intelligent Transport Systems (ITS)." [Online]. Available: https://www.itsstandards.eu/tc278. [Accessed: 04-Jun-2019].
[35] Cross-Cutting Technical Committee, "J2735 - Dedicated Short Range Communications (DSRC) Message Set DictionaryTM," SAE International.
[36] M. D. Furtado, R. D. Mushrall, and H. Liu, "Threat Analysis of the Security Credential Management System for Vehicular Communications," in 2018 IEEE International Symposium on Technologies for Homeland Security (HST), 2018, pp. 1–5.