**Automotive Black Box:**

**Node Authentication Within an Automotive CAN Using the Length of Cable Between Nodes**

Author: Oliver Lewis

Supervisor: Eirini Anthi

10th May 2024

# **Abstract**

The Controller Area Network (CAN) is the main structure of communication within all modern automotive applications. It was designed in the late 1980’s by Robert Bosch GmbH (Bosch) [www.can-cia.org. (n.d.)] and is a simple, efficient, and robust method for communicating between connected devices. However, when it was designed security was not a key consideration within the standard. Given this, and the fact that internet-connected nodes have become more common, vehicles have become vulnerable to attacks.

Attacks range in their methodology, identification, and ease of prevention. One such attack, known as the ‘Bus-Off’ attack highlighted by [Cho, K-T., and Shin, K.G. (2016)], is exceedingly difficult to detect due to its design and the simplicity of the CAN bus. The ‘Bus-Off’ attack takes advantage of collision avoidance mechanisms within the protocol to force a “victim node” into the “Bus-Off” state. This attack, along with CAN injection, will be one of the focuses of this paper.

The overall goal of this research project is to create an automotive data logger aimed at identifying instances of attacks by creating a form of novel authentication within the CAN bus which would then allow attacks to be logged securely. Having a way to reliably identify unauthenticated messages and log the nodes involved can aid forensics and provide ways to eventually disable many attacks altogether.

Table of Contents

[Abstract 2](#_Toc166224362)

[1. Introduction 4](#_Toc166224363)

[1.1 Aims 5](#_Toc166224364)

[2. Background 6](#_Toc166224365)

[2.1 Problem 6](#_Toc166224366)

[2.2 The CAN Bus 6](#_Toc166224367)

[2.3 The ‘Bus-Off’ Attack 7](#_Toc166224368)

[2.4 Theory of the proposed solution 7](#_Toc166224369)

[2.5 Analysis of Related Work 9](#_Toc166224370)

[2.6 Research Questions 10](#_Toc166224371)

[2.7 Threat model 10](#_Toc166224372)

[3. Methodology 11](#_Toc166224373)

[3.1 Initial Approaches 11](#_Toc166224374)

[3.2 Collecting Data. 12](#_Toc166224375)

[3.3 Comparing Results. 14](#_Toc166224376)

[3.4 Measuring Collisions 17](#_Toc166224377)

[4. Creating a Prototype 19](#_Toc166224378)

[4.1 Designing the Circuit 19](#_Toc166224379)

[4.2 Designing the code 19](#_Toc166224380)

[4.3 Implementation 21](#_Toc166224381)

[4.4 Final Results 23](#_Toc166224382)

[5. Conclusion 27](#_Toc166224383)

[5.1 Key findings 27](#_Toc166224384)

[5.2 Methodology 27](#_Toc166224385)

[5.3 Limitations 28](#_Toc166224386)

[5.3 Future work 28](#_Toc166224387)

[5.4 Anticipated use cases 29](#_Toc166224388)

[6. Reflection on Learning 30](#_Toc166224389)

[References 31](#_Toc166224390)

[Appendix 32](#_Toc166224391)

[Acknowledgements 32](#_Toc166224392)

# 1. Introduction

Controller Area Networks (CANs) are small-scale communications networks deployed primarily in automotive applications such as consumer vehicles. The CAN protocol is designed with four key features in mind: simplicity, efficiency, robustness, and ease of access [Falch, M. (2022)]. These features have made it the ideal base for automotive communication for over 33 years. However, an unfortunate drawback to these key benefits is that they only leave a small amount of room for security. Despite having collision detection and fault tolerance, the CAN protocol does not have any form of authentication or permissions which means that any node on the bus can communicate freely with any other. This feature can leave it exposed to the risk of malicious attacks.

In the age of electric vehicles (EVs), the CAN bus has become a vital backbone for most of the functions within a car. On a traditional internal combustion engine vehicle, a large array of parts rely on the presence of an engine to function efficiently, including the power-assisted brakes, steering, and air conditioning. Given the lack of engines within EVs, an array of electric solutions have been implemented to replace this reliance. These solutions also provide driver aid features such as lane assistance and collision avoidance systems. Given these solutions are all electronic, it is most effective to communicate with them using a form of CAN bus as it is robust and fault tolerant while only requiring two wires, regardless of how many nodes are connected. These developments provide a vast range of benefits and have allowed cars to become a lot safer [Motor Match. (n.d.)] with the help of software implementations.

The inclusion of faster hardware and complex software within vehicles has created a need for internet connectivity for over-the-air (OTA) updates and patches. This connectivity has presented a significant issue within the CAN bus system. CAN bus networks are designed to be self-contained systems, meaning that originally the only way for a bad actor to interact with the network would be to install a device physically. These forms of attack are possible but exceedingly rare due to physical security methods in vehicles. As early as 2007, with the introduction of smart vehicles, attacks focused on the CAN bus were discovered [Lang, A., Dittmann, J., Kiltz, S. and Hoppe, T. (2007)] which allowed data to be sniffed from a car remotely. Additional vulnerabilities have been gradually discovered over the last 14 years.

A key attack format is a Denial of Service (DoS) attack in which areas of the network can be disabled by a remote attacker. One such method observed in this paper is the ‘Bus-Off’ attack. This attack works by exploiting the collision detection methods within the CAN bus. A sizeable portion of modern safety features rely heavily on the speed and robustness of the CAN protocol with the maximum latency of high-priority messages being less than 120 microseconds at 1 Mbit [www.computer-solutions.co.uk. (n.d.)]. Devices like crash sensors and collision avoidance devices can reliably send data fast enough to make a considerable difference to the survival rate of the occupants of the vehicle. It is difficult to implement any form of authentication in the interest of efficiency. This is what the ‘Bus-Off’ attack preys on.

This paper proposes to implement a form of device authentication on the bus without delaying any communication. Conventional methods of authentication, such as encryption, would not be effective as they require obscuring data which adds a layer of processing time. To prevent additional processing within the time-sensitive nodes, the goal of this paper is to create a data logger, such as a ‘black box’ that could monitor communications and provide a form of authentication which will act independently of the communications within the CAN bus. Therefore, it will not be able to prevent unauthorised messages from being sent, but it will be able to detect them.

As the built-in message ID of nodes can easily be spoofed, another method of differentiating each node on the system must be discovered. Authentication relies primarily on individuality, i.e. an aspect of each node that is unique to it. In an automotive application, the location of each node provides a good starting point, as each node will always be in the same location for a given vehicle, and therefore any attacks that aim to use an infected node to masquerade as another will be in the wrong location.

While measuring the physical location of nodes is technically feasible, it is not practicable as it requires GPS or other positioning technology which is too expensive for commercial applications. However, there is another unique aspect that comes from the location of each node that can be used instead, and that is the length of wire running to a given reference. Measuring the distance of the wire is much simpler as length is directly proportional to the voltage increase of signals along the wire. This means that the ‘black box’ simply needs to measure the voltage along the wire to validate the messages.

## Aims

- Attempt to measure the signal strength of messages from a given node at variable distances to provide a form of message authentication.

- To detect instances of the ‘Bus-Off’ attack with a high degree of accuracy using the varied distance between the victim node and the attack node.

- Once ‘Bus-Off’ attacks have been detected, analyse signal strength to determine the potential source of the breach.

- Depending on the success rate, deploy other tactics to reinforce reliability.

# 2. Background

## 2.1 Problem

The CAN bus protocol was designed as a trust-all solution, which means that any message sent along the bus does not need to be authenticated. Authentication, while good for security, can be detrimental to the proper function of high-speed communication. This is most obvious in instances such as the use of crash detection sensors. [Duan, L., Sun, G., Cui, J., et al. (February 2016)] shows an impulse graph of a collision within a vehicle, the area under the curve is a good metric of how serious the collision is. The overall goal of crash sensors and airbags is to transfer as little force to the occupants as possible. Given that high acceleration can be felt in a few milliseconds it is vital to reduce the amount of time it takes for the airbags to deploy and therefore cushion the occupants.

Vital communications like these are why it would have been very detrimental to implement authentication when the CAN protocol was originally designed. The need for security is increasing, but speed and simplicity will always be a priority within automotive communication. Given this, conventional methods of authentication cannot be reasonably implemented as they take a large amount of processing, and therefore, time.

## 2.2 The CAN Bus

The CAN protocol is a high-integrity serial communication based on a half-duplex differential signal that uses conventional binary logic to allow communication. The figure below shows a conventional CAN signal that would be seen when performing logic analysis:

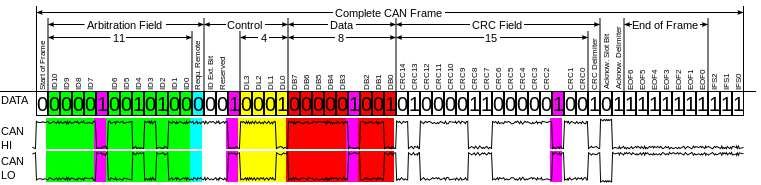


Figure 1: [Idrissi, Y., Fassak, S., et al. (November 2017)] A conventional CAN message.

CAN bus communication is based on differential signalling which means that there are two wires for communication, and the voltage between them is what is measured for a high or low bit. Within the CAN protocol the logical states (1 and 0) are recessive and dominant, which means that one is the default state on the bus and zero will always overpower one. This recessive and dominant design means that any node can easily detect traffic on the line.

[Libelium.com. (2020)] and [Kvaser. (n.d.)] describe the functionality of the CAN bus. When a signal is pulled low on the “Hi” line there is a set resistance the current must flow through to travel along the line. This resistance pulls the voltage up to around 4.5/5v and conversely down to 0v on the “Lo” line. However, when a signal is pulled high on the “Hi” line the controller brings the signal down to the voltage of 2.25v. This is the same for the “Lo” line as they split the reference voltage of 4.5/5v, which marks a recessive state. As this is achieved by essentially ‘shorting’ the lines to ground it is impossible to overwrite a dominant state on the CAN bus. Interestingly, however, other resistances such as the length of the wire can change this voltage, and most signals will never be exactly 2.25v.

This design allows the CAN protocol to include features for error handling and collision avoidance that greatly increase the resilience of communication. One key aspect of the communication is the arbitration field in which a node will send a unique ID to show priority (as well as origin). During this state, a device will send out a starting (dominant) bit that activates the line and then follows this with its ID. If another node attempts to send a message at the same time, they will compare IDs and the node with a higher priority will continue their frame while the other waits.

It is still possible for a collision to occur so there are error flags built into the protocol to help detect this. [CSS Electronics. (n.d.)] describes the error process. All nodes within a CAN bus are error-sensitive, meaning that when they are idle, they listen to the line, and if they detect any issues they can intervene. The main method of intervention is an error flag; the error flag is a set of six dominant bits which completely overwrites the bus and therefore flags all nodes to stop transmitting. On top of active error handling, each node within the CAN bus has a built-in error counter which allows each node to perform a form of simple self-diagnostic. If an error occurs the current nodes transmitting will increase a counter called the Transmit Error Counter (TEC), and depending on the value of the counter, the node can act accordingly.

There are 3 main error states for each node [www.can-wiki.info. (n.d.)] within a CAN: “Error Active”, “Error Passive”, and “Bus Off”. “Error active” is the default state when the TEC is below 127 in which the device follows the error-checking rules stated above. This response changes when the TEC increases above 127 into the “Error Passive” state. In this state, if an error occurs the node will send a recessive error flag of six bits which will be invisible on the line, so it does not interfere with other devices. If this is then overwritten again the node will increase the counter and wait. Finally, if errors continue occurring and the TEC of a node increases above 255 then the node is pushed into a “Bus Off” state in which the controller will stop any communications and must follow a recovery sequence to continue communications.

## 2.3 The ‘Bus-Off’ Attack

The ‘Bus-Off’ attack is a form of Denial-of-Service attack (DoS) in which the collision detection and avoidance system of the CAN protocol is exploited to disable a victim node from communicating.

The attack, described in detail by Cho and Shin in their paper, [Cho, K-T., and Shin, K.G. (2016)] aims to disrupt communication during arbitration. Starting in Phase 1, the attack node listens on the line for a message from the victim node. Once a message occurs the attack node proceeds to emulate the same message but changes one bit so that the attack message is dominant while the victim is recessive, to ensure a collision across the bus. This process continues until both nodes are at the limit of entering the passive error state, which marks the transition from Phase 1 to Phase 2. During this transition, the victim node will end up in the error passive state and the victim will attempt to send a recessive error frame; however the attacker will be transmitting its frame by design, and the victim will have to wait. This means that the attacker can reduce its error counter and places itself one step ahead of the victim node for Phase 2.

The last phase of the attack is what finally disables the victim node. Since the attack node is no longer in an error passive state but the victim node is the same process of forced collision occurs as in Phase 1. The error counter for the attack node decreases while the error counter for the victim increases until the victim is forced into the “Bus Off” state.

## 2.4 Theory of the proposed solution

As the ‘Bus-Off’ attack relies on spoofing the ID of the victim node, one suggested method to defeat it would be to provide some form of authentication so that it is impossible to spoof messages.

This paper suggests a method of authentication using the length of wire between nodes, i.e. the signal strength, as a way of differentiating the source of a given message. This method of authentication is not just useful for the ‘Bus-Off’ attack as it can be implemented as a measure to prevent CAN injection attacks which are commonly used for vehicle theft in the UK [Autoblog. (n.d.)].

The resistance of a section of wire is directly proportional to its length in a linear fashion, the longer a wire the higher the resistance. This can be shown by the equation:

Where R is resistance, Rho (ρ) is Resistivity, L is Length, And A is Area.

Due to Ohms's law, this also affects the potential for current to flow (Voltage), which means that the voltage of a high signal within the CAN bus can be affected by how far away one node is from another. This difference between voltages is what this research aims to use to authenticate messages efficiently.

The above equation has a few aspects that are important to understand to be used reliably to determine distance and create a model to extrapolate data. The first aspect is area, area refers to the cross-sectional area of the current carrier. Fortunately, it is common practice in automotive applications to use a standard gauge (thickness) of wire for CAN applications around 22 AWG or 0.34mm2 [High-Performance Academy. (n.d.)] which can be substituted into the equation as 3.4x10-7 for m2. Other thicknesses can be used however, as it is completely possible to vary the design to a different thickness or to remove dependency on thickness to focus on voltage alone instead.

The next aspect of the equation is Rho or resistivity, this is also well standardized as copper is commonly used as the core for wiring. Copper has a known resistivity of 1.67x10-8 at around 20 degrees Celsius [Matula, R. A. (n.d.)] which can again be substituted into the formula.

Given these two substitutions, it is possible to create a linear equation that relates the length of wire to a resistance:

A number with numbers on a white background

Description automatically generated with medium confidence

This is the first step in developing the model for distance estimation, the next step is understanding how the resistance relates to the voltage across the lines.

Voltage is directly related to resistance as shown in Ohm’s law (Voltage = Current \* Resistance), which means that assuming a constant current of around 20 mA [Texas Instruments. (n.d.)], given the spec of the protocol, a linear model of voltage against length can be created as follows:

With the addition of the minimum voltage (2.25v) and simplification:

A black text with numbers

Description automatically generated

This graph should model the change in voltage over a given length, but as the units are in metres and volts, it will need to be modified to suit the relatively minor change in distance (mm) and voltage (μV).

Converting to the correct units gives:

A black text on a white background

Description automatically generated

This equation allows relating the recorded voltage to a length so will become the basis of the authentication. To ensure reliability further testing will be performed to understand how interference and loading affect the voltage recording to ensure the results are reproducible.

## 2.5 Analysis of Related Work

The proposed solution aims to provide a set of functionalities that could prevent a range of attacks, not only the ‘Bus-Off’ attack. Another vulnerability of note is an exploit being used to steal Jaguar Land Rover products such as Range Rovers [Autoblog. (n.d.)], which places a new unauthorized node on the network to unlock and eventually steal the vehicle. Most commercial solutions to this attack are simple but well-engineered immobiliser units such as the ‘ghost’ [www.autowatch.co.uk. (n.d.)] which work to disable key features of a vehicle such as power to the engine control unit or disable the fuel pump to prevent the vehicle from starting without a key or keycode being present. Despite some of these solutions using inbuilt buttons within the car to be as undetectable as possible they act as a ‘band-aid’ solution to the underlying issue of a lack of authentication with the CAN bus.

Many research papers have considered various methods of authentication within CANs but have mostly been focused on cryptographic methods. Cryptography has been a tried-and-true method of authentication for millennia and lightweight solutions have been proposed [Luo, J.-N., Wu, C. and Yang, M.-H. (2021)] that work to provide authentication within automotive networks. Most solutions, while effective, require a substantial change to the fundamental function of the network and would be costly or feasibly impossible to implement in pre-existing vehicles.

There is one solution, however, that aims to be backwards compatible with pre-existing vehicles and is known as vatiCAN [Nürnberger, S. and Rossow, C. (n.d.)]. VatiCAN is a fully proposed protocol by Stefan Nürnberger and Christian Rossow that aims to implement many features including spoof detection and authentication for legacy devices. The spoof detection is implemented by having each node listen for its own ID during communication and signalling an error if it detects it. While this solution seems very promising and defeats the reliance on message spoofing involved in the ‘Bus-off’ attack, it is unclear whether a node could detect this information during arbitration. On top of this, the implemented authentication for existing systems is limited to selecting important nodes as it relies on adding a separate frame to messages so that they can be verified. This method, while seemingly robust comes at a cost of congestion on the line which is why the number of authenticated nodes is limited.

The second aspect of research is automotive forensics. There are many papers and commercial solutions focused on general automotive forensics which can communicate and collect data from all over the CAN bus, including often vulnerable head units and infotainment systems. However, the ‘Bus-Off’ attack is one attack that goes mostly under the radar. Given its design, it is incredibly difficult to detect, let alone understand which nodes are being affected. One solution highlighted by [Cho, K-T., and Shin, K.G. (2016)]is to recognise the ‘Bus-Off’ attack using the high number of consecutive collisions on the line in a short period of time; this method provided some success in identifying the ‘Bus-Off’ attack, however it is very easy for an attacker to simply slow the rate in which the attack node forces collisions and therefore make the attack harder to detect. Expanding on this method of detection, [Takada, M., Osada, Y., Morii, M., et al. (August 2019)] proposed a counterattack which can force the attack node into the “Bus-Off” state before the victim node and therefore prevent the attack. This method seems particularly successful but relies on the ability to detect the ‘Bus-Off’ attack reliably.

Although the ‘Bus-Off’ attack has been somewhat reliably detected, a new method of attack proposed by Gedare Bloom [Bloom, G. (2021)] aims to further obscure the attack and removes one of the key features relied upon: the frequency of errors. This solution also removes a vital identifier in which the compromised node must send a clear message during the transition to Phase 2 which prevents any logical analysis from reasonably identifying the attack.

## 2.6 Research Questions

1. CAN bus controllers have a variable current output which means that the voltage drop will not always be constant. Therefore, it is vital to test that the voltage drop is measurable across a reasonably short span otherwise it would not be feasible to use this method for authentication.

2. How do collisions affect signal strength? As data is originating from two separate places will it be possible to measure two distinct distances or will the signal strength of one node overwrite the other?

3. Reliability. This novel method has many variables to contend with including interference. It may be possible to measure distance, but it might not be reliable at high speed or when collisions occur.

4. The ‘Bus-Off’ attack. The attack is designed to be as stealthy as possible; will it be possible to consistently detect an occurrence?

5. Speed of processing. CAN signals can travel at 1Mbit per second [Falch, M. (2022)] so will it be possible to record both the message content and its voltage at the same time without data loss from either?

6. If collisions affect the signal in unpredictable ways, can we still detect the ‘Bus-Off’ attack?

## 2.7 Threat model

The proposed solution aims to be a catch-all solution in terms of providing authentication within the CAN bus. This therefore assumes that the network is set up to be completely insecure and nodes can be installed remotely or physically at any point during operation. On top of this, the goal is to understand the effects on existing systems so the CAN bus used for experimentation was set up as it would be in any current practical application with no modification unless attempting to force collisions.

# 3. Methodology

## 3.1 Initial Approaches

Due to limited research into the method proposed, it was important to gather evidence that supports the initial hypothesis. In order to gather supporting evidence a test CAN network was constructed in which it could be possible to reliably measure voltages between the high and low signals while being able to vary cable length.

The test network was comprised of two CAN controller modules and two Arduino Pro Micros, one pair was set up to be a sender and the other pair was set up to be the receiver. The Arduinos signal the CAN controllers to allow messages to be sent and received; a small piece of code was created to continuously send the same message from one controller to another. Initially, this code was set up to send a continuously repeating message including the initialisation data and a string of 64 ones, testing in the first experiment showed that it was important to also vary the message content, so the tests were repeated with a string of 64 zeros instead. A diagram of a circuit board

Description automatically generated

Figure 2: The circuit diagram for the first test layout

The figure above shows the first experiment layout with the voltmeter measuring across the “Hi” and “Lo” lines.

A diagram of a computer

Description automatically generated

Figure 3: The circuit diagram for the second test layout

The figure above shows the second experiment layout with the voltmeter measuring across the “Hi” line and ground.

Above are the circuit diagrams of the setup for both experiments; the voltage was first sampled across the “Hi” and “Lo” wires. Firstly, with the message of all high signals and then with the message of all low signals. Although not included in the diagrams, when using an Arduino to measure voltage a 10 Mega Ohm resistor was used in series alongside the Arduino in order to prevent any adverse effects on communication.

Secondly, the voltage was measured across just the “Hi” line of the bus using the same methods as with the first test. Ordinarily, measuring voltage using the layout in the second experiment would be impractical as a reference cable would need to be run to every node, and in some systems that could be 100+ cables. However, due to automotive design, the chassis of the vehicle acts as a common ground and therefore any ground source can be used as the reference point for measuring voltage consistently.

## 3.2 Collecting Data.

The figures below show the components used in testing and how they were set up. The main variation between these images and the circuit diagram is the use of a 10 Megaohm resistor on the A0 pin.

A circuit board with wires connected to it

Description automatically generated

CAN Controllers

Controller Arduinos

Voltmeter

Figure 4: The Circuit built for the first test

In the above image, the voltmeter is set up to measure the voltage across the “Hi” and “Lo” wires as shown in the wiring diagram for the first test.

A circuit board with wires connected to it

Description automatically generated

CAN Controllers

Controller Arduinos

Voltmeter

Figure 5: The Circuit built for the second test

The above image shows the voltage being measured across the “Hi” line by using one of the ground pins of the controller Arduino as a reference.

Initial testing showed that the change in voltage would be around 3mV per 50mm which is difficult to detect with a conventional multi-meter. To ensure that high-resolution information was collected the analogue pins of an Arduino were used instead; an Arduino Uno R4 [‌Arduino® UNO R4 Minima. (n.d.)] had to be used specifically which has a 14-bit analogue resolution and therefore allowed measurements to be taken in increments as small as 0.0003051944 V.

The sender continuously sends a string of 64 ones (or zeros) to the receiver then pauses for 100 milliseconds and repeats this cycle indefinitely. During this time, the voltmeter logs the voltages to be measured to ensure a sizeable number of messages are recorded to allow consistency across the samples. Multiple instances of the string of high signals were recorded for each length of wire, but the best example with the least interference and variation was chosen to allow for a fair comparison.

Within the test environment, every variable was controlled and measured to ensure the only variation between each result was due to the length of the wire.

The following control variables were included in both experiments:

* The message content between nodes to ensure there were the same number of samples between each test.
* The location and proximity to potentially interfering devices; especially as the second test was only measured across one line was important to ensure that there was as little interference as possible and that any potential interference, such as from the measuring equipment, was kept as consistent as possible.
* The thickness of the wire was important to keep consistent; 22 AWG or 0.34mm2 wire was used each time as it is very commonly used in CAN applications.
* The location of measurement compared to the receiving node. Voltage measurement was always taken at the input into the CAN controller to ensure the length of the wire does not deviate from what is expected.

Within the experiments, the only desired dependent variables are that

* The potential difference across the “Hi” and “Lo” lines of the CAN bus for the first experiment and the potential difference between the end of the “Hi” line and ground for the second experiment.
* A relatively short selection of wire lengths was chosen in the experiments in an attempt to determine the minimum accuracy of the measured signals. This would mean that the model would have to be extrapolated but it would be possible to understand with a high degree of certainty what the shortest possible length to be measured would be.

## 3.3 Comparing Results.

The data collected during the research has been included in my GitHub repository: https://github.com/ThePandoConnection/CANAuth

**Test 1:**

Figure 6: The Average voltage readings for the first test

When sending the same message over different distances there is a noticeable difference in the average voltage across the CAN “Hi” and “Lo” wires.

The method of testing voltage across the “Hi” and “Lo” lines revealed a flaw that would not provide reliable results. Due to the design of the CAN protocol, the default state (high voltage on “Hi” and low voltage on “Lo”) on each line remains mostly unaffected by the difference in voltage. However the active state is affected, this in turn creates an unexpected effect where the message content changes how length affects the average voltage.

The figure below shows the difference between average voltages in data collected in 50mm and 100mm sections. The independent variable within these results is the content of messages being sent. The graph on the left shows the voltages of a message of 64 ones, and the graph on the right shows the voltages of a message of 64 zeros.

Figure 7: The average voltage between 50mm and 100mm with the content of the message varied

When a one occurs the “Hi” line is pulled low, and the “Lo” line is pulled high, which reduces the voltage across the two lines. If resistance is added with additional cable the voltage increases on just the “Hi” line as the “Lo” line is at its maximum and therefore the voltage across the two lines increases as cable length increases.

Contrary to this, when a zero occurs the “Hi” and “Lo” lines remain in a passive state. As the voltage on the “Hi” line is already at its maximum only the voltage across the low line can be affected by the length of the wire. The increasing resistance from the length of the wire, therefore, decreases the voltage across the two lines as shown by the figures above. This means for an average signal with a similar number of ones and zeros it would be impractical to attempt to differentiate by average voltage.

It is potentially possible to remove the reliance on message content by focusing on just dominant signals on the line. However, a dominant signal produces a near zero voltage so, when measuring using a device such as an Arduino, the measurements are much more likely to be affected by interference or would require a very high resistance on the measuring lines which would be relatively difficult to implement.

**Test 2:**

Given the results of the first test, another more suitable method was needed to measure the distance between nodes. Therefore, the next test aimed to measure the voltage across just one line. This would lose some of the fault tolerance that comes with the two-wire approach of the CAN protocol but would allow for a more controlled analysis of signal strength.

Figure 8: The average voltage across the 4 lengths in the second test

The figure above shows the increase in voltage across the varied lengths of wire. The graph indicates an increase of around 2 - 4 mV per 50mm which, while small, is significant compared to the maximum potential length of a wire in a given CAN bus of around one thousand meters.

This method is also mostly unaffected by message content as the average voltage can be collected based on only inputs that are around 2.25 volts (the base voltage for a high signal) as the values are not near zero. This means that the message content only affects the variation in voltage across the different lengths if the message is all zeros, which is the default state, and therefore not an issue.

Using this data to create a best-fit line as shown in the figure above produces the linear formula:



The data in the graph has been modified against the data collected so that the units were even, when readjusting for x100 microvolts which is what was collected originally the equation comes out as:

A black text on a white background

Description automatically generated

This strongly resembles the theoretical model and proves that each variable within the experiment is either known or controlled. There is a small potential for issues over greater lengths as the potential current output in the CAN protocol can double from the numbers gathered here. However, this would have a small impact on the results as each sample set collected is relative to one another so in a full production system this could be easily accounted for.

## 3.4 Measuring Collisions

Collisions are a large part of the ‘Bus-Off’ attack and may still pose a large threat to this form of authentication. The majority of the identifying features of the ‘Bus-Off’ attack are a large number of collisions which has previously been used to identify an occurrence of the attack. However, it is hard to predict what the effect of a collision will be on the voltage across the line. Given this, a method to test collisions was needed.

A diagram of a computer

Description automatically generated

Figure 9: The circuit diagram for the collisions test

A set-up similar to the initial experiment was used to understand the effect on voltage when a collision occurs to see if it could also be used to identify the ‘Bus-Off’ attack. The only change compared to the initial experiment was that both CAN controllers were given the same ID and set to send the same message. Giving the controllers the same ID and message is a vital step in ensuring a collision as ID is what determines priority within the line. During Arbitration, each controller sends their ID to allow a priority to be created. If, however, the IDs are the same both controllers will continue to send messages at the same time causing a collision.

All nodes within a CAN bus attempt to listen to errors and will report them as soon as possible. This is done by pulling the lines high to clear all communications and allow for the nodes to reset. Unfortunately, it is not possible to ensure that a particular node in the collision will consistently send an error flag. While the flag could be cleanly measured to provide authentication, it would be difficult to decern which node sent it. Given this, the only valid time to measure voltage for authentication is during arbitration, which is only a small window of time. Attempts were made to differentiate data with arbitration, but as only a few bits at most are sent during arbitration often the data collected was largely unreliable so could not be realistically used to authenticate collisions.

This is not entirely detrimental to understanding and identifying the ‘Bus-Off’ attack. Due to the substantial number of collisions needed in the attack, it may still be possible to detect spoofing by chance but there is a much more reliable alternative in case of failure during early detection. As previously stated, there are 2 phases of the attack and Phase 2 holds a key requirement that can be exploited using the voltage. During the transition to the second phase, it is required that the attack node sends a complete message without collision to reduce its counter to continue its attack, however, this message will be sent with an invalid ID and therefore should be able to be picked up by the ‘black box’.

# 4. Creating a Prototype

## 4.1 Designing the Circuit

Despite being able to differentiate between distances in a test environment, creating a real-world prototype presents some considerable challenges to overcome. Firstly, is the order in which information needs to be handled. It is particularly important to know the message content, for the ID, as well as the voltages to ensure accurate distance measurement; this is difficult using a conventional Arduino as they are single-core devices and cannot handle collecting two sets of data concurrently. This could be easily solved with a different microcontroller such as an RP2040 which is a dual-core device however it only has an 8-bit analog-to-digital converter (ADC) [the Raspberry Pi Foundation. (January 2021)] and therefore cannot measure the resolution of voltages required.

Finding an off-the-shelve high-resolution multicore solution proved impractical so the solution opted for within this paper is shown below:

A diagram of a circuit board

Description automatically generated

Figure 10: The circuit diagram for the prototype

In this layout, there are two Arduinos and one multicore device to handle parallel processing (in this case a conventional PC). One device will measure voltage while the other will record the messages from the bus. Although not shown here a 10 Mega Ohm resistor was included on pin A0 as the Arduino was interfering with the line. This was also due to the lack of support for change interrupts on digital pins 0 and 1 later iterations of the design used D11 and D12 instead of D0 and D1 as shown.

## 4.2 Designing the code

The code can be found at the GitHub repository: https://github.com/ThePandoConnection/CANAuth

The code was designed as a demonstration of the accuracy of this system of authentication. Until the prototype was developed it was impossible to gauge the reliability and accuracy of measurement without being in a fully independent system.

The main challenge came from ensuring the three devices would integrate well with each other while still being able to integrate within an automotive ecosystem. This led to expanding the code built for experimentation in a way that would be simple and low power while also robust and able to keep up with the rate of information being collected.

There were three main aspects to the code base required to create a functional prototype which can be broken down into:

**CAN message receiving code:**

This code came primarily from the library included with the CAN controller module [docs.longan-labs.cc. (n.d.)] which required minor adaptation from the standard example including changing the digital pins and only sending the ID to serial as that is the only useful data. The sending code included was also modified from the example code in this library.

**Voltage measurement:**

The method of measuring voltage had been implemented at the start of testing. The intention was to create code to take measurements from the analogue to a digital converter and to convert this value back into its base form. The Arduino chosen uses a 14-bit ADC for high-resolution measurements, converting the output from this ADC requires understanding the range of measurement 0 – 5 volts and dividing it by the total possible values (2^14). This produces this code:



The resolution chosen was 1.6383 ((2^14 – 1)/10,000) due to the limited resolution of floating-point numbers which led to rounding issues especially further below the decimal point. This result is then simply output to the serial line.

**Parallel processing and comparison:**

Recording and comparing two data streams at once is always difficult, especially with live data streams like CAN bus information. Initially, it appeared that the best solution to ensure reliability was to implement multi-threading; one thread is dedicated to reading CAN messages off the bus, and the other thread is dedicated to reading the voltage from the Arduino. The intention was to have one thread monitor message information and as soon as it saw activity it would send a start flag to the voltage monitoring thread so that the message could be analysed.

Given the need for effective and efficient prototyping Python was chosen as the basis for the main threads controlling both Arduinos, although slower in operation it is easy to write quickly and has many libraries such as Pyserial and the Threading library which could be used to communicate and provide serial support without implementing whole new methods. Python is, however, much slower at run time than equivalents such as C or C++ and given the potential need for speed when measuring live data, it may prove more effective to switch if a prototype cannot easily be made efficient in Python.

The final key consideration was converting the incoming average voltage into a length measurement. The previous experimentation and research had produced an accurate equation for the estimate of voltage compared to the length of the cable. Given this, the initial process when creating the prototype was to rearrange the formula for a usable format. Shown here:

A math equation with numbers and lines

Description automatically generated

The magnitude of some of the values was modified to fit the resolution chosen for the voltage measurement coming directly from the Arduino (i.e. using 100x μV instead of μV).

With this layout, the initial plan was to have the ‘black box’ store the ID of the signal and compare it to a known length. 50mm for ID 1, 100mm for ID 2, 150mm for ID 3, and 200mm for ID 4 were chosen as this reference. Both this and the voltages collected would be used by the controller system to compare the length measured against the length expected. If they matched then no errors would occur, but if they did not then that could easily be flagged. Although it was outside of the scope of this paper given the time and limited access to resources, it would be entirely possible for the ID references to be collected in a form of initialization when connected to a new unknown CAN bus, allowing the system to function with any CAN-based vehicle dating back to 1990.

## 4.3 Implementation

Despite initial plans requiring two simultaneous threads in order to ensure the live data was accurately recorded, fairly late in the process of implementing these threads it was discovered that the CAN controller and Arduino had a buffer for message content. This buffer means that all information received and recorded is not sent until after the end of the CAN frame. This, on top of the discovery that the voltage measured by the voltmeter will only return to the recessive state outside of a CAN frame (as the sample rate is not high enough to decern individual bits) allowed for the implementation to be drastically simplified.

A diagram of a mathematical equation

Description automatically generated with medium confidenceInstead of transferring flags and having to synchronize threads the two measuring devices could act independently, and the data be compiled in a single thread. The code analyses the stream of voltages coming from the voltmeter and waits until it sees a voltage below a given value that is likely not recessive. Each successive value is then stored until a recessive voltage is seen again as this marks the end of the frame. Just after this happens the ID of the message is sent from the CAN controller and calculations can be done to see if the ID matches the recorded voltages. The flowchart below shows the layout of the prototype:

Figure 1111: A flowchart to show the layout of the prototype code

Using the equation found from the previous experimentation allowed the different lengths to be differentiated from each other. However, it consistently overpredicted the length the shorter the wire was. Initial concerns were that the variability of signal strength that comes from interference was the cause. After more testing, it was concluded that the unexpected values were due to a variation between the initial experiments and the prototype. The prototype, unlike the experiment, required the inclusion of termination resistors to be in place which help prevent reflections along the line. These resistors with a value of 150 Ohms are what caused the measured length to increase. So, the below equation was derived to counter the effect.

A black text with numbers

Description automatically generated

Once implemented, this equation appeared to have an adverse effect as the longer the wire got the further out the results would be. It was therefore decided to revert to the original as it only had a minor effect on short distances so would be much more effective.

In this version of the code, the error detection was implemented relatively simply as it was assumed that node 1 was 50mm, node 2 was 100mm and so on. Given this, to test the results the recorded length was divided by 50 to obtain the calculated ID and if the difference between that value and the ID was not less than 0.5 and error was flagged, and the code stopped. For example, 4 – 225/50 = 0.5 so an error occurs as it is too close to 250. This is a very unlikely setup in a real CAN topology as the ID is often used for priority so in a real implementation it would be preferable to have an initialisation stage where each length is recorded compared to the ID of the node and stored in a table. This table could then be referred to later with a margin of error for each node for error handling.

## 4.4 Final Results

A circuit board with wires connected to it

Description automatically generated

CAN Controller

Controller Arduino

Voltmeter

Figure 12: The Assembled prototype

Each length of wire was tested using the prototype ‘black box’ to see not only how the implementation functioned but also to provide data on accuracy. Many samples were gathered with each length to see how much variation was present in the system.

The figures below show the results output by the code for the 4 different lengths of wire as tested previously, each figure is a screenshot of a few select samples from the output of the code which shows in order: the ID (kept the same in this example), the average voltage, and the expected length.

A computer screen shot of a number

Description automatically generated

Figure 13: A snapshot of the prototype output at 50mm

A computer screen shot of a number

Description automatically generatedThe results from the first 50mm reading were higher than expected, this was described in 4.3 but despite this, the values are still very much usable. It would be relatively unlikely that a 50mm span is used in an automotive CAN bus.

Figure 14: A snapshot of the prototype output at 100mm

The accuracy greatly increased at 100mm which is also demonstrated in the table below, there were still some outliers for all lengths (shown in the dataset in the GitHub repo) but often the outliers were only 100mm out which is still very usable for differentiating values.

Figure 15: A snapshot of the prototype output at 150mm

A screenshot of a computer

Description automatically generatedA computer screen shot of a number

Description automatically generatedThe accuracy did not change much at 150mm which shows the dependability of the results. It is worth noting that as seen in these images the voltage alone could be used for authentication, but the variation was much larger in voltage than in length.

Figure 16: A snapshot of the prototype output at 200mm

The accuracy decreased slightly at 200mm; this was also the hardest length to set up. This is likely because CAN wires are usually twisted and although the wires in each test were also twisted the need to remove wires easily meant they could not be twisted as effectively as possible.

Each result shows a peak variation of around 30 mm in measurement. This result shows clearly that the signal strength across the bus is not only possible to be accurately measured but is also reliable enough to be used as a form of authentication.

|  |  |  |
| --- | --- | --- |
| Length (mm) | Average Variation from Expected Result | Standard Deviation between samples |
| 50 | 27.75 | 9.149 |
| 100 | 6.25 | 4.160 |
| 150 | 7.25 | 4.475 |
| 200 | 17.75 | 8.489 |

Figure 17: A table to show the variation and deviation of the results

The figure above shows the data relating to the variation from the expected value within these samples.

The testing period was not without complications, however. Firstly, it often took a few attempts to set up the system and when doing so occasionally the voltages were consistently below what was expected. The hardware used in this system was designed to be much more temporary and flexible than what would be found in an automotive application. Given this reduced robustness, it is likely that interference or improper communication had a stronger effect on the results than was otherwise expected.

The next complication that was present once the system was operating correctly was the occasional outlier result that would be well above or below the standard deviation and expected value by over 100 per cent. These outstanding results may be due to unexpected interference which would briefly cause a spike during communication. A neat solution to this would be to implement some form of filter that could isolate outlying data while being collected so that it would not have as strong of an effect on the average voltage.

# 5. Conclusion

## 5.1 Key findings

Within this paper, the overall goal was to implement a form of authentication in a way that prevents congestion on the line and is easily retrofitted to any pre-existing CAN bus network. Although still very much in its infancy, the concept of verifying frame origin using the voltage of the high signals within the frame has been proven to be effective and can identify and differentiate different lengths of cable (at least between the tested lengths).

Using the equation gathered in 4.2:

A math equation with numbers and lines

Description automatically generated

It was not only possible to differentiate between signals it was possible to identify the distance away from the receiving device.

A large array of samples were collected, and it was shown that the deviation of results from the expected value was rarely more than 30mm with a variation between each result being less than 10mm at peak. It is well within the possibility that a device calibrated to a network could reliably identify each node with relative certainty with no negative effect on the operation of the network.

The two main attacks focused on in this paper revolve around spoofing the ID of a given node. It is fair to say that spoof detection has been achieved to some extent with some room for improvement. As a result of this, it is reasonable to assume that a device such as this one, when inserted into a CAN bus, can effectively prevent bus injection as well as the ‘Bus-Off’ attack proposed by Cho and Shin without requiring any modification to an existing network or limiting communication rates.

Despite this, there is still a way to go when it comes to thorough testing. The automotive environment is one of the toughest on an engineered system and can include high interference and huge temperature differentials. This means that while the idea may work in principle and is more than worth pursuing in a wider context, it is important to test the overall resilience, which could not be achieved within this paper.

## 5.2 Methodology

Robust data collection was at the heart of this paper which is reflected within the results. Every effort was made to ensure that the results gathered were as repeatable as possible with as many variables being controlled to a high standard between each test.

The importance of a substantial dataset was at the forefront of the ideology behind the research and every step was taken to ensure that a large number of samples was collected to ensure that any recognised trends were robust and not a result of outlying information. On top of this, each test was performed multiple times to ensure that each device was functioning correctly.

Another aspect of data collection that was very important to the research was ensuring that each aspect was repeatable. Each variable was controlled between every test to ensure that the only variation was due to the independent variables and all the same devices and cables were used to ensure no variation in the relatively small measurements.

Despite this, some aspects were impractical to control such as interference and variability within the setup simply because of the small resolution of data which was required. This meant gathering clean data took much longer than anticipated to achieve fair results but was still eventually achieved.

## 5.3 Limitations

The system is not without limitations. Firstly, despite being able to function consistently when set up properly, the setup of the initial prototype was less reliable than hoped. This means although operational the solution in its current state could not be implemented as plug-and-play. Although untested, this is likely due to the temporary nature of the hardware as each of the five individual devices used had to be easy to dismantle for effective prototyping.

On top of this, limited testing was performed with larger distances as the main focus was to find the minimum resolution of this method. Due to the structure of the CAN bus, it is unlikely that additional length will deviate much from the model used but it would be advantageous to test the effectiveness over a greater distance.

Unfortunately, due to limited resources, the system could not be tested on any real automotive hardware. Although the CAN protocol does not vary much between conventional and automotive practices testing in a large-scale production network would provide key insight into the effectiveness of the device especially when it comes to combining timing with robust operation.

Finally, with the prototype being implemented in Python using 3rd party libraries it will unlikely be as effective in an embedded system, the vision for the system would be to base it all within a single device and while Python is a brilliant development tool there are better alternatives for low power and robust systems that tend to appear in automotive applications.

## 5.3 Future work

The project in its current state has many limitations as a commercial product but has fundamentally proven the idea which was always the original intent. Given this, there is a good selection of future work needed to make a viable product.

Firstly, and most importantly, the system has yet to be thoroughly tested outside of a designed environment. Although this should not have a significant effect on the outcome of the experiment there will likely be some bugs present that were simply undetectable due to the lack of access to a vehicle for testing. It is strongly advised that any further development should involve a real vehicle CAN to ensure proper operation.

The next step in development revolves around vehicle testing as well as how the ID verification code is set up. Currently, the ID verification is specific to how a network is set up and would not work in most real-world cases. While this design is entirely feasible if the system is implemented alongside a newly developed CAN by knowing the ID of each node hard coding is often not ideal. A much better solution for any network would be to implement an initialisation step that communicates with each node on the system to then measure their location and create a lookup table instead.

An additional aspect that would need to be implemented would be how a failed authentication scenario is handled. Currently, an error is flagged in the system and there is no further action. In cases such as CAN injection, this is not ideal as the message is still received and the vehicle would still be unlocked. One idea would be to have a few cases where the node waits for an OK signal from the ‘black box’ before an action occurs such as when the vehicle is locked. This would need to be handled correctly as safety-critical devices would need to be excluded from this due to the time-sensitive nature of those components. Another complementary implementation would be to have a fail-safe mode for when the vehicle is under motion. When a vehicle is in motion the only real access to the CAN is via the internet if a failed authentication is detected during this case there could be a system that disables all internet-connected nodes to prevent further tampering. These systems would need to be robustly explored as a poor implementation could be more detrimental to implement than having no authentication at all.

Finally, is the hardware implementation. The current device is built around open-source off-the-shelf hardware which is more than capable of performing the tasks asked of it, but creating a fully custom system would be much more effective and would allow for better integration of voltage reading and a more robust connection to a CAN bus. On top of this, a custom device could use an On-Board Diagnostics 2 (OBD2) port which could standardise it to most vehicles built after 1996.

## 5.4 Anticipated use cases

With this implementation being able to be retrofitted into any vehicle fitted with the OBD2 standard it has a wide range of use cases. The main interest as stated in 2.5 is the ‘Bus-off’ attack and CAN injection. Although the ‘Bus-off’ attack is relatively uncommon in real-world attacks CAN injection is a major issue facing a lot of people around the UK. Therefore, the initial anticipated use case of the device would be to prevent vehicle break-ins in an effective retroactive manner once full testing has been achieved.

On top of preventing attacks the implication for vehicle forensics is worth considering. Vehicle forensics is a fairly new aspect of digital forensics, and it is hard to tell how many accidents are caused by some form of malicious software attack but having a system that can monitor spoofed messages and eventually expand into full data logging would be particularly useful for understanding vehicle systems in all contexts including self-driving and fully automated settings.

The final potential use case would be for vehicle diagnostics. Although further investigation would be required, some CAN errors can be hard to diagnose if there is a break or small fault somewhere in the bus that causes sporadic communication. Being able to detect variations in signal strength will be useful in identifying early warning signs and potentially the location of the damage.

# 6. Reflection on Learning

This paper has provided me with a great deal of insight and has helped me develop a wide range of skills. Having the opportunity to analyse and create a completely novel prototype from scratch presented many unexpected challenges which were incredibly valuable for expanding my knowledge.

Firstly, was the development process. The project started as a single idea inspired by A-level physics and a relatively simple understanding of the CAN protocol. Despite having a base knowledge of the CAN protocol and common automotive practices, a large amount of research had to be conducted to confirm the validity of the idea. This has given me a breadth of new knowledge in not just the CAN protocol, but also a completely new skill set revolving around electrical and embedded engineering. This project was a stark contrast to previous work I had done that revolved around expanding an existing idea and would therefore require a lot more work to get to a design and development stage. This process taught me a lot about how best to approach tasks such as this as well as the importance of thorough research as one wrong assumption would set me back a couple of days of progress.

On top of this, I have learned a lot about creating resilient and robust datasets. The data recorded throughout this paper not only required very high-resolution measurements but was so susceptible to small variations that it was incredibly important to focus on, maintaining absolute control over each variable. Each choice made for each experiment had to be well thought out to the extent that the wrong hardware or set-up could be the difference between real results and completely anomalous trends.

The most important part of my development throughout the project was understanding the volatility of new research, especially given the time constraints. Despite having a proof of concept to back up my idea, the process of moving from proof of concept to prototype felt like a monumental step and was just as uncertain as the initial development. This taught me a lot about the best practices when it comes to project management, as too much time spent in the wrong place could have seriously affected the overall outcome of the project. With all the uncertainty, this once again demonstrated the importance of collecting robust data and making sure my methods were as airtight as possible.

The final aspect of learning was general documentation. Often for simplicity and to manage time I tend to produce a prototype with minimal documentation as it is not the finished result. However, for this project, it was paramount to ensure that proper documentation was included in all forms so that the experiments were as repeatable as possible. Once a prototype had been developed there was still the task of creating this report, documenting code, and ensuring each dataset was presentable. The main focus was always to display each key detail and to demonstrate my findings in a repeatable, informative, but also readable way. This was much more challenging than I anticipated as I had performed so much research and had so much data to present that it was very difficult to walk the line between too much information, and assuming a base knowledge the reader would not have. But through the creation, review, and alteration of drafts, alongside the guidance of my supervisor, I believe I achieved an informative and accessible representation of my work.

# References

‌Arduino® UNO R4 Minima. (n.d.). Available at: <https://docs.arduino.cc/resources/datasheets/ABX00080-datasheet.pdf> [Accessed 27/03/2024]

Autoblog. (n.d.). Thieves are now stealing cars via a headlight ‘CAN injection.’ [online] Available at: [https://www.autoblog.com/2023/04/18/vehicle-headlight-can-bus-injection-theft-method-update/?guccounter=1](https://www.autoblog.com/2023/04/18/vehicle-headlight-can-bus-injection-theft-method-update/?guccounter=1%20) [Accessed 18/04/2024]

www.autowatch.co.uk. (n.d.). Ghost Immobiliser. [online] Available at: [https://www.autowatch.co.uk/veh-sec/ghost-2-menu](https://www.autowatch.co.uk/veh-sec/ghost-2-menu%20) [Accessed 26/04/2024]

Bloom, G. (2021). WeepingCAN: A Stealthy CAN Bus-off Attack. doi: <https://doi.org/10.14722/autosec.2021.23002> [Accessed 29/04/2024]

www.can-cia.org. (n.d.). CAN in Automation (CiA): History of the CAN technology. [online] Available at: <https://can-cia.org/can-knowledge/history-of-the-can-technology> [Accessed 07/05/2024]

www.can-wiki.info. (n.d.). can\_faq:can\_faq\_erors - CAN Wiki. [online] Available at:[http:/www.can-wiki.info/doku.php?id=can\_faq:can\_faq\_erors](http://www.can-wiki.info/doku.php?id=can_faq:can_faq_erors)[Accessed 19/04/2024]

Cho, K-T., and Shin, K.G. (2016). Error Handling of In-vehicle Networks Makes Them Vulnerable. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. [online] DOI: [https://doi.org/10.1145/2976749.2978302](https://doi.org/10.1145/2976749.2978302%20) [Accessed 12/03/2024]

www.computer-solutions.co.uk. (n.d.). CAN and CAN FD - a brief tutorial for Embedded Engineers. [online]. Available at: [https://www.computer-solutions.co.uk/info/Embedded\_tutorials/can\_tutorial.htm](https://www.computer-solutions.co.uk/info/Embedded_tutorials/can_tutorial.htm%20) [Accessed 14/02/2024]

CSS Electronics. (n.d.). CAN Bus Errors Explained - A Simple Intro [2023]. [online] Available at: <https://www.csselectronics.com/pages/can-bus-errors-intro-tutorial> [Accessed 28/04/2024]

Duan, L., Sun, G., Cui, J., et al. (February 2016). Crashworthiness design of vehicle structure with tailor rolled blank. Available at: <https://www.researchgate.net/publication/282427177_Crashworthiness_design_of_vehicle_structure_with_tailor_rolled_blank> [Accessed 08/03/2024]

Falch, M. (2022). CAN Bus Explained - A Simple Intro (2021). [online]. CSS Electronics. Available at: [https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial](https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial%20) [Accessed 01/05/2024]

High-Performance Academy. (n.d.). CAN Bus wiring size. [online] Available at: [https://www.hpacademy.com/forum/canbus-communications-decoded/show/can-bus-wiring-size/](https://www.hpacademy.com/forum/canbus-communications-decoded/show/can-bus-wiring-size/%20) [Accessed 01/05/2024]

Idrissi, Y., Fassak, S., et al. (November 2017). A secure protocol for session keys establishment between ECUs in the CAN bus. Available at: <https://www.researchgate.net/publication/322058468_A_secure_protocol_for_session_keys_establishment_between_ECUs_in_the_CAN_bus> [Accessed 02/04/2024]

Kvaser. (n.d.). Controller Area Network (CAN BUS) Protocol | Free Course. [online] Available at: <https://www.kvaser.com/can-protocol-tutorial/> [Accessed 01/05/2024]

Lang, A., Dittmann, J., Kiltz, S. and Hoppe, T. (2007). Future Perspectives: The Car and Its IP-Address – A Potential Safety and Security Risk Assessment. Lecture Notes in Computer Science, pp.40–53. [https://doi.org/10.1007/978-3-540-75101-4\_4](https://doi.org/10.1007/978-3-540-75101-4_4%20) [Accessed 14/02/2024]

Libelium.com. (2020). Hardware | CAN Bus Networking Guide. [online] Available at: [https://development.libelium.com/can\_bus\_networking\_guide/hardware](https://development.libelium.com/can_bus_networking_guide/hardware%20) [Accessed 29/02/2024]

docs.longan-labs.cc. (n.d.). SERIAL CAN FD MODULE - Longan Docs. [online] Available at: [https://docs.longan-labs.cc/114992978/#test-the-board](https://docs.longan-labs.cc/114992978/%23test-the-board%20) [Last Accessed 03/05/2024]

Luo, J.-N., Wu, C. and Yang, M.-H. (2021). A CAN-Bus Lightweight Authentication Scheme. [online] 21(21), pp.7069–7069. doi: <https://doi.org/10.3390/s21217069> [Accessed 12/03/2024]

Matula, R. A. (n.d.). Electrical Resistivity of Copper, Gold, Palladium, and Silver. NIST. Available at: <https://srd.nist.gov/JPCRD/jpcrd155.pdf> [Accessed 02/04/2024]

Motor Match. (n.d.). Ways that our cars have become safer | most important safety features. [online]. Available at: [https://www.motormatch.com/blog/the-top-ways-that-our-cars-have-become-safer-than-ever-before/](https://www.motormatch.com/blog/the-top-ways-that-our-cars-have-become-safer-than-ever-before/%20) [Accessed 14/02/2024]

Nürnberger, S. and Rossow, C. (n.d.). -vatiCAN - Vetted, Authenticated CAN Bus. [online] Available at: [https://www.iacr.org/archive/ches2016/98130234/98130234.pdf](https://www.iacr.org/archive/ches2016/98130234/98130234.pdf%20) [Accessed 29/04/2024]

the Raspberry Pi Foundation. (January 2021). “RP2040 datasheet”. Available at: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>‌ [Accessed 27/03/2024]

Takada, M., Osada, Y., Morii, M., et al. (August 2019). Counterattack Against the Bus-Off Attack on CAN,” Available at: <https://ieeexplore.ieee.org/document/8827010> [Accessed 12/03/2024]

Texas Instruments. (n.d.). Available at: <https://www.ti.com/lit/ds/symlink/sn65hvd230.pdf> [Accessed 01/05/2024]

‌

# Appendix

Given the size of the data collected, it was impractical to include it all within this paper. All code and related documents can be found at: <https://github.com/ThePandoConnection/CANAuth>

# Acknowledgements

I would like to thank Rachel Cooper for her continued moral support and for listening to me and attempting to understand the contents of this paper without a single drop of technological knowledge.