Master of Science in Engineering: Computer Security
August 2019

Secure Automotive Ethernet

Balancing Security and Safety in Time-Sensitive Systems

Martin Lang

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Engineering: Computer Security. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Martin Lang
E-mail: martin@langnet.se

University advisor:
Assistant Professor Martin Boldt
Department of Computer Science and Engineering

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Acknowledgements

I want to express my special thanks to my university advisor Martin Boldt for being a thorough reviewer of my work and an excellent academic advisor. Although he does not have the domain knowledge of the automotive, he could provide me with information regarding academic writing and IT-security.

I would also like to express my thanks to Pierre Kleberger, who was my advisor from Combitech AB that provided me with information regarding the automotive domain and automotive security.

In addition, I would like to thank Adi Karahasanovic at Combitech AB who guided me during a time when Pierre had parental leave and together with Pierre used their expertise to help me construct a plausible scenario and topology used for this thesis.

Abstract

Background. As a result of the digital era, vehicles are being digitalized at a rapid pace. Autonomous vehicles and advanced infotainment systems are just parts of what is evolving within the vehicles. These systems require more information to be transferred within the vehicle networks. As a solution for this, Ethernet was suggested. However, Ethernet is a 'best effort' protocol which cannot be considered reliable. To solve that issue, specific implementations were done to create Automotive Ethernet. However, the out-of-the-box vulnerabilities from Ethernet persist and need to be mitigated in a way that is suitable to the automotive domain.

Objectives. This thesis investigates the vulnerabilities of Ethernet out-of-the-box and identifies which vulnerabilities cause the most significant threat concerning the safety of human lives and property. When such vulnerabilities are identified, possible mitigation methods using security measures are investigated. Once two security measures are selected, an experiment is conducted to see if those can manage the latency requirements.

Methods. To achieve the goals of this thesis, literature studies were conducted to learn of any vulnerabilities and possible mitigation. Then, those results are used in an OMNeT++ experiment making it possible to record latency in a simple automotive topology and then add the selected security measures to get a total latency. This latency must be less than 10 ms to be considered safe in cars.

Results. In the simulation, the baseline communication is found to take 1.14957 ± 0.02053 ms. When adding a security measure latency, the total duration is found. For Hash-based Message Authentication Code (HMAC)-Secure Hash Algorithm (SHA)-512 the total duration is 1.192274 ms using the upper confidence interval. Elliptic Curve Digital Signature Algorithm (ECDSA) - ED25519 has the total latency of 3.108424 ms using the upper confidence interval.

Conclusions. According to the results, both HMAC-SHA-512 and ECDSA - ED25519 are valid approaches to implement as an integrity and authenticity security measure. However, these results are based on a simulation with many potential error factors and should be verified using physical hardware to ensure that these measures are valid.

Keywords: Automotive Ethernet, Information Security, Integrity, Time Sen-

sitive System

Sammanfattning

Bakgrund. Som en påföljd av den digitala eran, så har fordon blivit digitaliserade i ett hastigt tempo. Självkörande bilar och kraftfulla infotainmentsystem är bara några få av förändringarna som sker med bilarna. Dessa system kräver att mer information skickas genom fordonets nätverk. För att nå dessa hastigheter föreslogs Ethernet. Dock så är Ethernet ett så kallat 'best-effort' protokoll, vilket inte kan garantera tillförlitlig leverans av meddelanden. För att lösa detta har speciella tillämpningar skett, vilket skapar Automotive Ethernet. Det finns fortfarande sårbarheterna av Ethernet kvar, och behöver hanteras för att tillämpningen skall vara lämplig för fordonsindustrin.

Syfte. Denna studie undersöker vilka sårbarheter som finns i Ethernet 'out-of-the-box' och identifierar vilka sårbarheter som har värst konsekvenser ur perspektivet säkerhet för människor och egendom. Två säkerhetsimplementationer väljs ut för att se över vidare de kan användas för kommunikation i bilar.

Metod. För att nå arbetets mål, så genomfördes en literaturstudie för att undersöka sårbarheter och potentiella motverkningar. Studiens resulat användes sedan i en simulering för att kunna mäta fördröjningen av en enkel topologi i en OMNeT++ miljö. Sedan addera den tiden med exekveringstiden för säkerhetsimplementationerna för att få en total fördröjning. Kommunikationstiden måste vara mindre än 10 ms för att räknas som säker för bilar.

Resultat. I simuleringen, så ger mätningarna en basal kommunikation på 1.14957 ± 0.02053 ms. När säkerhetsimplementationerna tillsätts så får man den totala kommunikationstiden. För HMAC-SHA-512 mäts den totala kommunikationstiden till 1.192274 ms genom att använda den övre gränsen av konfidensintervallet. För ECDSA - ED25519 mäts tiden till 3.108424 ms.

Slutsatser. Enligt resultaten så är både HMAC-SHA-512 och ECDSA - ED25519 möjliga alternativ för integritets- och äkthetstillämpningar i fordorns kommunikation. Dessa resultaten är dock framtagna ur en simulering med många felfaktorer och bör verifieras med hjälp av fysisk hårdvara så mätningarna är sanningsenliga.

Nyckelord: Automotive Ethernet, Informationssäkerhet, Integritet, tidskänsliga system.

Contents

Acknowledgments **i**

Abstract **iii**

Sammanfattning **v**

1 Introduction **1**
1.1 Security versus Safety . . . 1
1.1.1 Automotive Safety . . . 2
1.1.2 Automotive Security . . . 2
1.1.3 Automotive Ethernet: Finding the balance . . . 4
1.2 Research Questions . . . 5
1.3 Scope . . . 5
1.4 Limitations . . . 6
1.5 Contribution . . . 7
1.6 Structure of Thesis . . . 8

2 Background **9**
2.1 Electronic Control Unit . . . 9
2.2 Control Area Network . . . 9
2.3 Local Interconnect Network . . . 10
2.4 Flexray . . . 10
2.5 Media Oriented Systems Transport . . . 10
2.6 Automotive Ethernet . . . 10
2.6.1 Time-Triggered Ethernet (TTE) . . . 11
2.7 OMNeT++ . . . 12

3 Literature Study **15**
3.1 Literature Study Method . . . 15
3.2 Related Work . . . 16
3.3 Vulnerabilities found . . . 22
3.3.1 Network and System Access . . . 22
3.3.2 Traffic Confidentiality . . . 24
3.3.3 Traffic Integrity . . . 24
3.3.4 Frame Availability . . . 24
3.3.5 Frame Integrity . . . 24
3.3.6 Frame Authenticity . . . 24
3.3.7 Denial of Service (DoS) . . . 25

3.4 Results . . . 26
3.5 Identified Gap . . . 26

4 Countermeasures for Vulnerabilities found in Automotive Ethernet 27
4.1 Method of Finding Potential Countermeasures . . . 27
4.2 Potential Security Measures to Mitigate Ethernet Vulnerabilities . . . 28
4.3 Potential Safety Measures to Mitigate Ethernet Vulnerabilities . . . 29
4.4 Potential Mitigation Measures Results . . . 30

5 Finding suitable security measures using a simulation 31
5.1 Selection of Security Measures . . . 31
5.1.1 Authenticity . . . 31
5.1.2 Integrity . . . 32
5.1.3 The final candidates . . . 32
5.2 Creating the Topology . . . 34
5.3 Preparing the tools . . . 36
5.3.1 Test environment . . . 37
5.4 Setting up the Scenario . . . 38
5.5 Setup of the Simulation . . . 40
5.5.1 OMNeT++ . . . 40
5.5.2 Security Implementations using Nettle TLS . . . 44
5.6 Extracting the Results . . . 45
5.6.1 Extracting Data . . . 45
5.6.2 Statistical calculations . . . 46
5.7 Simulation Results . . . 47
5.7.1 Baseline . . . 47
5.7.2 HMAC-SHA-512 . . . 48
5.7.3 ECDSA - ED25519 . . . 48
5.7.4 Total time of communication . . . 49

6 Analysis and Discussion 51
6.1 Research Question 1 . . . 51
6.2 Research Question 2 . . . 52
6.3 Research Question 3 . . . 52
6.4 Discussion . . . 53

7 Conclusion 55

8 Future Work 57

References 59

A Table of Baseline latencies 63

B Table of HMAC-SHA-512 latencies (in seconds) 67

C Table of ECDSA - ED25519 latencies (in seconds) 71

List of Figures

1.1 The C.I.A. triangle of information security. [1] 3

2.1 A screenshot of the development environment. 12

5.1 Approximation of automotive network topology of a car. 35
5.2 Small simulation topology taken from OMNeT++ Integrated Development Environment. Note that phyX is interface name for the module it is connected to. Every phyX<->phyX connection here is a copper wire connecting physical interfaces together. 38
5.3 The Ned-file of ADASDC that shows the different modules in the higher level module. 40
5.4 A snippet from OMNeT++ IDE showing the handleMessage function of the Tic module. 41
5.5 A snippet from OMNeT++ IDE showing the handleMessage function of the Toc module. 41

6.1 The distribution plots for the security measure latencies. 53

List of Tables

3.1 Opportunities 19
3.2 Challenges 20
3.3 RSA latencies on STM32 micro controller 21

4.1 Potential countermeasures against the security threats found in the literature study. 28
4.2 Potential countermeasures against the security threats found in the literature study. 29

5.1 Partial baseline results 47
5.2 Baseline results total 48
5.3 HMAC results 48
5.4 ECDSA results 48
5.5 Final results using upper interval of confidence interval. B = Baseline, S = Signing, V = Verification, T = Total and % is percentage security measure latency out of total latency. 49
5.6 Final results using average values. B = Baseline, S = Signing, V = Verification, T = Total and % is percentage security measure latency out of total latency. 49

List of Abbreviations

ABS Anti-lock Braking System. 2, 9

ACL Access Control List. 28

AD Autonomous Driving. 22

ADAS Advanced Driver Assistance Systems. 2, 16, 18, 26, 31, 34, 35, 37, 38, 42, 63–65

ADASDC ADAS Domain Controller. ix, 37–40, 42, 43, 45, 47, 63

AES256 Advanced Encryption Standard 256 bits. 21

ARP Address Resolution Protocol. 24, 25, 28

BDC Brake Domain Controller. 37–40, 42, 43, 45, 47, 63–65

BPDU Bridge Protection Data Unit. 28

CAN Control Area Network. vii, 2, 3, 9, 13, 16, 20–22

CAN ID CAN Identifier. 10

CDG Critical Domain Gateway. 35

DC Domain Controller. 6, 35, 36, 38, 39, 42

DHCP Dynamic Host Configuration Protocol. 24, 25, 28

DMZ Demilitarized Zone. 35

DNS Dynamic Host Service. 25

DoIP Diagnosis over IP. 16, 21

DoS Denial of Service. vii, 18, 22, 25, 28

DSA Digital Signature Algorithm. 32, 33, 52, 55

EBCM Electronic Brake Control Module. 9

ECDSA Elliptic Curve Digital Signature Algorithm. iii, v, viii, xi, 33, 34, 36, 39, 44, 47–50, 52, 54, 71, 72, 74

ECM Engine Control Module. 9

ECU Electronic Control Unit. vii, 6, 9, 13, 17, 18, 21, 22, 27, 29, 34–38, 42, 43, 54

EMI Electromagnetic Interference. 10, 11

ESP Electronic Stability Program. 2

FTDMA Flexible Time Division Multiple Access. 10

GPS Global Positioning System. 10

HMAC Hash-based Message Authentication Code. iii, v, viii, xi, 29, 32–34, 36, 39, 44, 47–50, 52, 54, 55

HoliSec Holistic Approach to Improve Data Security. 21

HSM Hardware Security Module. 16, 17, 22

IDS Intrusion Detection System. 16, 28

IP Interet Protocol. 24, 25, 30

IPS Intrusion Prevention System. 28

IPsec Internet Protocol Security. 17

LAN Local Area Network. 23

LIN Local Interconnect Network. 3, 10, 22

MAC Message Authentication Code. 29

MAC Media Access Control. 24, 25, 29, 30

MACsec MAC Security. 16, 17

MOST Media Oriented Systems Transport. vii, 3, 10

OBDII On-Board Diagnosis port version 2. 2

OEM Original Equipment Manufacturer. 42, 54

RC Rate-Constrained. 11, 13, 40, 43

RFI Radio Frequency Interference. 4, 10, 11

RSA Rivest-Shamir-Adleman cryptographic algorithm. xi, 21, 33

SHA Secure Hash Algorithm. iii, v, viii, 33, 34, 36, 44, 47–50, 67, 68, 70

STP Spanning Tree Protocol. 25

TCP/IP Transport Control Protocol / Internet Protocol. 4, 12, 13, 16, 26

TDMA Time Division Multiple Access. 10

TLS Transport Layer Security. viii, 17, 28, 33, 34, 36, 44

TT Time-Triggered. 11, 40, 43

TTE Time-Triggered Ethernet. vii, 11, 13, 16, 22, 40, 42, 45, 57

V2X Vehicle-to-X. 22

VLAN Virtual Local Area Network. 18, 19, 23, 28, 30, 35, 36

VPN Virtual Private Network. 28, 36

Chapter 1

Introduction

This chapter will introduce the reader in some fundamental aspects of the area which have been investigated. Firstly the reader has been introduced to the subject at hand, then the research questions are stated with an explanation. Lastly, the thesis scope is presented with limitations before the last section, where the structure of the thesis is presented.

1.1 Security versus Safety

According to the Oxford dictionary, security is *'The state of being free from danger or threat'* and safety may be referred to as *'The condition of being protected from or unlikely to cause danger, risk, or injury'* [2]. Safety has since the early cars been considered a vital aspect that needs to be dealt with accordingly [3]. However, with the modernization of vehicles, the previously offline vehicles are being connected with modern communication technologies, thus making the internal devices in a vehicle accessible for a malicious user to exploit. This thesis will look at the security requirements to ensure that the vehicle is not being abused by unauthorized personnel with malicious intents, and how to realize measures according to those requirements without compromising the safety requirements of a vehicle. In this thesis, we will use two definitions of security and safety to make the Oxford definitions more applicable in the automotive domain.

Security is the state of a vehicle's system being free from danger or threat from malicious use, which means that there should be measures to prevent malicious activity against the vehicle's systems. A compromise in security could cause the system to get incorrect instructions and perform an action such as steering into a wall. Countermeasures can include standard digital security measures as well as physical separation.

Safety is a state where the vehicle acts as intended promptly. Meaning that the system should perform a braking procedure if such a message is being received, and it should do it promptly so the braking will have its intended effect. A compromise in safety can be regarded as the message not being received or not delivered in time. Countermeasures can include redundancy, time-critical communication networks, and fail-safe measures.

1.1.1 Automotive Safety

Safety has traditionally been regarded as the most relevant property for vehicles, and it has been enhanced by Advanced Driver Assistance Systems (ADAS) [3]. Previously physical access was required to perform malicious acts toward a vehicle and its passengers. Therefore, systems that prevent physical fault and ensures proper functionality has been prioritized, e.g., Anti-lock Braking System (ABS) and Electronic Stability Program (ESP). To ensure the reliability of vehicular communication, several different buses and systems have been implemented for this very function. Control Area Network (CAN) is one of those vital buses that can be found among those critical buses and systems.

Safety relies on knowing how the vehicle will perform, the touch of the steering wheel should reliably affect the direction of the wheels in an expected direction instantaneously. The buses and systems in the vehicle are built and designed to ensure such reliability. CAN is a viable bus for intra-vehicular communication because CAN is very reliable and ensures the timely delivery of information [4]. The essential safety requirements in a vehicle are considered end-to-end latency, frame availability, and network performance [3]. This thesis focuses on the effects of viable security measures on end-to-end latency

Vehicular traffic can be divided into two categories, time-sensitive traffic, and regular traffic. The regular traffic has no strict deadline for transmission. Regular vehicle-related traffic such as locking the doors or changing the temperature of the Air Conditioner has only strict requirements of reliability. The time-sensitive traffic, however, has rigorous requirements in timely delivery, reliability, and integrity. Regarding end-to-end latency, there is a maximum time of 10 milliseconds [5]. This requirement is set to ensure that time-sensitive traffic is kept in real-time to minimize the time until the vehicle performs the critical reaction, e.g., braking or adjust the direction of wheels.

1.1.2 Automotive Security

Information security relies on three core properties - Confidentiality, Integrity, and Availability [6]. Confidentiality is commonly reached through encryption, which prevents unauthorized people from getting access to restricted information. Integrity is reached by using means to verify the validity of information and ensure that information has not been tampered with. Availability is about making sure information is available when needed, i.e., servers are active, or information is accessible. These fundamental pillars are what information security is based around. The essential security requirements in a vehicular network are considered authenticity, frame integrity, and network separation [3]. The main security focus of this paper is authenticity.

All information that travels through the buses of a vehicle is relevant to these pillars, but have previously been relegated because of the lack of access to the network for most users. Previously attackers needed to get physical access to the On-Board Diagnosis port version 2 (OBDII) port to get access to the backbone network. However,

vehicles are becoming more connected using technologies such as 4G, Bluetooth, and Wi-Fi. These wireless technologies open up dangerous attack vectors.

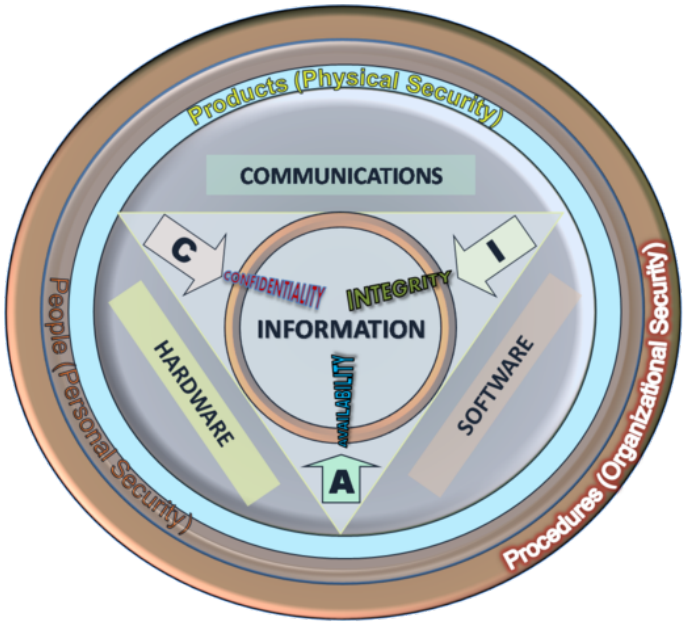

Figure 1.1: The C.I.A. triangle of information security. [1]

In a system where measures have been implemented to prevent cybersecurity attacks, the risk of a successful attack is reduced. However, because of the relegation for information security in vehicular networks, opening up such attack vectors causes substantial security threats. A car's functionality is directly responsible for human lives. A critical malfunction of the vehicle's systems can cause fatal incidents, and if an attacker gets access to the network, they might be able to control the vehicle to do malicious actions. In other words, insecurities in a connected vehicle directly affect the safety of the vehicle.

It is vital to study the security of communication in the vehicles to be able to mitigate possible security vulnerabilities [5]. Today the buses CAN, Flexray, Local Interconnect Network (LIN), and Media Oriented Systems Transport (MOST) are used to communicate information within the vehicle in most vehicles. [7]. None of these protocols have been designed with information security in regard. However, there are several studies on how to secure such a network [8, 5, 9]e.g., by encrypting all information on the network and use session keys [8].

Because of the increased need for security measures and bandwidth, the current buses are about to reach their limits. CAN has a limited bandwidth of 1 Mbps and a tiny payload of 8 bytes [7, 5]. It is challenging to design a suitable security protocol with such constraints on the payload size. To enable security measures to be

implemented with ease, another communication network is required.

1.1.3 Automotive Ethernet: Finding the balance

The most common network standard around the globe is the Transport Control Protocol / Internet Protocol (TCP/IP) which is commonly applied over Ethernet. The fact that Ethernet is the most common network interface protocol results in it being well tested and have multiple possible security measures already designed.

One of the significant problems with Ethernet is that is it a so-called *'best effort'* communication protocol [7]. This implies that is it not trustworthy in reliability and timely deliveries [7]. Another issue is that Ethernet is affected by Radio Frequency Interference (RFI) for it to be used in a safety critical real time system[7]. Because of this, researchers and developers have worked to solve these problems to utilize the vast strengths of Ethernet[10, 11].

Automotive Ethernet is a special design of Ethernet that fulfills the requirements in radio frequency resistance and includes protocols to enable real-time communication [7]. The next step is to find the balance between safety and security in regards to latency. There are set requirements for safety-critical communication, and without exceeding the requirements, it is needed to implement security measures to mitigate attacks against the system [7]. The challenge is not to fulfill the safety requirements while using security measures. It is to find suitable security measures that are secure enough to use while not increasing the latency time past the time set by requirements.

With the development of digitalized vehicles, the security requirements are increasing more than ever in the automotive domain [3]. However, the security cannot compromise the safety, as a vehicle without one or the other will not be safe for anyone to ride. It does not matter whether there is a safety or security issue that causes an accident; either way, there are human lives at stake. Thus, safety and security both need to be prioritized when designing a vehicle network.

1.2 Research Questions

This thesis will investigate vulnerabilities in Automotive Ethernet and how security measures for authenticity and integrity may affect end-to-end latency; finding an answer to this is the focus of the thesis. The thesis will suggest security measures that, according to a simulation, is sufficiently fast and secure according to set requirements. The first two research questions will primarily be answered using a literature study together with IT-Security concepts. The third question will be answered using the simulated experiment by analyzing how selected security measures affect the latency of the selected scenario. The selected security measures will be chosen depending on the findings of RQ2.

RQ1 *What security vulnerabilities can be found in an Automotive Ethernet network out-of-the-box, and what risks do they convey?*

For Automotive Ethernet to be viable for use in the automotive domain, it is vital to be aware of any known security vulnerabilities that may cause security risks and lead to safety issues.

RQ2 *Which security measures can mitigate identified vulnerabilities from RQ1 in Automotive Ethernet?*

Said vulnerabilities in Automotive Ethernet might lead to the technology not being viable for the automotive domain. Therefore, it is crucial to find possible measures that can mitigate those vulnerabilities and thus make the technology usable.

RQ3 *Which security measures are suited for time-critical communication over Automotive Ethernet without compromising safety concerning latency?*

Although a security measure might resolve the security vulnerability, it can still cause problems concerning safety in terms of latency. Automotive safety relies on time-critical applications in the vehicle being on time and authentic.

The overall goal of this work is to investigate secure Automotive Ethernet communications without sacrificing safety-critical aspects.

1.3 Scope

The scope of this thesis will focus on how latencies are affected in time-critical communication (e.g., braking and steering) when applying security measures suggested in RQ2. In addition, all time-critical communication will be conducted over Automotive Ethernet, with other words, Automotive Ethernet is not only used as a backbone; it is present throughout the whole vehicle. The security measures that are within the scope will primarily enforce the integrity and authenticity of messages. Other automotive applications are not included in this thesis, e.g., infotainment and lights. Authenticity is considered the most emerging security requirement [3], and

therefore, the genuineness of messages is of the utmost importance in this thesis. Integrity and timely delivery are considered the most emerging safety requirements [3], thus making latency and message integrity also of the utmost importance for this thesis.

1.4 Limitations

There are limitations to the conducted work. These range from difficulty accessing information to lack of proper documentation. The topic in question is very secretive as there is a competitive market that does not seem to wish for their research to be spread to competitors according to the attempts done to get access to information. Even through contacts of Combitech, it was not possible to gain access to information other than the limited work that is published.

The thesis topic is state of the art, and as far as it is known, no vehicles are using a similar method of communication today. There are no similar studies presented online. However, there is, research in the area, and it is clear that Ethernet is the future of vehicular communication [7, 10]. Ethernet is only partly used today and is mainly used in non-critical communication, e.g., multimedia [7]. There may be uncertainties of how the future actually will look, but this work is based on concepts that are being investigated [12, 4, 13].

Experimenting in this area may be complex and require specific hardware which is hard to retrieve. For this paper, we were unable to access hardware such as Automotive Ethernet Electronic Control Unit (ECU)s and decided to perform a simulated experiment instead. This experiment is planned and prepared to be as similar to real hardware as possible. However, because computer hardware may differ from automotive hardware in a multitude of aspects, it is unknown if all aspects of the hardware can be simulated. Some factors may affect a simulation compared to an experiment on hardware, and some of those factors may not have been included here in the limitations.

The simulation library (OMNeT++) and the modules (INET and CoRE4INET) used are relatively complex to learn [14]. Learning a new tool with limited documentation increases the time to prepare the simulation. The process of trying to get access to hardware took several months, and therefore, there was limited time to learn the simulation library.

Something else that is worth mentioning, because it is unknown how the firmware of the ECUs is, is impossible to determine the amount of time that it takes for an ECU or Domain Controller (DC) to process something. Therefore, rough assumptions have to be done with support from automotive security experts.

Because the simulation is run on an operating system that may consume processing power and memory, it may affect the latencies of both the baseline simulation and security measure process. This is dealt with by using statistical calculations to find

latency values that are reasonable using the result values.

It is unknown how well optimized the OMNeT++ simulation code is, and as pointed out during the defense, parts of the code that have been added might be removed due to optimization which might affect the results. If the code has been removed due to optimization, the firmware functionality other than communication is not included in the baseline latencies. However, even if the baseline would take 7.5 times longer time, the resulting latency would still be under the set safety requirements.

1.5 Contribution

This thesis shall contribute to the knowledge of time-critical communication in the automotive domain. Specifically, on how standard security measures in Ethernet can be applied to Automotive Ethernet, and how those security measures add to the latency in communication. Firstly, this thesis will provide the automotive security community with an extensive list of vulnerabilities in Automotive Ethernet, as there appear not to be one available to the public. Then the author will create a list of security measures that can mitigate those vulnerabilities. Lastly, the author will attempt to create an experiment to measure how a selection of those security measures affect latency over Automotive Ethernet and if those measures are fast enough to be under the requirements concerning latency that a real-time system needs in regard to safety. If those security measures are applied in the automotive domain, it could be the first step towards having Automotive Ethernet as the lone network in modern vehicles. If the time-critical communication can be delivered over Automotive securely, then any communication in vehicles should be able to be secured enough according to the standards of the Computer Security community.

Unlike existing research where Ethernet is only used as a backbone and confidentiality is considered necessary [5, 15, 16, 17], the primary focus of this thesis is to ensure frame integrity and frame authenticity. This is to minimize the amount of processing that needs to be conducted for the communication to keep the latency low.

1.6 Structure of Thesis

In Chapter 2, a little background is provided to some of the less known areas that are discussed in this thesis, primarily vehicle-related subjects.

In Chapter 3, , the project's literature study is presented. Firstly, the method of how papers and articles were found, and then what they contained before the found vulnerabilities are listed, and thus answering RQ1. At the end of Chapter 3, an identified gap is presented, to allow the reader to understand what needs to be researched more in the future and what additional contribution this thesis can provide.

Chapter 4 is a chapter that provides the reader with information regarding the method of selecting possible security measures to counteract the vulnerabilities found in the literature study. At the end of Chapter 4, a list of potential security measures is presented for the reader to understand how the threats in Automotive Ethernet may be countered, and thus answering RQ2.

The simulated experiment is presented in Chapter 5 where OMNeT++ is used to simulate a modern vehicle's critical domain and communicate between different subdomains with security measures. At the end of the chapter, the results of the simulation are presented in order to answer RQ3. There will be a selection of security measures from the security measures used to answer RQ2, and those selected will be used for the simulated experiment.

Analysis and Discussion are presented in Chapter 6 where the research question answers are discussed as well as the thesis' validation.

At the end of the thesis paper, the Conclusion and suggested Future work is presented in Chapter 7 and Chapter 8. In conclusion, the thesis is rounded off and answers whether the chosen security measures are possible to use in an automotive domain. In future work, the missing puzzle pieces are presented for other researchers to follow to improve the knowledge of the area further.

Chapter 2

Background

In this chapter, a short description of certain areas is being described, which will provide the reader with the required knowledge to understand the study that this thesis has conducted.

2.1 Electronic Control Unit

Electronic Control Unit (ECU) is an embedded device which in many ways can be compared to a digital computer. An ECU can be placed connected to any device in the vehicle, such as the engine, automated operations in the car, and passenger comfort systems. It can also keep a check on the performance of the vehicle, which can be used to warn the driver of malfunctioning devices [18].

The ECU is made up of hardware and firmware. The main components usually consist of a microcontroller or processing chip together with storage such as EPROM or Flash memory. The firmware is stored in the storage and can be updated [18].

Two of the most common ECUs in vehicles are Engine Control Module (ECM) and Electronic Brake Control Module (EBCM). The ECM handles processes such as fuel injections and ignition. The EBCM is used in the ABS [18].

2.2 Control Area Network

The most common networking technology used in today's vehicles is Control Area Network and was designed by Bosch in the 1980s. CAN is able to have multiplex communication between ECUs, which reduces the amount of wiring needed within the vehicle [4]. It is used in areas where strict response times for frames and timely delivery of commands are needed, e.g., engine and transmission control [7]. Despite that, CAN is used in areas unrelated to driving. Cars usually have multiple CAN networks - high-speed (500Kb/s – 1Mb/s) networks and low-speed (125–250 Kb/s). The high-speed CAN is used for more driving related data while the low-speed variants are mostly used for *body and comfort* functionality within the passenger compartment [4].

CAN is a serial communication protocol created to fulfill the needs of real-time control systems. It operates over twisted-pair copper wires and provides high fault tolerance and integrity of transmitted data. CAN is a multi-master bus where all

nodes can send broadcast messages and have no addressing. Instead, messages are addressed by the CAN Identifier (CAN ID). CAN ID is unique throughout the network and identifies the content of the frame and its priority. A lower CAN ID is directly translated to a higher priority because of how the physical medium is being used. A logical 1 bit will stop the transmission while a dominant logical 0 bit will continue transmitting [4]. This behavior causes lower IDs to be prioritized over higher values.

2.3 Local Interconnect Network

LIN is a cost-efficient automotive communication network used where low-speed communication is sufficient [19]. The low-cost is the result of simplicity in the communication controllers and the low requirements in computational power and memory [4]. LIN is a master-slave serial communication system with one master and several slaves on the same bus. Traffic is sent over a shared physical medium and with a low transmission rate of 20 Kb/s. It is most commonly used for comfort functions such as mirrors and seats [7].

2.4 Flexray

Flexray is a time-triggered protocol, available defined by ISO 17458 standard, which is used in several cars [4]. It is a deterministic, fault-tolerant, and high-speed bus among automotive communication networks. The protocol fulfills the requirements for x-by-wire applications such as drive-by-wire, steer-by-wire, and brake-by-wire [20]. Flexray uses a communication cycle that consists of static time-triggered windows using Time Division Multiple Access (TDMA) for each node and a dynamic window using Flexible Time Division Multiple Access (FTDMA) divided into several mini-slots [4].

2.5 Media Oriented Systems Transport

Media Oriented Systems Transport offers audio and video streaming capabilities and other applications such as Global Positioning System (GPS) GPS and entertainment systems. It was developed for automotive multimedia applications and consists of three versions providing different levels of bandwidth (25, 50 and 150 Mb/s) [4]. MOST has the advantage of having relatively high bandwidth for the automotive domain at a very high cost, and MOST will likely be among the first networks to be replaced by Ethernet [7].

2.6 Automotive Ethernet

Automotive Ethernet connects a vehicle's different components using a physical wired network. It meets the requirements in Electromagnetic Interference (EMI)/Radio

Frequency Interference (RFI) emissions and susceptibility and also provides the vehicle with higher bandwidths [7]. It contains components that ensure that latency requirements are fulfilled and that time synchronization is done on time [7].

Although Ethernet has been around for over 30 years, it has not applied to vehicles for more than a few years[7]. This is because Ethernet does not by itself meet the requirements in EMI/RFI because it has too much Radio Frequency Interference (RFI) "noise" and is easily manipulated by "noise" from other devices within the vehicle [7]. Other than that, Ethernet was unable to guarantee that latency was kept down in the microsecond range for point-to-point communication, which is problematic as the most critical functions of a vehicle are time-critical. Lastly, Ethernet did not include time synchronization and bandwidth allocation for different streams of data. All of these requirements had to be implemented for Ethernet to be applicable [7].

In Automotive Ethernet, there are physical implementations that provide the line of communication resistance against EMI/RFI, thus allowing Automotive Ethernet to fulfill the set Automotive EMI/RFI requirements [7].

2.6.1 Time-Triggered Ethernet (TTE)

Ethernet is a 'Best effort' protocol that has measures to avoid physical collisions of data communication. It does, however, not provide reliability and timely delivery. These two points make Ethernet a poor choice for the automotive domain. However, with the implemented measures added in Time-Triggered Ethernet, it is easily applied to the automotive domain. If regular Ethernet traffic is considered 'Best Effort' (BE) traffic, then Time-Triggered Ethernet has two additions compared to Ethernet.

Time-Triggered (TT)-traffic is used for communication that requires dedicated time slots and has high requirements for a timely delivery [10]. In the automotive domain, TT-traffic may, for example, be used for steering and sensor data. TT-traffic time slots are always reserved for TT-traffic, which ensures that TT-messages are delivered with as little latency as possible.

Rate-Constrained (RC)-traffic is designed for applications that require timely delivered traffic but is not continuous [10]. RC-traffic is always prioritized when there is no TT-traffic time slot. Thus, making RC-traffic prioritised over regular BE-traffic.

In addition to the two new traffic types in Time-Triggered-Ethernet, there are several other fault-tolerance implementations, e.g., time synchronization, scalability, multiple inconsistent faults, arbitrary end system failures, and self-stabilization [10].

2.7 OMNeT++

OMNeT++ is a modular, extensible C++ simulation library for discrete event simulation which is widely used by the academic community, and industrial settings as well [14, 4].

It is mainly created for network simulations and comes with an Eclipse based Integrated Development Environment (IDE) with a graphical run time environment. It runs on Windows, Mac OS X, Linux, and other UNIX-based operating systems [14]. Using OMNeT++, a programmer can simulate a network using Ethernet and run their own applications at endpoints by writing the C++ code required.

OMNeT++ can simulate all sorts of digital networks, including wired and wireless networks [14]. With additions, the library can easily simulate communication over Transport Control Protocol / Internet Protocol (TCP/IP) and real-time communication such as Automotive Ethernet [14]. By default, the library comes with the possibility to perform simple signal communication and uses a NED topology description language which allows for visual and precise topology designs [14]. It also has an interactive graphical user interface where the simulation is being run.

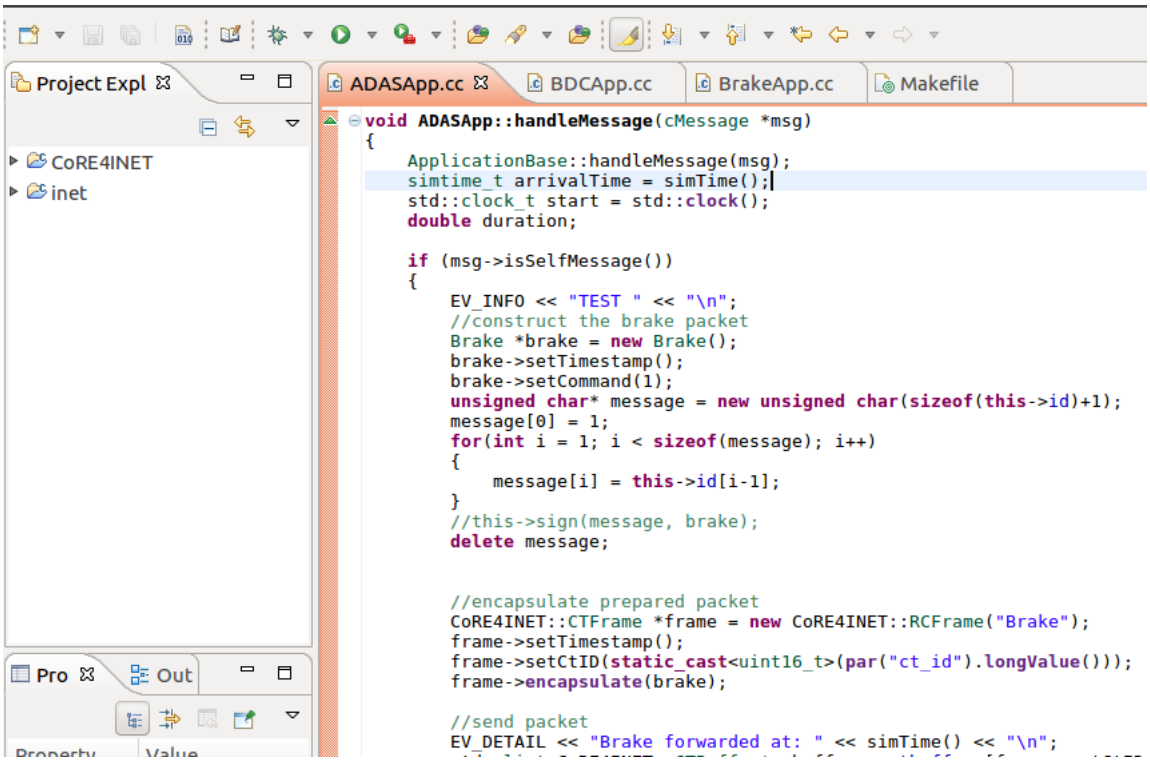

Figure 2.1: A screenshot of the development environment.

For this thesis, the additions INET framework and CoRE4INET is being used. INET allows for Ethernet, and TCP/IP communication and CoRE4INET adds additional functionality to INET such as real-time communication. If programmed correctly, any network can be implemented in OMNeT++ and any communication. In regard to this thesis, some basic examples are used as a foundation for Time-Triggered Ethernet (TTE) communication, which is a fundamental part of Automotive Ethernet. From those examples, the author then manufactures simulated ECUs that communicate using RC-frames. When a frame is received the program application for that specific device starts a timer, and then runs the C++ code that has been written for that module. When the C++ code is finished, the total duration for those calculations is added to the simulation time, which then allows for the ECU applications to modify the simulation time. This allows for the simulation to include any processing conducted in the ECUs and not only the communication latencies.

In addition to CoRE4INET, several other additions can be used to simulate vehicular communication [21]. For example, there are additions to implement CAN, Flexray and vehicular traffic simulators [21].

Chapter 3

Literature Study

This chapter describes the work that is connected to answering the research questions for this paper. This section will make it possible to answer RQ1 and make the foundation to answer RQ2. This thesis primarily tries to answer RQ3; however, for RQ3 to be answered, the research questions preceding it will need to have been answered. The research questions are answered using a literature study as a foundation that looks into the current problems with Automotive Ethernet and how Automotive Ethernet may be used in the future. In addition to a literature study, acquired knowledge in the area of cybersecurity, and further research will be used in complement to the literature study to answer RQ2 in the next chapter. In the following sections, a method is described concerning how information was gathered for the literature study. Later, the results of the study are presented as related work, which shows relevant work that has been conducted concerning Automotive Ethernet and Automotive Security. Lastly, in this chapter, a statement of the identified gap is found. The identified gap states what knowledge is missing or what needs to be further researched before this thesis. The following Chapter 5.7: Results will then use the related work as a foundation to answer RQ1 and RQ2.

3.1 Literature Study Method

This literature study is based on the generic six step method that describes the process of finding relevant articles and literature and extracting data [22]. As the work for this thesis has already stated research questions, the first step is therefore ignored. The second step is to search for literature that will be reviewed for relevance. This is being done by searching in literature databases using keywords. In this case, access to BTH Summon (Blekinge Institute of Technology library database) is used in combination with Google Scholar. Keywords used are, for example, *Automotive Ethernet*, *Security*, *Safety*, *Control Area Network*, *Vehicle*, *Car*, *Secure Communication* and *Time-Critical* in different combinations. To add additional sources of information, Google Search Engine was used to fill the gaps left by published papers. To get the most accurate information, primarily conference papers and journal articles were searched for during the study. Also, all papers had to be published in 2010 or later, as this thesis focus is so state-of-the-art.

According to the third step of the six-step method, once several sources have been located, the sources are then screened for inclusion [22]. Sources are eliminated in iterations, where the first iteration only looks at the papers abstract. If the abstract

is irrelevant for this thesis, then it is being excluded. The second iteration looks at the conclusion and results of the paper if their findings are interesting to this thesis. The third iteration looks at the method, to see if the way the results are acquired in a reasonable way. Looking at the quality of the paper is also a part of the six-step method, where step four is to assess the quality of primary studies [22]. After that, as step five, the remaining papers are fully read, and finally, the data is extracted from the included sources. Lastly, the data is being analyzed in step six [22].This method of information gathering may need several iterations as new questions or obscurity may arise.

In some cases, the primary sources references may be used to find more information using a method called backward snowballing [23]. If a primary source mentions something that is just briefly explained, their source may have a more extended explanation. For this thesis, this method is being applied, although not extensively as there are few references of sources used that are recent (2010 or later). As the literature study progressed, more keywords were added as the method of answering RQ3 became clearer. For example, the use of OMNeT++ was discovered in a primary source and was then further investigated as a possible way to simulate an automotive network. Keywords added in later iterations include *Time-Triggered Ethernet*, *Simulation*, *OMNeT++*, *MAC Security (MACsec)*.

3.2 Related Work

Ján Laštinec [4, 5] was kind enough to share his dissertation thesis, where he provides details regarding the current automotive domain and the rising security threats [4]. He describes that the transmission from the current in-vehicle networks to Automotive will occur in "3 generations". First, it will be used in Diagnosis over IP (DoIP) functionality, later it will be used for ADAS, and eventually, Ethernet will be implemented as a backbone [4].

In his analysis, J. Laštinec also investigates the current security in automotive networks, where matters such as the fact that CAN has no protection for unauthorized read, is vulnerable to denial of service attacks and does not ensure the authenticity of the messages is insecure [4]. These and several other vulnerabilities prove that CAN is not reliable whenever accessed by a malicious user. The new trend of connecting vehicles to the Internet is, therefore, a significant security issue as it may open up CAN to be reached from anywhere. In his summary of the analysis, he states that it is difficult to design new innovative functionalities for the current in-vehicle topology and that the current protocols have not been designed with security in mind [4].

Further, he describes the current state-of-art technologies in vehicles. He presents information regarding hardware, software, and communication protocols security technologies available. Those technologies include, for example, Hardware Security Module (HSM), CAN Intrusion Detection System (IDS), tunneling technologies.

For his own work, he has experimented with analyzing the delays of standard TCP/IP

protocols such as Internet Protocol Security (IPsec) and TLS in different varieties. The result from the experiment shows that TLS with an already established session was the fastest secure communication over a simple topology [4]. He also mentions the ongoing transition from the current topology to a more security-based topology with domains and gateways.

Lastly, he presents an experimental evaluation of the security measures from his experiment. Using the simulation environment OMNeT++, he performed communication tests over three different topologies. For this evaluation, he used One-to-One, Buffered, Timed, and Priority traffic and measured the delay. As a result, he explains that Ethernet/IP technologies are the most likely candidate for future in-vehicle networks and that IPsec using ah provides suitable performance and security properties for Automotive intra-vehicular communication [4].

Another research team from Germany have conducted a study on using TLS to secure in-vehicle networks [24]. They discuss four separate scenarios where communication requires some kind of security measure - Uni-Cast, Inter-Domain, External, and Multi-cast Communication. They suggest the use of certificates rather than TLS-PSK cipher suites as well as the implementation of HSMs to keep track of non-exportable private keys. In their experiment, they assume that all ECUs need a certificate, thus making the creation of public and private keys a drawn-out procedure.

There have been several researchers looking into the standard 802.1AE as a possible solution to deal with communication security in vehicles. 802.1AE is also known as MACsec and provides a similar kind of protection as IPsec on Layer 2 (Data-link) using tunneling technology. A research team from Italy and the United Kingdom have researched MACsec to provide security to the Ethernet backbone in a vehicle [16, 17]. Another research duo from Korea has researched a possible extension to the current MACsec standard that makes it end-to-end rather than host-to-host as it is by defaultn[15].

As described by both research groups, MACsec can provide confidentiality, integrity, and authenticity of data sent in the automotive network [16, 15]. However, as described by B. Carnevale et al. it does not guarantee protection against intrusions, but it can identify whether a message is allowed or not on the network, thus finding misbehavior [16].

In the case of a MACsec extension as described by J. Choi and Y. Han, they suggest a solution where an authentication server is being used to authenticate the communication allowing for the communication to be kept secure over the layer 2 switches without having to create new tunnels on each link [15]. In their experiment, they are using Raspberry pis as OVS switches and ECUs.

A relevant paper written by Chung-Wei Lin and Huafeng Yu [3] at Toyota discusses the increased need for security in current and future automotive Ethernet-based networks [3]. Ethernet allows for higher bandwidths, increased payload, and is a well-known standard. It does, however, bring specific problems regarding security. Ethernet was not designed to be secure, and therefore, there are four major problems, according to Lin and Yu's paper. These four threats are divided into four categories - Network access, attacks against traffic confidentiality, attacks against traffic integrity, Denial of Service [3].

In their paper, they suggest three solutions to mitigate those threats. Those solutions consider end-to-end latency and authenticity to be the primary safety and security requirements. Those solutions are secret key management, frame replication, and elimination, and lastly Virtual Local Area Network (VLAN) segmentation. Using their suggested solutions allows for efficient secret key handling, redundant lines of communication, and certainty of arrival. Lastly, VLAN segmentation separates communication over one standard switched network to separate VLANs. They conclude that in modern vehicles, security needs to be in the design along with safety because it is not possible to have safety without security [3]. So even if there is a possible competition between the two, cooperation is required to some extent.

In a second paper from 2016, H. Yu and C. Lin, present several security concerns for automotive communication [25]. . According to the researchers, there are still five open problems – Cross-Layer Security, Limited Resource, Hard design constraint, Plug-and-play Architecture, and unstable connection [25]. All of these issues are present in the time-critical domain in some way or another. These issues cause concerns to be of interest while researching security in the automotive domain.

A Finnish group of researchers conducted a comprehensive survey of Ethernet Security [26]. They present some general information regarding Ethernet and provide information about Ethernet today. Once they have set the foundation, they introduce the reader to the Ethernet threats where unauthorized joins, VLAN joins, and Man-in-the-middle attacks are among a few they mention [26]. Lastly, they provide the reader with existing security solutions to mitigate Ethernet security threats. These notes are interesting when researching vulnerabilities in Automotive as it complements the findings of H. Yu and C. Lin (which cites T. Kiravuo et al.) [26, 3].

There is another research team that looks at VLANs as a possible security measure for the automotive domain. F. Smirnov et al. have presented a paper on Automatic Optimization of the VLAN partitioning in Automotive communication networks [27]. Using their optimization, they reduce the latency and link load while also improving the VLAN separation of ECUs [27].

Christopher Corbett et al. [13] presents a paper where Automotive Ethernet is audited as a candidate for current and future inter-vehicle communication [13]. With new technologies such as autonomous vehicles and ADAS that require large amounts of bandwidth and are dependent on real-time communication [13]. They present Ethernet as a protocol with security issues. In the paper, they provide the reader

with explanations of the opportunities and challenges that Ethernet causes in the Automotive domain. See Table 3.1 for a description of opportunities and Table 3.2 for a short description of all challengers.

Table 3.1: Opportunities

Detection capabilities	The topology and services of an automotive network is very static which enables further detection possibilities.
Extended bandwidth	With greater bandwidth, better authentication and encryption can be used. Also software and configurations can be distributed faster.
Stricter separation	With the use of IEEE 802.1Q, a simple separation can be used to create VLANs which previously required its own bus. With the use of e.g. firewall concepts, communication paths can be restricted for the packets in the network.

Table 3.2: Challenges

Increased number of potential attacks	Ethernet is well-known, and is extensively described in literature and knowledge of the technology is widespread. Therefore the number of potential attackers (malicious or not) may cause problems in a Ethernet-based network.
Fewer resources	Compared to the regular environment Ethernet is used in today, vehicles are restricted due to lower processing power and memory compared to a modern computer.
Gained attention	Due to vehicles being used daily in very large numbers, there may be a number of commercially attracted interest groups that may intend to exploit software or hardware flaws.
Accessibility	Unlike larger networks, all devices are not distributed in different geographical locations. Whoever is in control of the vehicle have full access to the network infrastructure.
Security diversity	A vehicular network is a very heterogeneous system where most devices are unique in the network. Therefore it can be tricky to apply a mutual security mechanism for the whole system.
Safety compliance	Unlike a regular Ethernet network, Automotive Ethernet networks are safety compliant. It is important that applied security does not impact the safety of the vehicle as safety is prior to security in the vehicle domain.

They also discuss the combination of several security mechanisms to increase security in future vehicular Ethernet networks. New technologies that previously have not been applied to the automotive domain are used in an Ethernet-based network. Higher bandwidth and larger payloads of frames make security protocols more applicable, and anomaly detection can be used considering the static identity of the network and its services [13].

Philipp Mundhenk et al. [8] have been researching for a suitable lightweight authentication procedure that works in CAN-based networks [8]. They combine asymmetric and symmetric encryption algorithms to get secure communication. They propose the addition of a security model that will be used in three separate steps to enforce

secure communication [8]. First, the security module advertises its certificate to all devices, which then will be used to authenticate ECUs using asymmetric cryptography. Once the ECUs have been authenticated, they can become authorized for a stream. Stream authorization occurs via symmetric cryptography and once authorized for a stream; ECUs can send information between each other securely.

The suggestion includes the usage of Rivest-Shamir-Adleman cryptographic algorithm (RSA) and Advanced Encryption Standard 256 bits (AES256) for cryptographic algorithms. The latencies have been measured on an STM32 microcontroller using hardware implementations for encryption and by using the software. They present a table where different key lengths give different operations latency. Table 3.3 shows a summary of a table Mundhenk et al. use to present the latencies of RSA on the STM32 microcontroller [8]. The idea is that this processing time will be conducted either at the ignition of the vehicle or as they are required, depends on the latency requirements of the application.

Table 3.3: RSA latencies on STM32 micro controller

Key length	**Public Key Operations**	**Private Key operations**
512 bits	0.206s	0.886s
1024 bits	0.709s	4.977s
2048 bits	2.626s	33.181s

The stream authorization latencies are measured on different communication systems and compared. The setup consists of two ECUs with hardware support for symmetric cryptography and AES256. The worst communication system regarding stream authorization latencies was determined to be CAN (low-speed) with 2.43 ms, and the best one was Automotive Ethernet with 0.35 ms [8].

One of the first areas that started gaining interest in Automotive Ethernet was DoIP [9]. One part of the diagnosis is firmware updating, which takes a long time for devices in CAN-based networks. Varun Chellamuthu [9] has been looking at the implementation of Ethernet for diagnosis and a camera-based application over Ethernet [9]. According to the author, Automotive Ethernet is suited to act as the backbone of a vehicle's networks and to connect devices within domains in the vehicle. The results show that a CAN-based network can be migrated to an Ethernet-based approach [9].

A recent publication from the Holistic Approach to Improve Data Security (HoliSec) [12] project provides the automotive research domain with a reference architecture [12]. They state in their related work that the EVITA project that occurred during 2008-2011 is not a suitable architecture. This is because the current trend within the automotive industry is moving towards fewer more powerful nodes, which are not reflected in the EVITA project [12]. This is also supported by a publication on

IHS's website in 2017 where the author L. De Ambroggi states that automotive architectures are moving from a flat CAN-based network towards a domain-controller based Ethernet network [28], which is also supported by the reference architecture from D. Zelle et. al.s research [24]. The Authors A. Yadav and C. Sandberg propose an architecture where there is a distinct separation into different domains e.g., driving control, Vehicle-to-X (V2X), and engine/brakes. They include the external interfaces of a vehicle's communication technologies. Besides, to presenting their reference architecture, they provide the reader with some issues and threats that need to be managed, e.g., CAN priority denial of service, low bandwidths, and remote control via smartphones [12].

As late as January 2019, a paper was published about the recent advances and trends within the embedded systems within the automotive domain, where authors L. Lo Bello et al. describes the challenges for the leap from hands-on driving to Autonomous Driving (AD) [29]. One of the crucial things they mention in their paper is the development and implementation of Automotive Ethernet using for example Time-Triggered Ethernet. In addition, they also discuss security issues and countermeasures for in-vehicle network communications. In regards to security, Secure by design is mentioned as an essential matter. Currently, each LIN, CAN, or most controller can send messages to any other car controller [29]. In other words, malicious users can communicate with any controller in a vehicle as soon as they can send and read messages from a controller within the vehicle.

The authors mention a list of security countermeasures that may, in the future, prevent the malicious acts of malware and hackers. The list includes - Cluster subnetworks into security islands (domains), Embed HSMs to improve the security capabilities of current and new controllers in the vehicle, Use signature mechanisms for controller authentication in all new units, Cluster different ECUs into classes depending on how easily they can be attacked, Implement intrusion detection mechanisms, Implement gateway firewalls [29]. These items need to be taken into account when researching the area of Automotive Ethernet.

3.3 Vulnerabilities found

According to the research of T. Kiravuo et al., there is a large amount of vulnerabilities of Ethernet, which is then inherited by Automotive Ethernet [26]. They describe the threats toward Ethernet as four primary categories, *Network and System Access*, *Traffic Confidentiality*, *Traffic Integrity* and *Denial of Service* [26]. In addition to T. Kiravuo et al. research, the following sub-sections will describe the found vulnerabilities and how they could be used to cause harm. Note that all these vulnerabilities are for Ethernet out-of-the-box.

3.3.1 Network and System Access

For any attack, network access is a requirement. In Ethernet, this can be achieved, for example, by connecting devices to the network or gaining control over existing devices

[26]. This primary vulnerability category includes eight different vulnerabilities [26]. Those are as following:

Unauthorized Joins

Because of the easy-to-deploy design that Ethernet has been designed by, anyone can connect to an unconnected port on a switch. Meaning that anyone with access to the networks switches via physical access, via an available wall socket or via connecting new device between or instead of a current network device will have access to the network and can communicate [26].

Unauthorized Expansion of the Network

Because of the design of Ethernet, it allows users to expand the network easily by connecting a switch or access point of their own to a current network access port [26]. Thus, allowing other people to connect to the network.

VLAN Join

The use of VLANs can be used to separate devices into different Virtual Local Area Networks. Moreover, if a switch listens for management protocols for VLAN, then a host can act like a switch allowing it to join all VLANs [26].

VLAN Tagging and Hopping

There are methods where a user may create frames or packets to bypass VLAN separation. Either by using the *Double Tagging* technique or by sending a packet via a layer 3 device i.e., router [26]. *Double Tagging* implies that the sender of a frame sets the VLAN-id of the frame to the other VLAN and then puts that first frame into a second frame with the current VLAN-id. As the frame is being sent over the switched network, the first switch will strip off the senders VLAN frame, causing the first VLAN frame to be sent as if a device sent it on the correct VLAN. VLAN hopping includes simply sending a packet to a device on another VLAN via a layer 3 device and then having it routed to that VLAN.

Remote Access to the LAN

A malicious user may use methods such as social engineering to get access to the Ethernet device from a higher layer, allowing the malicious user to send and receive frames on the network remotely [26].

Topology and Vulnerability Discovery

When a user has access to the network, they can scan the network to detect its topology and possible vulnerabilities [26].

Break-ins

Ethernet can be used as a medium to perform attacks towards other devices [26]. Commonly, higher layer application vulnerabilities are used to perform the attack, which can lead to the attacked device either be malfunctioning or being under the malicious user's control.

Switch Control

Switches are often shipped with either no credentials or default passwords. If an attacker manages to get control of the switch, traffic can be managed by alter traffic routes, controlling the Spanning Tree Protocol domain or shutting down a link [26].

3.3.2 Traffic Confidentiality

By default, there is no protection against unauthorized read in Ethernet networks [26]. Any frames that are sent to a device is by default readable, and there are methods where the switch is forced to broadcast all traffic on the network, causing the malicious user to be able to read all traffic sent on the network [26].

3.3.3 Traffic Integrity

There is also no protection against loss of traffic integrity by default in Ethernet networks [26]. There are four primary threats to traffic integrity.

3.3.4 Frame Availability

Ethernet is a best effort protocol, meaning that the frame will potentially be delivered as soon as possible, and in some scenarios, that may be never [7].

3.3.5 Frame Integrity

In Ethernet, there is no default verification of frame integrity, meaning that interference, malicious users, or malfunctioning devices may cause frames to change from its intended structure.

3.3.6 Frame Authenticity

When an out-of-the-box Ethernet message is being sent, there is no reliability concerning frame authenticity. This leads to the receiver not knowing if the frame originated from the source IP address or if it was an unknown sender using a spoofed IP address.

Address Resolution Protocol (ARP) and Dynamic Host Configuration Protocol (DHCP) Poisoning

ARP-poisoning is conducted by a malicious user sending an ARP-reply to the sender of a frame using the intended receiver's IP address and with the attackers Media

Access Control (MAC) address [26]. Thus, allowing the attacker to read packets intended for another device. Similarly, the attacker can detect DHCP broadcast requests for IP address' and reply faster than the DHCP server. This allows for the attacker to chose it's own preference of gateway and Dynamic Host Service (DNS) server, making it possible for the attacker to control the victims traffic as they please [26].

Man-in-the-Middle

Man-in-the-Middle (MITM) attacks are simply attacks where all traffic between two hosts are sent via the attacker's device. This can, for example, be achieved by performing a double ARP-poisoning attack where the two devices are made to believe that their counterpart is located at the attacker's MAC address [26]. MITM attackers can potentially alter packets in the traffic between the victims.

Session Hijacking

Commonly, once a session has been established, no additional authentication is performed. This makes it possible for malicious users to hijack a session if the attacker have access to read the packets between devices [26].

Replay

By default, there is no protection against replay attacks in Ethernet [26]. A replay attack uses already sent frames to have an effect when sending them again. For replay attacks, the attacker does not need to be authenticated or know the encryption key, as long as the message is still valid, it will be received and processed by the receiver.

3.3.7 Denial of Service (DoS)

There are two primary types of DoS attacks, *Resource Exhaustion Attacks* and *Protocol Based DoS* [26]. There is by default no protection against DoS attacks in Ethernet [26].

Resource Exhaustion Attacks

Resource Exhaustion Attacks are for example when a target is being flooded with too many frames, causing frames to stress the processor and/or fill the RAM, which causes the target device to be overloaded and denied to perform its actual task [26].

Protocol Based DoS

Spanning Tree Protocol (STP) switches can be flooded with STP control messages, causing the network tree to be brought to a halt in regards to communication [26].

3.4 Results

The results from the literature study can be used to answer RQ1. Section 3.3 contain a list of numerous security vulnerabilities in Automotive Ethernet out-of-the-box, and with it, we have answered RQ1. These threats need to be mitigated to avoid any potential dangers to human lives and property. There may be several countermeasures that could mitigate those threats, which will be investigated in Chapter 4 in order to answer RQ2.

3.5 Identified Gap

Several sources point toward Ethernet being the future standard for Automotive communication [5, 4, 15, 16, 17, 3, 13, 9, 29]. However, they are primarily focusing on the nearest future and cases where Automotive Ethernet is used as a backbone, for infotainment systems, and ADAS. This, however, leaves a gap regarding whether Automotive Ethernet is applicable in the time-critical domains. For example, for the braking and steering domains. There is a limited amount of research looking specifically at the most time-critical aspects of automotive communication, and in most cases, the research only looks at Automotive Ethernet as a backbone. [5, 4, 15, 16, 17, 9, 29]. Therefore, a study should be conducted where the assumption that Automotive Ethernet is used entirely throughout the vehicle, and see if using standard TCP/IP security measures may be used to provide sufficient security for a time-critical system where safety is highly regarded.

For this thesis, the main focus is to answer which security measures would be sufficient in regards to integrity and authenticity in a scenario where Automotive Ethernet is used throughout the whole vehicle. The reason why integrity and authenticity are primarily focused upon is that in the time-critical domain, unauthorized read is not the primary concern. Instead, unauthorized write is considered the most critical issue, where an unauthorized program or device may send malicious commands to devices in the vehicle which may cause fatal incidents. This thesis aims to find an answer to this by answering RQ3 in Section 1.2: Research Questions.

Chapter 4

Countermeasures for Vulnerabilities found in Automotive Ethernet

As presented in Chapter 3.3, there are numerous threats related to Ethernet. Those vulnerabilities have been inherited by Automotive Ethernet and thus cause a critical issue for the implementation of Automotive Ethernet in modern vehicles. To avoid any potential malicious attacks towards ECUs, different countermeasures will be required. In the following sections, the method of how countermeasures have been identified will be presented, and then each vulnerability will be listed with potential countermeasures.

4.1 Method of Finding Potential Countermeasures

In order to identify countermeasures for the found vulnerabilities from Section 3.3, the learned skills during the education will be used in addition to the found references. In order to find references regarding countermeasures, the literature study in Chapter 3 will be used in combination with additional searching using the method from the literature study.

When selecting possible security measures as countermeasures to mitigate vulnerabilities, the main criteria is that the measure prevents or complicates an attack using said vulnerability. For example, encryption does not necessarily prevent unauthorized read. There could be several reasons why someone unauthorized was able to read the message.

4.2 Potential Security Measures to Mitigate Ethernet Vulnerabilities

After analyzing potential security measures, a list of suggested possible security measures is listed below. Note that specific countermeasures may be applicable for certain threats and that some threats may require specific implementations of a security measure.

Security Threat	Potential Security Measures
Unauthorised Joins [26]	Switch Port Security, Require authentication to communicate.
Unauthorised Expansion of the Network [26]	Switch Port Security
VLAN Join [26]	Switch port access mode.
VLAN Tagging and Hopping [26]	Switch port access mode, Disable routing between VLANs.
Remote Access to the LAN [26]	Security training, Anti-malware, Firewall
Topology and Vulnerability Discovery [26]	Disable ICMP
Break-Ins [26]	VLAN, Access Control List (ACL), IDS/Intrusion Prevention System (IPS)
Switch Control [26]	Switch port access mode, Change/set default passwords to secure password, Ignore Bridge Protection Data Unit (BPDU) on non-switch ports
Unauthorised Read [26]	Encrypted data
ARP and DHCP Poisoning [26]	IP Router[26], Static ARP [30], DHCP Snooping [31], Firewalls [30]
Man-in-the-Middle [26]	Static ARP, Virtual Private Network (VPN), Authenticated Communication.
Session Hijacking [26]	TLS[32], Random Session Key [32], Renew Session key often [32].
Replay [26]	Session keys, One-Time-Pad encryption for each message.
Resource Exhaustion Attacks [26]	Software security, IDS, /IPS
Protocol Based DoS [26]	Ignore BPDU on non-switch ports, Switch Port Security, Switch Port Access Mode.

Table 4.1: Potential countermeasures against the security threats found in the literature study.

These and potentially more security measures are possible countermeasures for known vulnerabilities in Automotive Ethernet by default. However, there are also safety vulnerabilities that need to be taken into account. As stated in Chapter 3: Literature Study, although Automotive Ethernet may resolve many of the physical challenges that Ethernet had, it does not solve all issues related to safety.

4.3 Potential Safety Measures to Mitigate Ethernet Vulnerabilities

There are potential safety threats that can be resolved by security measures. For example, without redundancy, there is a risk that potential critical messages are not delivered at all [3]. If one switch between two critical ECUs malfunction during driving, then it could lead to catastrophic results. To avoid this, measures related to safety needs to be looked upon. Moreover, because it is an Ethernet network, and there are solutions for this that relate to security, it will also be regarded as necessary in order to secure Automotive Ethernet for usage in modern vehicles. Note that these threats are related to Ethernet, but not often considered because most Ethernet communication does not rely on timely delivery.

Safety Threat	**Potential Safety Measures**
Frame Availability [3]	Multiple redundant paths.
Frame Integrity [3]	Multiple Frames to Compare, Hash Checksum.
Frame Authenticity [3]	Authenticated Communication, Session Keys, Digital Signatures, Message Authentication Code (MAC) (not to be confused with MAC), e.g. HMAC

Table 4.2: Potential countermeasures against the security threats found in the literature study.

4.4 Potential Mitigation Measures Results

As seen in Table 4.1 in the section above, there are several occurrences of Switch Port Security and Switch Port Access Mode. Switch Port Security is when a switch has configurations on its ports to protect the network. It can, for example, be by having a static MAC-address or a *sticky* MAC-address assigned to the port. Switch Port mode security measures where each switch port is configured to either be in Access mode or Trunk mode, where Access mode only allows one specified VLAN on that switch port, while trunk mode allows any VLAN. This may suggest that layer 2 security measures will be very relevant to solve some of the issues of Ethernet, which makes sense as Ethernet is a layer 2 protocol.

Another reoccurring security measure is Firewalls, which is a powerful network security tool. Firewalls can be used to prevent or allow specified traffic depending on IP-addresses, port, state of communication, and more. Firewalls can be located on both lower levels and upper levels, and thus look at values, stateful communication, and application data to prevent breaches.

Something else that is being brought up several times in Table 4.1 and also in Table 4.2 in the previous section is session keys. A session key is a key that is only used for a limited amount of time and is often negotiated using asymmetric cryptography at an earlier stage of communication. In addition, authenticated communication is mentioned, which can also be done using this mentioned technique.

Lastly, Hashsums, Digital Signatures and Message Authentication Codes are mentioned in Table 4.2. As stated in Chapter 1: Introduction, message integrity and authenticity are considered the most critical requirements for this thesis.

Chapter 5

Finding suitable security measures using a simulation

In the previous chapter, several security measures were found to mitigate the identified vulnerabilities. In this chapter, a few of those measures are selected and tried using an experiment conducted through simulation. This chapter will, therefore, describe why those measures were chosen. The methodology described in this Chapter will produce results that aim to answer RQ3 from Section 1.2. In order to do so, details regarding automotive topologies, low latency security measures, and simulation environment need to be investigated in order to create a potential valid answer for RQ3. Much of the method described in this chapter uses the knowledge gathered during the literature study presented in Chapter related: Literature Study. The results of the simulation will be presented at the end of this chapter in Section 5.7.

5.1 Selection of Security Measures

As a precondition to the selection of security measures, the assumption is made that all necessary key negotiations and key generations have been conducted at an earlier point of the communication, i.e., session keys, public keys, and private keys are generated and established at vehicle ignition before the selected security measures are being used, thus reducing heavy calculations at critical time deadlines. In other words, algorithms that are being chosen does not have to keep low latency for all calculations; only the ones being conducted *on-the-fly* while the vehicle is mobile. This also allows for asymmetric cryptography functions to be more viable, as they require a large number of calculations to, for example, generate keys.

For this thesis, the scope is limited to the time-critical domains e.g. steering, braking and ADAS. In addition to this, because of the nature of the time-critical domains, the main priority is message authenticity and message integrity. These can be achieved by several security measures as described previously.

5.1.1 Authenticity

Authenticity can be ensured by the different parties being able to send messages that prove the party to be authenticated. This can be achieved in several ways using cryptographic functionalities, for example, message encryption or digital signatures attached to the message. Using a shared key, both parties can encrypt and decrypt

messages, and as long as the key has not been compromised, both parties are authenticated and may send authentic messages. For digital signatures, a combination of a private and public key is being used in addition to a hash sum of the message. The sender creates a hash sum of the message and signs it with their private key, and then the receiver verifies that hash sum using the senders public key. As long as the public key of the sender is trusted, a message where the verification process is passed can be considered authentic.

The significant difference between encryption and digital signatures is the fact that encryption also protects an unauthorized read of messages. However, it comes with the drawback of having a shared key that cannot be compromised. A combination of the two methods can be found in HMACs where a shared key is being used in combination with a hash to provide authenticity.

5.1.2 Integrity

Hash sums are a standard method to ensure data integrity. Message integrity can, therefore, be ensured by the use of hashes, which can be compared to a hash of the message data. If the two hashes are a match, the data is most likely not altered. With more advanced hash functionalities, the risk of a collision is reduced. A collision is when two different inputs give the same hash.

5.1.3 The final candidates

Because both message authenticity and integrity is critical for the messages being sent, security measures based on hashes in combination with some authentication of the message is the best match. HMAC and Digital Signatures appear as the best choices using the set criteria. Out of those categories, there are a vast amount of different functions to choose from.

The list below contains a variety of different cryptographic functions to use in the two categories chosen above. HMAC is a symmetric cryptographic function and is, therefore, faster to calculate, but has the drawback of having to keep a shared key secret. Digital Signature Algorithm (DSA) uses asymmetric cryptographic functions which are slower than symmetric but removes the need for a shared key as it uses private and public keys.

HMAC	Digital Signatures
HMAC-SHA-0	RSA 512-bits
HMAC-SHA-1	RSA 1024-bits
HMAC-SHA-224	RSA 2048-bits
HMAC-SHA-256	RSA 3072-bits
HMAC-SHA-384	ECDSA 256-bits
HMAC-SHA-512	ECDSA 384-bits
HMAC-SHA3-224	
HMAC-SHA3-256	
HMAC-SHA3-384	
HMAC-SHA3-512	

From the list above, there is a large number of options. Out of these, two will be used in the simulation to try to answer RQ3. Out of comparability, one HMAC and one Digital Signature will be selected. Therefore, the two types will be selected separately.

However, there are a few criteria that both security measures must follow. Because this thesis focuses on the latency of the communication, the security measures must provide a minimal amount of additional latency, but still be considered safe from the perspective of the digital security community. As previously expressed, time-critical traffic needs to be delivered within 10 ms [5]. Moreover, regarding cryptographic functions security guidelines, we follow the NIST recommendations [33, 34, 35].

Digital Signature

The two primary choices of Digital signatures come from either digital signatures based on RSA or Elliptic Curve Digital Signature Algorithm (ECDSA) based on Elliptic Curve Cryptography. Both signature types use a hash function to create a hash which is used as the signature. Variants may range from DSA-SHA-1 to ECDSA-SHA-512.

Both types are based on the principle that it is easy to perform either public key operations or private key operations and incredibly hard to perform the other task. For RSA, private key operations take vastly longer time than public key operations, meaning that signing a message takes longer time than verifying a message [8, 24]. For Elliptic Curve Cryptography, the private key operations take a short amount of time, and the public key operations take considerably longer time [24].

To make an educated choice between the two types require knowledge of how processing power and primary memory looks in the automotive domain. Because there is a low amount of processing power and memory compared to a typical computer network, ECDSA looks promising for the automotive domain [24]. This is based on the fact that ECDSA provides the same amount of security on a smaller key size and require fewer heavy calculations than RSA.

Finally, ECDSA comes in some different versions using different curves. For this thesis simulation, the lightweight cryptographic library Nettle TLS is being used.

Also, Nettle comes with a prepared curve, namely ED25519. It has a signature length of 512 bits and uses 256-bit keys.

Hash-based Message Authentication Code (HMAC)

HMAC comes in many different types, many of which are not mentioned in the list above. However, the SHA family of hash functions are among the most commonly used ones. According to recommendations from NIST, SHA2-224, -256, -384 and -512 are all considered safe, and the same can be said for fixed-length SHA3 hash sums [33]. In addition to that, because SHA hash functions are relatively quick and do not differ vastly in latency between versions, all of them can still be considered for the final selection. With that in mind, there is a large number of options for the HMAC security measure. However, for comparability possibilities, the hash used should be the same length as in the ECDSA measure described in the previous section. This leads to the options of HMAC-SHA-256 or HMAC-SHA3-256. Also, if the hash function used in the Nettle TLS library should be taken into consideration, HMAC-SHA-256 is a clear choice. Moreover, with that, both security measures have been decided.

5.2 Creating the Topology

Unlike conventional computer networks, the automotive domain has static networks that most often does not change during the vehicle's lifespan. As described in the background of this thesis, there are currently four different network technologies in use in addition to the limited use of Ethernet. According to new studies, the current mesh topology that is often mentioned as the current topology is being phased out and a new star-like topology is becoming prominent, where functionalities are divided into domains [12, 36, 28, 37]. Because there is not much information published regarding this matter, and despite Combitech having contacts within the Automotive Domain, it was not possible to acquire any relevant information regarding modern topologies of vehicles. The published information, in addition to the domain knowledge of Combitech's Automotive Security Experts, will have to suffice to create a theoretical topology. In other words, there are no actual accurate topologies available which of the simulation can be based on. Instead, it was based on guidelines from research papers, and the domain knowledge of two Automotive Security Experts will be used to formulate a plausible topology which then will be used for the simulation. With that said, it is unclear if the created topology model would be accurate to a model used in an actual vehicle. However, without exact topologies or real guidelines, this topology is among the best approximations that can be done using the available data.

In combination with the divide into domains, it appears that fewer ECUs are being used and that they are more powerful in processing and memory, in addition to, gateways that will be used between these domains to split functionalities [12, 36, 29]. With that in mind, we assume that the network is divided into different domains, where for example, ADAS is one domain containing all the sensors and computa-

tional power used to determine actions to perform. If that is the case, then the ADAS domain will have a gateway connecting ADAS to other domains. For our suggested topology, we will call these domain gateways dc. Each domain has one Domain Controller (DC), which is then connected to other DCs. The dcs will be the primary processing unit within the domain and has direct connections to all ECUs within the domain. Direct links will ensure that the message comes from the correct source within the domain. These smaller ECUs can be for example a brake ECU that controls a physical brake appliance, or a sensor that measures the distance to objects in front of the vehicle.

These fundamental ideas are then combined with some secure by design input, i.e., separate critical devices from easy access devices, much like in a Demilitarized Zone (DMZ) in computer networking. The result will be a topology based on the trends within the automotive domain as well as some secure by design choices. Figure 5.1 shows the example of topology that has been created for this thesis. Primarily, the topology is divided into two domains, the Critical Domain, and the Backbone Domain. The Backbone Domain contains the telemetries unit, body domain, infotainment domain, and other non-critical domains and functions. The communication between domains can occur via separation using, for example, VLANs or using authentication and encryption. The Backbone Domain, however, is not of significance for this thesis as only critical communication is of interest.

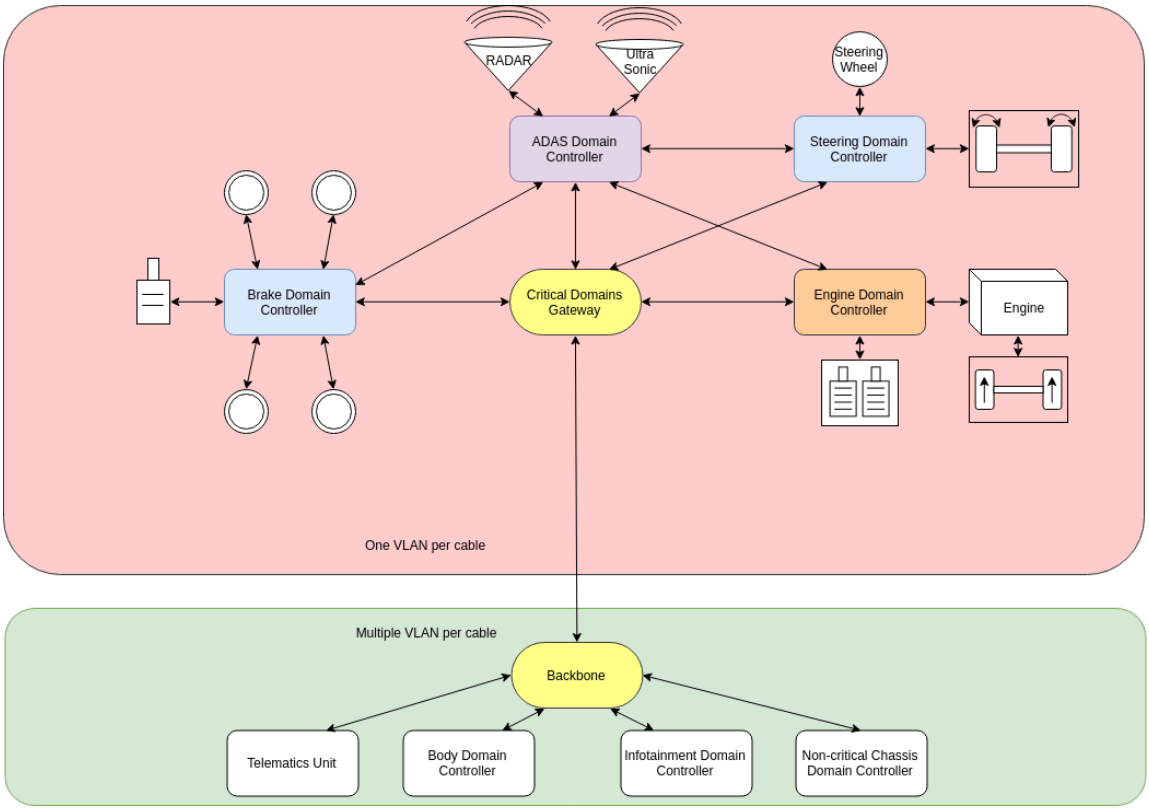

Figure 5.1: Approximation of automotive network topology of a car.

In the Critical domain, there are several sub-domains with a DC each connected and via a CDG. This gateway is the connection between the Critical Domain and

the rest of the vehicle's networks. The gateway is also used as a redundancy if for some reason the message is not delivered via the dedicated cables from one DC to another. In Figure 5.1, direct connections are shown between different domains, but there might be Automotive Ethernet switches in between, which allows for even more redundancy but also opens up a risk regarding the authentic origin of messages. However, the security measures mentioned in Section 5.1 should countermeasure that risk.

Unlike in the Backbone Domain, each cable used only contains one VLAN. This physical separation should provide some additional security from malicious actors. Within the sub-domains, i.e., Brake Domain, each device is directly connected to the DC. This should be sufficient for the DC to trust the information that is being received on that network port. However, as soon as the information leaves the domain, it needs to use security measures to prove the authenticity and integrity of messages. Another suggestion for this topology is that communication from the Backbone Domain is shut off thoroughly using a data diode. Allowing the Critical Domain to communicate with, for example, the infotainment system informing the driver of a malfunction or other general information, but prevents, for example, a connected Bluetooth device to access the Critical Domain. This suggestion causes a critical problem though, which is the ability to update critical ECUs. This can be solved by putting the vehicle in some *maintenance mode* that allows for information to flow from the oem to the ECUs via a Virtual Private Network (VPN) tunnel. While in this mode, the vehicle must be shut off so firmware updates can occur without disturbance and at the same time, no other communication may occur in the vehicle other than firmware from the VPN.

5.3 Preparing the tools

For the simulation, a few tools are necessary for it to function as intended. Firstly, the simulation must be conducted in an environment that can be used to simulate real-time scenarios. According to the papers of some researches, OMNeT++ has been used to do this in the automotive security domain [5, 4, 29]. For this simulation, OMNeT++ was downloaded as version 5.1.1. Also, in addition to the OMNeT++ library, INET and CoRE4INET is used as well on version 3.5 and 2018-02-03_00_00_08 respectively.

The OMNeT++ environment was downloaded and installed on an Ubuntu 16.04 Virtual machine with 3 GB primary memory and four virtual cores at 4 GHz. In this same environment, the security measures were run using a regular C++ program compiled with the compiler G++. The security measures use the lightweight cryptographic library called Nettle TLS. For this simulation version, 3.4 have been downloaded to enable the use of ECDSA and HMAC-SHA-512 to ensure the authenticity of messages.

5.3.1 Test environment

Because of the lack of hardware, these tests have to be conducted using a software environment. For this, a virtual machine environment has been selected using the Ubuntu 16.04 operative system. In this virtual machine, the OMNeT++ simulation environment and the CLI security measures are implemented. The virtual machine is allocated with 3 Gigabytes of RAM and four virtual cores at 4GHz each. Thus, making the environment powerful in terms of processing power compared to processors found in vehicles today [38].

According to a report from 2011 regarding one of Toyota's ECUs, the V850E1 microcontroller is one of the processing units that are being used in Toyota's vehicles [38]. This unit is said to have a processing frequency of about 66-100 MHz according to a page on Wikipedia [39]. However, for this study, we are primarily interested in domain controllers processing power. Domain controllers are cutting edge and require more processing capacity. NXP offers S32V234, a processor designed explicitly for ADAS processing and has a quad-core 1GHz processor specially designed to process heavy calculations and image processing [40].

Because the processing requirements are set higher for ADASDC and BDC, it is possible to assume that they would be implemented using a processor similar to the S32V234 processor and thus have a processor frequency of approximately 1GHz. The results created by a 4GHz processor will be adapted to what a 1GHz processor would perform, i.e., multiply any latency produced from the security measures by a factor of 4. This leads to security measures time from ADASDC and BDC being multiplied by 4. The presented results regarding security measures will have this factor already taken into account in Appendix B and Appendix C.

5.4 Setting up the Scenario

In order for a simulation to take place in the suggested topology, a scenario is needed to be crafted. The scenario needs to involve several Critical Sub-domains and simultaneously have a low number of ECUs involved to simplify the setup of the simulation. To match both of those criteria, an emergency brake procedure was selected which is an automatic emergency brake needed when the distance to an object in the path of the current vehicle has a drastically lower speed (or is stationary) and where applying the brakes should prevent impact. It is initiated by the ADAS domain which then sends a message to the brake domain. This scenario will be described in detail in this section.

Firstly, the devices needed for the scenario needs to be identified. Both ADASDC and BDC will need to be used as both those sub-domains are used for the scenario. In addition to the DCs, brake ECUs which control the actual braking appliance need to be included, as well as the sensor which will be used to measure distances. The topology in Figure 5.2 shows the topology which will be simulated, and it contains all the necessary devices for the scenario.

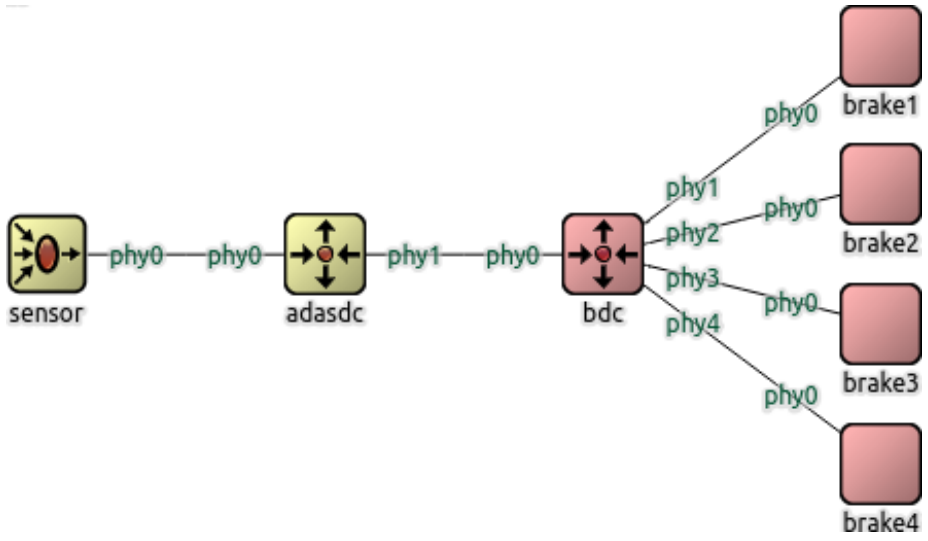

Figure 5.2: Small simulation topology taken from OMNeT++ Integrated Development Environment. Note that phyX is interface name for the module it is connected to. Every phyX<->phyX connection here is a copper wire connecting physical interfaces together.

The following segments will describe the role of all these devices during the scenario and how the communication will occur.

Sensor

The sensor ECU measures distance with intervals. The distances are continuously sent via point-to-point cables to the ADASDC, which then will process the information and take action if needed.

ADASDC

The ADASDC receives information from several sensors that use, for example, radar or ultrasonic. The data is then being processed to try to find different actions to perform. If the velocity of the vehicle is slightly higher than the vehicle located in front of the first vehicle, then the ADASDC will take action accordingly, for example by lowering the velocity of the vehicle by reducing the thrust or braking. In this scenario, the ADASDC will calculate the velocity of the object or another vehicle in front of the vehicle to determine if a collision is due. If a collision is due to happen, it will quickly calculate how soon such a collision may occur. If the only possible way to avoid a collision is to perform an emergency brake procedure, then the ADASDC will send an emergency brake message to the BDC. Emergency Brake procedure is when the vehicle reduces the velocity to zero in the fastest amount of time using the brakes. Before a message is sent to the BDC, the message is signed using the mentioned security measures to ensure message authenticity and integrity.

BDC

When the BDC receives an emergency brake message from the ADASDC, the BDC will verify the brake message using the previously mentioned security measures. If both the message integrity and message authenticity can be verified, then the BDC will react to the instruction by messaging all brakes over point-to-point cables to initiate an emergency brake procedure.

Brakes

Whenever a brake of the vehicle receives a braking instruction, it will see what kind of braking procedure is requested. In this case, an emergency brake procedure is requested meaning that all brakes should simultaneously reduce the velocity of the vehicle as promptly as possible. When this procedure is executed, the scenario is completed.

Different models

This scenario will be run using three different models. There will be one baseline without any security measures. The second and third model will consist of two different security measures implemented into the communication between the two DCs. When all the latency data is collected, it will be categorized into the baseline, HMAC, or ECDSA depending on its source. This is to make it easier to compare the data and to have the baseline by itself.

The security measures will only be applied between the ADASDC and the BDC. This is because, between those two devices, there might be switches and alternative routes to its destination, thus making it possible for other devices within the critical domain to be able to communicate incorrect or malicious traffic.

5.5 Setup of the Simulation

The following sections will describe what preparations were conducted in order to prepare the simulation. Primarily OMNeT++ and the security implementations need to be described.

5.5.1 OMNeT++

Once the scenario and the security measures have been decided. It is time to realize them using the tools specified in Section 5.3. In OMNeT++, there are pre-prepared examples that show how different implementation may look. When using the CoRE4INET addition, some examples use Time-Triggered Ethernet as a communication protocol. For this thesis, the Tic-Toc example was used as a foundation in creating the scenario. The Tic-Toc example has two devices, Tic and Toc. Each device has several code components attached to it; the following sections will describe how the example code was altered and why.

Applications are the piece of code that is being executed by the modules during the simulation, thus making it a vital part of the module's functionality. The Ned-file is a component mapping file that is used visually in the IDE. It shows how modules are connected. On the top level, Ned-files show the whole network being used. This view is presented in Figure 5.2 where the Sensor, ADASDC, BDC, and Brakes 1 through 4 is shown. By double-clicking any module, the user is sent to that modules Ned-file showing the different modules in that first module. See Figure ADASDC for the ned-file of ADASDC. Lastly, there are ini-files that are used to initiate the module, and it contains information such as physical addresses on interfaces, applications used, module connection/settings, and TT/RC-traffic settings.

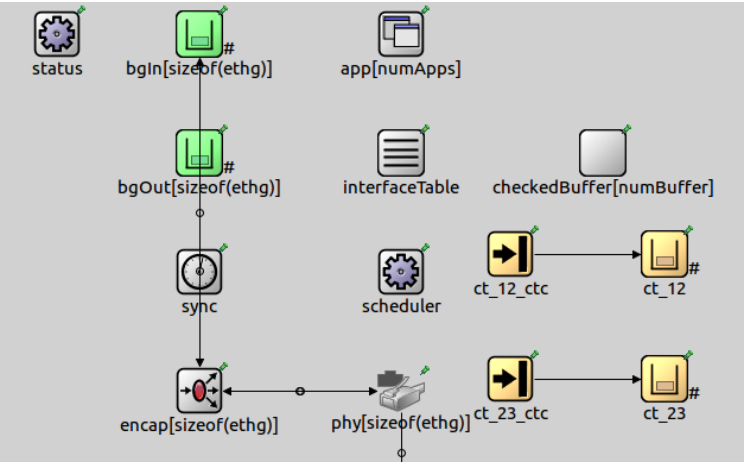

Figure 5.3: The Ned-file of ADASDC that shows the different modules in the higher level module.

Applications

```
void TicApp::handleMessage(cMessage *msg)
{
    ApplicationBase::handleMessage(msg);

    if (msg->arrivedOn("schedulerIn"))
    {
        Tic *tic = new Tic();
        tic->setTimestamp();
        tic->setRoundtrip_start(simTime());
        tic->setCount(static_cast<unsigned int>(par("counter").longValue()));
        CTFrame *frame = new TTFrame("Tic");
        frame->setTimestamp();
        frame->setCtID(static_cast<uint16_t>(par("ct_id").longValue()));
        frame->encapsulate(tic);

        EV_DETAIL << "Sending Tic Message\n";
        std::list<CTBuffer*> buffer = ctbuffers[frame->getCtID()];
        for (std::list<CTBuffer*>::const_iterator buf = buffer.begin(); buf != buffer.end(); ++buf)
        {
            Incoming* in = dynamic_cast<Incoming *>((*buf)->gate("in")->getPathStartGate()->getOwner());
            sendDirect(frame->dup(), in->gate("in"));
        }
        delete frame;
```

Figure 5.4: A snippet from OMNeT++ IDE showing the handleMessage function of the Tic module.

The Tic-Toc example has two applications, the TicApp, and the TocApp. These are bound to the module using the ini-files described later. These files are fundamental in the functionality of a module. At the beginning of every second, Tic sends a Critical Time Frame of the type Time-Triggered to the Toc module. Once the Toc module receives the message, it replies with a Critical Time Frame of the type Rate-Constraint. The Tic and Toc modules' handleMessage function is visible in Figure Tic and Figure Toc. If both modules were assigned the TocApp, neither would initiate the message bounce at all and nothing would happen forever.

```
void TocApp::handleMessage(cMessage *msg)
{
    ApplicationBase::handleMessage(msg);

    if (msg->arrivedOn("TTin"))
    {
        TTFrame *ttframe = dynamic_cast<TTFrame*>(msg);
        if (ttframe)
        {
            Tic *tic = dynamic_cast<Tic*>(ttframe->decapsulate());
            delete msg;
            bubble(tic->getRequest());
            par("counter").setLongValue(static_cast<long>(tic->getCount()));

            Toc *toc = new Toc();
            toc->setTimestamp();
            toc->setRoundtrip_start(tic->getRoundtrip_start());
            toc->setCount(tic->getCount() + 1);
            delete tic;

            CTFrame *frame = new RCFrame("Toc");
            frame->setTimestamp();
            frame->setCtID(static_cast<uint16_t>(par("ct_id").longValue()));
            frame->encapsulate(toc);

            EV_DETAIL << "Answering Tic Message with Toc Message\n";
            std::list<CTBuffer*> buffer = ctbuffers[frame->getCtID()];
            for (std::list<CTBuffer*>::const_iterator buf = buffer.begin(); buf != buffer.end(); ++buf)
            {
                Incoming* in = dynamic_cast<Incoming *>((*buf)->gate("in")->getPathStartGate()->getOwner());
                sendDirect(frame->dup(), in->gate("in"));
            }
            delete frame;
        }
        else{
            delete msg;
        }
    }
    else
    {
        delete msg;
    }
}
```

Figure 5.5: A snippet from OMNeT++ IDE showing the handleMessage function of the Toc module.

Using these two applications as a foundation, the scenario could be set up quickly.

By using TicApp as the foundation for the sensor application, it could send an RC-message on scheduled intervals and passing it forward to the next module, the ADASDC. The reason why RC-traffic was chosen instead of tt-traffic is based on a dialogue with an automotive security expert. The reasoning that because braking is performed when needed, not checked for in intervals. Thus, having RC-traffic making sense.

The ADASDC, BDC and the brakes are all based on TocApp. The ADASDC and BDC are in many ways identical in the simulation. They receive an RC-message on one of their ports and then process the message before sending it forward in the line of communication. The ADASDC passes it forward to the BDC, and the BDC passes four RC-messages forward, one to each brake in the vehicle. Lastly, the brake receives a message and processes it, but it does not forward it anywhere, at least not for this simulation, as the time that maTime-Triggered Ethernetrs is the time from the sensor to the actual braking process starts.

Each application that was created was called SensorApp or the corresponding for ADASDC, BDC, and brake. All brake ECUs use the same application as their functionality should be identical. In order to add the time of C++ functions, the OMNeT++ functions isSelfMessage() and scheduleAt() are used to add the forwarding message at a later time to allow the C++ code to execute.

In order for the simulation to accurately simulate vehicular communication over Automotive Ethernet, one must know how the firmware functions in the ECUs. However, since it proved impossible to get any information from the OEMs, certain assumptions had to be done. Because it is unknown how much processing power or code to be executed, each device has, a particular assumption has been made to simulate the time for each unit.

It is assumed that a processor with lower clock frequency will not be able to process the same amount of instructions as a processor with higher clock frequency. The Domain Controllers will have a vast amount of instructions that are being computed, making them more complex and have a higher requirement regarding clock frequency than, e.g., the brakes. Without any actual knowledge of the instructions executed for smaller ECUs, e.g., brakes, or gateway devices such as dcs, it is impossible to know precisely how much latency is caused by the execution of instructions. It is assumed that the amount of code in the Brake ECUs are significantly lower than the complex system that is ADAS. To make an estimation, specific C++ code was added between the reception of a message and the departure of the next after being processed. The example code used for this experiment is presented in Listing 4.1.

```
1 // Code inserted to increase latency in simulation
2
3 for(int i = 0; i < 100000; i++)
4 {
5   int k = i + i;
6 }
```

Listing 5.1: Added code to simulate latency cause by instruction processing

The reasoning why all units have the same amount of iterations in the loop is because the domain controllers have a higher clock frequency (1 GHz) while the brake ECUs have a much lower clock frequency (66 MHz). Thus, the baseline latency caused by processing in each device should roughly be similar based on the assumption that the ADASDC has to process roughly 15 times as many processing instructions compared to the brakes. This, however, may be a relevant factor that can cause incorrect values in the results because of the lack of knowledge regarding ECU firmware, so it is not possible to find more accurately without having information from inside vehicular companies. Besides, even if the correct firmware was available in the devices, it is not possible to down-clock specific devices in the simulation.

The different timestamps are then collected manually from the simulation software, and the measurements are normalized, so the time when the sensor sends the initial message is t=0.00000 seconds. This ensures that the data can easily be compared and used for statistical calculations to find standard deviation and calculate a confidence interval with a high probability concerning the total time of communication. Because the timestamps are collected manually, the baseline is limited to 50 iterations using OMNeT++, as it will still provide meaningful results.

Once the timestamps are collected, and the same statistical calculations are done as for the baseline, the worst, best, and average times will then be found. This will be done using the worst (most prolonged latency) baseline combined with the worst of each security measure, the best baseline with best of each security measure and then finally the total average for each measure added to the baseline.

Ned files

Ned-files are used in OMNeT++ to describe which low-level modules are a part of the current module. For example, the simulation may be considered one module, and it contains seven modules in this simulation - Sensor, ADASDC, BDC, Brake1, Brake2, Brake3, and Brake4. The Ned-file describes their connection to each other and may contain visual aspects of the module topology, as seen in Figure 5.3 and Figure 5.2. For this thesis' simulation, the Ned-files from the Tic-Toc example did not undergo any significant changes, except for ADASDC and BDC that were given more interfaces to manage their assigned task with dedicated links

Ini files

The ini-files are used for the configuration of modules. They are used to set which applications are used by one module and other settings such as network addresses and Time-Triggered Ethernet settings. In this thesis simulation, none of the TT-traffic settings are needed. However, all modules need RC-traffic settings to allow them to communicate, and for all modules, physical addresses need to be set.

5.5.2 Security Implementations using Nettle TLS

Because of a considerable problem that arose during the thesis work, it was not possible to implement the security measures directly into the simulation. However, together with supervisor Martin Boldt, the decision was made to perform the security measures separately and then adding that time together with the baseline time, which was achievable via the OMNeT++ simulation.

The implementations were done using the library Nettle TLS, which is a C-based cryptographic library with a vast amount of cryptographic functions. It was chosen based on the fact that it had predefined functions for HMAC-SHA and for ECDSA. It was also chosen for the fact that it is a low-level library designed to fit in more or less any context.

Both HMAC-SHA-512 and ECDSA - ED25519 were implemented in the same CLI executable, as it made a more structured program rather than having the two separately. The program consists of three functions with different tasks. The primary function consists of a loop where the iterations take place and for every iteration calls the function test. The function test contains the majority of the program, such as key generation, context setup, and actual security actions.

Firstly, the test function reads the message from the start parameters and stores it in a uint8_t array. Then creates similar arrays for the public key and private key for ECDSA and for the signature and fills the key arrays with a public key generated from a private key. Once the keys are in order, a time variable of the ctime library is set to current time before the ECDSA signing is performed. Once the signing is performed, another time variable is stored, and the difference is calculated. Similarly, to how the time was measured for signing, the same is done for verification. If the verification is successful, it is printed to the console in addition to the latency of both actions. Then the same is done for HMAC-SHA-512. When all four latencies are collected, the last function is called to store the latency results in a file which later can be used in statistical analysis as well as adding to the baseline to find the results.

5.6 Extracting the Results

In order for the data to be readable, and for it to be adequately analyzed, certain things need to be done. For the simulation, the data needs to be extracted. Once the data have been extracted, it needs to be analyzed. This section will describe the steps conducted in order to enable the results to be analyzed and then analyzed using statistical calculations and latency measurements. By using these timestamps, it is possible to determine how long time it takes for the message to travel from one device to another and how long it takes for the devices to process the message.

5.6.1 Extracting Data

In the baseline wriTime-Triggered Ethernetn in OMNeT++, there are specific time outputs to determine when the emergency brake message has reached a certain point in the communication line. The different points are visualized in Figure 5.2. Whenever the message leaves a unit (e.g., ADASDC) or enters a new one (e.g., a brake), there is a timer. So, there will be one time when the message leaves the sensor, one when it enters the ADASDC, one when it leaves it, for each device. That leaves a total of 9 timers. One for the sensor, two each for ADASDC and BDC. Finally, one time point per brake. The data from the security measure implementations are collected in a comma-separated file, which then can be brought into an Excel sheet for calculations.

With over 1000 data points that need to be included in the results, it is critical to use a structured approach to acquire them. Some points of data that are collected may differ from the mean. A possible reason for this is the experiment environment where the use of a processor core may differ from iteration to iteration depending on operative system usage and other applications on the virtual machine as mentioned in the limitations. Therefore, using statistical calculations, it is possible to find a 95% confidence interval, which is based on the standard deviation and use the upper interval for the results. This is because the maximum value risks being way past the mean in terms of latency while the upper interval may be closer to the truth. Using the upper interval also ensures that a latency that is worse than average, which then can prove whether the security measures are feasible to use in the automotive domain or not.

Therefore, the final results that will be used for comparison to the requirements set will be found by adding the upper interval limit of the total baseline together with the upper interval limit of the two different security measures (both signing and verifying).

5.6.2 Statistical calculations

Standard deviation is calculated by the following formula:

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{n}} \tag{5.1}$$

Where σ is the standard deviation. n is the number of iterations. x_i is the total duration of communication for every single iteration i. µ is the mean of all duration iterations.

The confidence interval can be computed using the following formula:

$$\mu \pm \lambda_{\alpha/2} \cdot \frac{\sigma}{\sqrt{n}} \tag{5.2}$$

When a two-tailed 95% confidence is what is being searched for, then $\lambda_{\alpha/2}$ is 1.96 which can be found in the quantiles of the Gaussian distribution. However, for a confidence interval to be statistically significant, the data must be normally distributed. Testing data can be done using the Shapiro-Wilk test [41].

5.7 Simulation Results

The collected data from the baseline and the different security measures are listed in Appendix A, B and C. The following section contains a summary of the results, followed by the standard deviation and confident interval for the total time of communication.

The results are divided into four categories: Baseline, HMAC-SHA-512, ECDSA - ED25519, and Total time of communication.

5.7.1 Baseline

These results are taken from 50 iterations of the simulation made in OMNeT++. All results are presented in milliseconds for each part of the line of communication. The reason only 50 iterations are conducted is because of the manual hand-on interaction that has to be conducted for the extraction of results. The timestamps need to be manually copied from the graphical user interface to a document.

Table 5.1: Partial baseline results

Partial baseline	Min (ms)	Average (ms)	Max (ms)	Std. dev. ρ (ms)
Sensor to ADASDC	0.00738	0.00738	0.00738	0.00000
ADASDC Processing	0.31200	0.43164	0.56400	0.05279
ADASDC to BDC	0.00738	0.00738	0.00738	0.00000
BDC Processing	0.39000	0.43272	0.42900	0.05529
BDC to Brakes	0.00738	0.00738	0.00738	0.00000
Brakes processing	0.21800	0.26307	0.36800	0.02056

According to the simulation, the time from one device to another is constant at 0.00738 milliseconds (ms). The time for the processing of each unit is similar for ADASDC and BDC around 0.432 milliseconds (ms) when looking at the average time. The processing time for the brake units is about 0.253 milliseconds (ms).

The following results take the total time of communication based on all iterations, i.e., the average, the minimum, and the maximum of the total time of communication from the baseline simulations into account. In other words, the minimum is not the minimum duration for each part of the communication duration, but instead, the minimum total simulated.

Table 5.2: Baseline results total

Baseline total	Min (ms)	Average (ms)	Max (ms)	Std. dev. ρ (ms)
Total time	0.94214	1.14957	1.38314	0.07408

Using the equation formulas *4.1* and *4.2*, found in Section 5.6.2, the standard deviation is found to be 0.07408 ms as presented in Table 5.2 and the 95% confidence interval is 1.14957 ± 0.02053 ms.

5.7.2 HMAC-SHA-512

The implementation of HMAC-SHA-512 has been iterated 200 times to gain accurate numbers. The results are divided into signing duration and verification duration. For HMAC-SHA-512, the only difference between signing and verification procedures is the comparison between the two produced hash-sums, thus making the verification take longer time than the signing

Table 5.3: HMAC results

HMAC-SHA-512	Min (ms)	Average (ms)	Max (ms)	Std. dev. ρ (ms)
Signing	0.00400	0.00918	0.04000	0.00309
Verification	0.00800	0.01174	0.07200	0.00589

Using the equation formulas *4.1* and *4.2*, found in Section 5.6.2, the standard deviation for signing is found to be 0.00309 ms presented in Table 5.3 and the 95% confidence interval is 0.00918 ± 0.00043 ms. For verification, the standard deviation is found to be 0.00589 ms as presented in Table 5.3 and the 95% confidence interval is 0.01174 ± 0.00082 ms.

5.7.3 ECDSA - ED25519

The implementation of ECDSA - ED25519 has been iterated 200 times to gain accurate numbers. The results are divided into signing duration and verification duration.

Table 5.4: ECDSA results

ECDSA - ED25519	Min (ms)	Average (ms)	Max (ms)	Std. dev. ρ (ms)
Signing	0.16400	0.39540	0.73600	0.08168
Verification	0.51200	1.50720	2.48800	0.17623

Using the equation formulas *4.1* and *4.2*, found in Section 5.6.2, the standard deviation for signing is found to be 0.08168 ms presented in Table 5.4 and the 95% confidence interval is 0.39540 ± 0.01132 ms. For verification, the standard deviation is found to be 0.17623 ms as presented in Table 5.4 and the 95% confidence interval is 1.50720 ± 0.02444 ms.

5.7.4 Total time of communication

By adding together, the upper interval limit from baseline with the signing and verification of the two different security measures separately, it is possible to find results regarding what latency may be expected if the proposed security measures are implemented.

Table 5.5: Final results using upper interval of confidence interval. B = Baseline, S = Signing, V = Verification, T = Total and % is percentage security measure latency out of total latency.

Upper interval	B (ms)	S (ms)	V (ms)	T (B+S+V) (ms)	%
HMAC-SHA-512	1.170104	0.00961	0.01256	1.192274	1.86%
ECDSA - ED25519	1.170104	0.40672	1.53160	3.108424	62.4%

As shown by Table 5.5, the total time of latency for HMAC-SHA-512 and ECDSA - ED25519 are both reasonable low and are far below the 10 millisecond requirements. However, ECDSA - ED25519 is by far more time consuming than HMAC-SHA-512, proven by the percentage of total latency that is caused by the security measures in Table 5.5.

Table 5.6: Final results using average values. B = Baseline, S = Signing, V = Verification, T = Total and % is percentage security measure latency out of total latency.

Average values	B (ms)	S (ms)	V (ms)	T (B+S+V) (ms)	%
HMAC-SHA-512	1.14957	0.00918	0.01174	1.17049	1.79%
ECDSA - ED25519	1.14957	0.39540	1.50718	3.05215	62.3%

The upper interval of the different confidence intervals is considered the most interesting as it would prove if the scenario would fail the set requirements. It may still be of interest to see how the average scenario would look for both security measures, which can be found in Table 5.6. Even in the worst-case scenario using this simulation and security measures, the total latency is far from the limit of 10 ms, thus answering RQ3. Note, however, that because of the different assumptions that have been made, it cannot be considered an answer with fully valid grounds, only as a clear guideline to what the actual implementation could look like.

The upper interval results for HMAC-SHA-512 are 0.021784 ms longer latency compared to the average. For the ECDSA - ED25519 results, there is a difference of 0.056274 ms in latency, where the upper interval values have the most prolonged latency compared to the average.

Chapter 6

Analysis and Discussion

This section contains an analysis and discussion of the results together with answers to the research questions put forth in this thesis.

6.1 Research Question 1

What security vulnerabilities can be found in an Automotive Ethernet network out-of-the-box, and what risks do they convey?

Using the results of the literature study in Chapter 3, it was made possible to give an answer to the first research question. The vulnerabilities of Automotive Ethernet out-of-the-box are numerous, and there are many security aspects that need to be taken into account for Automotive Ethernet to function as a secure network in modern vehicles fully.

It is surprising how many vulnerabilities there are in Ethernet out-of-the-box. However, there has come several supplements to Ethernet, which can be used to secure communication and infrastructure. Moreover, these vulnerabilities that are being mentioned are primarily on a low level, with advanced applications comes the risk of security vulnerabilities, identifying those are, however, out of this scope.

6.2 Research Question 2

Which security measures can mitigate identified vulnerabilities from RQ1 in Automotive Ethernet?

In Chapter 4, a list of potential countermeasures against the vulnerabilities found in the literature study is listed. This list has been put together with suggested security measures mentioned in the related work in addition to known security measures learned during the author's five-year-long education at BTH in Computer Security. The list presented in Chapter 4 may not be fully completed. However, some of the most critical threats have been investigated, e.g., frame integrity and frame authenticity.

As someone well understanding of potential security flaws and solutions, none of the countermeasures were in any way surprising. However, the primary challenge with this thesis has not been to identify countermeasures and implement them; it has been to learn of automotive implementations, learning about research conducted in the area of Automotive Security and finally to work with real-time simulations.

6.3 Research Question 3

Which security measures are suited for Automotive Ethernet without compromising safety in regard to latency?

Using the list found in Chapter 4, the most critical ones were selected from the perspective of time-critical communication. Frame integrity and Frame authenticity are most relevant when it comes to ensuring safe and secure time-critical communication. It is not needed to keep information hidden, rather knowing that information comes from the correct source and has not been altered in transition.

As reasoned in Chapter 5, both DSA and HMAC would be able to countermeasure threats towards frame integrity and authenticity. To learn if these measures are suitable security measures for time-critical communication in the automotive domain, the simulation and added security measures were compared to the set requirement in regard to latency. Using the data from the simulation, the programmed security measure program and statistical calculations, the results suggest that both security measures are applicable. The standard deviation for baseline, HMAC, and ECDSA is quite low, which means that most of the latencies are located near the mean. Having data centralized around the data may suggest that the mean is quite accurate, which is desirable. However, even though the baseline results are normal distributed, the security measures are not according to the Shapiro-Wilk Test. This makes confidence intervals for the security measures inaccurate and can lead to additional error factors.

With this in mind, it is possible to view the results as a distribution, and it is clear that the values are grouped together rather closely, it may not be the standard deviation *bell*-form, it does, however, show that most of the latencies are very close to

the mean constantly. See the plots in Figure 6.1 to get a more accurate presentation of the distribution.

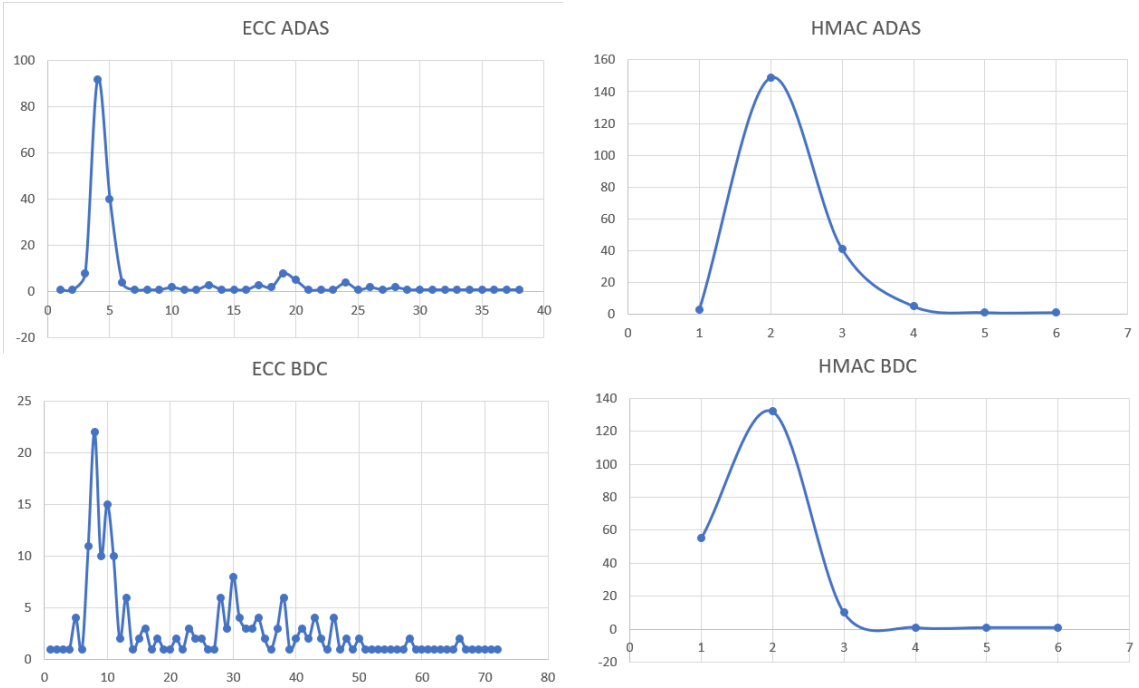

Figure 6.1: The distribution plots for the security measure latencies.

Despite the fact that the confidence intervals may not be accurate, it still shows that it is feasible that the selected security measures could potentially resolve authenticity and integrity threats. However, considering the vast amount of possible error factors in the simulation, it needs to be stated that these results are not reliable but do show a general direction in regard to latency when applying the selected security measures.

6.4 Discussion

The results presented in Chapter 5.7 may alter how standard security measures may be used in a future Ethernet-based network in the automotive domain. However, there are certain things that need to be taken into account regarding these results and its validity.

Because of the many potential error factors, it is crucial that these are declared and explained why this issue has occurred. In this thesis, the first primary issue that the project faced was that it was not possible to gain access to physical hardware to conduct the experiment. This led to the author having to turn to software-based experiments using a simulation. A simulation, however, can directly cause potential errors. If the simulation tool is not accurate, it could cause invalid results. To avoid this, the author compared different simulation programs and looked at what other researches had used in their experiments. OMNeT++ was therefore selected as it had the highest amount of usage within the automotive domain and has been claimed

on forums and in review to be a dependable tool.

The second issue that arose was that it was not possible to get information regarding modern/future topologies, ECU processing power, and the number of ECU firmware instructions from OEMs or published. This provided significant difficulties in designing the simulation. In order to do anything about this, the limited amount of published information regarding modern topologies and processing power had to be discussed with domain experts of the automotive security domain in order to create a plausible topology and find a processing power to design the simulation scenario. However, it was not possible to make any educated guesses regarding firmware instructions. Therefore, assumptions had to be done, which frankly is entirely without a proper basis, making the baseline latency a potential error factor.

The third issue that occurred during the project was the fact that problems with the OMNeT++ simulation development environment, which caused the compiler not to update the executable. Troubleshooting this error took several months before the author and supervisor decided to keep the executable as it was (only baseline) and calculate the security measures separately. Having the simulation and security measures separated causes another potential error factor.

Lastly, when the results were created, the baseline was considered being normal distributed; however, the ECDSA and HMAC latencies were not, which creates a potential error factor regarding the calculated confidence intervals. However, since the data is very grouped up except for some deviations, the mean can be considered relatively accurate. Either way, because part of the data is not normal distributed, the confidence interval cannot be trusted.

These potential error factors primarily caused problems to validity regarding the simulated experiment. However, both the first two research questions, RQ1 and RQ2, have not had any significant problems that arose and could cause problems with validity. Initially, there were too few academic references, which was resolved at a later stage when more previously undiscovered papers were added together with some new publications and some that were unavailable to the author.

With all these potential error factors in mind, there are several issues that can make the results questionable. Therefore, the results should not be taken as facts, but rather as a guideline toward potential future research that can be conducted in order to get accurate results. However, if the assumptions are somewhat accurate, these results could then be providing proof that theoretically, the suggested security measures would work in a time-critical communication network like the one in the topology suggested in Chapter method. Although to know for sure, a more extensive physical experiment should be conducted to confirm the potential theoretical proof. The only way to know whether the results from the simulated experiment is somewhat accurate is to have domain knowledge from inside an OEM.

Chapter 7

Conclusion

This thesis has investigated which security flaws exist out-of-the-box in Automotive Ethernet, and which risks they convey. With a primary focus on the vulnerabilities found in standard Ethernet and the flaws that would affect intra-vehicular communication in regards to safety, an attack against traffic integrity and attacks against traffic authenticity were identified as the essential threats.

In addition to the two primary threats focused upon in this thesis' experiment, in Chapter 3 provides a list of located vulnerabilities in Automotive Ethernet out-of-the-box and Chapter 4 provide a summary of potential security measures to mitigate those vulnerabilities. Two of those security measures were DSA and HMAC, which could both be used to mitigate threats to message integrity and message authenticity, which resulted in them being the subjects of testing in the simulated experiment.

Because of the limitations described in Section 1.4 in addition to the error factors mentioned in Section 6, there is much to suggest that the data created by the simulated experiment could be invalid and should, therefore, be treated as such. However, if the assumptions that were made for the simulated experiment is close to accurate, the results may be considered as accurately theoretically, but does require confirmation either way.

If these results end up being somewhat accurate, it can potentially provide a guideline towards future automotive security in the aspect of time-critical communication. According to the data found in this thesis, it is likely for both the security measures to be applied in the time-critical automotive domain.

This thesis has managed to answer all three research questions, although RQ3 will require additional work to be conducted for it to be confirmed. Connected vehicles can be secure in the future; however, there is work required to reach that goal. This thesis could be another step in that direction.

Chapter 8

Future Work

With the results and analysis in regard, the following areas are suggested as future work.

1. With the results and analysis in regard, the following areas are suggested as future work. Create a realistic line of communication using actual hardware capable of communication using Time-Triggered Ethernet and implement the security measures suggested in this thesis.
   Doing so would provide more realistic results and verify or disprove the found results. If a hardware-based experiment using said security measures would be successful, that could be a great leap towards a complete Automotive Ethernet network within vehicles.

2. Perform an extensive study where additional security measures are tested and involve the automotive industry in the work so that they can verify and comment on the assumptions made in this thesis. By performing an extensive study with more security measures, it would be possible to find the best solution in regard to security and safety. By involving the automotive industry, the simulated results may be more likely to be correct than the results in this thesis.

References

[1] John Manuel. The information security triad: Cia. second version. Available at https://en.wikipedia.org/wiki/Information_security#/media/File:CIAJMK1209.png Accessed: (2019/05/15).

[2] Oxford University. Lexico. Available at https://www.lexico.com/en Accessed: (2019/06/15).

[3] Chung-Wei Lin and Huafeng Yu. Invited: Cooperation or competition? co-existence of safety and security in next-generation ethernet-based automotive networks. *Design Automation Conference proceedings*, 2016(52), 2016.

[4] Ján Laštinec. *Security Extension of Automotive Communication Protocols using Ethernet/IP*. PhD thesis, Slovak University of Technology in Bratislava, Bratislava, Slovakia, 2016.

[5] Ján Laštinec and Ladislav Hudec. Comparative analysis of tcp/ip security protocols for use in vehicle communication. *Carpathian Control Conference*, 2016(17):429–433, 2016.

[6] Wikipedia Contributors. Information security. Available at https://en.wikipedia.org/wiki/Information_security Accessed: (2019/06/13).

[7] Ixia Worldwide. Automotive ethernet: An overview. Technical Report 915-3510-01, Ixia, Calabasas, California, May 2014.

[8] Philipp Mundhenk Sebastian Steinhorst Martin Lukasiewycz Suhaib A. Fahmy and Samarjit Chakraborty. Lightweight authentication for secure automotive networks. *Design, Automation & Test in Europe Conference & Exhibition*, pages 285–288, 2015.

[9] C. Varun and M. Kathiresh. Automotive ethernet in on-board diagnosis (over ip) & in-vehicle networking. *International Conference on Embedded Systems*, pages 255–260, 2014.

[10] TTTech ComputerTechnik AG. Time-triggered ethernet – a powerful network solution for multiple purpose. Available at https://paginas.fe.up.pt/~ee04007/tese/docs/TTEthernet_Article.pdf Accessed: (2019/06/15).

[11] IEEE. Ieee p802.3bp, 1000base-t1 phy task force. Available at http://www.ieee802.org/3/bp/ Accessed: (2019/06/15).

[12] Christian Sandberg Atul Yadav. Holisec reference architecture. Available at https://autosec.se/wp-content/uploads/2018/04/HOLISEC_D4.1.3_v1.0.pdf Accessed: (2019/03/10).

[13] Christopher Corbett Elmar Schoch Frank Kargl and Preussner Felix. Automotive ethernet: Security opportunity or challenge? *Lecture Notes in Infomatics*, pages 45–54, 2016.

[14] OMNeT++ official website. Omnet++ discrete event simulator. Available at https://omnetpp.org/ Accessed: (2019/06/13).

[15] Youn-Hee Han Ju-Ho Choi. Macsec extension over software-defined networks for in-vehicle secure communication. *Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018.

[16] Berardino Carnevale Luca Fanucci Samson Bisase Harman Hunjan. Macsec-based security for automotive ethernet backbones. *Journal of Circuits, Systems, and Computers*, 27(5), 2018.

[17] Beradino Carnevale Francesco Falaschi Luca Crocetti Harman Hunjan Samson Bisase Luca Fanucci. An implementation of the 802.1ae mac security standard for in-car networks. *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2, 2016.

[18] National Instruments. Ecu designing and testing using national instruments products. Available at http://www.ni.com/white-paper/3312/en/ Accessed: (2017/03/13).

[19] Vector. The lin standard. Available at https://vector.com/vi_local_interconnect_network_en.html Accessed: (2018/02/21).

[20] National Instruments. Flexray automotive communication bus overview. Available at http://www.ni.com/white-paper/3352/en/ Accessed: (2019/06/15).

[21] OMNeT++ website. Vehicular and distributed control systems for omnest omnet++. available at https://vehicular.omnetpp.org/Accessed:(2019/08/02).

[22] Guy Paré Spyros Kitsiou. Methods for literature reviews. Available at https://www.ncbi.nlm.nih.gov/books/NBK481583/ Accessed: (2019/06/15).

[23] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.709.9164&rep=rep1&type=pdf Accessed: (2019/07/26).

[24] Daniel Zelle Christoph Krauß Hubert Strauß Karsten Schmidt. On using tls to secure in-vehicle networks. *ARES '17 Proceedings of the 12th International Conference on Availability, Reliability and Security*, 12(67), 2017.

[25] Huafeng Yu Chung-Wei Lin. Security concerns for automotive communication and software architecture. *Conference on Computer Communications Workshops (INFOCOM WKSHPS): 2016 IEEE Infocom CPSS Workshop*, 2016.

[26] Jukka Manner Timo Kiravuo, Mikko Särelä. A survey or ethernet lan security. *IEEE Communications Surveys & Tutorials*, 15(3):1477–1491, 2013.

[27] Fedor Smirnov Felix Reimann Jürgen Teich Michael Glaß. Automatic optimization of the vlan partitioning in automotive communication networks. *ACM Transactions on Design Automation of Electronic Systems*, 24(1):1–23, 2018.

[28] Luca De Ambroggi. Ethernet in car: from multiplexed network to service network. Available at `https://technology.ihs.com/590689/ethernet-in-car-from-multiplexed-network-to-service-network` Accessed: (2019/06/15).

[29] Lucia Lo Bello Riccardo Mariani Saad Mubeen Sergio Saponara. Recent advances and trends in on-board embedded and networked automotive systems. *IEEE Transactions on Industrial Informatics*, 15(2):1038–1051, 2019.

[30] Venkatesh Sundar. Arp poisoning attack: How to prevent arp spoofing. Available at `https://www.indusface.com/blog/protect-arp-poisoning/` Accessed: (2019/08/01).

[31] Borja Merino. Defenses against dhcp attacks. Available at `https://www.securityartwork.es/2013/01/30/defenses-against-dhcp-attacks/` Accessed: (2019/07/31).

[32] Wikipedia Contributors. Session hijacking. Available at `https://en.wikipedia.org/wiki/Session_hijacking` Accessed: (2019/08/01).

[33] Morris Dworkin. Nist policy on hash functions. Available at `https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions` Accessed: (2019/07/30).

[34] Morris Dworkin. Digital signatures. Available at `https://csrc.nist.gov/Projects/Digital-Signatures` Accessed: (2019/07/30.

[35] Morris Dworkin. Nist sp 800-57 part 3. Available at `https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-57pt3r1.pdf` Accessed: (2019/07/30.

[36] Christoph Hammerschmidt. Functional safety: Safely delivering real-time performance. Available at `https://www.eenewsautomotive.com/content/functional-safety-safely-delivering-real-time-performance` Accessed: (2019/06/15).

[37] James Scobie Mark Stachew. Electronic control system partitioning in the autonoumous vehicle. Available at `https://www.eenewsautomotive.com/content/electronic-control-system-partitioning-autonomous-vehicle` Accessed: (2019/06/15).

[38] Pepe Vera Campello. What is the current specs and speed of the processors in the engine control units in most of the cars on sale currently. Available at `https://www.quora.com/What-is-the-current-specs-and-speed-of-the-processors-in-the-engine-control-units-in-most-of-the-cars-on-sale-currently` Accessed: (2019/06/15).

[39] Wikipedia Contributors. V850. Available at `https://en.wikipedia.org/wiki/V850` Accessed: (2019/06/15).

[40] NXP. S32v234: Vision processor for front and surround view camera, machine learning and sensor fusion. Available at `https://www.nxp.com/products/processors-and-microcontrollers/arm-based-processors-and-mcus/s32-automotive-platform/vision-processor-for-front-and-surround-view-camera-machine-learning-and-sensor-fusion:S32V234` Accessed: (2019/06/15).

[41] Statskingdom. Shaprio-wilk calculator. Available at `http://www.statskingdom.com/320ShapiroWilk.html` Accessed: (2019/07/28).

Appendix A

Table of Baseline latencies

ADAS referrers to when the message arrives at the ADASDC.
ADAS referrers to when the ADASDC has finished its calculations and forward the message.
BDC referrers to when the message arrives at the BDC.
BDC referrers to when the BDC has finished its calculations and forward the message.
Brakes referrers to when the message arrives at the brakes (which is simultaneously).
Brake1, Brake2, Brake3 and Brake4 referrers to when the brakes has finished its calculations performs the braking procedure.

@ADAS	ADAS	@BDC	BDC	@Brakes	Brake1	Brake2	Brake3	Brake4
0,00000738	0,00040438	0,00041176	0,00086976	0,00087714	0,00119714	0,00133314	0,00117014	0,00115414
0,00000738	0,00049338	0,00050076	0,00095276	0,00096014	0,00126014	0,00124114	0,00119714	0,00120514
0,00000738	0,00055738	0,00056476	0,00094576	0,00095314	0,00118514	0,00125214	0,00126014	0,00125914
0,00000738	0,00045738	0,00046476	0,00089976	0,00090714	0,00118914	0,00114114	0,00114914	0,00115814
0,00000738	0,00044738	0,00045476	0,00088976	0,00089714	0,00113914	0,00114514	0,00113114	0,00114714
0,00000738	0,00048538	0,00049276	0,00089176	0,00089914	0,00117714	0,00118614	0,00125014	0,00114414
0,00000738	0,00057138	0,00057876	0,00098176	0,00098914	0,00122314	0,00123214	0,00129214	0,00123114
0,00000738	0,00043638	0,00044376	0,00083376	0,00084114	0,00106714	0,00113514	0,00115114	0,00111814
0,00000738	0,00049538	0,00050276	0,00091076	0,00091814	0,00126114	0,00115514	0,00121714	0,00115314
0,00000738	0,00048438	0,00049176	0,00094176	0,00094914	0,00118614	0,00119214	0,00120314	0,00121914

@ADAS	ADAS	@BDC	BDC	@Brakes	Brake1	Brake2	Brake3	Brake4
0,00000738	0,00047438	0,00048176	0,00088876	0,00089614	0,00114014	0,00116514	0,00117414	0,00113914
0,00000738	0,00043338	0,00044076	0,00090976	0,00091714	0,00116414	0,00115014	0,00118414	0,00114014
0,00000738	0,00040138	0,00040876	0,00089276	0,00090014	0,00113614	0,00115214	0,00119414	0,00118614
0,00000738	0,00042538	0,00043276	0,00087776	0,00088514	0,00127314	0,00113114	0,00114214	0,00112014
0,00000738	0,00053138	0,00053876	0,00093576	0,00094314	0,00117814	0,00118614	0,00117614	0,00136214
0,00000738	0,00046938	0,00047676	0,00091276	0,00092014	0,00115514	0,00116814	0,00120314	0,00124714
0,00000738	0,00040438	0,00041176	0,00088476	0,00089214	0,00112514	0,00118414	0,00116214	0,00113014
0,00000738	0,00040638	0,00041376	0,00074276	0,00075014	0,00098614	0,00105814	0,00099214	0,00102214
0,00000738	0,00038338	0,00039076	0,00084476	0,00085214	0,00112414	0,00113914	0,00108914	0,00109414
0,00000738	0,00042338	0,00043076	0,00076976	0,00077714	0,00104814	0,00101814	0,00106214	0,00101114
0,00000738	0,00043238	0,00043976	0,00083376	0,00084114	0,00109814	0,00113014	0,00108014	0,00115114
0,00000738	0,00042338	0,00043076	0,00081976	0,00082714	0,00108714	0,00106114	0,00110414	0,00106614
0,00000738	0,00033838	0,00034576	0,00077476	0,00078214	0,00102014	0,00101714	0,00102014	0,00101314
0,00000738	0,00039438	0,00040176	0,00079076	0,00079814	0,00109214	0,00103314	0,00108314	0,00103114
0,00000738	0,00043838	0,00044576	0,00083376	0,00084114	0,00108114	0,00107214	0,00110814	0,00107414
0,00000738	0,00046038	0,00046776	0,00094876	0,00095614	0,00124514	0,00122414	0,00122414	0,00123014
0,00000738	0,00039838	0,00040576	0,00080176	0,00080914	0,00102814	0,00107714	0,00116414	0,00104814
0,00000738	0,00031938	0,00032676	0,00071676	0,00072414	0,00094214	0,00100614	0,00096414	0,00101314
0,00000738	0,00041238	0,00041976	0,00087076	0,00087814	0,00115114	0,00138314	0,00115814	0,00116914
0,00000738	0,00050238	0,00050976	0,00098476	0,00099214	0,00123414	0,00122514	0,00125014	0,00126514
0,00000738	0,00044238	0,00044976	0,00076676	0,00077414	0,00105514	0,00114314	0,00102014	0,00100814
0,00000738	0,00043838	0,00044576	0,00088076	0,00088814	0,00115614	0,00112214	0,00114714	0,00114414
0,00000738	0,00039538	0,00040276	0,00088676	0,00089414	0,00113614	0,00112514	0,00113414	0,00112514
0,00000738	0,00039838	0,00040576	0,00079376	0,00080114	0,00108614	0,00106814	0,00104414	0,00105314
0,00000738	0,00057038	0,00057776	0,00100776	0,00101514	0,00125214	0,00124714	0,00130114	0,00124914

@ADAS	ADAS	@BDC	BDC	@Brakes	Brake1	Brake2	Brake3	Brake4
0,00000738	0,00046238	0,00046976	0,00099276	0,00100014	0,00123914	0,00123114	0,00127414	0,00123314
0,00000738	0,00044238	0,00044976	0,00089976	0,00090714	0,00120814	0,00114214	0,00114814	0,00119914
0,00000738	0,00039638	0,00040376	0,00079976	0,00080714	0,00102514	0,00108314	0,00102514	0,00103814
0,00000738	0,00043738	0,00044476	0,00086876	0,00087614	0,00115014	0,00110914	0,00111414	0,00115414
0,00000738	0,00049938	0,00050676	0,00098776	0,00099514	0,00123514	0,00125114	0,00125414	0,00123114
0,00000738	0,00040638	0,00041376	0,00083876	0,00084614	0,00108614	0,00107714	0,00108514	0,00118914
0,00000738	0,00036138	0,00036876	0,00076076	0,00076814	0,00100614	0,00106514	0,00100914	0,00112214
0,00000738	0,00047338	0,00048076	0,00090976	0,00091714	0,00115514	0,00115514	0,00122514	0,00119014
0,00000738	0,00047838	0,00048576	0,00091276	0,00092014	0,00119514	0,00115114	0,00116214	0,00115614
0,00000738	0,00039938	0,00040676	0,00100276	0,00101014	0,00125114	0,00124214	0,00124714	0,00127814
0,00000738	0,00039738	0,00040476	0,00096176	0,00096914	0,00120814	0,00120014	0,00120714	0,00123914
0,00000738	0,00039738	0,00040476	0,00096876	0,00097614	0,00120114	0,00127714	0,00119714	0,00126314
0,00000738	0,00044338	0,00045076	0,00096476	0,00097214	0,00123514	0,00120414	0,00121214	0,00120714
0,00000738	0,00040738	0,00041476	0,00080476	0,00081214	0,00111314	0,00105214	0,00109814	0,00104414
0,00000738	0,00039838	0,00040576	0,00079376	0,00080114	0,00104214	0,00117914	0,00103714	0,00103314

Appendix B

Table of HMAC-SHA-512 latencies (in seconds)

Nr	Signing	Verification
1	0,000012	0,000012
2	0,000016	0,000016
3	0,000020	0,000012
4	0,000016	0,000016
5	0,000012	0,000012
6	0,000008	0,000012
7	0,000012	0,000016
8	0,000016	0,000012
9	0,000012	0,000016
10	0,000012	0,000012
11	0,000008	0,000048
12	0,000008	0,000008
13	0,000012	0,000016
14	0,000008	0,000012
15	0,000012	0,000012
16	0,000008	0,000012
17	0,000012	0,000012
18	0,000012	0,000016
19	0,000040	0,000016
20	0,000008	0,000012
21	0,000012	0,000008
22	0,000012	0,000008
23	0,000008	0,000012
24	0,000008	0,000012
25	0,000008	0,000012
26	0,000008	0,000012
27	0,000012	0,000012
28	0,000008	0,000072
29	0,000008	0,000012
30	0,000016	0,000012

Nr	Signing	Verification
31	0,000012	0,000016
32	0,000008	0,000008
33	0,000008	0,000012
34	0,000008	0,000012
35	0,000012	0,000012
36	0,000012	0,000012
37	0,000012	0,000016
38	0,000012	0,000016
39	0,000008	0,000008
40	0,000008	0,000012
41	0,000012	0,000012
42	0,000008	0,000012
43	0,000008	0,000008
44	0,000008	0,000012
45	0,000012	0,000008
46	0,000012	0,000012
47	0,000016	0,000012
48	0,000012	0,000012
49	0,000008	0,000012
50	0,000008	0,000008
51	0,000008	0,000012
52	0,000012	0,000012
53	0,000008	0,000008
54	0,000008	0,000012
55	0,000008	0,000012
56	0,000012	0,000012
57	0,000008	0,000008
58	0,000008	0,000012
59	0,000012	0,000012
60	0,000008	0,000012

Nr	Signing	Verification
61	0,000008	0,000012
62	0,000008	0,000012
63	0,000008	0,000012
64	0,000008	0,000012
65	0,000008	0,000008
66	0,000008	0,000008
67	0,000008	0,000012
68	0,000008	0,000012
69	0,000008	0,000008
70	0,000008	0,000012
71	0,000008	0,000012
72	0,000008	0,000008
73	0,000008	0,000008
74	0,000008	0,000012
75	0,000008	0,000012
76	0,000008	0,000012
77	0,000012	0,000008
78	0,000008	0,000008
79	0,000008	0,000008
80	0,000008	0,000012
81	0,000008	0,000012
82	0,000008	0,000012
83	0,000012	0,000012
84	0,000008	0,000012
85	0,000012	0,000012
86	0,000008	0,000008
87	0,000008	0,000012
88	0,000008	0,000012
89	0,000008	0,000012
90	0,000008	0,000012

Nr	Signing	Verification
91	0,000008	0,000008
92	0,000008	0,000012
93	0,000008	0,000012
94	0,000008	0,000012
95	0,000008	0,000008
96	0,000012	0,000044
97	0,000008	0,000012
98	0,000008	0,000012
99	0,000012	0,000008
100	0,000008	0,000012
101	0,000012	0,000012
102	0,000008	0,000008
103	0,000008	0,000012
104	0,000008	0,000012
105	0,000012	0,000012
106	0,000012	0,000012
107	0,000008	0,000012
108	0,000008	0,000008
109	0,000012	0,000012
110	0,000008	0,000012
111	0,000008	0,000008
112	0,000008	0,000012
113	0,000008	0,000012
114	0,000012	0,000012
115	0,000012	0,000012
116	0,000008	0,000008
117	0,000008	0,000012
118	0,000008	0,000008
119	0,000008	0,000008
120	0,000012	0,000012

Nr	Signing	Verification
121	0,000008	0,000012
122	0,000008	0,000012
123	0,000012	0,000012
124	0,000008	0,000012
125	0,000008	0,000008
126	0,000008	0,000012
127	0,000008	0,000012
128	0,000008	0,000008
129	0,000008	0,000008
130	0,000008	0,000008
131	0,000008	0,000008
132	0,000008	0,000008
133	0,000008	0,000008
134	0,000012	0,000008
135	0,000008	0,000012
136	0,000012	0,000012
137	0,000008	0,000008
138	0,000008	0,000012
139	0,000008	0,000008
140	0,000008	0,000008
141	0,000008	0,000008
142	0,000008	0,000008
143	0,000008	0,000012
144	0,000008	0,000012
145	0,000008	0,000012
146	0,000008	0,000012
147	0,000008	0,000012
148	0,000008	0,000012
149	0,000008	0,000012
150	0,000008	0,000012

Nr	Signing	Verification
151	0,000008	0,000008
152	0,000008	0,000012
153	0,000008	0,000008
154	0,000008	0,000012
155	0,000008	0,000012
156	0,000008	0,000012
157	0,000008	0,000012
158	0,000012	0,000012
159	0,000008	0,000012
160	0,000008	0,000012
161	0,000008	0,000008
162	0,000004	0,000012
163	0,000008	0,000012
164	0,000008	0,000008
165	0,000008	0,000012
166	0,000008	0,000012
167	0,000008	0,000012
168	0,000008	0,000008
169	0,000008	0,000008
170	0,000008	0,000008
171	0,000008	0,000012
172	0,000008	0,000012
173	0,000008	0,000012
174	0,000008	0,000012
175	0,000008	0,000012
176	0,000008	0,000012
177	0,000008	0,000012
178	0,000008	0,000012
179	0,000008	0,000012
180	0,000008	0,000012

Nr	Signing	Verification
181	0,000008	0,000012
182	0,000008	0,000012
183	0,000008	0,000008
184	0,000008	0,000008
185	0,000008	0,000008
186	0,000008	0,000012
187	0,000008	0,000012
188	0,000008	0,000012
189	0,000008	0,000012
190	0,000008	0,000012
191	0,000008	0,000008
192	0,000008	0,000012
193	0,000008	0,000008
194	0,000008	0,000012
195	0,000004	0,000012
196	0,000008	0,000012
197	0,000008	0,000012
198	0,000008	0,000008
199	0,000012	0,000012
200	0,000004	0,000008

Appendix C

Table of ECDSA - ED25519 latencies (in seconds)

Nr	Signing	Verification
1	0,000356	0,002488
2	0,000492	0,002196
3	0,000488	0,002168
4	0,000428	0,001964
5	0,000356	0,001932
6	0,000356	0,001872
7	0,000696	0,001788
8	0,000356	0,001788
9	0,000500	0,001780
10	0,000356	0,001772
11	0,000440	0,001768
12	0,000356	0,001752
13	0,000436	0,001748
14	0,000356	0,001744
15	0,000432	0,001740
16	0,000360	0,001736
17	0,000360	0,001736
18	0,000480	0,001724
19	0,000524	0,001720
20	0,000508	0,001716
21	0,000360	0,001696
22	0,000360	0,001680
23	0,000360	0,001656
24	0,000356	0,001632
25	0,000492	0,001624
26	0,000488	0,001624
27	0,000404	0,001620
28	0,000360	0,001616
29	0,000356	0,001616
30	0,000376	0,001612

Nr	Signing	Verification
31	0,000580	0,001604
32	0,000380	0,001604
33	0,000356	0,001604
34	0,000352	0,001604
35	0,000564	0,001600
36	0,000360	0,001596
37	0,000356	0,001596
38	0,000432	0,001592
39	0,000360	0,001592
40	0,000360	0,001592
41	0,000356	0,001592
42	0,000364	0,001588
43	0,000356	0,001588
44	0,000360	0,001584
45	0,000356	0,001584
46	0,000352	0,001584
47	0,000372	0,001576
48	0,000356	0,001576
49	0,000360	0,001572
50	0,000360	0,001568
51	0,000360	0,001568
52	0,000360	0,001568
53	0,000356	0,001568
54	0,000356	0,001568
55	0,000356	0,001568
56	0,000432	0,001564
57	0,000356	0,001564
58	0,000356	0,001564
59	0,000488	0,001560
60	0,000360	0,001556

Nr	Signing	Verification
61	0,000356	0,001556
62	0,000360	0,001552
63	0,000360	0,001552
64	0,000356	0,001552
65	0,000352	0,001552
66	0,000544	0,001548
67	0,000360	0,001548
68	0,000360	0,001548
69	0,000356	0,001544
70	0,000356	0,001544
71	0,000352	0,001544
72	0,000356	0,001536
73	0,000356	0,001536
74	0,000356	0,001536
75	0,000356	0,001536
76	0,000552	0,001532
77	0,000360	0,001532
78	0,000356	0,001532
79	0,000356	0,001532
80	0,000356	0,001532
81	0,000356	0,001532
82	0,000356	0,001532
83	0,000356	0,001532
84	0,000476	0,001528
85	0,000360	0,001528
86	0,000356	0,001528
87	0,000356	0,001524
88	0,000356	0,001524
89	0,000356	0,001524
90	0,000356	0,001524

Nr	Signing	Verification
91	0,000352	0,001524
92	0,000352	0,001524
93	0,000356	0,001520
94	0,000360	0,001516
95	0,000360	0,001512
96	0,000356	0,001512
97	0,000364	0,001508
98	0,000356	0,001508
99	0,000672	0,001504
100	0,000360	0,001504
101	0,000356	0,001504
102	0,000356	0,001500
103	0,000712	0,001496
104	0,000356	0,001496
105	0,000356	0,001492
106	0,000360	0,001488
107	0,000488	0,001456
108	0,000360	0,001456
109	0,000544	0,001452
110	0,000488	0,001440
111	0,000356	0,001440
112	0,000356	0,001440
113	0,000360	0,001432
114	0,000356	0,001432
115	0,000492	0,001420
116	0,000548	0,001416
117	0,000540	0,001416
118	0,000532	0,001416
119	0,000528	0,001416
120	0,000520	0,001416

Nr	Signing	Verification
121	0,000356	0,001416
122	0,000360	0,001412
123	0,000356	0,001412
124	0,000608	0,001408
125	0,000532	0,001408
126	0,000524	0,001408
127	0,000492	0,001408
128	0,000488	0,001408
129	0,000360	0,001408
130	0,000360	0,001408
131	0,000356	0,001408
132	0,000356	0,001408
133	0,000356	0,001408
134	0,000736	0,001404
135	0,000380	0,001404
136	0,000368	0,001404
137	0,000360	0,001404
138	0,000360	0,001404
139	0,000356	0,001404
140	0,000356	0,001404
141	0,000356	0,001404
142	0,000356	0,001404
143	0,000356	0,001404
144	0,000356	0,001404
145	0,000356	0,001404
146	0,000356	0,001404
147	0,000356	0,001404
148	0,000356	0,001404
149	0,000524	0,001400
150	0,000524	0,001400

Nr	Signing	Verification
151	0,000480	0,001400
152	0,000480	0,001400
153	0,000360	0,001400
154	0,000360	0,001400
155	0,000356	0,001400
156	0,000356	0,001400
157	0,000356	0,001400
158	0,000356	0,001400
159	0,000684	0,001396
160	0,000492	0,001396
161	0,000488	0,001396
162	0,000364	0,001396
163	0,000364	0,001396
164	0,000360	0,001396
165	0,000360	0,001396
166	0,000360	0,001396
167	0,000360	0,001396
168	0,000356	0,001396
169	0,000356	0,001396
170	0,000356	0,001396
171	0,000356	0,001396
172	0,000356	0,001396
173	0,000356	0,001396
174	0,000356	0,001396
175	0,000356	0,001396
176	0,000356	0,001396
177	0,000356	0,001396
178	0,000356	0,001396
179	0,000356	0,001396
180	0,000352	0,001396

Nr	Signing	Verification
181	0,000488	0,001392
182	0,000484	0,001392
183	0,000360	0,001392
184	0,000356	0,001392
185	0,000356	0,001392
186	0,000356	0,001392
187	0,000356	0,001392
188	0,000356	0,001392
189	0,000356	0,001392
190	0,000356	0,001392
191	0,000352	0,001392
192	0,000356	0,001388
193	0,000484	0,001372
194	0,000360	0,001372
195	0,000356	0,001372
196	0,000356	0,001372
197	0,000356	0,001368
198	0,000356	0,001280
199	0,000164	0,000768
200	0,000168	0,000512

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden