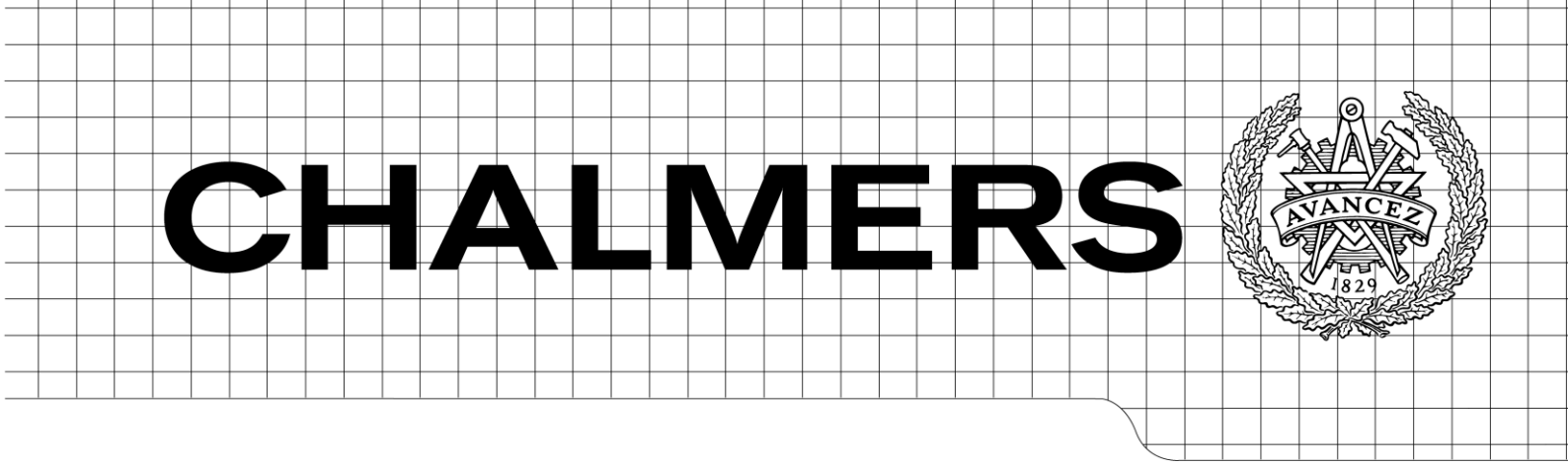


Security Analysis of Vehicle Diagnostics using DoIP

Master of Science Thesis in the Programme Networks and Distributed Systems

JOHAN LINDBERG

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, May 2011

Security Analysis of Vehicle Diagnostics using DoIP

J. Lindberg

Examiner: T. Olovsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden May 2011

Abstract

An upcoming trend in the automotive industry is to enable remote access to vehicles. This access opens up for many new applications, such as the possibility to perform vehicle diagnostics over the air. There are obvious benefits in being able to diagnose a vehicle remotely; a driver that experiences a problem with the car can just pull over to the side and call the workshop, which may perform diagnosis of the vehicle over the air.

So far, diagnostics have been performed using brand-specific protocols, but as the car is getting connected, IP-based networks may be used when communicating with the vehicle. The documents in ISO 13400 DIS (Draft International Standard), Diagnostics over IP (DoIP), describe a protocol for this type of interaction. The protocol may be used in environments with varying security characteristics. For example, a vehicle might be parked in a workshop and have a direct connection to the test equipment. The other extreme is a car at an arbitrary distance from the workshop, communicating over the Internet.

This work composes a security analysis of a DoIP system. An examination of the security environment is one part of this work. Furthermore, when connecting the car, new security issues must be considered. To ensure the continuous operation of safety-critical systems within the car, the vehicle along with its communication has to be protected. Therefore, this work contains a thorough investigation of the DoIP protocol. The report describes a set of required security attributes derived from safety aspects and discusses what is satisfied by the protocol. Since DoIP runs on top of TCP/IP, the inherited security issues are also taken into account.

Keywords: Automotive, DoIP, Remote Diagnostics, Security

Acknowledgements

First and foremost I would like to extend thanks to my two supervisors, Henrik Broberg from the industrial side and Pierre Kleberger from the academic side. They have helped me in providing advice, reviewing my writing, having regular discussions on the issues that have emerged during the work as well as making sure I have been on track. I would also like to thank my examiner, Tomas Olovsson, who helped me get off to a good start and who has provided assistance in reviewing the report. Another person who deserves to be thanked is Jan Nilsson, for providing general thoughts on the report as well as reviewing the more technical aspects of the DoIP analysis.

Furthermore, I am grateful to Volvo Car Corporation for providing the resources needed to carry out my work and specific thanks also go out to the people of the SIGYN II project who have all made me feel very welcome at the company.

Table of Contents

Abbreviations

ACK – Acknowledgment
ARP – Address Resolution Protocol
CAN – Controller Area Network
CCU – Communication Control Unit
CERT – Computer Emergency Response Team
CPNI – Centre for the Protection of National Infrastructure
CVSS – Common Vulnerability Scoring System
DIS – Draft International Standard
DHCP – Dynamic Host Configuration Protocol
DoIP – Diagnostic communication over Internet Protocol
DDoS – Distributed Denial of Service
DoS – Denial of Service
ECU – Electronic Control Unit
EID – Entity Identification
GID – Group Identification
ICMP – Internet Control Message Protocol
IP – Internet Protocol
IPsec – Internet Protocol Security
ISO – International Organization for Standardization
MAC – Media Access Control
MDI – Medium Dependent Interface
NACK – Negative Acknowledgment
NDP – Neighbor Discovery Protocol
NIST – National Institute of Standards and Technology
OCTAVE – Operationally Critical Threat, Asset, and Vulnerability Evaluation
OSI – Open Systems Interconnection
OSSTMM – Open Source Security Testing Methodology Manual
PDA – Personal Digital Assistant
SIGYN – Software In Global Yielding Networks
SSL – Secure Sockets Layer
TCP – Transmission Control Protocol
TKIP - Temporal Key Integrity Protocol
TLS – Transport Layer Security
TOE – Target Of Evaluation
UDP – User Datagram Protocol
VIN – Vehicle Identification Number
WEP – Wired Equivalent Privacy
WPA – Wi-Fi Protected Access

1 Introduction

1.1 Background

In the automotive industry, diagnostics refers to examining the correctness of the operation of a car. In the past this has been done by plugging a cable into a port located in the passenger compartment of a vehicle. This port gives access to the internal network of a vehicle over which the diagnostic messages then can be sent.

A trend over the past few years has been to start equipping vehicles with capabilities enabling the diagnostic services to be carried out remotely, over the air, without the need of connecting a cable directly to a port in the car [1]. Remote diagnostics of a vehicle has a multitude of advantages compared to the traditional wired model. Diagnostics can be performed without the vehicle being taken to a repair shop, which saves both time and costs [2]. To put the issue of costs associated with maintenance into perspective, the same paper also provides an example from the European Commission which states that 40% of the total ownership costs accumulated during the lifetime of a single vehicle arises from resources spent on repairs. It is thus clear that rationalization in this department can lead to big savings.

As noted in [3], the automotive industry has in recent years begun to move away from brand-specific technologies and towards a greater degree of standardization. Diagnostic technologies are no exception from this and, as mentioned in the article, having a common interface leads to cost reductions as well as the possibility for actors in the automotive industry to change between different products of the same type. Since these are both desirable features, it is clearly also a good idea to have a standardized protocol for remote diagnostics. The ISO (International Organization for Standardization) is in DoIP (Diagnostic communication over Internet Protocol) working on such a standard [4, 5, 6]. As the name implies the aim of the protocol is to be able to use existing IP-based networks to carry the diagnostic messages between repair shops and vehicles. Having such a vast infrastructure already in place does of course have obvious advantages compared to constructing new networks from scratch.

With the Internet being traversed, not only benefits are introduced; a vast number of security issues are added to the ones existing when only a wired point-to-point connection to a port located inside the car is used. The previously discovered issues of TCP/IP (Transmission Control Protocol/Internet Protocol) might result in new and potentially severe consequences when entering a previously non-connected environment – the car.

In [7] the authors have performed an experimental analysis of what attacks can be carried out if an adversary has access to the internal network of a vehicle. They do not examine the attack surface, but instead focus on the possible implications of a breach. The results of the tests are staggering as the experimenters are able to control large parts of the vehicle's functionality, for example being able to disable the braking capabilities of a car in motion. The consequences of such an attack can clearly be fatal and it is thus absolutely necessary that a malicious intruder is not able to gain access to the internal

network. In order to guarantee this, there is a need for thorough security analyses of all technologies that enable connections to a vehicle, with remote diagnostics being no exception.

The work described in this report is a sub-project of the SIGYN (Software In Global Yielding Networks) II research project carried out at Volvo Car Corporation. The SIGYN II project encompasses a wide range of issues related to vehicles being connected to modern communication infrastructures such as the Internet and cellular phone networks. Within the SIGYN II context, the project detailed in this report investigates security problems that might arise from using DoIP.

Since DoIP still exists only as a draft and not a standard, there has not been any previous security-related work published on the subject. This work is therefore an important part of helping to gain a greater understanding of the security aspects of the protocol. As the more general research area of security related to the connected car has only recently begun to emerge, this work will hopefully also help in improving that knowledge base as well.

1.2 Purpose

The overall purpose of the work described in this report is to present a thorough security analysis which answers the question: can DoIP be used for vehicle diagnostics in a realistically modeled environment while fulfilling all the security attributes required to ensure the correct operation of safety-critical systems and thus avoiding harm to vehicle, infrastructure and driver?

In order to answer this question all the individual parts of a remote diagnosis are considered in their respective contexts. This comprises the environments from, over, and to which the communication is carried out as well as the protocol, DoIP, used for the message exchanges.

1.3 Objective

The specific goals of the project are the following:

- To describe a realistic model of a system communicating over DoIP along with security attributes that ensure the correct operation of any safety-critical systems within the vehicle.
- To come to a conclusion whether the DoIP protocol is secure to use or not within the established environment.

1.4 Scope

This work only considers the latest draft of the DoIP protocol as of the start of this project, the one on which the voting procedure began 2010-09-13. The draft includes the three documents ISO/DIS (Draft International Standard) 13400-1, ISO/DIS 13400-2 and ISO/DIS 13400-3 [4, 5, 6]. The study of the protocol is purely theoretical in the sense that implementation-specific vulnerabilities are not considered. To put it in other words,

only design flaws are considered while implementation and configuration flaws are not within the scope of this work.

The only part of the attack surface of the vehicle that is researched is the DoIP edge node connecting the car to its surroundings [5]. Other entry points to the internal network, such as maliciously constructed aftermarket components a consumer willingly installs in his vehicle [7], are not examined.

The analysis considers the security issues that arise from the fact that DoIP runs over networks based on TCP/IP. It does however not take into account problems that arise from protocols running on top of DoIP. Since security problems related to the specifications of TCP/IP protocols are already well known, this part does not produce any new research but rather summarizes previously established results and puts them into the context of vehicular communication. As with the analysis of the DoIP protocol itself, implementation specific issues of the TCP/IP stack are not within the scope of the work which focuses on design problems.

Real-time computing related issues such as how the sending and re-sending of messages affects timing-constrained systems are not researched within this project. These aspects also include the potential of security algorithms having a negative impact on response times of components.

1.5 Structure of the report

The first chapter of the report is an introduction which states the problem, describes the purpose and defines the scope of the project. The report then continues with the second chapter in which an overview of related work is given. A brief analysis of assessment methodologies is then described and the output of that analysis is the method used throughout the project. After the method has been specified, chapter four defines the DoIP system which is analyzed in the project. The description of the system includes the DoIP protocol as well as the communicating entities. In chapter five threats to the defined system are described in order to get a clear view of what an attacker might be capable of. Chapter six follows with the bulk part of the report, which describes an analysis of the vulnerabilities found in the DoIP protocol. Chapter seven contains a discussion of the results, chapter eight consists of an outlook on future work in the field and chapter nine ends the report with conclusions that can be drawn from the work carried out in this project.

2 Related work

No academic reports on the security of the DoIP protocol have been produced. In fact, no academic writing regarding any aspect of DoIP has been produced. In the more general area of security within the field of remote diagnostics some research has however been published. This previous work generally deals with security issues in remote diagnostics from a more abstract and conceptual perspective as opposed to the work described in this report which is directed towards the specific technology that is DoIP.

In [8], the author first does an assessment of the security risks in vehicle diagnostics and software updates over wireless links. The results of the analysis are then used to provide the foundation for a series of security policies and requirements needed in creating a secure infrastructure for the type of communication described.

In the area of securing the communication between vehicles and infrastructure in a more general sense, not specifically addressing diagnostics, a larger amount of research has been produced. [9] is a paper discussing how to handle cryptographic key management in automotive communication where the computing devices are typically significantly less powerful than traditional PCs.

The paper [10] describes how a secure lightweight protocol for the diagnostics related area of firmware updates over the air can be constructed. [11] continues the work on firmware updates over the air by presenting a framework for self-verification of these types of updates.

Continuing with articles on the more general issue of security in automotive communications, [12] motivates why vehicular networks need to be secured. The authors construct a threat model and thereafter propose a security architecture which they proceed by evaluating. [13] is another article that proposes a security architecture for vehicular communication systems complete with a walkthrough of security requirements necessary in the presence of a modeled attacker. The article discusses vehicle-to-infrastructure communication as well as vehicle-to-vehicle communication.

There are a number of papers describing how specific technologies can be used for the protection of automotive networks. These include research on honeypots [14] to gather information about attacker behavior as well as research on intrusion detection systems [15]. Some articles have also been produced describing results from experimental and theoretical analyses of what can be done when security has been breached and a connection to the internal vehicular network of a car is available [7, 16, 17, 18].

Finally, [19] describes a method for measuring the security of vehicular communication systems in a model-based fashion and can thus be used for evaluation.

3 Specification of analysis method

In this chapter the process of specifying the analysis method used for this work is described. It should be noted that the method is a composition of ideas from different methods as well as terminology taken from other research. In other words, a single previously existing method is not followed.

The analysis and comparison of the methods are carried out on a relatively high level and specific details treated with less importance.

3.1 Specification of requirements

In this section the requirements on the analysis method are specified. That is, what features are sought from the method.

Requirements:

1) Well defined and established terminology. That is, there should not exist any ambiguities, and terminology that has previously been used in the field is preferred.

2) Compact. Since this project is limited in time the methodology should be compact and easy to digest in order to not detract from the main part of this work, which is the analysis itself.

3) Adaptable. The method should be easy to adapt after the needs of the project.

4) Technical focus. Many of the existing methods have a broader focus than what is necessary for this project [20, 21]. Here the technical issues are the focal point, and organizational problems less relevant.

3.2 Descriptions of methods

This section contains descriptions and explanations of the different analysis methods studied. The descriptions are made on a format where an introductory text is followed by a list of the steps to take as formulated by the different methods. These steps are then further elaborated on in a language more in line with that of this report.

3.2.1 Common Criteria (Protection Profile)

The following description is derived from the Common Criteria standards document [20], the book Using the Common Criteria for IT Security Evaluation [22] and an example of a PP (Protection Profile) published by NIST (National Institute of Standards and Technology) [23].

The steps performed in the evaluation, using the terminology of the Common Criteria, are the following:

1) Definition of scope
The creation of a PP begins by defining the purpose and scope of the PP. That is, to define what issues the particular PP deals with and when it is applicable.

2) TOE description
The next step is the first part of the actual analysis. The goal of this step is to create a description of the TOE (Target of Evaluation), which is a term used for the system that is being assessed in the analysis.

3) TOE security environment
In the third step a description of the security of the environment the TOE operates in is created. Describing the security of the environment encompasses making analyses of the threats to the TOE as well as assessing the organizational security policies in use.

4) Security objectives
In this step the analyst states how the TOE and its environment are to counter the threats described in the previous step where the security environment was established. That is, what types of controls and mitigation efforts are to be utilized.

5) Security requirements
In the final step it is stated of how the TOE and its environment are supposed to behave with respect to security. That is, what security attributes are fulfilled and to what degree.

To summarize, the main ideas of the protection profile can be generalized into the following abstract steps:

1) Description of the system.
2) Description of the environment of the system.
3) Threat assessment.
4) Description of security controls.
5) Description of the security requirements the system is supposed to fulfill.

3.2.2 OCTAVE Method

The description in this section is derived from the OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) method implementation guide v2.0 [21].

The OCTAVE method uses the terms "phase" and "process" to denote the different steps of the analysis. The assessment is divided into three major phases which in turn each consists of a number of processes.

1) Phase 1: Build asset-based profiles
In the first of the three phases the critical assets of the organization are identified along with their current protective measures. The result of this phase is a series of threat profiles, where the threats to each identified asset is described.

2) Phase 2: Identify infrastructure vulnerabilities
In the second phase the IT infrastructure is analyzed in order to find possible security problems that if exploited might lead to one or more of the critical assets being compromised in some way.

3) Phase 3: Develop security strategy and plans
With the results of the previous phases as input the final part of the assessment consists of conducting a risk analysis of the identified threats and then creating a security strategy to mitigate any issues.

The phases can be summarized in the following way:

1) Description of the system (to protect).
2) Threat assessment.
3) Vulnerability assessment.
4) Risk analysis.
5) Construction of security controls.

3.2.3 OCTAVE Allegro

OCTAVE Allegro is a more streamlined and compact OCTAVE method, and the information used in this section stems from the official sources [24].

The method is conducted by following the eight steps as specified below:

1) Step 1 – Establish risk measurement criteria
Analyze the impact to the organization if a risk to an asset was to materialize as a compromise. That is, what are the consequences if an attacker exploits a vulnerability.

2) Step 2 – Develop an information asset profile
In the second step, the objective is to create descriptions of the assets. Of importance here is to clearly define the borders of the assets in order to separate them from the rest of the system.

3) Step 3 – Identify information asset containers
Create descriptions of the system and the sub-systems that the different assets reside in. The sub-system here is not necessarily a piece of hardware or software, but might instead be a piece of paper or a human being.

4) Step 4 – Identify areas of concern
Make an analysis of the problem areas of each previously identified asset. That is, identify the vulnerabilities that have the potential of being exploited.

5) Step 5 – Identify threat scenarios
Connected to the previous step, a further analysis of the threats to the assets is carried out. In this part, the scenarios in which a threat can lead to a compromise are identified.

6) Step 6 – Identify risks
In this step, threats are connected to vulnerabilities in order to see what the different vulnerabilities might be exposed to.

7) Step 7 – Analyze risks
Analyze the risks and how they might affect the organization. This is a step in which measurements of the different risks are made in order to calculate their respective severities and probabilities of occurrence.

8) Step 8 – Select mitigation approach
Construct security controls in order to mitigate the risks that were found to lead to severe consequences for the organization.

These eight steps can be compressed into the following:

1) Description of the system.
2) Vulnerability assessment.
3) Threat assessment.
4) Risk analysis.
5) Construction of security controls.

3.2.4 NIST SP 800-30

This description is derived from the NIST-authored specification as found in [25].

The method defined in NIST SP (Special Publication) 800-30 consists of two major parts, risk assessment and risk mitigation, which are then divided into smaller parts as elaborated on below:

Risk assessment:

1) Step 1: System characterization
The assessment begins with the creation of a description of the system. In this step the borders of the system are defined. Then descriptions of the specific characteristics of the different parts of the system are made. This step could also be said to include the asset definitions that are more specifically defined in some of the other assessment standards.

2) Step 2: Threat identification
In this step, the threats to the system are identified. That is, the sources of possible threats are identified and then the potential they might have of exploiting specific vulnerabilities is evaluated.

3) Step 3: Vulnerability identification
The vulnerability identification step is a straight forward walkthrough of the flaws and weaknesses of the system.

4) Step 4: Control analysis
After the vulnerabilities have been identified, the current security measures are analyzed. This step contains an examination of both organizational and technical preventive efforts in place.

5) Step 5: Likelihood determination
In the fifth step, the probability of potential vulnerabilities being exploited is measured. This analysis takes the threat-source motivation and capability, the nature of a specific vulnerability, and the currently implemented controls as input to produce a likelihood rating as output.

6) Step 6: Impact analysis
The impact analysis is an examination of the potential impact to the organization if threat-sources were to exploit specific vulnerabilities. Topics such as data sensitivity and system criticality are considered in this step.

7) Step 7: Risk determination
The risk determination combines the results from the two previous steps in order to measure the risks. The objective is to get a quantification that is exact and easy to compare.

8) Step 8: Control recommendations
In this step, controls are constructed in order to mitigate the previously determined risks. This creation of controls is both a technical and organizational matter. That is, the resulting output might be a combination of both policy documents and technical controls such as firewall solutions.

9) Step 9: Results documentation
In the final step of the risk assessment procedure, the analysts document the findings from all the different steps in a structured report.

Risk mitigation:
Determine what risk mitigation efforts should be implemented and what risks can be accepted. This analysis for example includes comparisons of the costs associated with security breaches to the costs related to implementing controls and mitigation efforts.

The steps can be summarized into the following:

1) Description of the system.
2) Threat assessment.
3) Vulnerability assessment.
4) Analysis of existing controls.
5) Risk analysis.
6) Construction of security controls.

3.2.5 Other methods

Except from the previously compared methods, a couple of others were studied as well. These other candidates were however not chosen for the main comparison for different reasons:

CVSS:
The CVSS (Common Vulnerability Scoring System) [26] was considered but was ultimately dropped because of its focus on application-specific vulnerabilities.

OSSTMM:
The OSSTMM (Open Source Security Testing Methodology Manual) [27] was also under review but was dropped because of its verbosity and the fact that the key ideas are not strictly defined. Issues with the method are also noted by [28] which critiques the intuitiveness of OSSTMM.

ISO 27000:
The ISO 27000-family of standards, and foremost the ISO 27005 [29], was considered as well but did not make it into the final comparison since it is not freely available which had disadvantages both related to the cost itself, but to circulation as well. In [30] it is also noted that only the ISO 27001 and ISO 27002 can be seen as mature standards at this time.

3.3 Comparison and conclusion

As also noted in [31], the comparison conducted for this project shows that the listed methods tend to have a series of common steps. Steps that are common to all the analyzed methods, even though possibly carried out in different orders, are the following:

1) System description
An introductory part which describes the specific target being evaluated along with the surrounding system according to well defined boundaries. In more organizationally oriented methods, the output of this step might be descriptions of a series of systems, sometimes denoted assets, that are important enough to warrant protection.

The system description also contains definitions of security requirements on the system. That is, properties derived from the criticality and sensitivity of different areas of the system.

2) Threat assessment
An analysis of the threats, and their possible sources, to the system specified in 1).

3) Vulnerability assessment
An analysis of the vulnerable points in the systems. That is, components with flaws that could possibly be exploited if in the presence of a threat.

4) Risk analysis
An analysis of the risks to the system. A risk is a combination of the probability of a threat source exploiting a certain vulnerability and the resulting impact if successful.

5) Construction of security mechanisms
In the final step security solutions are devised in order to mitigate or completely abolish any identified risks. These controls are motivated by the risk, as identified in 4), of the required security properties, as specified in 1), being violated.

As these steps are present in all the well-established risk management methods studied, these are also mostly followed in the assessment made in this project. The risk analysis part is however less applicable than the others to this particular project as it is hard to measure the probabilities when dealing with a system that is yet to be implemented, and is of a type that has not previously been in large scale use. Thus, no statistical data is available to take foothold in and proceed from. The risk analysis is for this reason not within the scope of this work. Neither is the construction of security mechanisms as this project is a pure analysis and the creation of security controls external to the DoIP protocol are thus out of scope.

3.4 Description of analysis method used for the project

To begin with, the steps as specified in the previous section are followed. Having a short series of actions to carry out instead of using one of the studied methods from beginning to end, with all details followed in between, ensures that the requirements for compactness and adaptability are met. The technical focus is also kept since the steps are defined at a high level, and thus not filled with organizational or business-oriented demands.

In order to follow the demands on the terminology used during the analysis a couple of taxonomies have been chosen to define the meanings of the words used. Definitions of the incident and attack terms used throughout the paper are taken from [32] which is chosen on account of being an adaption of the CERT (Computer Emergency Response Team) taxonomy [33] focused on the automotive industry. For the description of anti-intrusion concepts, [34] is selected as it has been used by previous research, such as [35], in the field.

Where a more detailed description of the issues included in the above steps is needed, the NIST SP 800-30 is consulted. This method was chosen over the others for a number of reasons. The first being that its terminology fits well with the chosen taxonomies described in the previous paragraph. The second that it is structured in a clear and concise manner where the individual steps are easily mapped to those stated in the Comparison and conclusion section 3.3. The third and final being that it is a mature standard that has not changed since 2002.

4 System description

In this chapter a model of the entire system from endpoint to endpoint is defined and described.

DoIP is not supposed to be used as a stand-alone application layer protocol. The protocol should rather be considered as an interface between TCP/IP and higher level protocols, such as ISO 14229-1 [36], which define the specific diagnostic services. DoIP is in other words a container for the diagnostic requests and replies while they travel over traditional TCP/IP networks.

Table 1: Relation between layers and ISO documents.

Layer	**Document**
Application	E.g. ISO 14229-1
Transport	ISO 13400-2
Network	ISO 13400-2
Data link	ISO 13400-3
Physical	ISO 13400-3

In Table 1 above, a division of diagnostic protocol documents into the layers of the IP stack is depicted. The scopes of the individual DoIP documents are as follows. ISO 13400-1 contains general information and defines the use cases that are applicable to DoIP [4]. ISO 13400-2 describes the transport and network layer services [5]. It states requirements and defines different phases of communication in the protocol. ISO 13400-3 is a document specifically describing Ethernet based wired transmission [6]. The third document is the only one describing specific requirements on data link and physical layer technologies. Since there is currently not any draft available for other technologies, such as for wireless transmissions, this document is not further analyzed. The lower layers of the IP stack are instead treated on a more abstract level, in accordance with the use case definitions of ISO 13400-1, so that all lower layer technologies are analyzed with the same amount of detail.

The drafts are under heavy development, and the assumptions made in this project are not guaranteed to be valid for future documents such as the final standard. This work focuses on the drafts with a voting period that terminated on 2011-02-13 and all DoIP document references throughout this report point to these drafts.

4.1 Network endpoints

The network endpoint under the control of a workshop mechanic is called the external test equipment. It is also referred to as external test tool or external tester. This is the unit used to send diagnostic requests to the vehicles that are subject to particular tests. That is, the external tester sends a diagnostic request to a vehicle and then waits for a reply which will contain data about the status of the car.

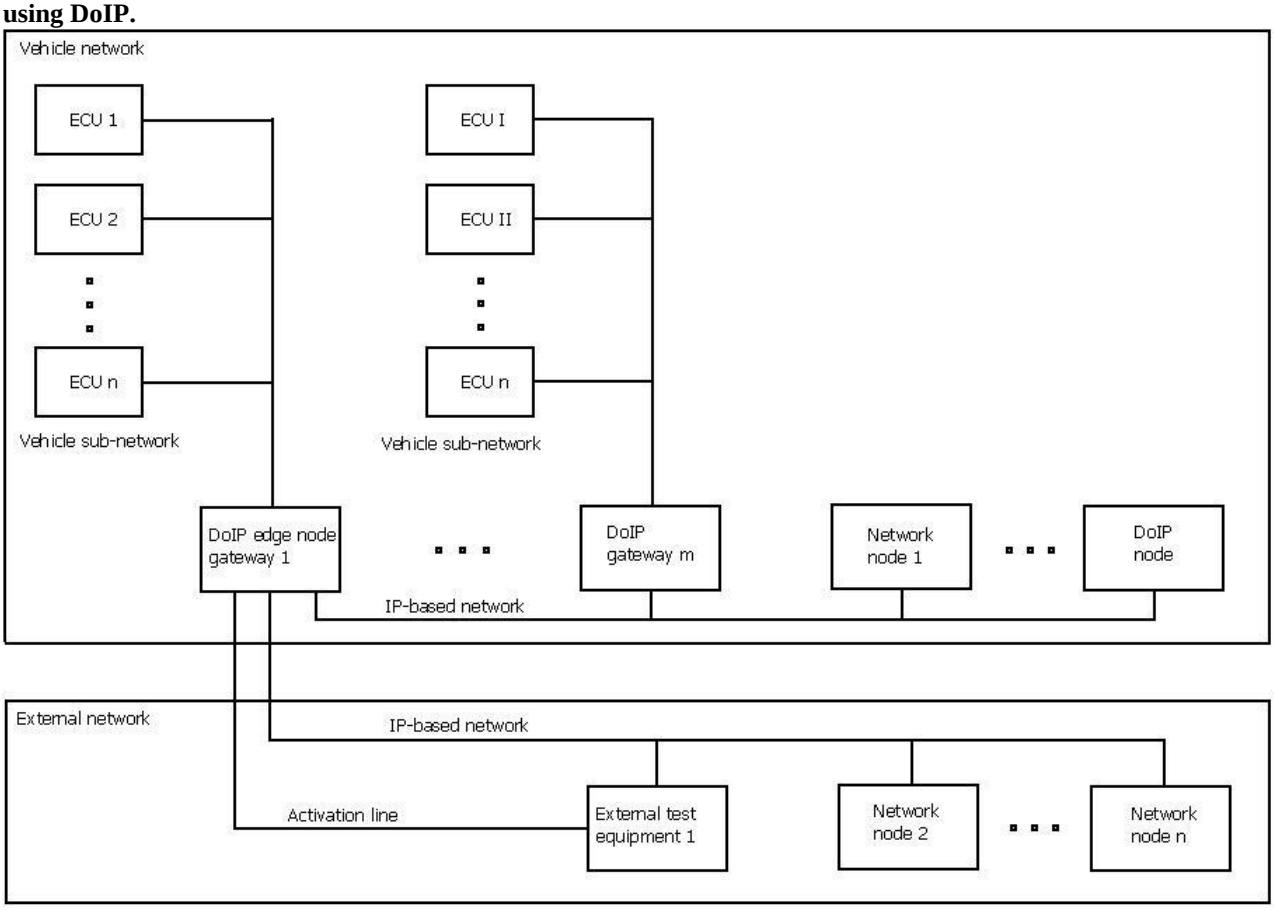
The external test equipment is physically a DoIP-capable device of any kind, for example a stationary PC, a laptop or a PDA [4]. Its storage and processing capacities are assumed to be powerful enough to not affect the choice of security algorithms. The

other network endpoint is the vehicle which is the subject of the diagnosis. The part of the vehicular network that receives messages sent using the DoIP protocol is called the DoIP edge node [5]. This is a logical concept that is implemented in the physical unit known as the CCU (Communication Control Unit) [37]. The storage and processing capacities of the nodes in the vehicular network are also assumed to be powerful enough to not affect the choice of security algorithms.

4.2 Logical view of a DoIP network

This section presents a brief overview of the logical model used by DoIP to describe a vehicular network along with how it is connected to external equipment.

Figure 1: Example architecture of an internal vehicular network connected to an external network using DoIP.



In Figure 1 above is a logical depiction of the DoIP network of a vehicle connected to an external tester. Of the two outermost boxes, the one placed on top depicts the vehicular network consisting of a series of nodes, some capable of communicating using DoIP and some not. The entry point for each sub-network of ECUs (Electronic Control Units) is the respective DoIP gateway which acts as a bridge between the DoIP network and for example CAN (Controller Area Network) networks that connect the more internal ECUs. The DoIP gateway that accepts external connections is called the edge node, and is among other things responsible for communication with the external test equipments during diagnostics.

The bottom one of the two outermost boxes depicts an external network, where the main point of interest is the external tester connected to perform diagnostics. The activation line is an Ethernet-specific feature and is not discussed further in this report. Although the name might imply that it is security-related, the feature is only there to ensure reduction of power consumption as well as to decrease the electro-magnetic interference [6]. The remaining external network nodes are ordinary non-DoIP related networked nodes. They might for example be workstations responsible for other tasks than communicating with the vehicles in a repair shop. For example billing systems.

A final, but important, remark is that the IP-based network is an arbitrary IP-based network of any kind. That is, it is not necessarily a cabled network even though the figure might suggest so.

4.3 Physical and link layer assumptions and characteristics

The DoIP draft standard does not impose many requirements or restrictions on the link or physical layers. The technologies that are listed in this section are therefore taken from the use case definitions in part one [4] of the documentation and are described here because of their differing characteristics with respect to security. These are interesting to note since one of the possible configurations, described in the Network configurations section, consists of only a single Ethernet cable, while others consist of multiple links where wireless links might be included. The security characteristics of those two extremes are quite different [38].

An Ethernet cable is in essence a private medium. There are some possibilities to listen in on the communication passing over a cable by reading the emitted radiation if the medium is not shielded well enough [39]. This is however an extreme case, and in this work it is assumed that a cable will not leak. It is also assumed that the cable is under some form of supervision, eliminating the possibility of the cable being split and any type of device inserted.

WLAN is essentially an open medium, not considering protective mechanisms such as encryption. In other words a message sent can be eavesdropped as well as altered by anyone within the transmission range. Seeing as advanced antennas are easy to construct [40] the range should generally not be seen as an obstacle for a person wishing to catch a signal.

When it comes to protection of the WLAN traffic, three main methods exist. These are WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) and WPA2 together with different methods for encryption and key assignment [41]. Of these, WEP is considered to be entirely broken [42]. Furthermore, WPA together with pre-shared keys suffers from an off-line dictionary attack. Implementations of WPA and WPA2 utilizing the TKIP (Temporal Key Integrity Protocol) protocol for encryption also have known vulnerabilities [43].

In [42] it is stated that WEP was used in as many WLANs as 76% and 85% for two big city regions in 2006. The adoption of WPA and WPA2 should surely have increased since then. But considering this, along with the weaknesses of certain configurations of

both WPA and WPA2, and for reasons of simplicity and consistency, the assumption made in this work is that wireless traffic is in general not guarded in any way at the physical or link layer. In other words, mechanisms have to be provided at higher levels in order to enforce security.

Depending on whether IPv4 or IPv6 is used, different technologies are in place for services such as address resolution. For version four, ARP (Address Resolution Protocol) is used for the resolution between network and physical addresses. For version six, NDP (Neighbor Discovery Protocol) is instead used for these services.

4.4 Network and transport layer assumptions and characteristics

This section describes what protocols are used on the transport and network levels, and also details the characteristics of the transmissions made over these. The description is made in a high level fashion. For more specific requirements and details, part two of the DoIP draft documents [5] is referred to.

The network layer protocol that is used for transmissions within the specified system is the Internet Protocol. DoIP has support for both version four and version six, where IPv6 [44] is recommended but IPv4 [45] is included for compatibility reasons and easy integration into existing networks. Some characteristics of the Internet Protocol, regardless of version, that are relevant for this project as they have implications on security are:

- Public. That is, anyone can access and use the network and any communication might be sent over unsafe links where it can be seen and manipulated.
- Packets might be dropped or lost. At any point in the network any packet might be lost or simply dropped.
- Packets might be delivered out of order.
- Packets might be delivered erroneously. That is, they might be corrupted along the route.
- No bounds on delay or jitter. The delays might be infinitely long, vary infinitely often and infinitely much.

DoIP uses both of the two major transport layer protocols, TCP [46] and UDP (User Datagram Protocol) [47], but for different services provided by the protocol.

TCP is a protocol that provides reliable in-order delivery of packets. UDP on the other hand provides no guarantees on the delivery of transmissions made using it. Since it is less complex than TCP and also more aggressive, it is also somewhat faster and is therefore often used for purposes where delivery in time is more important than delivery at all. In DoIP communication, the UDP protocol is mainly utilized for **identification** messages while TCP foremost is used for the transport of **routing** and **diagnostics** messages. The different message types are described in sections 4.7 and 4.8.

To summarize, it is important to note that data transmitted using the DoIP protocol might be subject to packet loss, out-of-order delivery, delay and jitter while travelling

exposed over a public medium where interference and eavesdropping is possible. Depending on what transport layer protocol is used for the current service, some of the issues might be solved by that layer while others must be treated by the application.

4.5 Application layer assumptions and characteristics

No assumptions are made regarding the technology used above DoIP on the application layer for the diagnostic transmissions. The protocols above DoIP might, from the perspective of this work, be of arbitrary nature and considered out of the scope of this work. Rationally however, some kind of diagnostics protocol is likely to be used. An example of such a communications protocol would be ISO 14229-1 which specifies "Unified diagnostics services" as stated in [36].

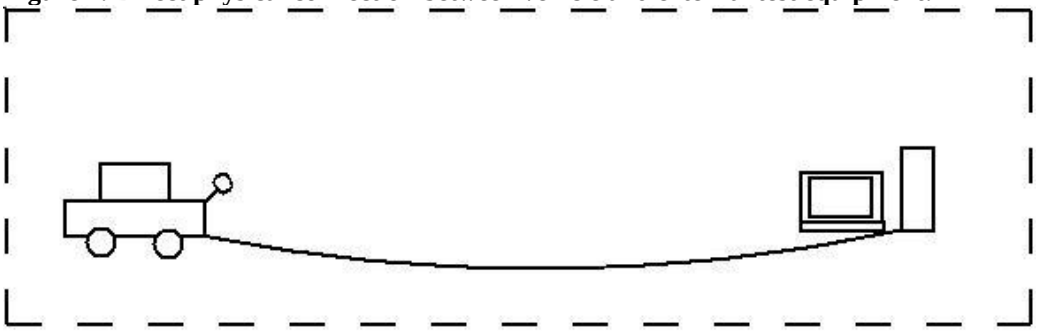
The DoIP draft does specify the use of DHCP (Dynamic Host Configuration Protocol) for configuration of the nodes before their communication. Issues related to this protocol are hence also included in this work, even though DHCP is only used for configuration of hosts prior to the actual diagnostic transmissions taking place.

4.6 Network configurations

This section contains descriptions of the four possible types of configurations a DoIP network can have according to the draft documentation [4]. The different connections are described at a high level without going into different details about the possible topologies since these can be arbitrarily many. The individual nodes and characteristics of the transmission mediums are further defined in other sections of this chapter. Each image presented depicts an example of the specific configuration and there are of course other possibilities as well. For example, all the networked connections might potentially include the Internet.

Direct physical connection between one vehicle and an external tool
In the least complex case, a single tester is connected to a single vehicle by a direct physical connection. In this scenario, the physical connection always consists of a cable since the DoIP draft demands that: "It is also always clear by the physical connection with which vehicle the external test equipment is communicating" [4]. This can clearly not be guaranteed using a wireless medium.

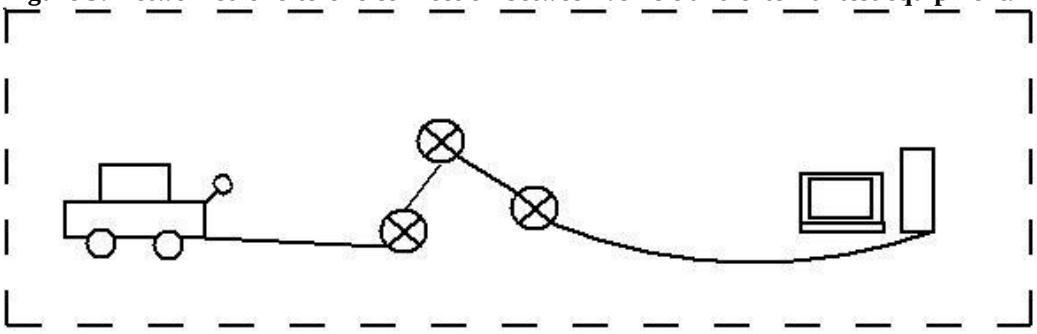
Figure 2: Direct physical connection between vehicle and external test equipment.



In Figure 2 above is a depiction of the direct cabled one-to-one connection between a vehicle and a tester. The tester in this particular example is a PC.

Networked connection between one vehicle and an external test equipment
The second scenario also describes a one-to-one connection, albeit over a different type of connection. In this configuration the connection passes over an IP-based network positioned in between the two communicating endpoints. The exact size or topology of this network is not defined by the draft standard and it might thus have arbitrary characteristics and might consist of anything from a single router to the Internet.

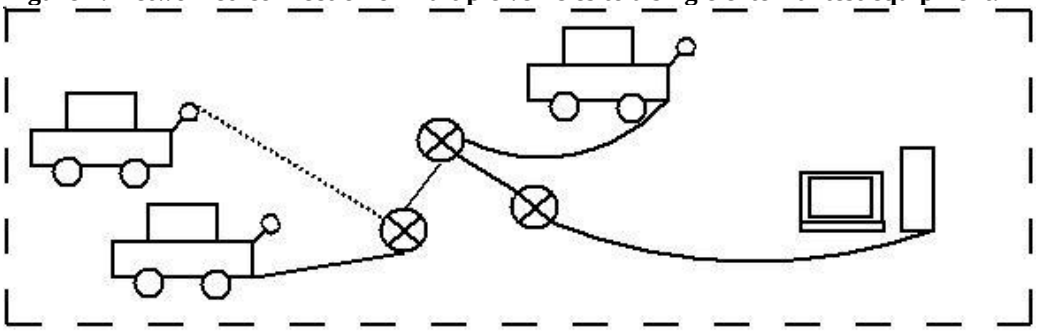
Figure 3: Networked one-to-one connection between vehicle and external test equipment.



In Figure 3 above is an example of what a networked one-to-one connection might look like. The connection in the figure passes over three different routers before reaching its final destination. The endpoints of this image are, just as in the previous scenario with the direct cabled connection, a vehicle and an external test equipment in the form of a PC. In this example, all intermediate connections consist of wired links but they might just as well be wireless. As previously stated, there are no limits on the number of links contained in the intermediate network.

Networked connection between multiple vehicles and one external test tool
In this scenario, the networked one-to-one configuration of the previous sub-section is expanded to a connection of the type many-to-one, where a number of vehicles are connected to a single external tester.

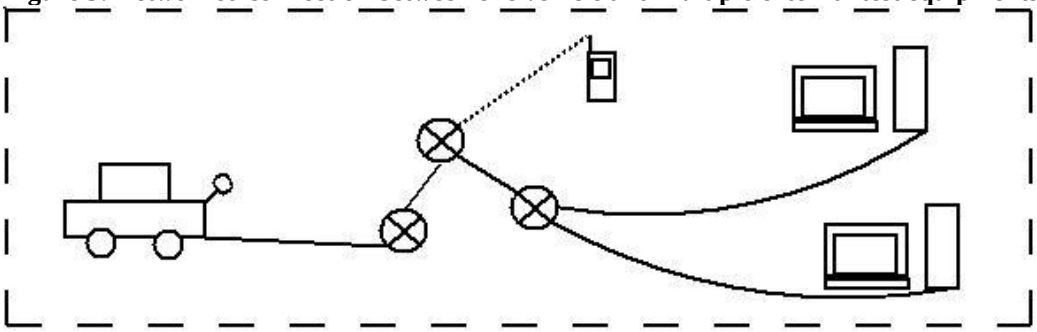
Figure 4: Networked connection of multiple vehicles to a single external test equipment.



In Figure 4 above, an example of a networked connection between a single tester and multiple vehicles is depicted. In the example, there is a mixture of different link types with the dotted line indicating a wireless communication link. The tester is once again modeled as a stationary PC.

Networked connection between one vehicle and multiple test tools or test applications on a single physical tool
In the final configuration, the scenario is the opposite of the previous one; that is one vehicle is connected to multiple test tools.

Figure 5: Networked connection between one vehicle and multiple external test equipments.



In Figure 5 depicted above is an example of a connection between one vehicle and multiple test tools. In this particular case, the test equipment consists of two PCs and one PDA, each carrying out a separate diagnostic session with the vehicle.

4.6.1 Summary of network configurations

All the possible DoIP network configurations can, according to the use-cases specified by the first DoIP draft document, be abstracted into four general network models [4]. All the models have endpoints made up of either vehicles or testers. The differences between the configurations are mainly based on two parameters. The first one being whether there is a networked connection or a direct cabled connection in between the endpoints. The second one being the multiplicity of the endpoints.

The four possible configuration scenarios are:

- Direct physical connection between one vehicle and an external tool
- Networked connection between one vehicle and an external test equipment
- Networked connection between multiple vehicles and one external test tool
- Networked connection between one vehicle and multiple test tools or test applications on a single physical tool

4.7 DoIP phases of communication

This section describes the major phases of the DoIP communication. The division into phases is not stated clearly in the draft documents but is made in the work described in this report since it is a relevant way to split the communication into smaller parts that are easier to analyze separately. For a specification of all the fields contained within the messages mentioned, the second part of the DoIP draft documents is referred to [5]. In order to make the assessment easier to follow, these fields will also to some extent be shown in the security analysis of the DoIP protocol. Table 2 contains a listing of the three phases which are then further elaborated upon in the paragraphs following underneath.

Table 2: Phases of communication in DoIP.

Phase
Announcement/Identification
Routing activation
Diagnostic communication

In the first phase, **Announcement/Identification**, the vehicle and its individual DoIP entities are identified by the external test equipment. This process can be carried out in two different ways. The first one of these is when the tester simply receives the **Vehicle announcement** messages sent out by newly configured DoIP entities. The message contains relevant identification information such as VIN (Vehicle Identification Number), EID (Entity Identification), GID (Group Identification) and logical address [5].

The second identification possibility is useful if the tester has missed the original announcement messages. Then the tester can send out **Vehicle identification request** messages requesting the DoIP entities to identify themselves. A message from a responding DoIP entity will be of the same format as the **Vehicle announcement** messages.

Before diagnostic messages can pass from a tester, through a DoIP gateway in the vehicle and on to the internal network, routing on a TCP_DATA socket in the gateway must be activated. The routing is activated with a simple request-response scheme where the tester sends a **Routing activation request** containing the logical address of the tester along with the type of activation it seeks. The DoIP entity then responds with a **Routing activation response** containing its logical address, along with the requester's, and a response code noting if the activation attempt was successful or not. Logically, this phase is referred to as the **Routing activation** phase.

When the routing has been activated, the **Diagnostic communication** phase can start. That is, a series of diagnostic messages sent from the tester to the DoIP gateway which in turn routes the packets on to the logically addressed ECUs as specified in the messages. These transmissions might then be followed by diagnostic messages, containing response data, sent in the opposite direction. All requests do however not produce responses. Important to note here is that the actual diagnostic requests and responses are not specified by DoIP, but by higher layer protocols, such as ISO 14229-1 [36]. DoIP only specifies containers for the diagnostic messages, but does not define the diagnostic functions.

4.8 DoIP message groups

This section contains descriptions of the different groups of messages used in the DoIP communication. These groupings are specified in the technical description of the protocol, in the second part of the draft documents [5], and are shown in Table 3 below with further descriptions following.

Table 3: DoIP message groups and the byte values identifying them.

Payload type value (identifier)	Message group
0x0XXX	Node management
0x4XXX	Vehicle information
0x8XXX	Diagnostics

The first of the message groups is called **Node management**. As the name implies, this group consists of messages used for node administration. Looking at the communication phases, the messages of the announcement and routing activation phases belong to this category together with the alive check messages used by DoIP entities on the vehicle side to check if a connected tester is still active.

The second of the three categories is the **Vehicle information** group. The messages in this group are used to gather entity and vehicle specific information that might be useful before performing diagnostics. This might for example include retrieving the diagnostic power mode in current use along with other operating conditions of the vehicle.

Finally, the third group, **Diagnostics**, contains the purely diagnostic messages. That is, the encapsulation of the commands sent by the application layer diagnostics protocol in place.

4.9 Assets to protect

The assets identified as those that are important to protect, with respect to DoIP communication, are the endpoints of the previously described networks of the configurations section. In other words, the vehicle and the external test equipment are the assets that need to be protected.

Since this work is about security issues related to DoIP, the real asset of the vehicle is the internal network of the car. As explained earlier, the vehicular network is connected to the outside world through the DoIP edge node located in the CCU. Communication deemed legitimate by the gateway is then routed on to the internal network. Studies [7, 16, 17] have shown that the actions that can be performed given this type of access might indeed result in severe consequences.

The tester itself needs to be protected since an attacker in possession of a legitimate test device in essence would have direct access to the internal networks of vehicles, albeit through some kind of communication medium or network. The databases the tester potentially might have access to are also included in this asset. That is, customer records etc.

As noted, the assets within the system that need to be protected are the vehicle and the external test equipment. However, to summarize, one can in a way say that it is really the communication that is the asset that is to be protected. If the messages transmitted are guarded in accordance with the requirements specified in the following section, the endpoints will as a consequence also be protected with respect to DoIP-related security issues.

4.10 Security requirements

The taxonomy used to specify the security attributes in this report is the same as used by the European Commission-funded project called EVITA (E-safety vehicle intrusion protected applications). The properties are specifically taken from the document [48] produced by the project. At its core, the taxonomy has the common CIA (Confidentiality Integrity Availability) model [39] which is extended with a number of extra properties. The taxonomy has been chosen as it is clearly relevant for the automotive industry.

The basic terms defined by the EVITA project are used throughout this work, although adapted to fit with the characteristics of this specific type of system where communication external to the vehicle is focused on. It should be noted that some of these terms are largely overlapping, but all are still included for the sake of consistency.

Table 4: List of security attributes.

Attribute
Data origin authenticity
Integrity
Controlled access (authorization)
Freshness
Non-repudiation
Privacy/anonymity
Confidentiality
Availability

Table 4 shows a list of the security attributes in the order in which they are described in this section.

Data origin authenticity
This property ensures that the source of a message is verifiable. The receiver of a DoIP message should in other words be able to authenticate that the claimed source is actually from where the communication came.

Applicability to DoIP in the specified system:
When fulfilled, this **data origin authenticity** will make sure that the vehicle can verify that diagnostic requests come from a trusted external test equipment, and the tester can in turn be asserted that it indeed gets responses from the vehicle it seeks to communicate with. That is, responses are not made by another entity on behalf of the vehicle.

Possible implications if the security attribute is not fully upheld:
If not fulfilled, a user with malicious intent could pretend to be an authorized party in order to have potentially dangerous commands accepted by a receiving entity. Depending on the diagnostic services provided by the protocol running on top of DoIP, the results might differ. If the commonly used ISO 14229-1 is in place, data might be written and thus malicious software uploaded [36]. As proven in [7, 17] this could for example allow the adversary to disable the brakes of a vehicle in motion.

Integrity
When satisfied, the **integrity** property guarantees that a message has not been altered, maliciously or by random chance (failures or physical effects), in transit. That is, the data received is identical to the data sent.

Applicability to DoIP in the specified system:
The integrity attribute ensures that an unauthorized party cannot modify commands or data being sent in the DoIP messages.

Possible implications if the security attribute is not fully upheld:
Modifications could for example mean that an attacker intercepts a message and exchanges a contained command for another. It also allows an attacker to alter software being transmitted and the scenario described under **data origin authenticity** might thus also occur if **integrity** is not guaranteed.

Controlled access (authorization)
This property describes how different entities are allowed to access resources, for example at other nodes. It is fulfilled if the entities allowed to perform certain actions are also the only ones able to perform them.

Applicability to DoIP in the specified system:
Authorization can be used to make sure that only legitimate external test equipment is allowed to be perform diagnostics on a vehicle.

Possible implications if the security attribute is not fully upheld:
There is an obvious danger in giving arbitrary actors the ability to execute diagnostic commands on a vehicle. Not restricting access means that the running example is valid in case of the breaking of this property as well since not enforcing **authorization** would mean that any software uploads would be accepted.

Freshness
The **freshness** property is satisfied if information received is always current. That is, a message received is guaranteed to not be a copy of a previously transmitted piece of information.

Applicability to DoIP in the specified system:
In the specified setting this property entails that a previously sent legitimate diagnostic request cannot be re-transmitted as is.

Possible implications if the security attribute is not fully upheld:
A command that is not dangerous given a certain scenario, might be potentially lethal in another. Say that a workshop mechanic sends a diagnostic command to start a routine [36] that releases the brakes of a car in order to test their functionality. This would of course be carried out under controlled conditions within a repair shop, but imagine that an adversary eavesdrops on the workshop network and records the message sent. If **freshness** is not ensured this legitimate message could be re-transmitted later on with malicious intent when the previously tested vehicle is at speed. It should also be noted

that this attack can be carried out even if the messages are guarded by encryption and authentication mechanisms as the adversary does not need to alter the transmission in any way, but can send the message as-is to produce the sought result.

Non-repudiation
Non-repudiation is an attribute which requires that an entity having performed an action cannot claim that it did not. In other words, actions can be traced and proven to have been performed by certain entities.

Applicability to DoIP in the specified system:
If damage to vehicle, passengers or surroundings arise as the result of one or several diagnostic messages the origin of said communication can be proven. This is for example useful in order to uphold legal accountability.

Possible implications if the security attribute is not fully upheld:
While **non-repudiation** does not help in preventing incidents from happening, it does ease the forensic work carried out afterwards in identifying the source of an attack. A problem that could arise from not ensuring the **non-repudiation** property would be to prove legal accountability in the case of security incidents resulting in accidents.

Privacy/anonymity
Privacy is a property assuring that information about a certain entity stays confidential. **Anonymity** is a special case of privacy referring to the confidentiality of the identity of an entity.

Applicability to DoIP in the specified system:
In a diagnostics system this property makes sure that information about a vehicle and its owner is not available to unauthorized parties.

Possible implications if the security attribute is not fully upheld:
The potential issues very much rely upon what kind of information is stored in and accessible from the vehicle. If sensitive data, such as credit card information or related details, is stored and accessible through diagnostics the consequences might be serious. Information about the state of the car is probably not very useful for the average attacker, but such issues might be considered in extreme cases.

Confidentiality
The property of **confidentiality** is a broader and more general concept of secrecy than privacy. This requirement pertains to the secrecy of all information transmitted, regardless of whether it can be connected to a specific entity or not.

Applicability to DoIP in the specified system:
A malicious user seeing the contents of the commands and data being sent can use this information in order to get a view of potential problems with the car. This could possibly later be used in order to launch an attack.

Possible implications if the security attribute is not fully upheld:
If the **authorization**, **freshness** and **integrity** properties are fulfilled only information that could possibly violate the privacy and anonymity properties is disclosed. That is, the information will most likely not help in mounting attacks with consequences that could be of potential danger to human beings, vehicles or surroundings.

Availability
Availability is a property that is satisfied as long as the service being investigated is functioning. That is, as long as the service is available.

Applicability to DoIP in the specified system:
The **availability** requirement is satisfied in the specified DoIP system as long as the diagnostic messages sent reach their intended targets which also process and answer the transmissions in accordance with the draft standard.

Possible implications if the security attribute is not fully upheld:
Breaking of the **availability** property will lead to annoyances, but disruption of diagnostic services is not likely to endanger human life, vehicle, or surroundings. It might however cause harm to the brand of the service provider.

4.11 Summary of the system

The endpoints of the system consist of *at least* one external test equipment performing diagnostics on *at least* one vehicle. These diagnostics are performed over a network which may be built from anything between a single cable to a network of arbitrarily many and mixed cables and wireless links even crossing the Internet. In the worst case, seen from a security perspective, the transmissions thus pass a public medium, the Internet, accessible to anyone with any intent.

In this system the communicating entities should satisfy the requirements specified in the previous section in order to prevent harm to human beings, vehicles, or the environment.

5 Analysis of environment

This chapter discusses the threats to the previously specified system in order to derive a model of the attacker. That is, a model describing what the possible actions an attacker might take are. Some inspiration for the structure is taken from [12] which also uses the motivation, method, and membership terminology. This work does however use a more detailed classification where the capabilities of the attacker are also connected to the various network configurations which are possible in a DoIP system.

5.1 Attacker motivation

The motivation of an attacker can be divided into two main broad classes, malicious and rational. A rational attacker is an attacker only attempting attacks that can produce benefits for himself. For example this class includes various kinds of chip trimming or tuning and other modifications to the own car. This class is however not dealt with in this work, since DoIP is an unlikely attack vector for this type of exploit as there are substantially easier methods of manipulation having full physical access to an vehicle.

The focus is on the second of the two classes, namely the malicious attacker class. The attacks coming from this class does not necessarily produce any benefits for the attacker and can thus be more random in their nature [12].

For more specific descriptions of the potential motivation of the attacker, [33] and its automotive adaptation in [32] is referred to since an in-depth analysis is not within the scope of the project.

5.2 Attacker method

The method of the attacker is divided into two separate classes, active and passive. A passive attacker is someone who simply eavesdrops on communication without disturbing or altering it in anyway. An active attacker on the other hand participates in the communication during the attacks. This activity might consist of intercepting and modifying messages being sent, or possibly deleting them, or even injecting new messages into the stream.

5.3 Attacker membership

The membership property describes if an attacker is a legitimate user of a network. An insider in the scope of this project would thus be an authenticated tester or vehicle. The case of an attacker in possession of an external test equipment is however not considered as it has already been proven what harm can be caused in such a scenario [7, 16, 17]. Attacks originating from vehicles that are a legitimate parties in the network are on the other hand considered. This could for example include a single vehicle attacking the tester, or a vehicle attacking other non-malicious cars in the same network.

An outsider is someone who is not considered a legitimate part of the network by the other entities.

5.4 Attacker capabilities

The capabilities of an adversary are to a certain extent controlled by the environment the attacker and its target operate in. The possibilities for an attacker are thus described in different sections separated according to the network configurations specified in the previous chapter.

Direct physical connection between one vehicle and an external tool
As it was assumed that direct communication over a single cable cannot be eavesdropped or affected in any way, the transmissions cannot be attacked. Since attacks originating from the external test equipment are considered to be out of the scope of this work, that only leaves attacks coming from the vehicle in this one-to-one connection. That is, the attacker legitimately connects to the tester which it then tries to attack.

Potential attacks (attack vector -> target):

- Vehicle -> Tester

Networked connection between one vehicle and an external test equipment
This scenario is similar to the previous one, but it has one major difference. The communication travels over a potentially insecure medium where an attacker may operate freely. The single attack opportunity from the previous scenario is still available, but the possibilities are thus extended with injection, deletion, manipulation, and eavesdropping of transmissions as well. An attacker can then use this vector in order to attack both a vehicle and a tester.

Potential attacks (attack vector -> target):

- Vehicle -> Tester
- Communication link -> Tester
- Communication link -> Vehicle

Networked connection between multiple vehicles and one external test tool
In this scenario, the possibility of multiple cars existing simultaneously in the system is added. The case might thus be that one of the cars is controlled by an attacker, while the others are not. The extension to the previous scenario is then logically that an attacker in control of a vehicle can attack another (potentially bouncing attacks off the tester in the process).

Potential attacks (attack vector -> target):

- Vehicle -> Tester
- Vehicle -> Vehicle
- Communication link -> Tester
- Communication link -> Vehicle

Networked connection between one vehicle and multiple test tools or test applications on a single physical tool
In this network configuration the messages are still travelling across a potentially public medium and shall thus be treated in the same way as in the previous scenarios. This setup is a slightly more advanced version of the networked 1-to-1 configuration with the added complexity of multiple testers, as opposed to the multiple vehicles of the previous scenario. As the testers are assumed to be secured in the sense that an attacker is not in control of one of them, the issues added are not as visible as in the previous sub-section. Here, an attacker in control of the vehicle might however attempt to abuse the relation between the different external test equipment.

Potential attacks (attack vector -> target):

- Vehicle -> Tester
- Communication link -> Tester
- Communication link -> Vehicle

5.5 Attacker resources

The computing capacity of the attacker is assumed to be close to unlimited. That is, the attacker is in possession of the best possible equipment available today to carry out the attacks. The attacker is neither restricted by transmission range in wireless communication systems.

Another important assumption is that the attacker is always able to have a legitimately acknowledged vehicle in network configurations with multiple cars. If a target vehicle of the attacker is in such a multi-car system, the attacker is thus also able to have a vehicle there. This vehicle of the attacker is not necessarily allowed to participate in communication with the others, but it is a legitimate party in the same network.

6 Vulnerability analysis

This chapter includes an overview of the vulnerabilities and security issues in the DoIP protocol as well as in the underlying services required by the protocol.

6.1 Security in services used by DoIP

This section contains descriptions of known and previously discovered problems with the security of the technologies utilized by the DoIP protocol. Because of the overwhelming amount of possible vulnerabilities, this walkthrough is not all-encompassing but rather aims to point to some issues that are serious enough to warrant protective measures and secured versions of the technologies.

It should also be noted that all these protocols are not used during the actual DoIP transmissions. DHCP, ARP, and NDP are for instance only used earlier, or in between, in order to setup the possibility for communication between different entities.

Security issues related to the technologies used that cannot lead to vulnerabilities propagating in a DoIP environment are not considered.

Table 5: Protocols utilized by DoIP.

Protocol
DHCP
ARP
NDP
ICMP
IP
TCP
UDP

Table 5 contains a listing of the protocols used by DoIP to provide auxiliary services. They are listed in the same order as their respective analyses are described throughout the section.

DHCP
A number of attacks can be performed against DHCP given access to the local network of the DHCP server. The authors of [49] mention a few and a couple of those are taken as examples here. Perhaps the most used attack is a starvation attack, where the resource being exhausted is the IP addresses available. In the starvation attack the malicious host spoofs messages in order to require as many of the addresses as possible. The result of this is a denial of service when legitimate hosts later are in need of addresses.

DHCP can also be attacked in order to re-route traffic. This is done by providing faulty address information. For example, the malicious host can answer to DHCP requests from legitimate users of the network and provide itself as the default gateway so that all traffic from the target is directed to the attacker which can then either keep the traffic or send it on, acting as a man-in-the-middle [50].

ARP
ARP spoofing, or ARP cache poisoning, is a type of attack where a malicious host on the local network tries to make another host accept faulty information into its ARP cache [51]. That is, the attacker might for example answer ARP requests on behalf of other nodes in a way so that translations between IP addresses and MAC addresses point to the attacker's own computer. Performing this attack on two hosts seeking to communicate with each other, an attacker might even place itself in between the parties in order to produce a man-in-the-middle attack without the affected hosts noticing. The attacker then has the possibility to shift or skew the communication in any way possible.

As with the DHCP attacks, these attacks do of course also require access to the local network of the target.

NDP
As the authors of [52] note, similar spoofing attacks as the ones using ARP can also be carried out in an IPv6 system using NDP for address resolution. Instead of sending faked ARP responses, spoofed neighbor solicitations and advertisements are sent to produce the same effect. That is, re-routing packets to an incorrect address.

As the NDP technology is also link-local, access to the local network of the target is required for the adversary to be able to perform attacks.

ICMP
The most common way of incorporating ICMP into attacks is to use it to probe, scan and fingerprint [53] a target in order to find vulnerabilities before the actual aggression. Not considering these types of issues, ICMP can also be used to create other types of more concrete problems. For example, ICMP route redirect messages can be spoofed in order to redirect traffic causing either a denial of service or easier interception of transmissions. Yet other attacks could include sending malformed ICMP packets to try to trigger implementation bugs leading to crash or freeze problems [54].

IP
Since IP is one of the most well-researched protocols, a lot of vulnerabilities have been uncovered over the years. For an in-depth look the CPNI (Centre for the Protection of National Infrastructure) sponsored paper [55], which discusses issues throughout the entire protocol, is referred to.

A goal of many attacks is to fill data fields with unexpected contents in order to trigger issues not thought of by the developers of an implementation. The point of these attacks is to trigger behavior causing crashes or other undesirable results. One example of such an attack is the so called LAND attack in which an attacker sends a packet with the target in both the source and the destination IP address fields. This is an old attack that has resurfaced in the past few years, for example causing problems for some versions of Windows Vista [56].

TCP
As for IP, there is also for TCP a CPNI sponsored report available which is very thorough about the issues of the protocol [57]. Because of the increased complexity of TCP compared to UDP, the intricacy of the potential attacks against the former might also be higher than those against the latter. With more mechanisms in place, there are also more mechanisms to be abused.

One class of attacks against TCP is called session hijacking, and as the name implies the objective of such an attack is to take over an endpoint of a TCP stream [58]. There are also simpler variants of the same theme, where an attacker tries to insert RST segments into the stream in order to force a shutdown. The latter attack results in a possible denial of service while the consequences of the former have the potential of being even more severe.

UDP
UDP suffers from the fact that it does not have any sequence numbers or a notion of session [47]. This makes it easy for an attacker to inject datagrams into ongoing dialog streams.

Another type of attack is the so called UDP bomb in which a malicious host tries to trigger an implementation bug by sending a datagram with header fields of illegal length [59].

The goal of both these attacks is to produce a denial of service for legitimate users.

6.1.1 Summary of security in services used by DoIP

The preceding text has, through its examples, proven that all technologies utilized by DoIP can be abused in one way or another. As secured versions of the protocols are not mentioned or recommended by the draft documents, the protocols utilized by DoIP can all be treated as attack vectors by a user with malicious intent. It needs to be pointed out that the protocol vulnerability descriptions are not complete in any way. Examples have merely been taken in order to show that the foundation of DoIP is not secured. As complete analyses of these TCP/IP protocols have been carried out by previous research, such assessments will not be repeated here.

The walkthrough of the protocols has also helped in identifying which types of attacks might be most common. In the examples, two main classes can be seen. The first one consists of pure denial of service attacks, while the other consists of different ways to control the flow or contents of transmissions.

6.2 Security in DoIP

This section describes security issues in the DoIP protocol itself. That is, not problems that are inherited from technologies used by the protocol but issues that stem from the specifications in the draft documents.

The analysis contains references to the different requirements (specified as [DoIP-xxx]), tables and state machines of the technical documentation in the draft standard. Even though the assessment of this section can be considered to be self-contained, it is recommended to read it together with the DoIP draft documents as this analysis would be overly verbose if each requirement was to be fully explained before its weaknesses and strengths are investigated.

As briefly mentioned in the previous paragraph, the specification of the DoIP protocol is built up around requirements, tables, and state machines that together determine the behavior of the communication. Generally, the sequence actions are taken in is stipulated in the state machines where the transitions are decided by rules specified in the requirements. These requirements are shorter textual descriptions which can be compared to the imperative terminology of RFC documents [60]. The tables then more precisely explain the data fields of the messages. As this is how the draft is constructed, the analysis is also to a large degree centered around examining specific requirements.

All messages are prepended by a special DoIP header, and an assessment of its data fields and handling is thus also included in this section. The analysis of the header is then followed by walkthroughs of the specific issues related to the different payloads a DoIP message might have.

6.2.1 DoIP header handling

This section describes issues and protective mechanisms related to the standard header of DoIP messages. It discusses topics related to the different fields and the handling of the header upon receipt of a packet.

Table 6: DoIP header fields.

Item	Starting position (byte)	Length (bytes)
Protocol version	0	1
Inverse protocol version	1	1
Payload type	2	2
Payload length	4	4

In Table 6, the DoIP header fields are shown. The **Inverse protocol version** is simply a product of a logical exclusive or operation between the **Protocol version** and the hexadecimal byte 0xFF.

Table 7: Protective measures specified for the DoIP header handling.

Reference (in DoIP draft documents)	Function	Protection against
[DoIP-031]	Ignore unwanted packets	Magnification attacks
[DoIP-039]	Ignore unwanted packets	NACK storms
[DoIP-040]	NACK policy	NACK storms
Table 14	Message discarding policy	Fingerprinting
[DoIP-042], Table 11	Handling of unexpected values	Fuzzing
[DoIP-043]	Input validation	Buffer overflow
[DoIP-044]	Input validation	Buffer overflow
[DoIP-045]	Input validation	Buffer overflow

Table 7 specifies those protective measures that are in place for the handling of the DoIP header according to the second draft document. The text that follows further down then expands upon these items and provides explanations as to what the thought behind these mechanisms might be.

Table 8: Potential security issues in the DoIP header handling.

Reference (in DoIP draft documents)	Type of weakness	Potential result
[DoIP-041]	Weak data integrity check	Unauthorized modification

Table 8 contains a listing of the different security issues found in the specification of DoIP. These are further discussed together with the control mechanisms in the text beginning with the next paragraph.

The requirement [DoIP-031] states that any packet with a multi- or broadcast IP address as its source should be ignored by the receiving node. From a security point of view this is good as it helps in protecting against certain types of magnification attacks where an attacker tries to send such a packet to a legitimate host in order to make it reply to the multi- or broadcast address [61]. The term magnification is used to describe the type of attack since the single packet sent with malicious intent produces a number of packets from the abused node.

In [DoIP-039], it is specified that DoIP entities shall ignore received **Generic DoIP header NACKs**. In other words, **NACKs** are only valid being sent from a DoIP entity to a tester, and not in the other direction. The ensuing requirement, [DoIP-040], prevents test equipment from sending **NACKs**. Together, these two statements help in protecting against so called storms. That is, a DoIP entity and test equipment sending repeated **NACKs** back and forth, each insisting on the other committing an error by sending the **NACK**. The intention of someone using this type of attack is to cause an exhaustion of the available network resources by making these transmissions consume otherwise available bandwidth.

As can be seen in Figure 7 of the second DoIP draft document, the requirements [DoIP-041] to [DoIP-045] encompass different kinds of sanity checks for the fields contained

within the header of a received DoIP message. Table 14 of the same document follows up by specifying the different actions that should be taken if the respective controls were to fail. That is, how to handle the message and connection. Having a clear definition of how to handle the closing of connections, and when to simply discard messages, helps in alleviating the possibility of diverging implementations resulting from specification ambiguities. From a security perspective, one of the advantages of this is that it reduces the potential for fingerprinting. In fingerprinting, an attacker tries to deduct which implementation, of what version and from which vendor, is in place. The advantage of knowing this can be to later on try to exploit previously found implementation-specific vulnerabilities.

Of the different sanity checking requirements, [DoIP-041] begins by defining that every DoIP entity should make sure that the value in the **Inverse protocol version** field actually is the inverse of the value found in the **Protocol version** field. According to the description found in Table 12 of the second draft document this inverse field is in place so that a DoIP entity should be able to know that it receives a correctly formatted message. In other words, it is a type of data integrity control mechanism. It is however not a very strong one as it only covers the first two bytes of the eight-byte header in the best case. Because of this weakness, the control can clearly not be seen as sufficient to ensure data integrity, even in the absence of malicious threats.

[DoIP-042] continues the header field controls by specifying how a DoIP entity should act when the **Payload type** is unknown. That is, where the type is not specified in Table 11 of the second DoIP draft document. Since these values are well defined there is no ambiguity surrounding what **Payload type**s a DoIP entity should accept either. Having this requirement and table in place is indeed good as a common type of attack against protocols is to simply send seemingly random messages with values that do not make sense in order to trigger vulnerabilities related to the handling of unexpected data [62].

The requirement [DoIP-043] specifies that every DoIP entity should check if the **Payload length** field in headers received exceeds the maximum DoIP message size supported by the receiving entity. This check enables the prevention of a type of overflow attack where an adversary would send oversized messages in order to reach memory locations otherwise inaccessible, or to simply cause some kind of malfunction [63].

Another overflow-related control is found in the requirement [DoIP-044] which defines that each DoIP entity upon reception of a transmission should perform a check against the **Payload length** field to see if acceptance of the message would cause the currently available DoIP protocol handler memory to be exceeded. This should not be confused with the issues discussed in the previous paragraph. Whereas this requirement deals with the DoIP protocol handler memory, the *total memory* available for the DoIP protocol implementation in an entity, the previous paragraph dealt with the maximum size a *single message* might have.

[DoIP-045] specifies that a DoIP entity receiving a message should control that the value in the **Payload length** field matches the expected length for the specific **Payload**

type defined in the message. In other words, the **Payload length** field value should be the same as is defined by the table describing the fields for the particular **Payload type**.

6.2.2 Vehicle announcement/identification

This section discusses topics related to the vehicle identification phase of the DoIP protocol. The phase consists of either a **Vehicle announcement** message or a **Vehicle identification request** followed by a **Vehicle identification response**.

The **Vehicle identification request** does not contain any data fields. There are however two variants to the payload that do contain data. These are **Vehicle identification request message with EID** and **Vehicle identification request message with VIN**. The only field contained within these are a six-byte EID and a 17-byte VIN respectively and these variants are thus used when a tester wants to reach an entity with a specific EID or a vehicle with a specific VIN.

Table 9: Vehicle announcement message payload / Vehicle identification response message payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
VIN	0	17	Yes
Logical address	17	2	Yes
EID	19	6	Yes
GID	25	6	Yes
Further action required	31	1	Yes
VIN/GID sync status	32	1	No

The data contained in the response, shown in Table 9, are all fields describing the DoIP entity that is either announcing its presence or responding to a previous request. The **Further action required** gives information about if there for example are DoIP entities with no initial connectivity or if a centralized security approach is used. The **VIN/GID sync status** is used by the DoIP entity to inform the tester about if all other entities in the vehicle have the correct VIN or GID configured.

Table 10: Protective measures specified for the vehicle identification phase.

Reference (in DoIP draft documents)	Function	Protection against
[DoIP-050]	Limit number of transmissions	DoS
[DoIP-051]	Limit concurrency of transmissions	DoS

In Table 10, the different protective mechanisms found for the vehicle identification phase are listed. It does not matter whether they are intended features or not, as long as they make a contribution to the overall security of the DoIP protocol.

Table 11: Potential security issues in the vehicle identification phase.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Table 18, Figure 8	Lack of authentication	Spoofing
[DoIP-125], [DoIP-011], [DoIP-135]	Specification ambiguity	Fingerprinting
Figure 23, Section 8.5.2	Specification ambiguity	Fingerprinting
Figure 23	-	DoS

Table 11 notes the different security problems found in the vehicle identification phase. The issues in this table are, together with the protective control mechanisms from the directly preceding table, further evolved upon in the following paragraphs.

An overall issue of this phase is the lack of authentication and assurance of integrity. Since there is no form of authentication specified before the vehicle identification phase, it is of course trivial for an attacker to create falsified responses with altered identification information. The adversary does not have to setup a connection either, since the identification phase is carried out over UDP. An attacker could for example respond with messages saying that the vehicle with the sought VIN resides at its own IP address. Another example of an attack would be to try to cause confusion by responding using another vehicle's IP address but changing certain parameters of the response from the correct ones. As no field is protected, every single one can be spoofed.

The requirement [DoIP-050] states that the amount of **Vehicle announcement** messages sent out should be limited. It also specifies the minimum time that should pass between each consecutive **Vehicle announcement** message. This is a nice feature from a congestion perspective or if an attacker for example tries to force re-occurring crashes. Say that the attacker uses a malformed packet to cause a restart (and thus reconfiguration of the DoIP entity's IP address). This attack is then, from a denial of service perspective, amplified by the number of times the host sends out **Vehicle announcement** messages since it is done upon each start-up. Likewise the announce interval helps keeping down the amount of simultaneous messages and it is thus harder for the attacker to exploit the mechanism to make a system help in congesting itself.

[DoIP-125] defines how **Vehicle announcement** messages should be sent. The purpose of this requirement is most likely to describe the port and address information used when sending this type of transmission, but it is unclear. The requirement mentions UDP_CONTROL which is previously undefined. Earlier in the document the notation with all capital letters and words separated by underscore characters has been reserved for port number definitions. There is however no UDP_CONTROL port defined in the UDP port number listing in Table 8 of the second DoIP draft document. The requirements [DoIP-011] and [DoIP-135] do however speak of "UDP control messages". What those actually are is something that is left undefined also in those descriptions. It might be the case that all "control packets" are sent over UDP, and that is the reason for the term. This is however unclear at best, and thus something that quite possibly could lead to different interpretations by implementers. For example, UDP_CONTROL might likely be interpreted as a port by some because of the notation used to write the term. A fact that on the other hand speaks strongly against

UDP_CONTROL being a port is that [DoIP-011] and [DoIP-135] mention UDP control, not using the caps and underscore notation, and also specify source and destination ports.

[DoIP-051] states that there should be a random delay between the reception of a **Vehicle identification request** and the sending of the corresponding response. This randomness makes it much harder for an attacker to perform a coordinated distributed denial of service attack abusing the identification mechanism. That is, an attacker simultaneously sending out **Vehicle identification request**s to multiple vehicles trying to make them help in flooding the network with their responses. This delay is however not specified in [DoIP-052] or [DoIP-053] for the identification messages where the EID or VIN is specified. Since an EID should be unique, it is not a problem for that case. Depending on the layout of the network and the number of DoIP entities in the vehicle, the case of identification with the VIN number specified might potentially be a problem, but it is unlikely that the amount of data produced by such responses would severely affect network performance. Possibly, the mechanism could be abused as a small part of a larger attack.

There is an inconsistency between Figure 23 and the description in 8.5.2, of the second DoIP draft document, when a node reports "sync status incomplete" in its **Vehicle announcement** message. In both the figure and the textual description the tester first starts a timer. When this timer runs out, the actions in the two cases do however differ. In the figure, the tester sends a **Vehicle identification request** as what seems like a limited broadcast to all DoIP entities, including those that previously reported valid VIN/GID. On the other hand, according to the textual description, the **Vehicle identification request** is only sent out to those nodes that reported VIN/GID as invalid. As explained earlier, even though this might not seem like a vulnerability that can be exploited for any specific gain for the attacker, ambiguities are generally bad as they might lead to differences in implementations of the same protocol. Fingerprinting made easier is one obvious result, but when different implementations are to interact with each other more severe consequences might be produced.

An attacker with access to the network can actually, even without spoofing another entity's address, exploit the synchronization feature. If the implementations active in the network follow the later of the two approaches mentioned in the previous paragraph, where a single node reporting "sync status incomplete" forces re-identification of all entities, a magnification attack with a severity relative to the number of nodes can be performed. During the announcement phase the attacker would simply always send out "sync status incomplete" in its **Vehicle identification responses**. This would then cause two effects. First of all, the identification of the other nodes fails and has to be performed again, thus producing a denial of service. Secondly, each such message sent out by an adversary triggers a new phase of **Vehicle identification requests** and **Vehicle identification responses** which takes up network bandwidth possibly affecting other communicating processes as well.

6.2.3 Routing activation

The routing activation phase is carried out when a tester seeks to enable routing of its messages via a DoIP gateway and on to the internal vehicular network. In this section, the issues covered, as well as those not covered, by the DoIP draft are presented.

Table 12: Routing activation request message payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address	0	2	Yes
Activation type	2	2	Yes
[Reserved for future use]	4	4	Yes
[Reserved for OEM-specific use]	8	4	No

The fields of the **Routing activation request** message are shown in Table 12. The logical address is the address of the source of the message. In other words the external test equipment sending the request. The activation type can for example be "default" or "WWH-OBD" [64].

Table 13: Routing activation response message payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address (from corresponding request message)	0	2	Yes
Logical address (of the entity sending this response)	2	2	Yes
Routing activation response code	4	1	Yes
[Reserved for future use]	5	4	Yes
[Reserved for OEM-specific use]	9	4	No

Table 13 contains the different fields of the **Routing activation response** message. The **Routing activation response code** is a hexadecimal number noting if the activation was successful, and if it failed the individual code further describes what type of failure occurred.

Table 14: Protective measures specified for the routing activation phase.

Reference (in DoIP draft documents)	Function	Protection against
[DoIP-059]	Access control	Access from unknown addresses
Figure 9, [DoIP-062], [DoIP-063]	Access control	Unauthorized access
Figure 9, [DoIP-151]	Handling of unexpected values	Fuzzing

In Table 14 is a listing of the protective measures in the routing activation related message exchanges. These individual controls are described in detail in the text following this series of tables.

Table 15: Potential security issues in the routing activation phase.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Table 21, Table 23, Figure 9	Lack of authentication	Spoofing
Table 24	Information disclosure	Attacks taking advantage of disclosed information
Figure 9, [DoIP-149], [DoIP-151]	Specification ambiguity	Fingerprinting
Figure 9, Figure 13, [DoIP-060]	Specification ambiguity	Fingerprinting

Table 15 contains security problems discovered in the routing activation phase of the DoIP protocol. These items are also individually elaborated upon in the following paragraphs.

The issue of spoofing is prevalent in this phase as well. An attacker could try to manipulate the logical addresses of messages sent over the network or simply create new transmissions containing false information. The routing activation handler in Figure 9 of the second DoIP draft document does however specify an authentication step before the registration of a certain address on a specific socket. If the authentication mechanism is in place the attacker cannot finish the routing activation successfully without having the correct information. If it is not, nothing stops the adversary from managing to create arbitrary mappings between sockets and valid logical addresses at the DoIP entity. This could for example be useful for an attacker seeking to cause a denial of service as this can be done by simply occupying all sockets available at a particular DoIP entity. If this resource exhaustion attack is carried out, the external test equipment about to perform diagnostics on the vehicle cannot even connect.

[DoIP-059] ensures that a DoIP entity does not accept routing activation messages originating from an unknown logical source address. This check by itself does not provide a very strong security, as address based security is something that can never be entirely relied upon because of the threat of spoofing. The control is however still a

hurdle for an attacker as a valid source address must be known in order to successfully activate routing.

The routing activation response code values of Table 24 in the second of the DoIP draft documents give detailed information about what went wrong if the activation procedure fails. This information can be of use to the attacker. To take a concrete example, one can look at the requirement [DoIP-059]. This attribute specifies that a DoIP entity should reply with a certain response code if the logical source address of a **Routing activation request** is unknown to the entity. By sending repeated requests, each with a different source address, an attacker can map which addresses are trusted by the particular DoIP entity as the entity produces other error codes for other types of issues. Since the activation step is defined later on in the state machine of Figure 9 in the second DoIP draft document, this probing can be carried out even with authentication in place.

An optional feature that deserves some mention is the "confirmation" described in Figure 9 and in the requirements [DoIP-062] and [DoIP-063]. The mechanism enables a DoIP entity to require confirmation from within the vehicle, for example by the driver, before registering a logical source address on a socket.

In the state machine for the routing activation handler in Figure 9 of the second DoIP draft document, there is a control to see if the **Activation type** in the message received is supported by the DoIP entity. The transitions in the state machine point to [DoIP-149]. This is however not correct as this requirement deals with the mapping between logical source addresses and TCP sockets. What should be referred to by the state machine is most likely [DoIP-151] instead. The [DoIP-151] requirement states exactly the type of check that is textually described in the state machine. Such a check is useful since an attacker cannot exploit the possibility that the protocol does not know how to deal with messages containing unknown **Activation type**s. An adversary could otherwise have tried to abuse the unspecified event to trigger unexpected behavior from the attacked implementation, in order to possibly cause a denial of service.

According to the state machine in Figure 9, of the second DoIP draft document, describing the routing activation handler, [DoIP-060] is always applicable when the socket handling requirements, according to Figure 13 of the draft, are not fulfilled. This requirement does however state that a specific negative response code should always be sent. That response code is only one out of a number of negative response codes defined in Table 24, also of the second DoIP draft document, for rejection by the socket handler. The effect of [DoIP-060] is thus that these other response codes of Table 24 are rendered obsolete if one is to follow the state machine from Figure 9. This is probably not the intention of the authors, but an implementer will surely be confused as to whether the state machine and requirements should be complied with when there is a table defining the different cases where different response codes should be sent according to what went wrong in the socket handler. The discrepancy could at the very least help an attacker in fingerprinting a particular implementation.

6.2.3.1 Socket handling

The socket handling is performed as part of the routing activation handling, but is specified in an individual section of the second DoIP draft document and is thus also separated out in this description.

Table 16: Protective measures specified for the socket handling.

Reference (in DoIP draft documents)	Function	Protection against
[DoIP-131]	Ignore unwanted messages	Unauthorized access
[DoIP-127], [DoIP-128], [DoIP-132]	Inactivity timers	Resource exhaustion
[DoIP-091], [DoIP-093], Figure 13	Access control	Resource exhaustion

In Table 16, the protective measures specified for the socket handling of the routing activation phase are noted. The items of the listing are further described in the text following the tables.

Table 17: Potential security issues in the socket handling.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Section 3.2, Section 7.2.1.1	Specification ambiguity	Fingerprinting
[DoIP-148]	Static resource allocation	Resource exhaustion

In Table 17 security issues found in the socket handling are listed. These issues are then analyzed further in the following paragraphs.

There is some confusion about the socket definition used, something which can lead to implementation-specific security issues. In section 7.2.1.1 of the second document of the DoIP draft, a socket handle is defined to be identified by the source and destination IP addresses along with ports and the transport layer protocol used for communication with the socket. The section 3.2 of the same document does however refer to RFC 147 for the socket definition. This is an older RFC that specifies sockets as unidirectional entities. The older definition is rarely used nowadays and sockets are generally bidirectional.

The requirement [DoIP-131] states that all messages not related to authentication or confirmation should be dropped before the connection is in the "Routing active" state. Not accepting possibly random messages before authentication of the sender is of course good from a security point of view as it prevents unauthorized access.

In [DoIP-127], [DoIP-128] and [DoIP-132] the behavior of two timers is specified. The first one is the **Initial inactivity timer** which is started when a socket is initialized. The second one is the **Generic inactivity timer** which is started when a **Routing activation request** is received on the socket. Together these two provide some defense against resource exhaustion attacks. By removing unresponsive associations upon timeout, the timers help in keeping resources available for new connections as the maximum amount

of sockets that can be allocated is limited. An adversary trying to make a DoIP entity unavailable by keeping all sockets occupied must thus continually perform attacks as connections are otherwise pruned.

According to the requirements [DoIP-091] and [DoIP-093] along with Figure 13 from the second draft document a single logical source address cannot occupy multiple TCP sockets at a single DoIP entity. These mechanisms prevent an attacker in possession of only a single legitimate address from performing resource exhaustion attacks using this attack vector. That is, the attacker trying to tie up multiple TCP sockets.

[DoIP-148] specifies that a DoIP entity should not accept more connections than what has been statically defined. That is, the number of available sockets is not dynamic in the sense that the DoIP entity could increase it whenever needed. This way of handling sockets is good if resources are limited and each extra socket allocation takes up a significant amount of the total memory and computing capacity available. On the other hand, it also makes it easier for an attacker to produce a denial of service by, in ways previously described, occupying multiple sockets thus not allowing connections from not yet connected testers.

6.2.4 Diagnostic communication

The analysis of the diagnostic communication has been divided into sub-sections according to the different types of messages as every message type is not necessarily part of every single diagnostic session.

6.2.4.1 Diagnostic message

The **Diagnostic** messages constitute the main part of the diagnostic communication. The messages are used as containers for the more specific diagnostic functions defined by higher layer protocols.

Table 18: Diagnostic message payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address (of the sender of this message)	0	2	Yes
Logical address (of the final destination of this message)	2	2	Yes
Diagnostic data	4	Maximum message size - 4	Yes

Table 18 shows the three fields of the **Diagnostic** message payload. The diagnostic data contained within the message is a request or response as specified by some higher layer protocol such as ISO 14229-1 [36].

Table 19: Diagnostic message positive acknowledgment payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address (of the receiver of corresponding **Diagnostic** message)	0	2	Yes
Logical address (of the sender of corresponding **Diagnostic** message)	2	2	Yes
ACK code (type of acknowledgment)	4	1	Yes
Copy of message being acknowledged	5	≤ (Maximum message size - 5)	No

In Table 19, the fields of the **Diagnostic message positive acknowledgment** are shown. A copy of the previous **Diagnostic** message sent can be included in this message for troubleshooting reasons.

Table 20: Diagnostic message negative acknowledgment payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address (of the receiver of corresponding **Diagnostic** message)	0	2	Yes
Logical address (of the sender of corresponding **Diagnostic** message)	2	2	Yes
NACK code (type of negative acknowledgment)	4	1	Yes
Copy of message being negatively acknowledged	5	≤ (Maximum message size - 5)	No

The **Diagnostic message negative acknowledgment**, shown in Table 20, follows the same structure as the **Diagnostic message positive acknowledgment** with the only difference being that the ACK code is substituted for a NACK code.

Table 21: Protective measures specified for the Diagnostic message exchange.

Reference (in DoIP draft documents)	Function	Protection against
[DoIP-072], [DoIP-074], Figure 10	Input validation, Message discarding policy	Buffer overflow
[DoIP-073], [DoIP-074], Figure 10	Input validation, Message discarding policy	Buffer overflow

In Table 21, protective mechanisms related to the **Diagnostic** message exchange are listed. The text underneath this series of tables gives further descriptions of each item.

Table 22: Potential security issues in the Diagnostic message exchange.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Figure 10	Lack of authentication	Unauthorized access
[DoIP-071]	Information disclosure	Attacks taking advantage of disclosed information
[DoIP-107], Table 30	Specification ambiguity	Fingerprinting

Table 22 lists the security problems discovered in the **Diagnostic** message exchange part of the diagnostic communication phase. These issues are further elaborated on in the following paragraphs.

There is no mention of any reliable authentication in the diagnostic message handler in Figure 10 of the second DoIP draft document. A control to see if the logical source address in the message is registered on the TCP socket the transmission was received is specified in [DoIP-070]. An attacker could however spoof this information, thus managing to get a potentially dangerous payload routed on to the internal network as no robust access control is enforced.

[DoIP-071] specifies that DoIP entities should reply with a negative response code when the logical target address (the address of the final destination node, located internally behind the DoIP gateway) of the message received is unknown. This information could be used to map the network behind the DoIP gateway. An attacker can send multiple messages with different final destinations in order to see if there are nodes with any of those addresses located behind the DoIP gateway the attacker is communicating with.

Another requirement related to forwarding is [DoIP-072] which prevents the DoIP gateway from routing transmissions on to internally located non-DoIP networks when the message size exceeds the maximum allowed on the network. The requirement helps in protecting against attacks with the goal of causing faulty behavior through overflows of message buffers.

[DoIP-073] protects against target buffer overflows within a DoIP entity. The requirement realizes this defense by stating that a negative response shall be sent when a **Diagnostic** message is too large to be copied into the destination buffer. That is, an attacker cannot send oversized diagnostic messages in order to cause malfunctions or otherwise unexpected behavior. While the requirement does not explicitly state how to handle the problematic message, the discard operation is instead noted in the requirement [DoIP-074] and the diagnostic message handler of Figure 10 in the second draft document of DoIP.

The requirement [DoIP-103] specifies a check to see if the target of a message is *currently* reachable. It also specifies sending a response to the sender of the message informing about the scenario. This information is clearly useful to an adversary who for example wants to spoof communication to make it seemingly originate from another entity. The attacker is helped in doing this as the messages defined in [DoIP-103] can be used to find out if a targeted node is currently responding to communication, and thus interfering with the attack.

A discrepancy between tables and requirements can be found when looking at [DoIP-107]. This requirement specifies that the same negative response code should be sent for the errors "unknown target network" and "transport protocol error", while Table 30 of the second DoIP draft document on the other hand specifies two different response codes for the conditions. This ambiguity is thus yet another example of an issue which can help an attacker in distinguishing between different implementations of the protocol, as the unclear specification might lead to divergent solutions.

6.2.4.2 Alive check request and alive check response

The alive check messages are used by DoIP entities on the vehicle side to control if an external test equipment is still active in a diagnostic session.

The **Alive check request** consists of an empty payload without any data fields.

Table 23: Alive check response payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Logical address (of the tester currently active on the socket on which the **Alive check request** was received)	0	2	Yes

The **Alive check response** payload, shown in Table 23, only contains the logical address of the tester responding to the **Alive check request**.

Table 24: Protective measures specified for the alive check messages.

Reference (in DoIP draft documents)	Function	Protection against

Table 24 presents the control mechanisms used to enforce security when transmitting alive check messages. Since no definitions in the draft documents were both security related and alive check specific, this table is empty.

Table 25: Potential security issues in the alive check messages.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Figure 14, Figure 15	-	DoS

In Table 25 is a listing of the security related problems of the alive check message exchange. As in previous sections, the entries in the table are further described in the following text.

An unsophisticated denial of service attack an adversary could carry out in the alive check message exchange would be to jam either the request or the response message. This type of message deletion results in the alive check failing and will, in accordance with Figure 14 and 15 of the second draft document, lead to the corresponding TCP socket being closed. As either the request or response is interrupted, no response will be received by the DoIP entity as it was either deleted or never sent.

Another possible abuse of the alive check message exchange would be for an adversary to send spoofed **Alive check response**s in place of the test equipment connected to a DoIP entity, even after the tester has ceased performing its diagnostics, in order to keep the TCP socket occupied. This type of resource exhaustion attack can be problematic as the number of TCP sockets does not scale dynamically but is instead fixed to a pre-defined amount and new connections to the DoIP entity could thus be prevented by the attacker.

[DoIP-134] states that alive check messages shall only be sent on connections that are in a registered state. That is, sockets where a routing activation has been performed successfully. The DoIP documents do however not state what should be done if this is not the case. That is, someone sends an alive check message on a socket that is not yet in a registered state. Since that scenario is not specified in the draft, implementations may lack the ability to handle such messages before routing activation.

6.2.5.3 Diagnostic power mode information request and response

The diagnostic power mode messages are used by the external test equipment to check the current power mode of a DoIP entity to ensure that diagnostics can be performed in a reliable manner.

The **Diagnostic power mode information request** consists of an empty payload without any data fields.

Table 26: Diagnostic power mode information response payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Diagnostic power mode	0	1	Yes

Table 26 shows the fields of the **Diagnostic power mode information response** payload. As can be seen in the table, the payload contains only one field. This field describes whether the responding vehicle is in the diagnostic power mode or not.

Table 27: Protective measures specified for the diagnostic power mode messages.

Reference (in DoIP draft documents)	Function	Protection against

Table 27 notes the protective controls defined specifically for the diagnostic power mode message exchange. Since none were found, the table is left empty.

Table 28: Potential security issues in the diagnostic power mode messages.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Table 34	Lack of data integrity mechanism	Modification

In Table 28, the security issues of the diagnostic power mode message exchange are listed. The single problem found is elaborated upon in the following paragraph.

The issue of spoofed responses containing false information is prevalent here as well. For example, an adversary could intercept a response and modify the diagnostic power mode information from "Ready" to "Not ready" and vice versa. This would possibly cause an interruption of the service, since the tester might not continue if the DoIP entity is reported as "Not ready".

6.2.4.4 DoIP entity status information request and response

The DoIP entity status information messages are used by external test equipment to fetch information relevant for performing diagnostics on a DoIP entity.

The payload of the **DoIP entity status information request** does not contain any data fields.

Table 29: DoIP entity status information response message payload.

Item	Starting position (byte)	Length (bytes)	Mandatory support
Type of node (of the responding node)	0	1	Yes
Maximum number of concurrently open TCP sockets allowed for DoIP communication (at the responding node)	1	1	Yes
Number of currently open TCP sockets for DoIP communication (at the responding node)	2	1	Yes
Maximum processable size of logical requests (at the responding node)	3	4	No

Table 29 shows the different fields of the **DoIP entity status information response**.

Table 30: Protective measures specified for the DoIP entity status information messages.

Reference (in DoIP draft documents)	Function	Protection against

Table 30 is empty as no protective mechanisms specific to the DoIP entity status information message exchange were found in the DoIP draft documentation.

Table 31: Potential security issues in the DoIP entity status information messages.

Reference (in DoIP draft documents)	Type of weakness	Potential result
Table 36	Lack of data integrity mechanism	Modification

The only issue in Table 31 might not seem like much, but it should be considered that the problem applies to all the fields of the **DoIP entity status information response** message. The spoofing issues will be further explained in the ensuing paragraphs.

As with the other diagnostic message types, most issues here are a consequence of the lack of authentication and integrity checking mechanisms. The absence of such controls makes spoofing trivial. As noted, every field in the **DoIP entity status information**

response message can be modified to cause some kind of disruption under the right circumstances.

To begin with the attacker could intercept the **DoIP entity status information response** message and alter the field describing the type of node the responding DoIP entity is, before sending the message on. By doing this, a tester can be made to believe that a DoIP gateway is a DoIP node or vice versa thus impairing the possibility of correctly performed diagnostics.

An attacker could also experiment with fields noting how many sockets are currently open as well as the maximum number of sockets that can be opened. The manipulation could for example include making the tester think there are available slots when there are not, which would result in the DoIP entity receiving unexpected messages from the tester. The attacker could also alter the numbers the other way around by setting the same number in both fields, even when there are sockets available, thus denying service. A final example could be for the attacker to change them into nonsensical numbers by letting the currently open number of sockets be larger than the maximum number of sockets. The last example could cause crash problems if the external test equipment's DoIP implementation does a check which consists of a simple subtraction in an unsafe programming language.

An adversary could intercept and alter the field containing the maximum processable size logical requests are allowed to have. Changing it to a higher value could have the effect that the tester upon receipt proceeds by sending messages that are of a larger size than the DoIP entity could handle. If the specific implementation in place is not constructed to handle the event where oversized messages are unexpectedly received, the result could be an overflow of the destination buffer.

6.2.5 Uncategorized issues

This section takes up some issues that did not fit into any of the other categories while at the same time not meriting sections of their own. At the same time, they are actual issues from the standard and are as such included in the analysis.

In section 7.5, "Communication environments and recommended timings", of the second DoIP draft document the authors clearly state that no specific timing parameters will be part of the standard. This is however contradicted by the fact that there is an entire section dedicated to timing parameters in 7.3. This type of inconsistency might lead to fingerprinting or interoperability issues as previously explained.

Throughout the draft there are inconsistencies in the use of variable parameters versus defined values. Most of the time a parameter name is given during descriptions rather than a defined value of the variable. However, on page 59 the value "three times" is written out for the number of **Vehicle announcement** messages a DoIP entity should send out after having had a proper IP address configured. In other sections of the draft, for example on page 28, the variable "A_DoIP_Announce_Num" is instead simply referred to.

6.2.6 Feedback on security requirements

This section reconnects to the security requirements stated in section 4.10 and discusses how well the DoIP protocol manages to uphold these.

Table 32: Security requirements and their statuses.

Attribute	Status
Data origin authenticity	Broken
Integrity	Broken
Controlled access (authorization)	Broken
Freshness	Broken
Non-repudiation	Broken
Privacy/anonymity	Broken
Confidentiality	Broken
Availability	Broken

Table 32 paints a grim picture of the security in the protocol. It should however be noted that these attributes are either seen as fully broken or not broken in any way. All attributes having their statuses set as "broken" does in other words not necessarily mean that there are no mechanisms at all in the protocol to enforce the different requirements. It simply means that no attribute is fulfilled in its entirety. It should also be pointed out that these statuses relate to the most general case of DoIP usage. That is, in a networked environment. To reconnect to the previously specified network configurations, this section discusses the statuses of the security attributes in all configurations except the 1-to-1 direct cabled scenario which has already been assumed to be secure on account of its link properties. As described in the attacker model created in the threat analysis of chapter four, it is also assumed that an adversary can always eavesdrop on and affect the workshop network either through abuse of wireless links or the Internet. A further elaboration of each requirement follows in the ensuing paragraphs. For examples of what results these breaches can produce in a real-world environment, the specification of the security requirements in section 4.10 is referred to.

The **data origin authenticity** requirement is not completely fulfilled. Some controls do exist to check the logical source addresses of messages to ensure that transmissions come from trusted parties. As there is no mechanism in place to prevent such information from being spoofed, the security attribute is however not enforced.

The only **integrity** mechanism in place is a weak control to discover corruption of the first bytes of the DoIP header. There are however no checksums or similar in place to fully prevent message corruption or alteration, and **integrity** is thus not guaranteed.

As previously noted, the only form of access control in place is the mechanism used to check the logical source addresses of received messages. The address is then matched towards some list, noting trusted hosts, kept by the DoIP entity. As **integrity** and **data origin authenticity** are not enforced, neither can **controlled access** be as it relies on identity information not being possible to spoof.

There are no features such as sequence numbering, nonces or timestamps in DoIP. None of the services utilized by the protocol can be said to provide reliable and non-spoofable replay attack defenses either. **Freshness** is thus not ensured by the protocol.

As **data origin authenticity** is not guaranteed, neither can **non-repudiation** be since it relies on being able to verify the source of a message.

There is no form of encryption in place for any part of the transmissions made by the protocol. As messages are sent in cleartext over a medium that can be eavesdropped, **privacy** is not ensured.

As mentioned in the previous paragraph, there is no encryption mechanism specified by the protocol. That being the case, **confidentiality** can clearly not be guaranteed which is proven using the same reasoning as for **privacy**.

Throughout the analysis different kinds of denial of service attacks have been taken as examples. As these are possible to perform, **availability** is not enforced. It should however be noted that it is very seldom the case that full **availability** can be guaranteed under all possible circumstances.

6.2.7 Summary of security in the DoIP protocol

In order to get a better overview of the types of vulnerabilities that exist in the DoIP protocol, the potential issues found were categorized into different classes. A classification procedure was applied to the protective mechanisms discovered as well. The mappings between issues, mechanisms and classes are shown in tables throughout the analysis section.

The most common vulnerabilities found included lack of authentication and data integrity assurance mechanisms. Other issues seen throughout the draft stem from ambiguities in the specification and missing sanity checks for validation of data in received messages.

Authentication mechanisms being left out for everything except routing activation means that spoofing is generally trivial. The result of this is that message fields containing some kind of identification information could be altered by an attacker wanting to pretend to be another node in the system. The information modified includes logical addresses, VIN numbers and EID or GID values as none of them are protected. It should also be noted that the access control in place for the routing activation is only specified for the actual activation procedure. Once a socket is activated no further robust authentication is made to control the origin of messages received on the socket.

Specification ambiguities include problems that are not necessarily technical problems with the draft but rather issues with the way the specification has been written. The category includes anything from vague or unclear parts to pure contradictions between different sections of the documentation. Issues such as these might lead to implementations not following the ideas behind the less well defined parts. These problems are also likely to increase the likelihood of an adversary being able to fingerprint the entity being targeted in order to later on be able to apply attacks abusing

implementation-specific weaknesses. The potential of successful fingerprinting is also greatly increased by the possible presence of OEM-specific fields.

The resource exhaustion category includes issues where an adversary tries to occupy some resource in order to make it unavailable for legitimate users. The resource might for example be storage or computing capacity at a target entity or it might be the bandwidth of a specific network.

State machines describing how received messages should be handled are included for all processing to be done by DoIP entities on the vehicle side. For the external test equipment however, no sequenced handling is defined. Parts of the DoIP header handling specified might be valid for the tester, but since the state machine in Figure 7 among other things includes the sending of NACKs, which test equipment is not allowed to carry out according to [DoIP-040], it should not be applied to external test equipment.

Finally, it should be noted that all messages as specified by the draft are sent in cleartext. That is, no encryption of any part of the transmissions is provided by the protocol.

7 Discussion

DoIP, without extra control mechanisms, is not secure to use in an arbitrary environment. That is the most general conclusion that can be drawn from the results of this work. If DoIP is to be used over public media, such as either over the Internet or over wireless links, protective measures need to be applied in order to fulfill the requirements stated in this report and thus guarantee the correct operation of safety-critical systems. As seen in the summary of the security analysis of DoIP from the previous chapter, there exists breaches of all requirements stated as the messages are generally not authenticated or encrypted in any way. These two types of features provide the basis for upholding almost all requirements except the availability related ones, and as was also shown in the previous section, there exists numerous possible ways of performing denial of service attacks in DoIP.

The stream of thought behind the security questions surrounding the protocol is hard to pin down. Had all kinds of protective measures been completely left out, the natural conclusion would have been that the authors did not see it as a mission of the protocol to be secure. As it is now, there are some mechanisms described in the draft documents. These are however not nearly enough to provide adequate security for a system where incorrect operation can lead to the endangerment of human life.

A possibility is that DoIP has been constructed to include functions that offer full security only in certain operating environments, for example when using a direct cabled connection. In such a scenario the authentication mechanism in the routing activation might be enough as the connection is broken and the socket has to be registered and authenticated all over if a cable is pulled out. This would then explain why the authentication and confirmation operations are only present for routing activation and not for the other phases. If this is the case that security is only provided under specific conditions, it does however need to be clearly stated in the final standard to avoid confusion.

Not counting the authentication and confirmation features of the routing activation, there are a couple of other protective mechanisms mentioned in the protocol specification. Most of these are however small features and not thorough enough to live up to the security requirements stated by this project. The question is then, whether these mechanisms were designed with security in mind or if those properties are simply products of mechanisms designed to withstand for example common errors arising from faults produced without malicious intent.

One issue that manifests itself throughout the DoIP specification is the lack of well-defined behavior for the tester side of the communication. The handling of messages by the external test equipment mostly seems to be up to the implementers of the protocol. If this lack of definition is intended or not is unclear. The consequences are most likely not as severe as missing details for the vehicle side, but adversaries could possibly exploit such weaknesses in order to "bounce" attacks off the tester in order to cause problems for a targeted vehicle.

A problem when working with security in a diagnostic protocol in particular, and troubleshooting protocols in general, is balancing how much information to communicate. In security, it is desirable to disclose as little information as possible, while it in diagnostics is the other way around. In order to troubleshoot correctly it is useful to retrieve as much information as possible about the erroneous system. In other words, higher security in a way means less useful diagnostic information.

A more general problem that all communicating systems have to face is denial of service attacks. Full protection against this class of attacks is almost impossible. How does one guard the air from malicious jamming signals when transmissions travel over a wireless medium for example? Having communication that can potentially be interrupted at any time poses questions about how to perform vehicular diagnostics over unsafe mediums. Is it defensible to allow diagnostics of serious issues to be performed on a vehicle in motion? It is not within the scope of this work to answer such questions, but they will arise upon future large-scale implementations of protocols such as DoIP. That being said, there are ways of mitigating attacks that attempt to exhaust different kinds of resources. In the DoIP draft, there are efforts such as activity timers and alive checks defined to reduce the possibility of legitimate entities being denied service. If these mechanisms are in place to increase the security of the protocol or if they are in place simply to mitigate the effects of nodes crashing or somehow otherwise failing to communicate properly in the middle of sessions is unclear.

While this project only considers design flaws, configuration and implementation-specific issues might be more prevalent. The type of assessment described in this report is still useful as it can provide the base needed to produce guidelines for future implementations. That is, guidelines specifying how the implementer should act in the presence of ambiguous specification details, and guidelines that note where security external to the protocol needs to be emphasized. Such policies could then help in reducing the number of possible problems resulting from the implementation. For them to help in mitigating problems such as fingerprinting they do however also need to be widely accepted and not internal to a company. This being said, an implementation can never be assumed to be secure simply because it is based on a secure protocol, and analyses of deployed systems are vital to ensure security. It also deserves to be mentioned that even if external security measures such as SSL/TLS or IPsec were in place to guard the communication, there are problems that these do not protect against. One example is the buffer overflow type of attack. To be secure against such issues robust input validation needs to be in place at the endpoints. Since DoIP specifies requirements on all layers beneath itself, inclusion of technologies such as SSL/TLS or IPsec might also require changes to be made to the DoIP specification in order for secured DoIP implementations to be compliant. With the issues described in this report in mind, it is however clear that the communication cannot be unguarded.

Finally, it should be noted that the current DoIP specification is after all a draft and not a finished standard. The version of DoIP that was reviewed in this project is still in the enquiry stage and a lot of things could change before it is approved as an ISO standard [65]. The critique presented in this report should thus not be assumed to be applicable to any other version than the one examined.

8 Future work

To begin with, ISO 13400 is still only a series of draft documents. As it is not yet a finished standard work still remains to be done. A first step could be to describe more well-defined message handling for the tester side of the communication. Other areas that need to be emphasized include clearing out the ambiguities as well as clarifying what security the future standard will provide. After such a statement has been made, actors in the automotive industry implementing the protocol need to consider whether or not external security measures should be added. Most likely, this is the case as there does not seem to be any efforts in the way of describing standardized authentication or access control in the ISO 13400 documents.

In providing external security, there are two main parts that need to be considered: securing the communication and securing the environment by protecting the endpoints from attacks using other vectors than DoIP and related protocols. In securing the communication, different tunneling technologies should be considered in order to fulfill the requirements stated in this report. Apart from the purely technical aspects of securing the communication between two entities, designers of the system must also consider how the identity management is handled and how the system would interact with vehicles of other brands. If there is no infrastructure in place to provide a way of verifying identities, authentication cannot be provided.

As mentioned, securing the environment is mainly realized by securing the endpoints of the communication. A lot of previous research has already been carried out on this matter, in a more general sense as well as focused on the automotive sphere. In other words, there is no need to develop new concepts. Existing ones do however need to be adapted to the demands of a DoIP system. As is elaborated upon later in this section, there is also the matter of computing and storage capacity restrictions of less powerful devices to take into account.

Research also needs to go into how the implementation of IP-based systems, such as DoIP, in the automotive sphere will affect the traditional vehicular networks from a timing perspective. IP is known to be an unreliable technology in the sense that timing guarantees cannot be provided. The internal networks of a vehicle are however sensitive to timing discrepancies as they rely on messages to arrive on time to ensure the correct operation of safety-critical systems.

Vehicular networks traditionally consist of ECUs with computing and storage capacities that are notably less powerful compared to modern PCs. Some research [66, 67] has already gone into how well less powerful computers such as embedded devices cope with modern security algorithms. Further experimentation does however need to be carried out on a larger scale considering the amount of vehicles that might communicate over vehicle-to-vehicle and vehicle-to-infrastructure systems in the future.

9 Conclusion

The purpose of the work described in this report is to be able to conclude whether or not DoIP is a protocol that fulfills the security properties required in order to ensure the correct operation of safety-critical automotive systems, even in an arbitrarily networked environment. As has been shown throughout the analysis described in this report, DoIP is in its current (as of the voting period 2010-09-13 – 2011-02-13) state clearly not a secure protocol and external security measures are required in order to avoid harm to human beings, vehicles, or property when a car is attacked by a reasonably skilled adversary. It should also be pointed out that it is not enough to simply protect the communication using solutions such as SSL/TLS or IPsec, since issues related to input validation and resource allocation must always be considered as well.

If DoIP is to be used in a safety-critical environment, two paths are possible. The first one is to redesign the protocol to include security measures to ensure all security requirements are fulfilled. The other, more adaptable and easier to implement, one consists of the following steps:

- Protect the communication using technologies such as SSL/TLS, IPsec or similar. The inclusion of this type of protection might however require changes to the DoIP specification.
- Create a specification describing the validation of all input derived from data included in received transmissions.
- Clearly specify security requirements and enforce control mechanisms on the endpoints of the DoIP communication in order to protect them from being compromised.

References

[1] T. V. Ramadasu. Trends in Automotive Remote Diagnosis. In *International Mobility Engineering Congress & Exposition 2005~SAE India Technology for Emerging Markets*, 23-25 October 2005.

[2] S. You, M. Krage and L. Jalics. Overview of Remote Diagnosis and Maintenance for Automotive Systems. In *2005 SAE World Congress*, 11-14 April 2005.

[3] Vector Informatik. Vehicle Diagnostics: The whole story. http://www.vector.com/portal/medien/cmc/press/PDG/Diagnostics_Congress_ElektronikAutomotive_200703_PressArticle_EN.pdf, 2007. Visited January 2011.

[4] ISO/DIS 13400-1:2010-09-13: Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 1: General information and use case definition.

[5] ISO/DIS 13400-2:2010-09-13: Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 2: Network and transport layer requirements and services.

[6] ISO/DIS 13400-3:2010-09-13: Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 3: IEEE802.3 based wired vehicle interface.

[7] S. Savage. Experimental Security Analysis of a Modern Automobile. In *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 447-462, 16-19 May 2010.

[8] D. K. Nilsson, U. Larson, and E. Jonsson. Creating a Secure Infrastructure for Wireless Diagnostics and Software Updates in Vehicles. In *Proceedings of the 27th international conference on Computer Safety, Reliability, and Security - SAFECOMP '08*, pp. 207-220, 22-25 September 2008.

[9] D. K. Nilsson, U. Larson and E. Jonsson. Low-cost key management for hierarchical wireless vehicle networks. In *Intelligent Vehicles Symposium*, pp. 476-481, 4-6 June 2008.

[10] D. K. Nilsson and U. Larson. Secure Firmware Updates over the Air in Intelligent Vehicles. In *International Conference on Communications 2008 - ICC '08*, pp. 380-384, 19-23 May 2008.

[11] D. K. Nilsson, L. Sun and T. Nakajima. A Framework for Self-Verification of Firmware Updates over the Air in Vehicle ECUs. In *GLOBECOM Workshops 2008*, pp.1-5, 30 November - 4 December 2008.

[12] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks - SASN '05*, pp. 11-21, 2005.

[13] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya et al. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine, IEEE*, vol. 46, no. 11, pp. 100-109, November 2008.

[14] V. Verendel, D. K. Nilsson, U. Larson and E. Jonsson. An Approach to using Honeypots in In-Vehicle Networks. In *2008 IEEE 68th Vehicular Technology Conference - VTC2008-Fall*, pp.1-5, 21-24 September 2008.

[15] T. Hoppe, S. Kiltz and J. Dittmann. Adaptive Dynamic Reaction to Automotive IT Security Incidents using Multimedia Car Environment. In *The 4th International Symposium on Information Assurance and Security - IAS '08*, pp. 295-298, 8-10 September 2008.

[16] D. K. Nilsson, U. Larson, F. Picasso, and E. Jonsson. A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay. In *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems - CISIS '08*, pp. 84-91, 23-24 October 2008.

[17] D. K. Nilsson and U. Larson. Simulated Attacks on CAN Buses: Vehicle virus. In *Proceedings of the Fifth IASTED Asian Conference on Communication Systems and Networks - ASIACSN '08*, pp. 66-72, 2-4 April 2008.

[18] T. Hoppe, S. Kiltz, A. Lang, and J. Dittmann. Exemplary Automotive Attack Scenarios: Trojan horses for Electronic Throttle Control System (ETC) and replay attacks on the power window system. In *Proceedings of the 23rd VDI/VW Gemeinschaftstagung Automotive Security*, pp. 165-183, 27-28 November 2007.

[19] M. Müter and F. C. Freiling. Model-Based Security Evaluation of Vehicular Networking Architectures. In *2010 Ninth International Conference on Networks*, pp. 185-193, 11-16 April 2010.

[20] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model Version 3.1, 2009.

[21] C. J. Alberts and Audrey J. Dorofee. OCTAVE Method Implementation Guide Version 2.0, 2001.

[22] D.S. Herrmann. *Using the Common Criteria for IT Security Evaluation*, Auerbach, 2003.

[23] R. Melton, T. Fletcher and M. Earley. System Protection Profile--Industrial Control Systems Version 1.0, NIST, 2004.

[24] R. A. Caralli, James F. Stevens, Lisa R. Young and William R. Wilson. The OCTAVE Allegro Guidebook, v1.0, 2007.

[25] G. Stoneburner, A. Goguen and A. Feringa. Risk Management Guide for Information Technology Systems (NIST SP 800-30), NIST, 2002.

[26] P. Mell, K. Scarfone and S. Romanosky. CVSS: A Complete Guide to the Common Vulnerability Scoring System Version 2.0, FIRST: Forum of Incident Response and Security Teams, 2007.

[27] P. Herzog. OSSTMM 3 – The Open Source Security Testing Methodology Manual, 2010.

[28] M. Prandini and M. Ramilli. Towards a practical and effective security testing methodology. In *2010 IEEE Symposium on Computers and Communications - ISCC '10*, pp. 320-325, 22-25 June 2010.

[29] ISO/IEC 27005:2008. Information technology -- Security techniques -- Information security risk management.

[30] The ISO 27000 Directory. ISO 27000 – ISO 27001 and ISO 27002 Standards. http://www.27000.org/, 2009. Visited January 2011.

[31] A. Syalim, Y. Hori and K. Sakurai. Comparison of Risk Analysis Methods: Mehari, Magerit, NIST800-30 and Microsoft's Security Management Guide. In *International Conference on Availability, Reliability and Security - ARES '09*, pp. 726-731, 16-19 March 2009.

[32] R. R. Brooks, S. Sander, J. Deng and J. Taiber. Automobile security concerns. *Vehicular Technology Magazine, IEEE*, vol. 4, no. 2, pp. 52-64, June 2009.

[33] J. D. Howard and T. A. Longstaff. A Common Language for Computer Security Incidents (SAND98-8667). *www.cert.org/research/taxonomy_988667.pdf*, 1998.

[34] L. R. Halme and K. R. Bauer. AINT Misbehaving: A taxonomy of anti-intrusion techniques. In *Proceedings of the 18th National Information Systems Security Conference*, pp. 163-172, 10-13 October 1995.

[35] D. K. Nilsson. How to Secure the Connected Car, Chalmers University of Technology, 2009.

[36] ISO 14229-1:2006. Road vehicles -- Unified diagnostic services (UDS) -- Part 1: Specification and requirements.

[37] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle and B. Weyl. Security requirements for automotive on-board networks. In *2009 9th International Conference on Intelligent Transport Systems Telecommunications - ITST '09*, pp. 641-646, 20-22 October 2009.

[38] R. Housley and W. Arbaugh. Security problems in 802.11-based networks. *Communications of the ACM - Wireless networking security, ACM*, vol. 46, no. 5, pp. 31-34, May 2003.

[39] C. P. Pfleeger and S. L. Pfleeger. Security in Computing, Prentice Hall, 2003.

[40] R. Flickenger. Antenna on the Cheap (er, Chip). http://www.oreillynet.com/cs/weblog/view/wlg/448, 2001. Visited February 2011.

[41] G. Lehembre. Wi-Fi security – WEP, WPA and WPA2. http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf, 2005. Visited February 2011.

[42] A. Bittau, M. Handley, and J. Lackey. The final nail in WEP's coffin. In *2006 IEEE Symposium on Security and Privacy*, pp. 386-400, 21-24 May 2006.

[43] M. Raggo. Top 5 myths about wireless protection. *(in)secure magazine*, vol. 22, pp. 55-58, September 2009.

[44] S. Deering and R. Hinden. RFC-2460: Internet Protocol, Version 6 (IPv6) Specification, *Request For Comments*, December 1998.

[45] J. Postel. RFC-791: Internet Protocol, *Request For Comments*, September 1981.

[46] J. Postel. RFC-793: Transmission Control Protocol. *Request For Comments*, September 1981.

[47] J. Postel. RFC-768: User Datagram Protocol. *Request For Comments*, August 1980.

[48] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet and G. Pedroza. Security requirements for automotive on-board networks based on dark-side scenarios, EVITA Deliverable D2.3, December 2009.

[49] H. Altunbasak, S. Krasser, H. Owen, J. Sokol and J. Grimminger. Addressing the weak link between layer 2 and layer 3 in the Internet architecture. In *29th Annual IEEE International Conference on Local Computer Networks*, pp. 417-418, 16-18 November 2004.

[50] OWASP. Man-in-the-middle attack. https://www.owasp.org/index.php/Man-in-the-middle_attack, 2009. Visited April 2011.

[51] C. L. Abad and R. I. Bonilla. An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks. *27th International Conference on Distributed Computing Systems Workshops*, 22-29 June 2007.

[52] J. Arkko, T. Aura, J. Kempf, V.-M. Mäntylä, P. Nikander and M. Roe. Securing IPv6 neighbor and router discovery. In *Proceedings of the 1st ACM workshop on Wireless security - WiSE '02*, pp. 77-86, 28 September 2002.

[53] J. M. Allen. OS and Application Fingerprinting Techniques. SANS Institute

InfoSec Reading Room, 2007.

[54] SANS Institute. ICMP Attacks Illustrated. http://www.sans.org/reading_room/whitepapers/threats/icmp-attacks-illustrated_477, 2001. Visited March 2011.

[55] F. Gont. Security Assessment of the Internet Protocol. http://www.bsdcan.org/2009/schedule/attachments/73_InternetProtocol.pdf, 2008. Visited March 2011.

[56] T. Newsham and J. Hoagland. Windows Vista Network Attack Surface Analysis: A Broad Overview. http://www.symantec.com/avcenter/reference/ATR-VistaAttackSurface.pdf, 2006. Visited May 2011.

[57] F. Gont. Security Assessment of the Transmission Control Protocol (TCP). http://www.cpni.gov.uk/Docs/tn-03-09-security-assessment-TCP.pdf, 2009. Visited March 2011.

[58] B. Harris and R. Hunt. TCP/IP security threats and attack methods. *Computer Communications, Elsevier*, vol. 22, no. 10, pp. 885-897, June 1999.

[59] G. De Laet and G. Schauwers. Network Security Fundamentals, Cisco Press, 2004.

[60] S. Bradner. RFC-2119: Key words for use in RFCs to Indicate Requirement Levels, *Request For Comments*, March 1997.

[61] K. Wooding. Magnification Attacks: Smurf, Fraggle, and Others. http://pintday.org/whitepapers/dos-smurf.shtml, 1998. Visited March 2011.

[62] P. Oehlert. Violating assumptions with fuzzing. *Security & Privacy, IEEE*, vol. 3, no. 2, pp. 58-62, March - April 2005.

[63] C. Cowan, P. Wagle, C. Pu, S. Beattie and J. Walpole. Buffer overflows: attacks and defenses for the vulnerability of the decade. In *Foundations of Intrusion Tolerant Systems - OASIS '03*, pp. 227-237, December 2003.

[64] ISO/PAS 27145-1:2006. Road vehicles -- Implementation of WWH-OBD communication requirements -- Part 1: General information and use case definition.

[65] ISO. International Harmonized Stage Codes. http://www.iso.org/iso/stage_codes.pdf, 2011. Visited April 2011.

[66] V. Gupta and M. Wurm. The Energy Cost of SSL in Deeply Embedded Systems. Technical Report TR-2008-173, SUN Microsystems, June 2008.

[67] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady. Security in embedded systems: Design challenges. *Transactions on Embedded Computing Systems, ACM*, vol. 3, no. 3, pp. 461-491, August 2004.