CHALMERS

Penetration Testing of Vehicle ECUs

Master of Science Thesis in the program networks and Distributed systems

VIJAIYA PRATHAP
ABHISHAKE RACHUMALLU

Department of computer science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2013

Penetration testing of Vehicle ECUs

Examiner: TOMAS OLOVSSON
Supervisor: TOMAS OLOVSSON
Chalmers University of Technology
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 August 2012

Abstract

The automobile industry has grown rapidly in the last few decades. The industry is moving towards electronics and software for better efficiency and results. These electronic components consist of hardware and software to control important operations like braking, engine control etc. The future automobiles will be highly sophisticated and extremely integrated with other devices like smart phones and tablets and update protocols like Firmware Update Over the Air (FOTA). It is also possible to have car to car and car to infrastructure communication which will open up for lots of security attacks on the Electronic Control Units (ECUs).

Given the future developments and increasing percentage of electronics in automobiles, it is vital to perform penetration testing of ECUs. A Fault or a failure in one ECU can affect the operations of the automobile and hence endangering the passengers. This occurs due to the properties of In-vehicle networks which are discussed here. In order to find vulnerabilities and access security of an in-vehicular system, it is necessary to perform penetration testing on ECUs.

This thesis focuses on how to perform penetration testing of ECUs by discussing the working of ECUs, its security mechanisms and discovering the vulnerabilities that exist in the ECU. Performing penetration test is essential for designing and building of new ECUs with better performance. The discovered attacks on ECUs are studied and appropriate countermeasures are suggested to patch the vulnerabilities.

Keywords

In-vehicle networks, embedded systems, Electronic Control Units (ECUs), penetration testing, Firmware updates over the air (FOTA)

Table of Contents

Abbreviations:

ECU	Electronic control Unit
FOTA	Firmware updates Over The Air
MOST	Media Oriented System Transport
CAN	Control Area Network
LIN	Local Interconnect Network
CSMA/CD	Carrier Sense Multiple Access/ Collision Detection
TDMA	Time Division Multiple Access
OBD	OnBoard Diagnostics
RSU	Road Side Unit
VANET	Vehicle Addhoc Networks
OBU	OnBoard Unit
PBL	Primary Boot Loader
SBL	Secondary Boot Loader
AUTOSAR	Automotive Open System Architecture
OEM	Original Equipment Manufacturer
PCM	Powertrain control Module
TCM	Transmission Control Module
BCM	Body Control Module
ABS	Antilock Braking System
ESC	Electronic Stability Control
HVAC	Heating Ventilating and Cooling
TCU	Telematics Control Unit
EBCM	Electronic Brake Control Module
TPM	Trusted Platform Module
FPGA	Field Programmable Gate Array
SoC	System on Chip

List of figures:

List of tables:

Penetration testing of vehicle ECUs

1 Introduction

It all started with invention of wheel where early humans can move heavy objects with ease. Then the invention of motors run by fuel giving birth to early cars. As the years passed the complexity of cars gets increased as the passenger comfort levels are also enhanced. In the last few decades automotive industry has seen a rapid increase of electronic components. Gradually other components such as brakes, steering etc. have started to use electronic components to increase efficiency. Today due to the excessive presence of electronic components and software in cars, security attacks directed towards automobiles are not in distant future.

Koscher[1] has demonstrated various security attacks possible in an automobile. A modern automobile is said to have more lines of code than a jet aircraft. In the future, there might be communications between cars and road side units (RSU) through wireless or other protocols. Hence it would be possible to perform firmware updates, send and receive information about traffic and many other things. These new features will make the system more vulnerable to attacks from hackers. If they are targetted carefully, it can have devastating effect on safety of pasengers. Therefore security needs to be addressed in the automotive communication networks.

1.1 Background

With the development of information technology, it has managed to reach out to every industry possible. The automobile industry is no exception. Everyone wants all the comforts and services we can get in a car. There have been efforts made in automobile industry to enhance communication between cars by wireless. By doing this, it would be possible to do Firmware updates over the air (FOTA) with which the user need not go to the service center to update the software in car. In automobile industry, efforts are going on to enable communication between automobiles that is every automobile acts like a node in a network. This is called connected cars. In the recent years, the concept of connected cars is getting popular. There are two types of communication in vehicles.

- Inter vehicle communication
- In-vehicle communication

Inter vehicle communication is implemented using Vehicle ad hoc networks (VANETs). It shares information about the environment such as traffic, weather and road conditions between vehicles. It consists of On Board unit (OBU) which is located inside the vehicle and Road side unit (RSU). The In-vehicle network consist of electronic control units (ECU) which are interlinked by various bus system technologies like Controller Area Network (CAN), Local Interconnect Network (LIN), Flexray and Media Oriented Systems Transport (MOST).

By using these protocols, the amount of wiring in a vehicular network is significantly reduced. These protocols are used depending upon the criticality of the operation and the amount of data transfer required. Important operations such as braking, steering, engine control, electronic gearbox are performed using the CAN and Flexray protocols due to its high capacity to handle data. In this paper, we have focused on stability of individual ECU's to various attacks.

1.2 Objectives

The objectives of this project is to

- Explain about concepts such as in-vehicle networks, embedded system and ECUs to have better understand of the background.
- To understand the security mechanisms and working of ECUs in detail.
- Investigate the existence of various insecurities and vulnerabilities that exists in the ECU.
- Provide a framework for penetration testing vehicular ECUs.
- To discuss the countermeasures for the found vulnerabilities.

1.3 Limitations

In this work we restrict ourselves to the possible vulnerabilities that exist in the ECU and also the possible vulnerabilities which can occur in the future due to technological developments. We didnt had access to the ECU and hence this work is purely theoritical. We also discuss about approaches to penetration testing and doing a white box testing is out of the scope of this master thesis. We dont have input from manufacturers, the exact implementation of security in ECU is unknown.

1.4 Methodology

The following are the steps to be performed to achieve the mentioned objectives

- To have a thorough understanding of vehicular networks.
- Understand the working of ECUs in automobiles.
- To analyze the present security implementations in ECUs.
- Propose realistic attack scenarios on ECU's.
- Suggest possible countermeasures

1.5 Structure of the report

This report begins with an introduction to in-vehicle networks and its security vulnerabilities in chapter 2. Then we move on to talk about embedded systems, general security concepts and taxonomy of attacks in chapter 3. We discuss and define ECU, its components, security mechanism followed by ECU classifications.

In chapter 5 we discuss the various attacks on ECUs and its vulnerabilities. Then we define penetration testing and discuss about the ways in which we perform penetration testing. In chapter 8 we talk about the possible counter measures for the attacks discussed in chapter 5. The last chapters go for conclusion and future work.

2 In-Vehicle network

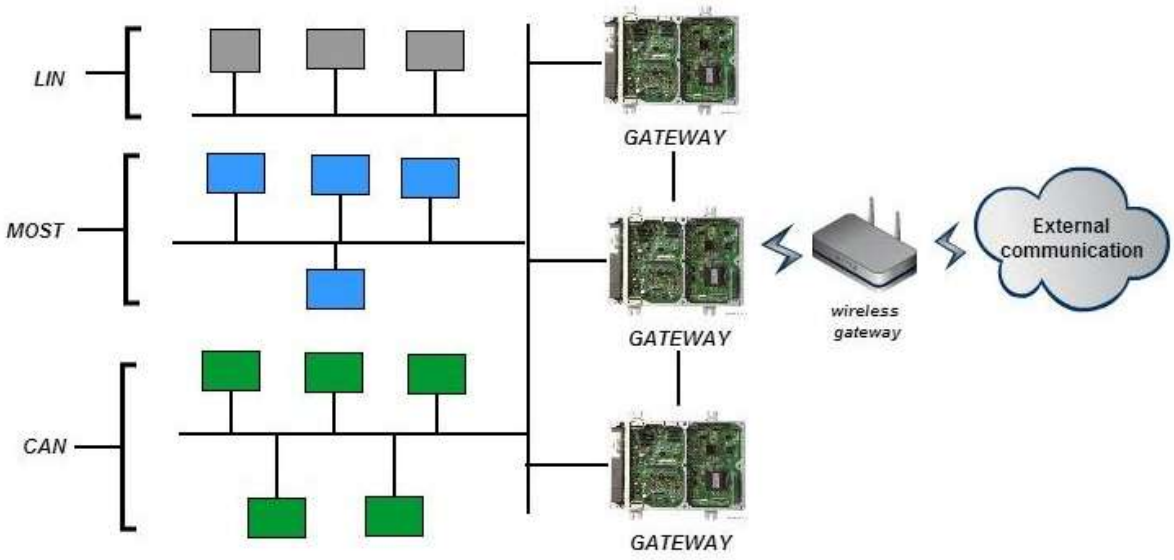

Figure 1: In-Vehicle network

2.1 Background

The increase in number of ECUs in automobiles made the automobile manufacturers to develop communication protocol which could increase the efficiency of communication and reduce the amount of wiring inside the vehicle. The different types of protocols for communication between ECUs are CAN, LIN, MOST and Flexray as shown in figure 1. Communications between these protocols are performed using gateway ECUs. The firmware or software in the ECU must be constantly updated and tested during the servicing of the vehicle. The vehicle network is connected to a computer using the Onboard Diagnostics (OBD) port through which each independent ECUs and their communication can be analyzed through appropriate software. Table 1 shows properties of in-vehicle network in tabular form.

2.2 CAN

The Controller Area Network (CAN) protocol is an event triggered bus system for serial communication with data rate of upto one megabit per second. It can operate even if some of the nodes are defective as it allows redundant networks through a multi-master architecture i.e even if one master node fails, another master node can takeover. The CAN messages are classified by their identifier. It does not have a specific recipient address. The CAN protocol transmits the message to the intended ECU by broadcasting i.e sending the message to all the ECUs and the intended ECU responds appropriately by looking at the message type.

The CAN protocol uses CSMA/CD (Carrier Sense Multiple Access/ Collision Detection) access control method which is decentralized, reliable and priority driven. The CSMA/CD ensures that every time a top priority message is transmitted, it is always transmitted first. The CAN

protocol offers an error mechanism that detects transfer errors, interrupts the erroneous transmissions with an error flag and initiates the retransmission of the affected message [2].

2.3 LIN

The LIN (Local Interconnect Network) protocol is a single-wire sub network for low-cost, serial communication between smart sensors and actuators with typical data rates upto 20 kBit/s. It is used because it is much cheaper than other bus systems and also when the bandwidth and versatility of a CAN network is not required. It has single master, which controls the collision free communication with up to 16 slaves.

Incorrect LIN messages are detected by using parity bits and checksums. LIN provides sleep mode operation to consume less power. This feature can be misused by attackers by sending malicious sleep frames which can completely deactivate the corresponding subnet until a wakeup frame is sent from a higher level CAN bus [2].

2.4 MOST

The MOST (Media Oriented System Transport) protocol is a serial high speed bus typically used for developing automotive multimedia networks for transmitting audio, video, and control data. It uses fiber optic cables for transmission of data. The peer to peer network is connected by plug and play up to 64 nodes in ring, star or bus topology. It offers data rate of 24 MBit/s for synchronous transmission and 14 MBit/s for asynchronous transmission. In MOST messages, the sender and receiver addresses are always clearly mentioned.

The access control for synchronous transmission is done by TDM (Time Division Multiplexing) and for asynchronous transmission is done by CSMA/CA. Since MOST device handles role of the timing master, malicious time frames disturb and interrupt the MOST synchronization mechanism. Continuous bogus channel requests reduce the remaining bandwidth and are a feasible jamming attack on MOST busses [2].

2.5 Flexray

The Flexray protocol is a deterministic and error-tolerant high speed bus, which is used for future safety related high speed automotive networks. It has a data rate of up to 10 MBits/s. The Flexray network is flexible and expandable and has upto 64 point to point connected nodes. Both optical fibers and copper lines are suitable for physical transmission medium. For priority driven control of asynchronous and synchronous transmission, Flexray uses cyclic TDMA (Time Division Multiple Access) method.

The ways of error tolerance in Flexray are through channel redundancy, checksum and independent bus guardian that detect and handle logical errors. Fake errors messages can be created and directed at components to deactivate them using the bus guardian. It is also possible to completely deactivate the Flexray network when there is an attack on common time base [2].

Bus	LIN	CAN	Flexray	MOST
Adapted for	Low-level subnets	Soft real time	Hard Real-time	Multimedia Telematics
Target Application Examples	Door locking, Climate regulation, Power Windows, Light, Rain sensor	Antilock braking system, Driving assistants, Engine control, Electronic gear box	Brake-by-wire, Steer-by-wire, Shift-by-wire, Emergency systems	Entertainment, Navigation, Information services, Mobile Office
Architecture	Single master	Multi master	Multi master	Multi master
Access control	Polling	CSMA/CA	TDMA, FTDMA	TDM, CSMA/CA
Transfer mode	Synchronous	Asynchronous	Synchronous, Asynchronous	Synchronous, Asynchronous
Data rate	20 kBits/s	1 MBit/s	10 MBit/s	24 MBit/s
Redundancy	None	none	2 channels	none
Error Protection	Checksum, Parity bits	CRC, Parity bits	CRC, Bus Guardian	CRC, System service
Physical Layer	Single-Wire	Dual-wire	Optical Fiber, Dual-wire	Optical Fiber

Table 1: Properties of automotive bus systems

3. Embedded Systems Security

An embedded system is a special purpose computer within a mechanical or electrical system, which are designed to meet specific needs. Basically all embedded systems consist of hardware and software, which performs the necessary action. Different types of embedded systems exist according to the function it performs. Today embedded systems are used in almost all industries such as defense, consumer, automobiles, telecommunications etc. The embedded systems in comparison to normal computers and laptops consist of low configuration and runs highly efficient code. According to Turley [12], only 2 percent of microprocessors produced in the world are used in normal computers, rest is used in embedded systems.

3.1 Security concepts

Embedded systems are used in almost every industry and it is handling critical information, security issues of embedded systems are to be considered. Security is defined as a protecting the system against unauthorized use, access, disclosure, modification or disruption in order to provide three core principles of confidentiality, integrity and availability.

- **Confidentiality:** The information in system is available only to the intended recipient. It shouldn't be possible for unauthorized access of critical information. If the data is not kept secret, it can result in heavy losses.
- **Integrity:** The information in the system should not be able to change or modify by unauthorized people. If the system doesn't has integrity, a malicious user or command can change the intended working of system causing harm to the user or the device.
- **Availability:** The system must be able to perform and information must be available for proper use of the system. Lack of availability results in denial of service to the users.

The above mentioned properties are the primary attributes for Security in any system. Secondary attributes are that composite of primary attributes i.e they share the properties of two or more primary attributes [13].

- **Authenticity:** The information given to the system must be true i.e the integrity of message content and its origin is not compromised. The proof of authenticity is done by authorization. The proof might be something a user knows (eg password), or a user has (eg keycard), or user is (eg biometric scan).
- **Accountability:** It should be possible to ascertain who has accessed the information and make them responsible for the accessed information in the system.
- **Non-Repudiability:** It shouldn't be possible for the user to deny the information given to the system. This is achieved through digital signatures.

3.2. Attack Taxonomy

Howard et al. [14] have introduced common language for computer and network security terms to classify information and events into a common taxonomy. They have given some general terms as well as structure for the incidents. The attack scenario given in [14] consist of five logical steps that are given in an attack matrix are Tool, Vulnerability, Action, Target and Unauthorized Result. In short, an attacker might use a tool to exploit a vulnerability to perform an action on target to achieve unauthorized results.

Tool: It is a means or software used by which vulnerabilities are exploited or rules are violated.

Vulnerability: It is a flaw or weakness in a system that might lead to security breach by a malicious user.

Action: It is an activity to gain result by either a user or a process. It is perceived to be directed towards the target.

Target: It is categorized into two entities- logical entity such as account, process, or data and physical entity such as component, computer, network, or internetwork.

Unauthorized attack: To gain unauthorized result by the action.

Attack: Attack happens when a series of steps on a computer or network achieve an unauthorized result.

Event: An event is a discreet change of state or status of a system or device.

Incident: Group of attacks that can be distinguished from other attacks because of the distinctiveness of the attackers, attacks, objectives, sites and timing.

Two more things which need to be discussed are the attackers and their objectives.

Figure 2: computer and network incidents

An attacker is an individual who initiates an attack to gain his/her objectives. The objectives are the final goal of such attackers. Wolf et al [2] has presented 3 categories of attackers: car owners, garage personnel and third party. Out of these three, garage personnel are the most powerful attackers as they have the technical expertise as well as complete access to the automobile.

Taking all these mentioned above and addressing other issues, Hoppe et al [15] proposed the picture given below for attack taxonomy for vehicle environment.

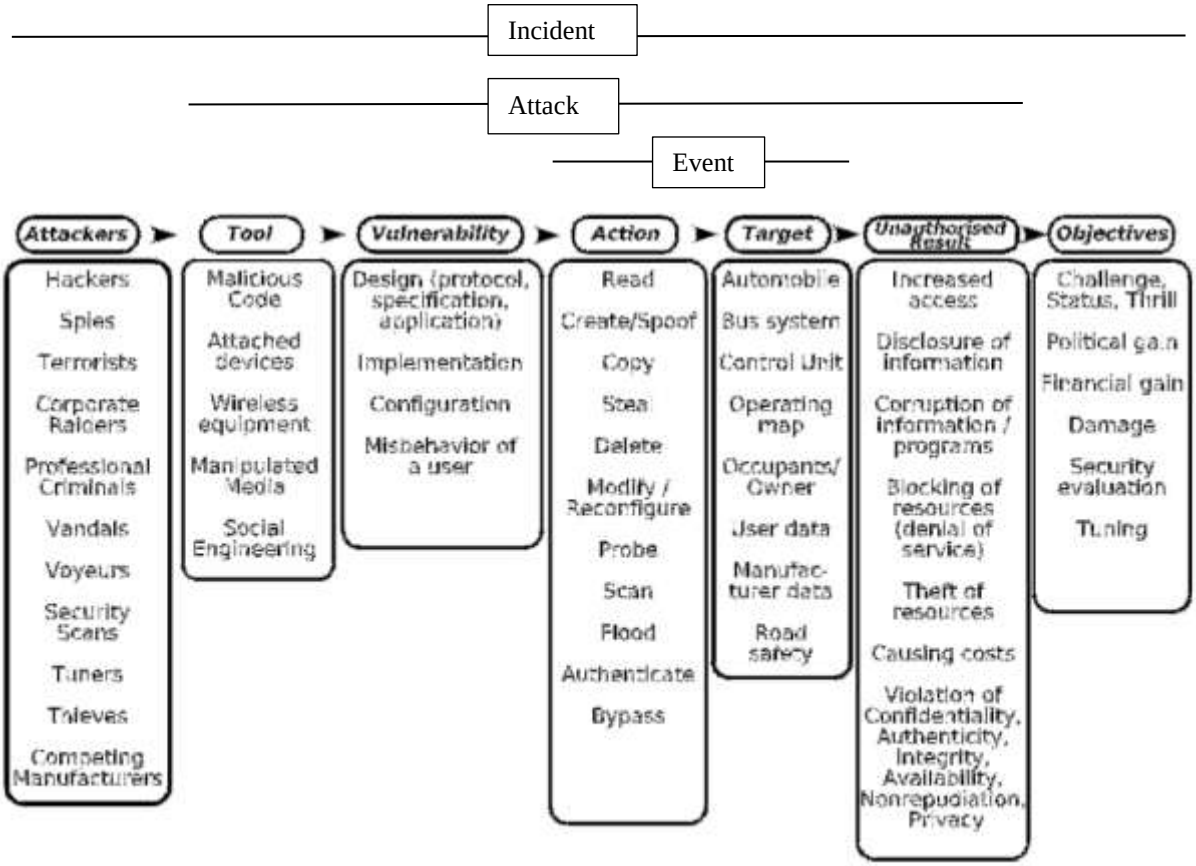

Figure 3: CERT taxonomy adapted for use in automotive environment

3.3. Threats and attacks

The threats faced by any embedded systems are of two levels: Top level attacks and bottom level attacks. The top level attacks are classified into three main categories based on their function objectives.

- **Privacy attacks:** These attacks are performed to gain access to sensitive information stored or communicated within an embedded system.
- **Integrity attacks:** The data or code associated with the embedded system is changed in these attacks.
- **Availability attacks:** By misappropriating system resources, normal functioning of the system is disrupted. It causes the system to be unavailable for normal operation.

These are the top level attacks in an embedded system. The second level of attacks is based on agents or means used to launch the attacks [16]. These attacks are software attacks, physical or intrusive attacks and side channel attacks. These attacks are explained in detail in the subsequent paragraphs. The classification of attacks on embedded systems is shown in the figure below.

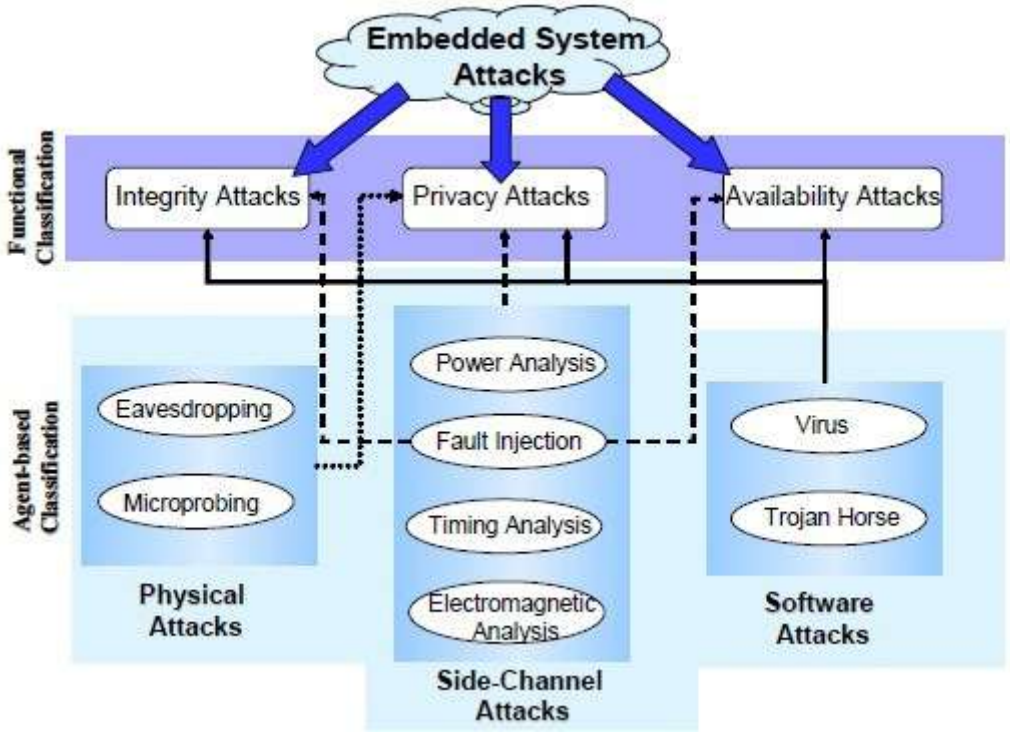

Figure 4: Taxonomy of attacks on embedded systems

3.3.1. Software attacks

Software attacks are a major threat to embedded systems as it does not require expensive infrastructure. Malicious software agents mount software attacks by exploiting weakness in the end system architecture which arises due to either vulnerabilities or exposures [16]. Vulnerability allows an attacker to gain control of the system where as an exposure is an entry point that an attacker may indirectly exploit to gain access. The software attacks are performed in many ways such as code injection, bufferoverflows and attacking cryptographic protocols [17].

3.3.2. Microprobing or Physical attacks

Physical or intrusive attack is to damage or gain the knowledge of working of embedded system device by physical tampering. Ravi et al [16] has provided some steps for performing physical attacks. They are listed below

- The first step is De-packing, that is to remove the chip of package by dissolving the resin covering silicon using fuming acid.
- The second step is reconstructing layout using a systematic combination of microscopy and invasive removal of covering layouts.
- The third step is to use techniques such as manual microprobing or ebeam microscopy to observe the values on the busses and interfaces of the components in a depackaged chip.

Physical attacks on chip requires expensive infrastructure and hence it is hard to use. However once they are performed, they are used as precursors to the design of successful non-invasive attacks. Some examples are, layout reconstruction is done before performing electromagnetic radiation monitoring. Similarly the knowledge of ROM contents which includes cryptographic routines and control data that can provide information that can help in the design of non-invasive attack to an attacker.

3.3.3. Side channel attacks

A side channel attack is any attack based on the information gained from the physical implementation of a system. Various side channel attacks are Power analysis, Timing attacks and electromagnetic analysis. The power analysis attack monitors the power consumption by the system. As power consumption is data dependent, the key used can be inferred from this attack [16].

Timing attacks exploit the observation that the execution times of cryptographic computations are data dependent and hence can be used to infer the cryptographic key [19]. The Electromagnetic analysis attack measures the electromagnetic radiation emitted by the device to reveal sensitive information. It was shown that using electromagnetic analysis the screen contents of a video display unit can be reconstructed [20].

3.3.4. Fault injection attacks

Fault injection attacks are performed by varying the external parameters and environmental conditions of a system such as voltage, temperature, clock, radiations etc.. to induce faults in the components. Ravi et al [16] has given ways in which faults injected in the system can compromise its security. They include Availability attack, integrity attack, privacy attack and pre cursor attacks.

3.3.5. Reverse engineering attacks

Reverse engineering attack understands the inner structure of device and learns to emulate its functionality. An attacker who performs is able to reverse engineer the device, gets similar capabilities of a device manufacturer. Reverse engineering is an invasive attack, hence it may not be possible to reuse the same device after trying to reverse engineer. Dr. Sergei [18] describes the possible approaches in performing this attack.

4. Electronic Control Units

The Electronic Control Units (ECU) is an embedded system used in automobiles, which performs a variety of operations in automobiles. These are electronic components which perform required actions by getting different information through sensors and it has its own processor and memory for calculation and storage of values. Hence they are also called as the computers of an automobile.

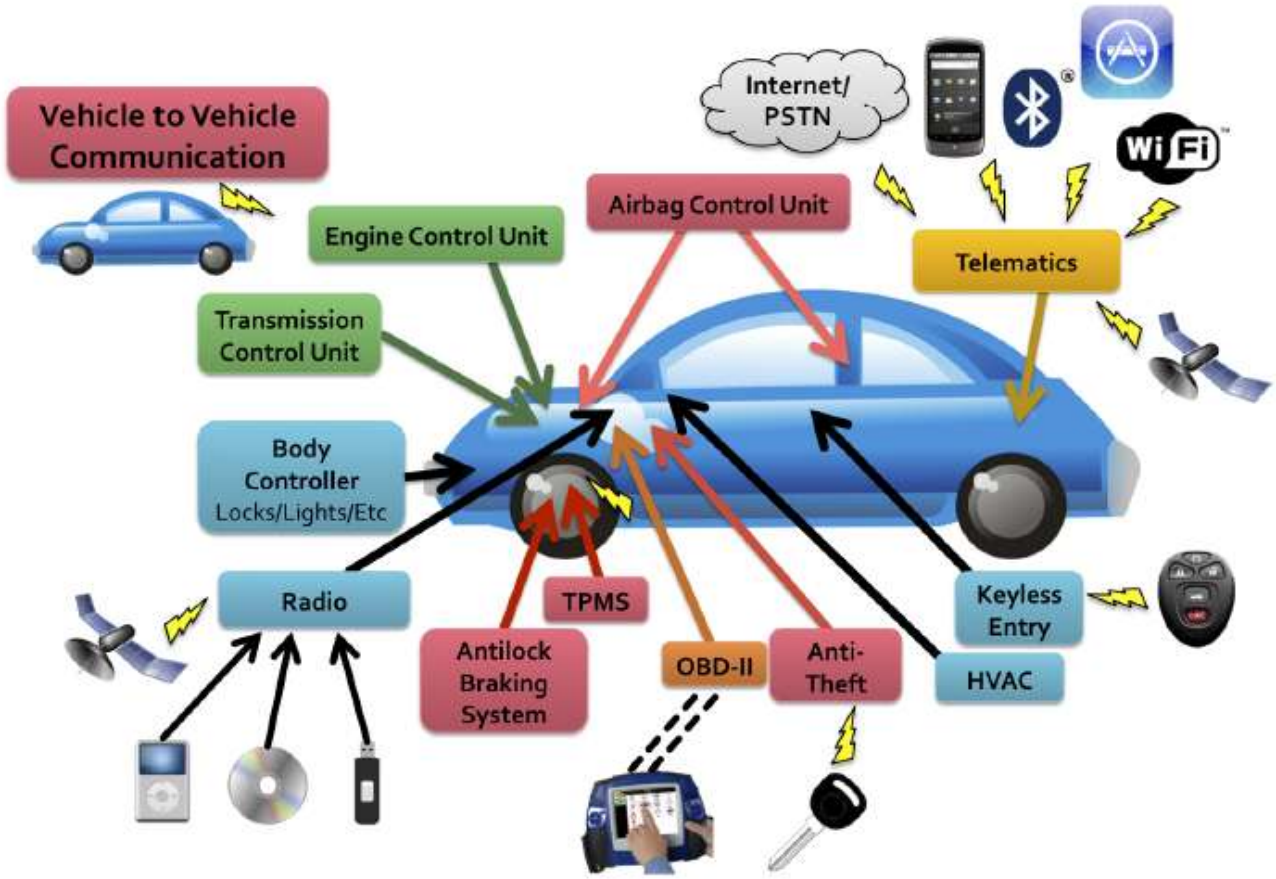

Figure 5: List of components used in cars

The functionality of each component in the automobile is controlled by an ECU with specific firmware. Gradually other mechanical components are replaced with electronic components controlled by software. The value of electronic components in cars will be 40 percent of its total value by 2015 [3]. With the ever increasing number of ECUs, various third party software for the electronic components have emerged. Hence the need for authentication of ECUs and protecting its integrity is very vital in the present automobile industry.

4.1 ECU components

The Electronic Control Unit (ECU) is an embedded computer that is constructed from printed circuitry and consists of many electrical components run by firmware. It controls and regulates various functions in a vehicle. The ECU is connected to a network of other ECUs through busses and gateways. It sends and receives signals through sensors and actuators. The sensors

convert physical input such as speed, temperature, etc.. to electrical signals which can be processed by the processor in ECU. Actuators are devices which converts electronic signals from the processor in ECU to physical signals which can be understood by the components in vehicle.

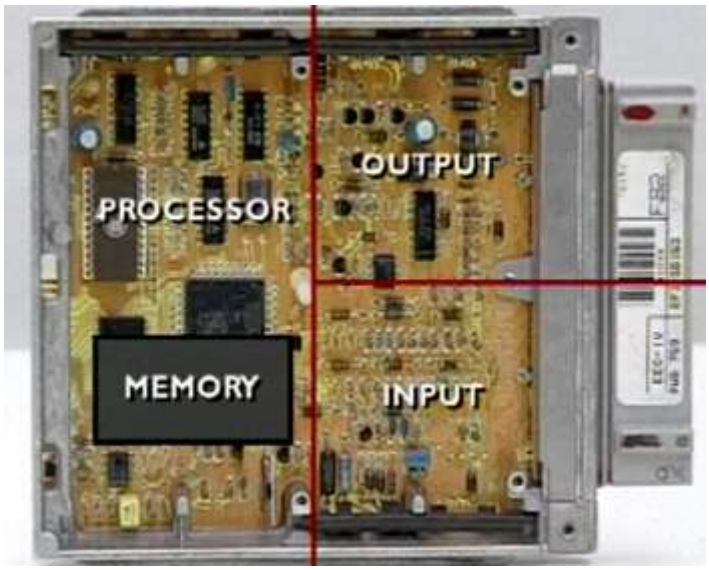

Figure 6: components of a typical ECU

A modern ECU might have a 32 bit 40 MHz processor and the code in ECU might take upto 2MB. A typical desktop computer has a processor which runs at 1000 to 2000 MHz of speed and with atleast 2 GB of memory. This shows that the individual ECU has less computing power in comparison to a desktop computer. The processor takes up input from the sensors, and sends appropriate signals to the actuators by analyzing the input and the data from the memory of ECU.

Memory:

The memory stores the code of ECU which defines its complete behavior. A typical ECU consists of two kinds of memory. They are flash memory and RAM memory. The flash memory does not lose the data stored in it, when the power is switched off. Hence it is a non-volatile memory. It cannot be overwritten as it has a primary boot loader but other parts of the memory can be erased and reprogrammed.

The ECU first executes the boot loader. The bootloader consists of two separate parts. They are: primary boot loader (PBL) and secondary bootloader (SBL). When an ECU is started, the application software in the flash memory is executed by PBL. The PBL has a very small memory size (16k); hence it cannot be modified after the unit is produced. It is also not suitable for programming or update data in flash memory. The SBL is downloaded by the primary loader into RAM and is then activated. Then it becomes responsible for the management of the flash memory i.e. erasing or programming and the software download process [9] as shown in the figure 8.b.

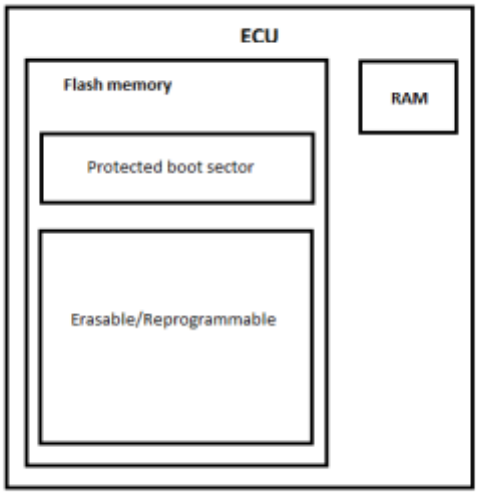

Figure 7: ECU hardware memory map

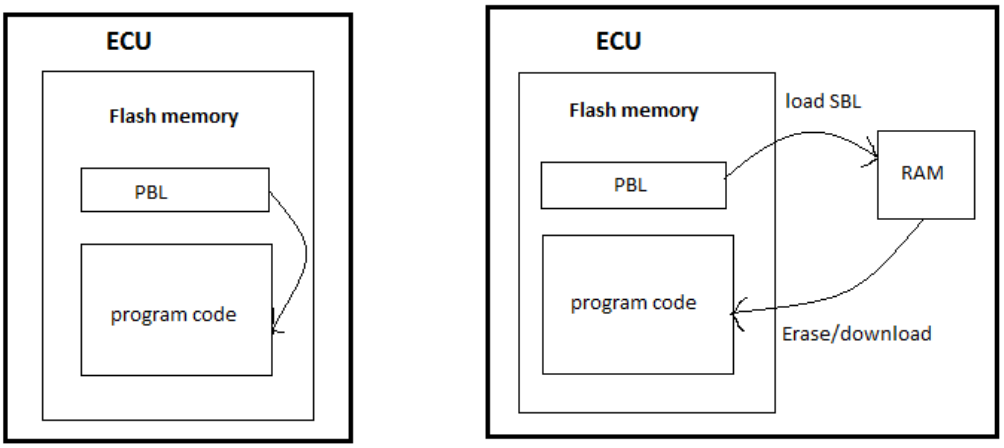

Figure 8: ECU's execution modes

4.2 Software Architecture

The automotive industry has witnessed rapid induction of electronic components which runs by software. 30 years ago, tiny bits of software was introduced in cars to control the engine. There were mostly coded in machine code or the programming language C. The amount of software in a car today is about million lines of code, implements about 270 functions a user interacts with, deployed over 70 embedded platforms which amount to around 100MB of binary code [10].

The current requirements in the automotive industry has risen to a great extent such that new technological breakthrough is required to fulfill customers and legal requirements. In Order to address this, various OEM manufacturers and tier 1 suppliers have collaborated and formed the Automotive Open System Architecture (AUTOSAR). It is an open and standardized software architecture developed jointly by automobile manufacturers, suppliers and tool developers. The goals of AUTOSAR are

- Standardization of basic software functionality of automotive ECUs.
- Scalability to different vehicle and platform variants.
- Transferability of software.
- Support of different functional domains.
- Definition of an open architecture.
- Collaboration between various partners.
- Development of highly dependable systems.
- Sustainable utilization of natural resources.
- Support of applicable automotive international standards and state-of-the-art technologies.

To achieve the given technical goals of AUTOSAR such as modularity, scalability, transferability and re-usability of functions, AUTOSAR provides a common software infrastructure based on standardized interfaces. AUTOSAR enables configuration process optimization and wherever necessary to allow local optimization to meet runtime requirements.

4.3 Security mechanisms

The ECU has challenge response protocol to authenticate communication between the user and the ECU. Since by rule no encryption algorithm can reside in an ECU, both the challenge and its response are in the memory of the ECU [1]. Firstly a user asks for a seed from the ECU. The ECU responds with a seed to the user where he computes the key with a seed to key algorithm and sends it back to the ECU where the calculated key is checked [23]. If this stage is successfully passed, the ECU is open for modifications. The challenge response protocol is shown in the figure below.

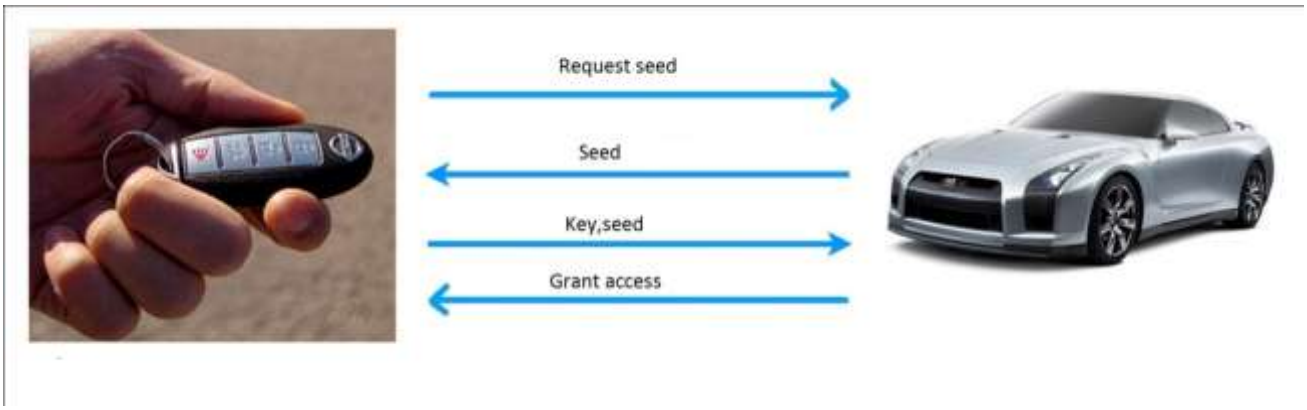

Figure 9: ECU challenge response protocol

It is also known that the software in ECU has the signature of the manufacturer to authenticate itself. In general, the software image is hased and decrypted with manufacturer's private key to form signature. This is then authenticated by the device using its public key to

encrypt the signature and check if the encrypted signature and hash of software image are the same.

4.4 Firmware updates over the air (FOTA)

The software in ECUs should be frequently updated with latest versions to enhance the performance of the vehicle. Usually these software updates are performed through OnBoard Diagnostic (OBD) protocol. The software updates are performed using any software tool for communicating with ECU installed in a laptop which is connected through OBD cable. The current research activities in vehicular on board IT architectures follows two basic trends. They are unification of network communication and centralization of functionality. Hence in future it is possible to perform software updates on ECUs through wireless.

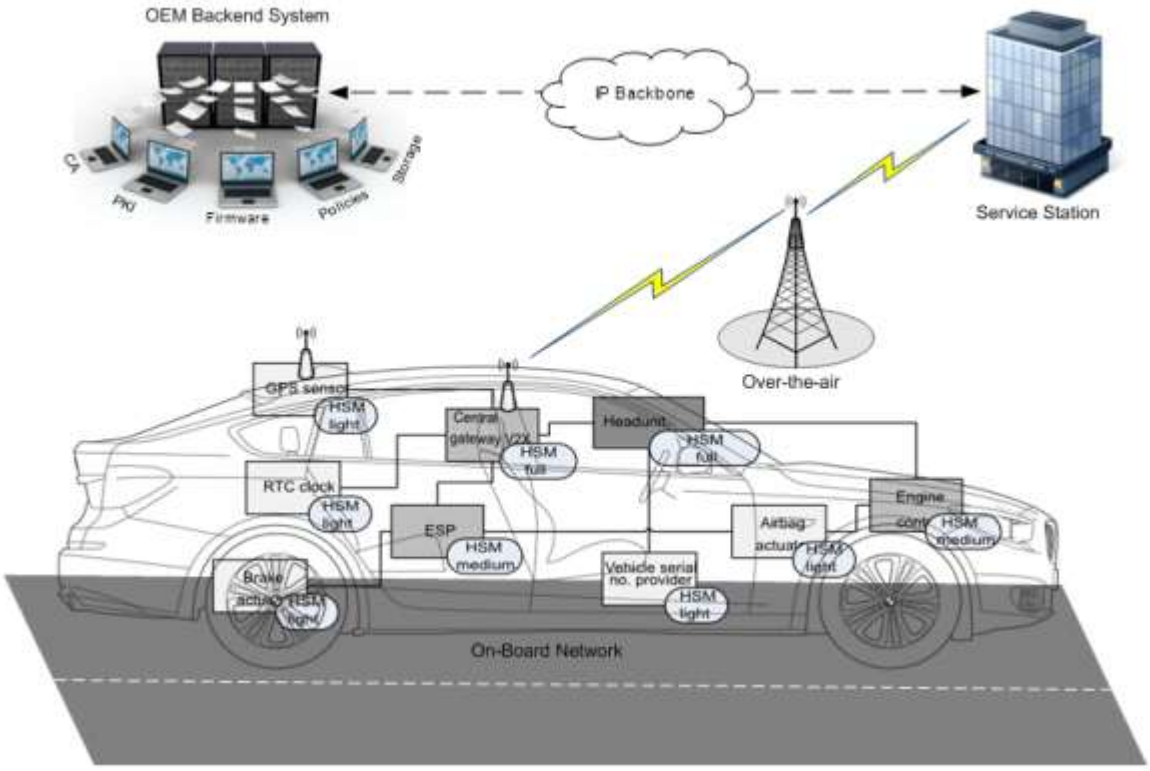

Figure 10: Firmware Updates over the air

A pictorial representation of FOTA is given in figure 10. The automobile gets the updates from Original Equipment Manufacturer (OEM) through the service station via internet. The service station sends the update to the wireless gateway in the automobile which communicates with the various ECUs and the appropriate software is loaded in it. There exist technical and practical difficulties in implementing FOTA. Due to the transfer of data over the air, FOTA opens to possible cyber-attacks on the automobiles. Idrees et al [11] suggests a protocol for securing firmware updates by hardware and software modules. These modules provide various security mechanisms like encryption, decryption, signing etc. Nilsson et al [5] suggests a protocol which involves using hashing chains.

5. Classifications

ECUs are connected to each other through different protocols like CAN, LIN, MOST and Flexray. Communication between each protocol is done through gateway ECUs. Nilsson et al [4] classifies ECUs according to safety integrity levels, i.e the ECUs are classified according to the value of damage by attacks on the ECUs as shown in table 2.

ECU category	SIL
Powertrain	4
Vehicle safety	4
Comfort	2
Infotainment	1
Telematics	1

Table 2: Ecu classification with SIL values

In their conclusion, they say that powertrain and vehicle safety ECUs should be provided with more protection if we are to introduce remote connectivity. Let us now discuss each category of the classification in detail.

5.1 Powertrain:

The powertrain consists of critical components such as engine and transmission controls. There are three computers in powertrain such as Powertrain control module (PCM) which acts as a brain of the whole automobile, Transmission Control Module (TCM) which controls modern electronic transmission of messages between various ECUs and Body control Module (BCM) responsible for monitoring and controlling various electronic accessories. When a fault occurs in PCM, the check engine light on the dashboard will appear. Failure of powertrain ECUs can cause critical damage as these ECUs are responsible for working of critical functions such as acceleration, braking etc.

5.2 Vehicle safety:

Vehicle safety is of two types. They are passive safety and active safety. Passive safety features are built inside vehicles to minimize the harm during an event of a crash. Ground breaking passive safety features include seat belts and air bag. In modern automobiles, air bag deployment, seat belt sign in dashboard are controlled by these vehicle safety ECUs.

With significant advances in technology, it is now possible to prevent accidents and to minimize the damage in case of unavoidable accidents. Various devices or features which comes under active safety are Anti-Lock Braking system (ABS), Electronic stability control (ESC), Brake assists etc. The ABS is used to prevent locking of wheels when the driver hits the brake hard at slippery conditions. The ESC is used to control the vehicle in case of a blind turn, a driver might do

an oversteer or an understeer. ESC negates the driver error and prevents the vehicle from lopsiding.

5.3 Comfort:

The comfort ECUs provides driver assistance systems and passenger comfort systems. Driver assistance systems such as Lane, park, and speed assist systems which are used to monitor the car for not crossing out of lane by providing visual or audio warnings to a driver and help the driver to park the car in parking spots by using sensors. Intelligent speed assistance system regularly watches over the speed limit of car, which should not to exceed the local speed limit by providing warnings.

Passenger comfort systems consist of Automatic climate control which controls heating, ventilating and cooling (HVAC) systems in the car. Automatic windshield wipers are used to keep the windshield clean from rain water, snow, fog and dust. The Automatic headlight alters the lighting of head lamps according to driving situations and curves in the road. A breakdown in these ECUs is not as critical compared to the powertrain and vehicle safety ECUs.

5.4 Infotainment:

The Infotainment systems are gaining more popularity in the cars and provide a myriad of services which includes navigation system which provides turn-by-turn directions by using GPS or electronic maps through display and voice interfaces. Entertainment systems such as AM/FM radio, CD/DVD players, TV, surround sound, game consoles and controls for, headlights, air conditioning, wipers etc. Bluetooth streaming that allows Bluetooth enabled phone or a device to connect with vehicle and one can control functions like checking text messages, answering calls, music from phone etc., with a smart display screen in dashboard or LCD display depending on car variant. According to a survey from a market researcher, every new car built in Europe will have internet connection by 2013 [6]. A failure in these ECUs is not considered as a violation of safety issues.

5.5 Telematics:

The telematics system provides an interface between the user and the mechanical or electrical components. The Telematics Control Unit (TCU) is a small computer about the size and weight of a paperback book, that communicates with automobile ECUs and GPS satellites and accesses telematics services such as Automatic crash notification, vehicle tracking, remote door services, traffic assistance, diagnostics and much more. The telematics unit gets the information about the ECUs from the CAN network and stores them. However for this information to be useful, it should be possible to communicate with external environment.

There are different manufacturers for telematics units such as Hughes, Onstar and many more. Hughes telematics has released a white paper on automotive telematics which is described in [7], explains the telematics architecture, telematics services and telematics control unit in detail. The telematics network topology says that communication between the car and person at call center can be done by four concentric circles of communication. They are Bluetooth, GSM or GPRS, wireless LAN and through satellite.

6 Penetration Testing

In this section we give a brief description of penetration testing. Penetration tests are necessary to detect vulnerabilities in system or network, and the knowledge of past and present vulnerabilities of the system is essential for securing it. Although temporary patches to the equipment provide security, penetration tests are useful to detect various vulnerabilities which are present in the system.

6.1 What is penetration testing?

Penetration testing is an accepted procedure that is performed to find vulnerabilities in hosts, networks and application resources using known attacks. Penetration testing also known as Pen test, is a method which is used to check if the existing and new applications, networks and systems are vulnerable to a security risk that could allow unauthorized users to access resources.

A penetration test usually includes the use of attacking methods conducted by trusted individuals. The tests can range from a simple scan to identify IP addresses of systems to identify vulnerabilities or exploiting known vulnerabilities. The results of the tests and attacks are documented and presented to the owner of a system in order to address the identified vulnerabilities [34].

One must keep in mind that penetration testing is not a fully security audit, rather it gives security at a single moment in time. Penetration tests are performed in order to secure sensitive information and resources from hackers or unauthorized individuals. For example, a company asset might be compromised if vulnerabilities in the system are exploited by unauthorized persons. Penetration testing is either used to create awareness of security issues among higher management or to test intrusion detection and calculate responses. The aim of penetration testing is to identify vulnerabilities before being exploited.

6.2 Internal vs. external penetration testing

A penetration tests typically simulates either inside attacks or outside attacks.
There are two types of penetration tests: Internal and external

An external *test* is a traditional approach that lets a remote attacker to attack the network from outside. The attacker targets Internet connected systems that are connected to internal networks to find potential weaknesses. For example, by opening services on internal servers or gaining access to network devices (routers, firewalls). The attacker gathers required knowledge about internal systems and network from various resources to perform such an attack.

Internal tests are attacks that are performed by a person who has an authorized access to the internal network or through social engineering. Internal tests are intended to identify the vulnerabilities that exist for systems that are accessible to authorized network connections that reside within internal perimeter. Inside attackers are more dangerous than external attackers

because insiders already have necessary information about the network to exploit it, whereas external attackers do not have it in their initial stage.

6.3 The process and methodology

To perform penetration testing, there exists four phases as shown in the figure below.

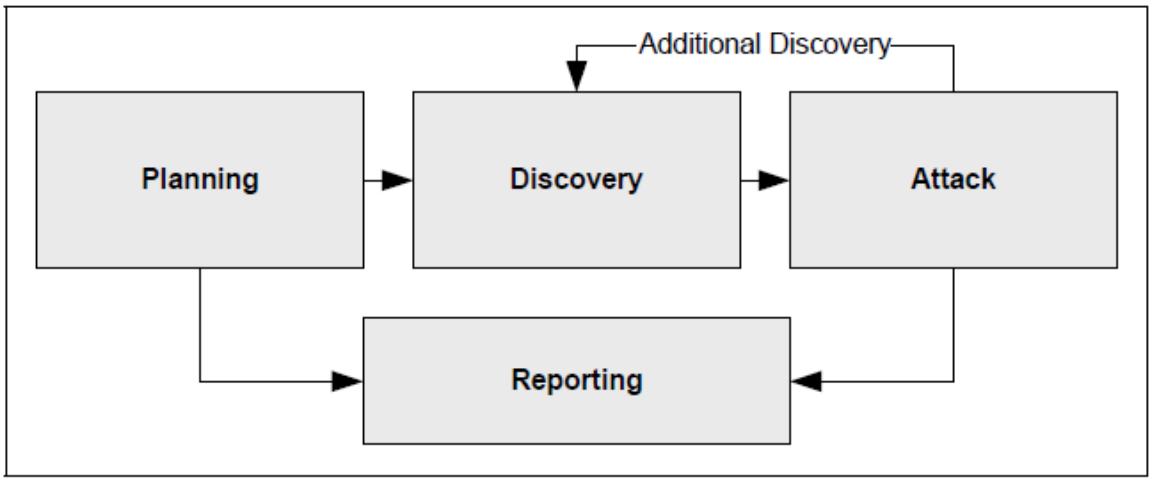

Figure 11: phases of penetration test methodology

- **Planning:** In the planning phase, the objectives of penetration testing is determined, rules are established, the resources that need to be tested and staff who are identified. The foundation work is done in this phase to achieve for successful penetration testing [35]. No actual test is done in this phase.

- **Discovery:** The discovery phase consists of two parts;
  - **Information gathering and analysis:** After finishing the planning phase, the next step is to gather as much information as possible about the targeted system using various tools. The actual testing is started in this phase.
  - **Vulnerability detection:** The information gathered from the previous step is used to determine the existing vulnerabilities in the targeted systems. The vulnerabilities found are compared with a vulnerability database to see if we already have a countermeasure for the attack. The collection of exploits and vulnerabilities found are important to perform successful penetration testing [34].

- **Attack:** This is the important phase of penetration testing. The vulnerabilities that are identified in the previous step are used in this phase when trying to exploit the system. If an attack is successful, then it is verified and measures are proposed to mitigate the risks. If testers are able to exploit the vulnerability successfully on the target system, they can install more tools on target system to gain access to additional systems or resources on the network. These attacks have to be conducted on multiple systems in order to determine the

level of access that an attacker can gain. The following figure shows the individual steps of this phase [35].

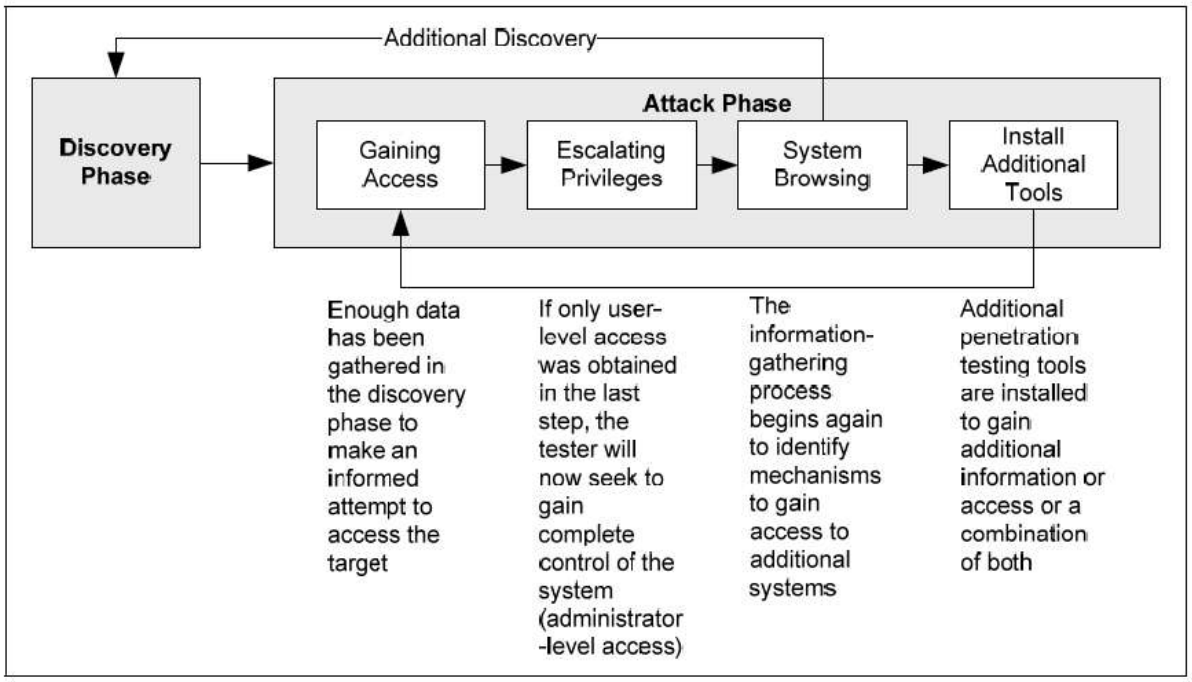

Figure 12: Attack phase

- **Reporting:** The reporting phase is done together with the other three phases. It is important to generate a comprehensive report that includes details of how penetration testing is done, collections of vulnerabilities that are found during the tests and approaches to resolve those vulnerabilities. This report can be used to analyze how an adversary can exploit weaknesses and will also give guidance how to mitigate the discovered attacks in the system or network.

- **Cleaning Up:** This cleaning up process is done to clean the mess that has been made as a result of penetration test. E.g. removing user accounts that are created during the process of penetration tests.

6.4 Rules of behavior

The rules of behavior for the penetration testing are governed by an agreement between the target organization and test team to ensure that there is a common understanding of the limitations, constraints, liabilities and possible compensations throughout the penetration tests [36].

The rules of behavior define the scope of tests by defining targets, rules and time frames. They also address the type of tests to be performed, risks involving in tests, limitations of approaches and techniques of attacks. The 'rules of behavior' also permits to proceed with the penetration test [36].

6.5 Criteria of success

The rules of behavior should be developed with success criteria in mind. It is important to determine the time frames and conditions to perform penetration testing. Once the success criteria have been achieved, penetration attempts should be terminated immediately and safely.

The different goals of penetration test include [36]:

- Access to internal resources
- Reading and modifying restricted files
- Reading and executing transaction data
- Controlling network management systems
- Access to any user accounts
- Access to supervisor privileges
- Demonstrating ability to control resources.

It is essential to have well defined success criteria, as a failure to properly defined conditions will result in and misconceptions that could lead to a false sense of security.

6.6 Penetration Approaches

There are three types of approaches for penetration testing.

- Zero knowledge: In this approach the test team will not have any information about target system. So the test team should start gathering information to proceed further in performing penetration testing. This test provides most realistic penetration test.
- Partial Knowledge: In this approach the test team is provided with information from target organization to perform tests. This type of test is chosen when target organization wants to test on specific attack or to test on specific targeted network or host.
- Full Knowledge: In this approach the test team has as much information about the target system as possible. This test is used to simulate an attacker who has familiar knowledge of target organization's systems like a current employee.

6.7 Limitations

Penetration testing is just a snapshot of the targeted systems and network at a moment. There are severe restrictions on penetration testing especially on time and cost issues. It is not intended to be a complete evaluation of security, since the targeted organization may add or change functionalities in the targeted system or network for business purposes, so many security attacks and configuration issues may not be identified during tests. The amount of data that is collected during the given time period is an important element to evaluate the validity of a penetration testing [33] [36].

7 Attacks on ECUs

The ECUs are embedded systems which are vulnerable to most of the common attacks possible in embedded systems. Due to high performance requirement of ECUs in automobiles, security is not focussed sufficiently. In the future, when FOTA is feasible, it would be possible to change the contents of ECU software from a remote location. This might give rise to many possible attacks by malicious users which could jeopardize the normal operation of an ECU. Hence it is vital to discuss the security loopholes in an ECU to perform penetration testing.

7.1 Brute force

Using any diagnostic software, it is possible to read the contents of ECU and perform required changes to it. However some operations are disabled by the automobile manufacturer inorder to protect the working and integrity of a vehicle. To unlock all the operations, a separate key should be fed to the ECU. Koscher et al [1] says that all ECUs have some fixed seeds and corresponding keys and every next attempt is done after 10 seconds. With the given scenario, an attacker can crack an ECU around 7 days by using all possible key combination for each seed.

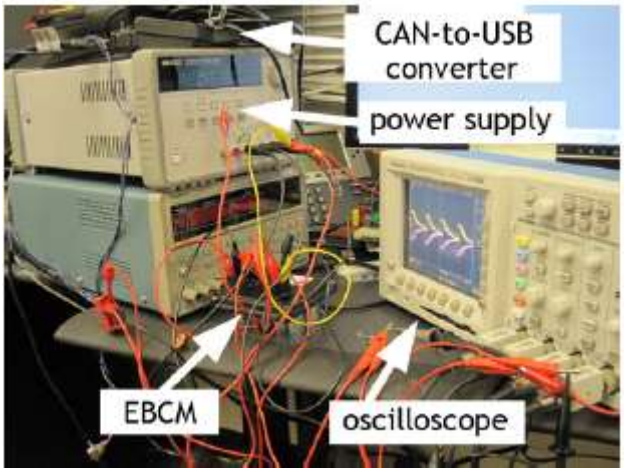

Figure 13: Bench setup of brute force attack

The required to break the key reduces considerably if the ECU is reset after every two attempts and if we can physically remove the ECU. Koscher et al [1] uses Electronic Brake Control Module (EBCM), CAN to USB converter and an oscilloscope to break the key of EBCM as shown in the figure. By using the above setup, the ECU is cracked in about one and half days.

7.2 Reverse engineering

Reverse engineering is the process of discovering the technological principles of a device, object or a system through analysis of its structure, function and operations [22]. In this context, it involves taking a device or software and analyse the working of each components in it. Reverse engineering gives way to pirate software and cloned parts which could have big economic impact on the automobile company.

Since the software in ECU is not encrypted due to memory constraints, It can be easily read by using appropriate software. The software loaded in the ECU has an image inorder to detect any changes in the software. Even though we cannot change the software, we can copy both the software and its image into a new ECU [21]. An attacker having physical access to an ECU can disassemble its parts and study the working of each and every components.

7.3 Software manipulation

ECUs employ the use of signatures to detect illegal modification of software. The ECU software is signed by the manufacturer by a private key and the signature is verified by the public key stored in ECU. An attacker can replace parts of the ECU software image or the entire image on external storage, causing the ECU to execute software which is not authorized. Hence a unique signature is used by calculating a unique digest value over the software image and the manufacturer uses the private key to sign the software image.

An attacker who wants to inject a new software image can create a new public private key pair and uses the private key to sign the software and stores the public key, new software image and signature in the system [21]. The OnBoard Diagnostic protocol in automobiles can be abused to download malicious software which can perform illegal operations. Software in ECU can also be illegally modified by abusing onboard diagnostic commands such as WriteMemoryByAddress() function provided we can break the challenge response protocol [23].

7.4 Attacking the memory

An ECU memory consists of important data which are required for the functioning of automobile components. The memory capacity of an ECU is usually very low and the software in ECU is written in unsafe programming language like C, Hence it is possible to perform buffer overflow attacks on ECUs [24]. A malicious user can inject unexpected error into the memory by power shutdown, injecting wrong values etc. These failures might cause corruption of data stored in the memory and hence disturb the integrity of the data [21].

The memory of an ECU is also subjected to offline attacks. We can disassemble the ECU and takeaway the memory chip using anti-static mat and a soldering iron and using an EEPROM reader we can get the complete software image. Now to make sense of the software image, we need to use a disassembler. Hence contents of the memory can be easily read out and it is not secure [25].

7.5 Unauthorized hardware

The usage of Unauthorized parts to communicate with ECU can have economic impact on the automobile company and it can also endanger the safety of automobile as the parts might not be tested. The component authentication is one-way i.e the ECU authenticates the component but not the other way round. The ECU sends a challenge to the component to identify itself. The component responds to the challenge and hence the device is authenticated by the ECU [21]. If an attacker is able to replace the secret in ECU, then the attacker can connect his non original part to the ECU.

7.6 Attacks on ECU communication

The ECUs in an automobile communicate with each other by using protocols such as CAN, LIN, MOST or Flexray. Wolf et al [2] has pointed out the various security loopholes in these protocols. The ECUs in these protocols are connected through gateway ECUs. The gateway is the most safety critical ECU in the in-vehicle network as attacks such as drop, flood, modify, read, replay and spoof can be performed [26].

Communication with ECUs can be either internal or external. If ECUs communicate within the in-vehicle network then its internal communication whereas if communication is done

outside of in-vehicle networks its external communication. Koscher et al [1][27] has explained and performed experiments showing inadequate security in both internal and external communication.

8 Countermeasures

The possible attacks and vulnerabilities of ECUs were explained and investigated in the previous sections. The huge number of scanning tools available and easily accessibility to hardware makes the threat to ECUs very real. In the future, with the introduction of FOTA, remote attacks on ECU are a possibility. In the following section, we try to discuss possible countermeasures to the vulnerabilities discovered from the previous section. When designing or developing new system, the vulnerabilities found from penetration testing should be patched.

8.1 Security Module

The Security module provides necessary security methods such as encryption, decryption, generation and verification of signatures, hashing, and secure storage of cryptographic keys. The Security module is implemented in two ways: Hardware implementation or Software implementation. A hardware implementation of the security module is more effective as it provides higher level of security compared to software implementation, since software implementation of security can be broken easily [29]. Wolf et al [28] has mentioned the following requirements to be fulfilled by a security module.

- **Unclonable:** A security module must be unclonable. The identity of the vehicle must be binded to the security module so that it cannot be faked, manipulated or cloned. It should be impossible to install the security module in another car to change its identity.
- **Secure Key storage:** A security module must be able to store keys in a protected way. The secret keys stored in the module must not be read or modified by an attacker.
- **Secure computations:** The security module must be able to securely perform cryptographic operations to prevent leakage of cryptographic secrets into unprotected areas.
- **Alarm channel:** The security module must be able to give notice in the case of a security breach.

A security module can be based on any of the three approaches: customized security controller, Trusted Platform Module (TPM), or Field-programmable gate array (FPGA). The properties of these approaches such as flexibility, cost, security level and standardization are given in the table below.

	Trusted Platform module (TPM)	Customized security controller	Field-programmable gate array (FPGA)
Standardized	Yes	No	No
Flexibility	Very limited	Yes, until release	Yes, even after release
Cost	Medium	Low (high volumes)	High
Security level	High	Adaptable	Medium high

Table 3: Properties of security module approaches

8.2 Software Protection

The software in the ECU must be secured from attacks by malicious users. To provide effective software protection, Wolf et al [28] has given the following requirements must be provided by the developers.

I. Only original software must be accepted by the vehicle. No malicious or unauthorized software must be loaded that changes the defined behavior of the vehicle.
II. Only authenticated parties must be able to alter data.
III. The compromise of a single control unit should not affect the entire system.
IV. The required computational performance on the side of the control unit must be minimal.

I and II are necessary whereas III and IV are desirable properties. Though it is not possible to prevent reading the code of the ECU through various tools, we could employ obfuscation techniques which will make the code look complex and hence reverse engineering the software will be difficult [30]. To detect software piracy techniques such as digital watermarking and fingerprinting can be used.

8.2.1 Secure Software Download

It is possible to use the flashing mechanism in the ECU to inject malicious software. Hence it is vital to have a secure software download mechanism. The requirements discussed in the previous section, can be addressed by using digital signatures. The common practice of employing signature is to sign the entire hash of software. There are various methods of hashing like SHA-1, RSA exponentiation etc.

	Code size	Time
SHA-1 hashing	1132	680 kB/s
RSA exponentiation w/small public key	2368	11 ms
RSA verification (16 kB code)	3500	34 ms

Table 4: RSA signature verification on ARM7TDMI at 40 MHZ

From the table above it shows that RSA is an appropriate fit as it allows very fast signature verification. The following steps are given for a secure software download and given in the figure below [28].

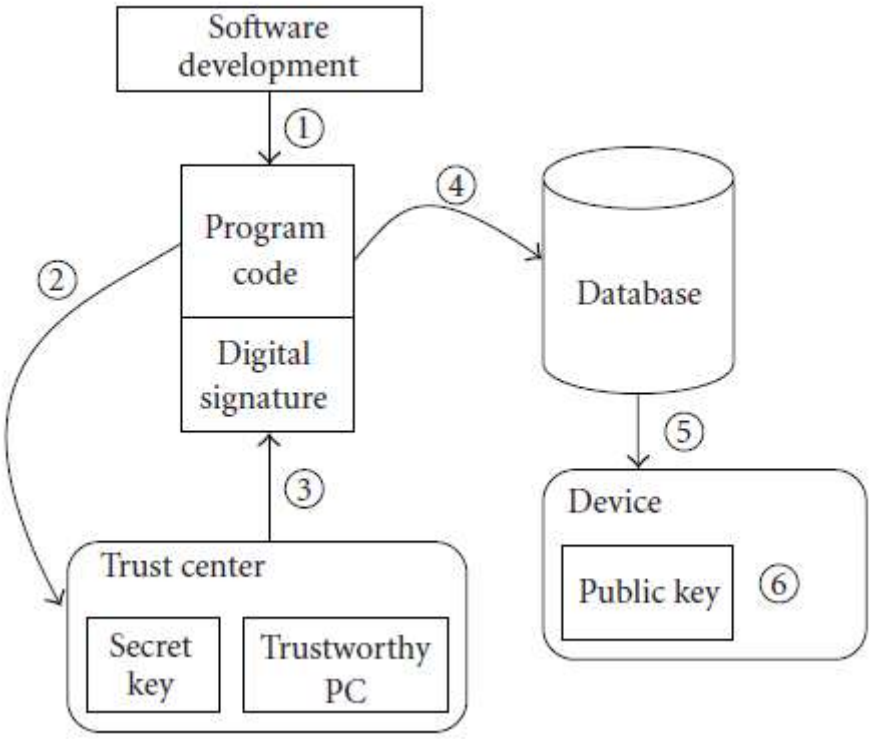

Figure 14: Secure Software download

I. After software development, the program object code is passed to a trust center.
II. The trust center signs the object code using its secret key to form a signature and it is passed back to attach itself to the program code.
III. The code and signature is now stored in a database that has versions of different ECUs.
IV. The signature is verified by the public key stored in the device to which the code is to be uploaded.

8.3 Hardware Protection

The ECU hardware can be protected against hardware manipulations by mechanical countermeasures deploying special component constructions. These constructions can be proprietary that could fit only into cars of a single manufacturer or constructions that require proprietary tools and equipment. This approach is uncomfortable and provides only minimal hardware support.

Weimerskirch et al [31] discusses about detection of faked or bogus vehicle components by using a small computing tag attached to each component to logically bind security and safety related parts to a specially protected security module. These component identification schemes rely on the tamper-evidence of the computing tags that are tightly integrated into critical components that can communicate with each other and on the tamper resistance of the security module.

Wolf et al [28] suggest integrating all critical hardware cores completely into a single protected chip inorder to protect the hardware effectively. To compromise such a system on chip (SoC), we need to use physical and chemical methods which are highly sophisticated and expensive.

8.4 Forensic Protection

It should be possible to investigate the crime scene involving automobiles. The communications between various ECUs with timestamp must be stored in a secure log which can be used to analyze the failure of the system. Under no circumstances, the contents of the log be changed or tampered with. Nilsson et al [32] uses the model proposed by Carrier and Spafford for forensic investigation of vehicular networks as the model connects physical and digital crime scene investigations. With both physical and digital investigations, it may provide complementary evidence when investigating vehicle incidents. The five phases of forensic investigations are readiness phase, deployment phase, physical crime scene investigation phase, digital crime scene investigation phase and presentation phase.

8.5 Securing communication

The communication between ECUs must be secured both internal threats from the in-vehicle network and external threats from outside the in-vehicle network. Many approaches for securing the communication protocols are suggested. Wolf et al [2] has given ways through which authentication and encryption of messages be performed so that only the intended recipient can have access to the message. He also says about the usage of gateway firewalls for a complete vehicular bus communication security. Larson et al [26] discusses about an approach towards specification based attack detection, which involves having a detector for every ECU and sends appropriate signal if a particular attack occurs.

9. Conclusions

The rapid increase in number of electronic components and dependence on software has made automobiles users vulnerable not only to safety threats but also to security threats against information stored in the car. We have discussed how to perform penetration testing on ECUs, by describing in-vehicle networks, embedded systems, possible attacks and countermeasures to prevent those attacks. We have also analyzed the classifications of ECUs, which ECU holds important information and attack on which ECUs might lead to catastrophic failure of the system.

The ECUs are easily subjected to attacks from malicious users as authentication and encryption of important information is either not done securely or not even addressed. It is also very easy to communicate with an ECU through various software tools via onboard diagnostic cable and also an attacker can have physical access to it. This gives rise to various attacks like brute force, signature manipulation, reverse engineering etc.

With rapid strides in developing firmware update over the air (FOTA), it is possible to perform software attacks on ECUs in a car from a remote location. Hence it is vital to look at the possible vulnerabilities and insecurities of the system before sophisticated protocols and connections such as FOTA are introduced in the automobile environment. The given countermeasures must be carefully studied and implemented keeping in mind the memory constraints and processing capability of the ECU.

Hence we hope our thesis will be useful for anyone who wants to perform penetration testing of ECUs. Finally the attacks on ECUs must be performed to complete penetration testing of ECUs.

10. Future work

This thesis has opened up lot of interesting opportunities to carry out research on cyber security in automobiles. Possible directions in which this thesis can be extended are:

- Get access to ECU and software to communicate with it. Try to perform the attack on the discussed vulnerabilities.
- Perform simulated remote attacks on ECUs assuming the availability of FOTA and car to car communication.
- Collaboration with industry can yield better results from the tests to be performed.
- Investigate the possibility to implement the countermeasures discussed here.

References:

1. karl koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno, Stephen Checkoway, Hovav Shacham, and Stefan Savage"experimental security analysis of a modern automobile", Department of Computer science and engineering, University of Washington, and University of California.
2. Marko Wolf, Andre Weimerskirch, and Christof Paar. "Security in Automotive Bus Systems". escrypt GmbH,Bochum, Germany {mwolf,aweimerskirch,cpaar}@escrypt.com
3. Electro to Auto forum http://e2af.com/trend/071210.shtml October 2012
4. Dennis K.Nilsson, Phu H. Phung and Ulf E. Larson, "Vehicle ECU classification based on safety-security characteristics" Department of Computer science, Chalmers University of Technology, Gothenburg, Sweden.
5. Dennis K. Nilsson and Ulf E. Larson, "Secure Firmware Update over the air in Intelligent Vehicles", Department of Computer science and engineering, Chalmers University of Technology, Gothenburg, Sweden.
6. "Cars get internet connection", EETimes automotive Europe, http://www.automotive-eetimes.com/en/cars_get_internet_connection_study_says?cmp_id=7&news_id=212001156 novemeber 7th 2008 || 212001156.
7. White paper on automotive telematics from Hughes, http://hsc.com/Portals/0/Uploads/Articles/WhitePaper-Telematics633833564780651074.pdf October 2012
8. http://www.cdxetextbook.com/fuelSys/efi/ecu/ECU.html September 2012
9. Volha Bordyk, "Analysis of software and hardware configuration management for pre-production vehicles" master thesis in software engineering and technology.
10. Manfre broy, Ingolf H. Kruger, Alexander Pretschner, and Christian Salzmann. "Engineering Automotive Software"
11. Muhammad Sabir Idrees, Henrik Schweppe, Yves Roudier, Marko Wolf, Dirk Scheuermann, and Olaf Henniger, "Secure Automotive On-Board Protocols: A Case of Over the Air Firmware Updates", EURECOM, Escrypt GmbH, Fraunhofer SIT.
12. J. Turley. The two percent solution. *Embedded systems design*, December 2002.
13. Armin Wasicek, "Embedded security at a glance: Security concepts for embedded systems" Vienna university of Technology, Institut for Technische Informatik, Technical report 182-1/ 2007/70 April 2013
14. J.D. Howard and T.A. Longstaff, "A common language for computer security incidents" no. Sandia Report: SAND98-8667, 1998.
15. Tobias Hoppe, Jana Dittmann, "Sniffing/Replay attacks on CAN Busses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy" april 2012
16. Srivaths Ravi, Anand Raghunathan and Srimat Chakradhar, " Tamper Resistance Mechanisms for Secure Embedded Systems", NEC laboratories America, Princeton, NJ 08540
17. Sri Parameshwaran, Timan Wolf, "Embedded systems security – overview", March 2013
18. Dr Sergei Skorobogatov, "Physical attacks on Tamper resistance: progress and lessons", http://www.cl.cam.ac.uk/~sps32/ARO_2011.pdf University of Cambridge, London April 2013
19. P.C. Kocher, "Timing attacks on implementation of Diffie-Hellmen, RSS, DSS and other systems", Advances in cryptology – CRYPTO'96, springer-verlog lecture notes in computer science vol 1109, pp 104-113, 1996.

20. W.Van Eck, “Electromagnetic radiation from video display units: an eavesdropping risk?” Computers and security vol 4, no 4, pp 269-286. 1985.
21. Eran Rippel, “Embedded security challenges in automotive designs”, Discretix Technologies, Ltd. November 2012
22. Eilam Eladad, “Reversing: Secrets of reverse engineering”, Jhon willey & sons. Pg 3. ISBN 978-0-7645-7481-8
23. Automotive Diagnostic command set User Manual, National instruments, January 2007.
24. Zili Shao, Qingfeng Zhuge, Yi He, Edwin H.-m. Sha, “Defending Embedded Systems against Buffer overflow via Hardware/Software”, Department of Computer science, University of Texas at Dallas.
25. “ECU Hacking 101 blog”, http://ecuhacking101.blogspot.se/2010/12/some-disassembly-required.html April 2013
26. Ulf E. Larson, Dennis K. Nilsson and Erland Jonsson, “An Approach to Specification-based Attack Detection for In-Vehicle Networks”, Department of Computer Science and Engineering, Chalmers University of Technology.
27. Stephen Checkoway, Damon McCoy, Brian Kanto, Danny Anderson, Hovav Shacham, and Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno, “Comprehensive Experimental Analyses of Automotive Attack Surfaces”, University of California, San Diego and University of Washington.
28. Marko Wolf, Andre Weimerskirch and Thomas Wollinger, “State of the Art: Embedding Security in Vehicles” April 2007.
29. P. Van Oorschot, “Revisiting software protection”, in proceedings of the 6th international conference on information security (ISC ‘03), vol. 2851 of lecture notes in computer science, pp. 1-13, Bristol, UK, October 2003.
30. C. Linn and S. Debray, “Obfuscation of executable code to improve resistance to static disassembly”, in proceedings of the 10th ACM conference on computer and communication security, pp.290-299, Washington DC, USA, October 2003.
31. A. Weimerskirch, C. Paar, and M. Wolf, “Cryptographic component identification: enabler for secure vehicles”, in proceedings of 62nd IEEE vehicular Technology Conference(VTC ‘05), pp. 1227-1231, Dallas, Texas, USA, September 2005.
32. Dennis K Nilsson, Ulf E Larson, “Conducting Forensic Investigations of Cyber attacks on Automobile In-vehicle Networks”, Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg Sweden.
33. SANS Institute Infosec Reading Room: “Penetration Testing – Is it right for you?” SANS institute 2002
34. SANS Institute Infosec Reading Room: “Conducting a Penetration Test on an Organization”, SANS Institute 2002.
35. Karen Scarfone, Mugugiah Souppaya, Amanda Cody and Angela Orebaugh, “Technical guide to Information Security Testing and Assessment”, National Institute of standards and technology, September 2008.
36. SANS Institute Infosec Reading Room: “Guidelines for Developing Penetration Rules of Behavior”, SANS Institute 2001.