Literary Review

Michiel Willems

July 21, 2018

1 OBD Overview

1.1 OBD purpose

The goal of OBD2 is to Help Technicians Properly Diagnose and Repair Complex Problems [15]. To do this the protocol introduces:

- Standardisation: information is communicated in s standardized format to allow for 1 tool to be used on many vehicles.
- Certification: Every vehicle manufacturer required to submit certification application for review and approval, which includes a detailed description of how the OBD2 protocol was implemented.
- Helps lowering emissions by identifying emission controls in need of repair.

1.2 Fault Detection

Every vehicle has a couple of specific values that don't change, and act as a kind of signature for each specific vehicle. These include[15]:

- The vehicle identification number (VIN).
- Readiness profile: A given make/model/year should have a specific readiness profile.
- ECU address allocation should be unique.

If a change in value of this data is detected, this could indicate possible fraud. Other so called 'fingerprinting' techniques are the CAL ID and CVN (which are also make/model/year specific), the communications protocol used (e.g. LIN, CAN, MOST and FLexRay) and the parameter ID's which are supported inside the vehicle.

2 Security, Vulnerability and Protection of Vehicular On-board Diagnostics

2.1 Threats

Automobile hacking has become one of the major concerns for software security today. The attacker can take control of your car and its various features and functions without having physical access to the car. Everyday a new method of hacking is developed, which challenges the existing security solutions [6]. Currently four out of ten car thefts in major cities involve some form of car hacking. Immobilisers used in 100 different models from the likes of Volvo, VW, Audi and Fiat especially models that come with a starter button instead of a key were found vulnerable to hacking by thieves with access to a computer. The researchers were banned from publishing the report for two years by car manufacturers due to its sensitive nature. [16].

2.2 Points of Entry

2.2.1 Physical

- Ripping apart the dashboard.
- **Door Locks and Key Fobs:** Since 1995 EU legislation demands that all new cars come standard with an electronic immobiliser. This device only allows the vehicle to start when it is provided the right credentials - but thieves can wirelessly steal all of the information from a car key in seconds. [6]

2.2.2 Wireless

- **Bluetooth:**
- **Cd Player:**
- **Tire Pressure Monitoring System:**
- **Unauthorized applications:**

2.3 OBDII port

In order to provide comprehensive, easy-to-use self diagnostic and reporting functionality for in vehicle ECUs, a standardized On-Board Diagnostic (OBD) interface was developed in the 1990s and today OBD is deployed

worldwide and legally mandatory in the US and Europe. [6] On one hand, OBD enables digital access to public data, for instance to emission control and error codes and on the other hand, OBD enables access also to hidden manufacturer-specific ECU settings, for instance to theft protection or engine control [6].

2.4 OBD port threats

It has been found out that a set of messages and signals that could be sent on cars CAN bus (via OBD-II) to control key components (e.g. lights, locks, brakes, and engine) as well as injecting code into key ECUs to insert persistent capabilities and to bridge across multiple CAN buses [13]. Due to commonly available tools and information, automotive manufacturers face an increasing amount of ECU manipulations that might affect vehicle safety, legal applications (e.g., exhaust gas treatment) or undermine aftermarket business models. Furthermore, critical data are stored in the ECUs such as crash data, data for insurances, or warranty indicators. Such data is also very attractive for malicious manipulations. For example, data such as vehicle speed, seat belt status, brake pedal position etc. are typically recorded in the seconds before a crash. A driver who has been involved in an accident could be motivated to change the recorded data to indicate that the brakes were applied when they really were not.

2.5 Security Solutions

2.5.1 Seed Key Algorithm

The seed key algorithm (like the name implies) applies a secret key value to calculate the response key from the seed. Only the person with the correct secret key can gain access to the diagnostic service of a specific ECU. The problem with this algorithm is the fact the same ECU in different cars will have the same secret key. Another problem is that the secret key material is often stored in unprotected memory. If enough keys are made public this would undermine the security of the entire algorithm.

2.5.2 Two-Way authentication method

This algorithm is an extension of the seed-key algorithm. In addition to requiring possession of the secret key, a message will be sent to the client of the vehicle whenever access is requested (Cellular or via Internet). Without acknowledgement from the user the seed is dropped and access is denied.

The above process adds a layer of safety as a result of keeping the client informed at every stage.

2.5.3 Timer Method

The timer method is again an extension of the Two-Way authentication method. It Exploits the time brute force methods and other algorithms take to crack a 16-bit long seed key, as well as giving more autonomy the car owner by giving the global seed directly to the client, who in turn must enter the key to complete the authentication process. As soon as a security access request is sent by the tester, the timer will be started. As soon as the timer runs out, a message or a notification alert is sent to the client informing about the malicious activity, as well as aborting the authentication process.

3 Security Crash Test Practical Security Evaluations of Automotive Onboard IT Components

3.1 Threats

- manipulating the steering wheel or brakes.
- distracting the driver by triggering odd vehicle behaviour.
- odometer manipulation.
- changing crash data, data for insurances, or warranty indicators.
- access to private data such as vehicle location, credentials to online services, or mobile payment data
- theft of vehicles or valuable vehicle components such as airbags or head units, e.g. by abusing diagnostics commands to reprogram a new key.

3.2 Automotive Security Assurance Levels

In 2015 Stephanie Bayer, Thomas Enderle, Dennis Kengo Oka and Marko Wolf introduced the "Automotive security evaluation assurance levels (ASEAL) which define up to four discrete security testing levels that determine (i) the size of security evaluation scope, that means which security analyses and tests have to be executed for a certain ASEAL and (ii) how deeply and thoroughly these security analyses and tests have to be executed. The goal for ASEAL is to make security evaluations comparable and, in consequence,

to make it possible to assign standardized levels of minimum security assurance to each automotive onboard IT component. [13] A comprehensive summary of the proposed levels is shown in figure 1.

Figure 1: ASEAL levels.

Evaluation Category	Type of Automotive Security Evaluation		Scope and Depth of Automotive Security Evaluation for each ASEAL			
			ASEAL A	ASEAL B	ASEAL C	ASEAL D
Theoretical Security Analyses	TRA: Security Threats and Risks Analysis		TRA1	TRA2	TRA3	TRA4
	SDA: Security Design Analysis		SDA1	SDA2	SDA3	SDA4
	DEV: Security Development Analysis		--	--	DEV1	DEV2
	DEP: Security Deployment & Processes Analysis		--	DEP1	DEP2	DEP3
Practical Security Testing	FST: Functional Security Testing		FST1	FST2	FST3	FST4
	VUL: Vulnerability Scanning		--	VUL1	VUL2	VUL3
	SYF: Systematic Fuzzing			SYF1	SYF2	SYF3
	Penetration Testing	LPA: Logical Penetration Attacks	--	--	LPA1	LPA2
		IPA: Invasive Penetration Attacks	--	--	IPA1	IPA2
		ORA: Organizational Pen. Attacks	--	--	ORA1	ORA2

4 On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle

The OBD-II port was created to provide consumers with choice and control over their purchase. At the same time, this freedom must be balanced with thoughtful conversations on how to limit adversaries access to vehicle internals

4.1 Recommendations for security of IT components in cars.

- **Separate CAN communication from the network stack:** The OBDII data that is sent should separated from the network stack and allow an application to send a request only from a list of pre- chosen OBD-II commands.
- **Sign and encrypt firmware updates:** Firmware updates should always be cryptographically signed and encrypted to prevent firmware modification by an attacker.
- **Be secure by default:** It is recommended that every third-party device that ends up in the vehicle (e.g. ECU's developed in another

factory) is shipped in the most secure configuration. The reseller can then modify the configuration as needed.

- **Obey the principle of least privilege:** Additional features introduce additional risk. This means that every device should only implement just enough connectivity to perform it's desired function.

4.2 The connected car infrastructure

The connected car can be described as a vehicle with one or more external wireless commu- nication possibilities, which connects the vehicle to an external network. These external networks can have many different configurations (e.g. local network vs connection to the internet) and purposes (e.g. remote diagnostics, software download, media streaming, etc). Figure 2 shows the infrastructure of the common connected car. vehicular communication can be divided into 2 groups: Vehicle-to-vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, collectively known as Vehicle-to-X (V2X) communication. The network can also be implemented in a couple of ways: Wifi-technology, Cellular communications and others (e.g. WAVE- or ETSI ITS-protocol).

4.3 ISO 13400 Diagnostics over IP (DoIP)

DoIP uses IP, TCP, and UDP, both in unicast and broadcast. The diagnostics messages themselves are not specified by DoIP as it's purpose is to transmit these messages over IP-based networks. Figure 3 shows the protocol stack of the DoIP protocol. [18]

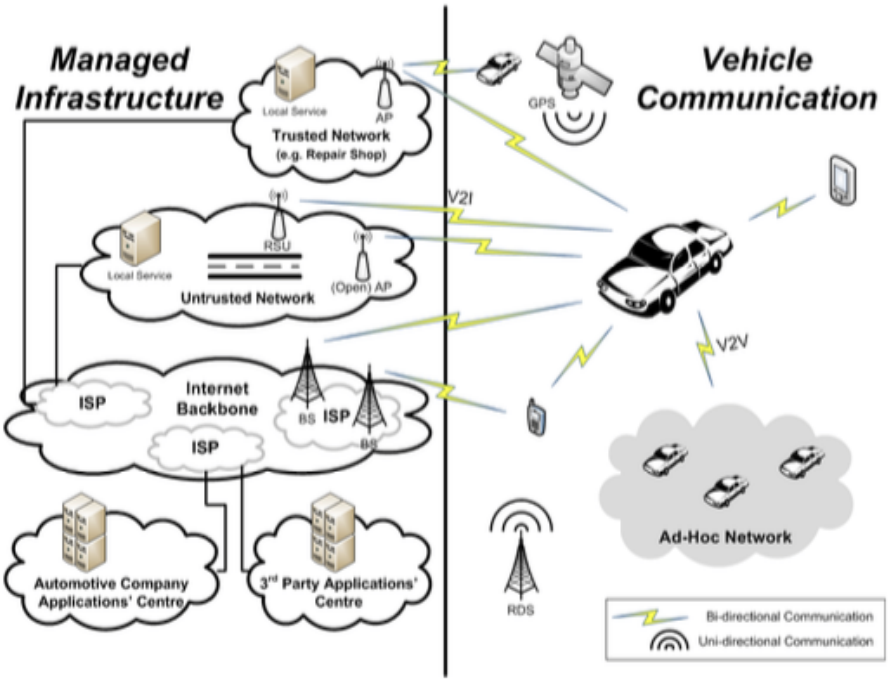
4.4 ISO 14229 Unified Diagnostic Services (UDS)

ISO 14229 defines an application layer protocol for diagnostics of vehicles. 26 services are provided and with these services, it is possible to read and write data to ECUs, reset ECUs, and upload firmware. [17]

5 An In-Depth Analysis of the Security of the Connected Repair Shop

In 2015 Pierre Kleberger, Tomas Olovsson, and Erland Jonsson of the Department of Computer Science and Engineering of Chalmers University of

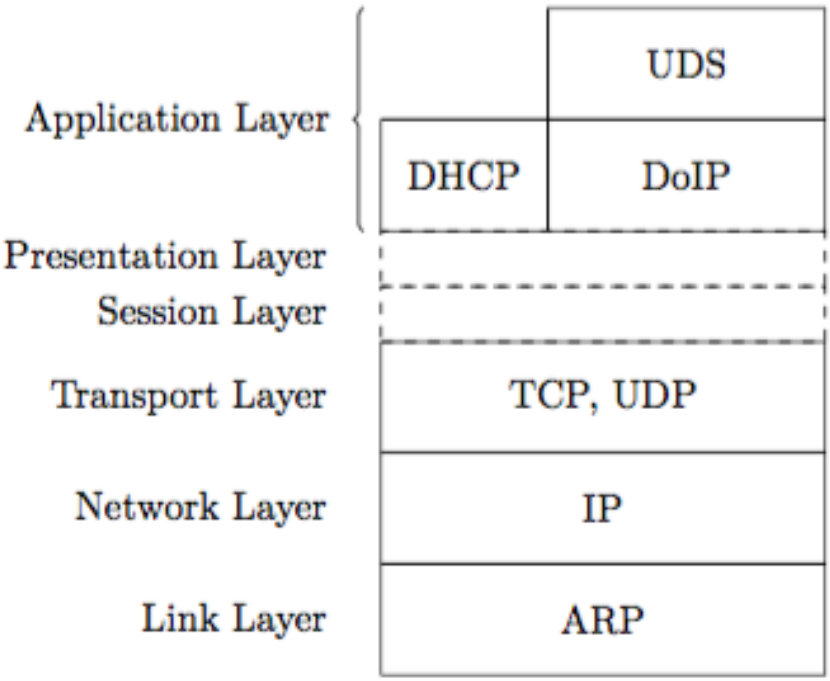
Figure 2: Connected car infrastructure



Technology in Gothenburg, Sweden, presented a security analysis of connected repair shops. They used the model given in figure 4. After a detailed analysis of possible threats (using the threat, vulnerability, and risk analysis (TVRA) method defined by ETSI.) they came up with the following security countermeasures [3].

- Data should be stored and transmitted encrypted.
- Strong authentication is needed. Private/public keys, with or without certificates, and Kerberos-like authentication mechanisms can be used.
- Traffic separation should occur in the repair shop network. This helps in dealing with misbehaving software and helps contain software bugs. Traffic seperation can be logical (e.g. VLAN technology) or cryptographical (e.g. using cryptographic keys and encryption to separate traffic). Cryptographic encryption is preferred since it is more flexible.
- Digital signatures and message authentication codes (MACs) can be

Figure 3: DoIP protocol stack



used to verify the source of and the integrity in the communication.

- Timestamps can be used to prevent replay attacks.

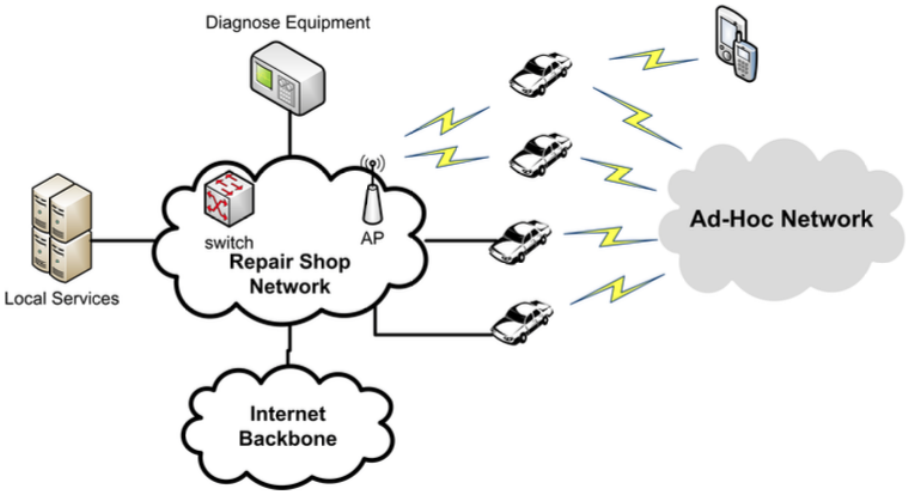
All of these countermeasures are clearly applicable to other V2I and V2X models (e.g. automatic gas payment, V2V data sharing, etc).

6 Comprehensive Experimental Analyses of Automotive Attack Surfaces

6.1 access

- Entertainment: Disc, USB and iPod.
- Remote Keyless Entry.
- RFID car keys.

Figure 4: Connected Repair Shop Infrastructure



- Emerging short-range channels: A number of manufacturers have started incorporating 802.11 WiFi in their automobiles.
- Dedicated Short-Range Communications (DSRC) standard: In such systems, forward vehicles communicate digitally to trailing cars to inform them of sudden changes in acceleration to support improved collision avoidance and harm reduction.
- broadcast channels: The modern automobile includes a plethora of broad- cast receivers for long-range signals: Global Positioning System (GPS), Satellite Radio, Digital Radio, and the Radio Data System (RDS) and Traffic Message Channel (TMC) signals transmitted as digital subcarriers on existing FM-bands.
- Addressable Channels: Perhaps the most important part of the long-range wireless attack surface is that exposed by the remote telematics systems (e.g. cellular).

Figure 5 shows a summary of the attack surface capabilities of each of the entry points discussed earlier.

Figure 5: Attack surface capabilities

Vulnerability Class	Channel	Implemented Capability	Visible to User	Scale	Full Control	Cost	Section
Direct physical	OBD-II port	Plug attack hardware directly into car OBD-II port	Yes	Small	Yes	Low	Prior work [14]
Indirect physical	CD	CD-based firmware update	Yes	Small	Yes	Medium	Section 4.2
	CD	Special song (WMA)	Yes*	Medium	Yes	Medium-High	Section 4.2
	PassThru	WiFi or wired control connection to advertised PassThru devices	No	Small	Yes	Low	Section 4.2
	PassThru	WiFi or wired shell injection	No	Viral	Yes	Low	Section 4.2
Short-range wireless	Bluetooth	Buffer overflow with paired Android phone and Trojan app	No	Large	Yes	Low-Medium	Section 4.3
	Bluetooth	Sniff MAC address, brute force PIN, buffer overflow	No	Small	Yes	Low-Medium	Section 4.3
Long-range wireless	Cellular	Call car, authentication exploit, buffer overflow (using laptop)	No	Large	Yes	Medium-High	Section 4.4
	Cellular	Call car, authentication exploit, buffer overflow (using iPod with exploit audio file, earphones, and a telephone)	No	Large	Yes	Medium-High	Section 4.4

7 CANtoolz

[19]

8 Securing the Automobile: a Comprehensive Approach

In particular, the CAN bus, one of the primary buses used in modern automobiles, is not particularly safe or se- cure. There is no protection against babbling idiots on the bus, in which a node denies service to other nodes, either because of a fault or in a malicious denial of service.

9 On securing the connected car

9.1 Automotive business constraints

Business constraints can be in tension with developing secure systems, particularly those that have a great impact on cost, in [1] a couple of constraints are highlighted:

- Part cost.
- Size and weight (especially when available space is limited, e.g. sports cars).

- Legacy integration: Since car components are designed to have long lifetimes, a lot of new designs might have to communicate with legacy components.
- Timing requirements: Additional security measures might violate timing constraints.
- Standardisation: A single manufacturer cannot always afford to break away from industry standards in order to improve security.

They also propose a couple of changes to the ECU supplier model in order to make automobiles more secure:

- Independent evidence: Car Manufacturers should require independent evidence that security objectives are met by a suppliers software.
- Collaboration: The security of the manufactured car will improve if the third party suppliers are allowed to embed their own engineers within the manufacturing team.
- Open software: This might seem counter-intuitive but open software is subject to public analysis, which in turn should reveal any vulnerabilities, thereby improving it's security.
- Liability: Finally, there is precedent for holding automotive manufacturers liable for harm caused by their software [20].

They also propose some ways of improving the software of ECU's:

- Software Development: A couple of improvements in the development of software are proposed:
  - Coding Standards (e.g. MISRA C standard [21]).
  - Static Analysis (static analyzers traverse the source code of a program and alert the user to possible flaws).
  - Memory Safe Programming (e.g. use safe programming languages like RUST).
- Testing:
  - Fuzz testing (random testing input).
  - property based testing (testing data generated in accordance with property specification, e.g. QuickCheck [22]).

- Formal Verification:
  - Program Verifiers (e.g. Verifast [23]).
  - Glue code generation.
- Runtime Assurance:
  - System Specialization (least privilege).
  - Data Integrity.
  - Runtime Verification (formal verification + testing).
  - Software Fault Containment Regions.
- Sanitizing inputs:
  - Radio systems.
  - Media systems.
  - Telematics systems.
  - Wireless key systems.
  - Vehicle to Vehicle Communication.

10 Scalable CAN security for CAN, CANopen and other protocols

10.1 CANcrypt

CANcrypt uses a CAN feature that allows two devices to exchange a hidden bit that is not visible to other CAN devices. This allows generating pairing keys that only the two devices know. CANcrypt uses a dynamic 64-bit key to cover the longest possible secure data block, 8 bytes. From this key, a pseudo one-time pad is generated and changes frequently. CANcrypt does not protect against DOS attacks.[24]

11 VulCAN: E icient Component Authentication and So ware Isolation for Automotive Control Networks

In December of 2017 three researchers (Jo Van Bulck, Jan Tobias Mhlberg and Frank Piessens) at the KULeuven university in belgium, presented VulCAN. VulCAN is a generic design for vehicular message authentication, software component attestation and isolation. Their solution is distinguished

from previous work (e.g. CANcrypt[24], VetiCAN[28] and Leia[27]) by relying on trusted hardware and a minimal Trusted Computing Base (TCB). This TCB relies heavily on the SANCUS[25] security architecture (also developed at KuLeuven).

11.1 SANCUS

Many embedded platforms lack the standard security features (e.g. privilege levels and virtual memory) present in high-end processors. SANCUS was developed to tackle this problem. the goal of SANCUS is to provide network embedded systems with remote attestation and strong integrity and authenticity guarantees with a minimal hardware TCB. Figure 6 shows the SANCUS system model. The infrastructure provider (IP) provides a number of nodes (N_i) on which software providers (SP_i) can deploy software modules $SM_{j,k}$. This model is applicable to many ICT systems today (in the case of VulCAN this system of course a CAN network). Any system that supports installation of software modules by several software providers must implement measures to make sure that the different modules can not interfere with each other in undesired ways. To enforce this requirement SANCUS provides 4 security properties:

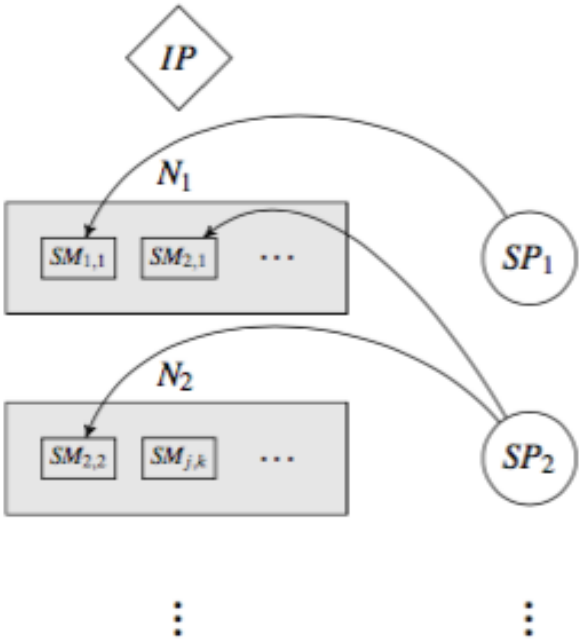
- **Software module isolation:** Every software module has a runtime state that is not modifiable by other modules. The only way other modules are able to interact with it is by calling one of its designated entry points.
- **Remote attestation:** A software provider can verify with sufficient assurance that a specific software module is loaded on a specific node.
- **Secure communication:** A software provider can communicate with a software module while authenticity, integrity and freshness guarantees are maintained.
- **Secure linking:** A software module on a node can link to and call another module on the same node without interference by other software on the same node.

For more information on SANCUS confer [25].

11.2 VulCAN

VulCAN was designed to compartmentalise every ECU into a small group of trustworthy authenticated software components (It's easy to see that this

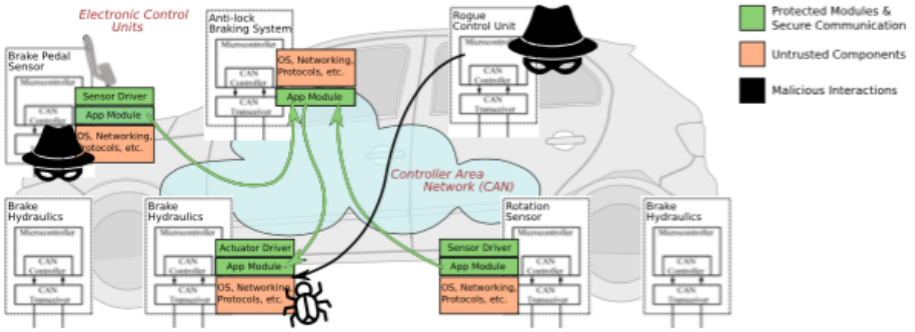
Figure 6: SANCUS system model



is where the SANCUS design was implemented). Figure 7 shows An example CAN network scenario where an authenticated trusted path is set up between a software component that senses a braking pedal and wheel rotations, over an Anti-lock Braking System (ABS) component, all the way to a brake hydraulic actuator component. To fulfil the requirements of such a system VulCAN complies to the following requirements:

- **Message Authentication:** The system uses MAC's (message authentication codes) to prove the message was indeed sent by a trusted sender component.
- **Lightweight Cryptography:** Lightweight cryptography is a must because of strict timing constraints (e.g. it is obvious that the example used in figure 7 should perform as fast as possible) and because of the computational limitations of the embedded devices.
- **Replay Attack Resistance:** the authentication scheme is immune

Figure 7: VulCAN example [26]



to replay attacks (a malicious agent injecting a previously sent message in the hope of it being falsely authenticated). This is ensured by using short term session keys, and a monotonically increasing counter or nonce as a source of freshness in the MAC computation.

- **Backwards Compatibility:** Legacy unmodified applications without authenticated communication should continue to function. To this end the system broadcasts the authenticated message in plain text, and afterwards constructs and transmits authentication data on a different CAN identifier, effectively decoupling the authentication metadata from the original message.

While the above requirements are mostly met by two other recent CAN authentication protocols [27] [28], a number of system-level guarantees were added for the system to be applicable in an in-vehicle CAN network:

- **Real Time Compliance:** the system adheres to stringent real-time deadlines.
- **Component Isolation:** Effectively what SANCUS guarantees.
- **Component Attestation:** Also Guaranteed by SANCUS.
- **Dynamic Key Update:** It is possible for broken ECUs to be replaced at a distrusted automobile repair shop (again this is where SANCUS comes in).

To learn more about VulCAN confer [26].

12 Security and privacy in vehicular communications: Challenges and opportunities

12.1 Safety Requirements

The ISO 26262 [29] standard defines the functional safety requirements that must apply during the complete lifecycle of every automotive electronic/electrical system that is safety-related. Among others (cf [29]) it introduces the Safety Integrity Level (ASIL), a risk-based approach to determine potential hazard in a vehicle operating scenario. There are four levels: ASIL-A, ASIL-B, ASIL-C and ASIL-D. The risk increases from ASIL-A to ASIL-D, where the ASIL-A level indicates the lowest risk while the ASIL-D level is the highest one.[30]

12.2 AUTOSAR

AUTOSAR is a software architecture for the automotive sector. This architecture aims to assist with the development of vehicular software.

12.3 Different Communication protocols used inside vehicles

Figure 8: Different data communication buses in modern cars [30]

Table 1
Summary of characteristics and use of different data communication buses in modern cars.

Bus name	Transfer rate	Used for	Bus features	Hard/soft real-time	Access control
CAN	1 Mbit/s	OBD2, power-train, chassis and body electronics	Multi-master serial bus and Low-cost protocol	Soft real-time	CSMA/CA
CAN-FD	8 Mbit/s		Similar to CAN with longer payload		
LIN	20 Kbit/s	Body electronics (including mirrors, power seats and accessories)	Broadcast serial bus, master-slave communication and cheaper than CAN	Hard real-time	Polling
FlexRay	10 Mbit/s	High-performance power-train and safety (drive by wire, active suspension and adaptive cruise control)	Multi-master serial bus, 1-master; up to 16 slaves, expensive protocol and 2 channels	Hard real-time	TDMA
MOST	150 Mbit/s	Rear-view, cameras, infotainment and multi-master bus	Ring topology, supports 64 devices and very high cost	Hard real-time	CSMA/CA TDM
Ethernet (BroadR -Reach)	100 Mbit/s	Cameras, infotainment and on-board diagnosis	Cheaper than MOST, more expensive than CAN and lightweight wiring and CSMA/CD	Soft real-time	TDMA TDD

References

[1] Lee Pike, Jamey Sharp, Mark Tullsen, Patrick C. Hickey and James Bielman, *Securing the automobile: A comprehensive approach*, 2015.

[2] Dan Klinedinst, Christopher King, *On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle*, 2016.

[3] Pierre Kleberger, *On Securing the Connected Car*, 2015.

[4] Brian Russell, Aaron Guzman, Paul Lanois, Drew Van Duren, *Observations and Recommendations on Connected Vehicle Security*, Cloud Security Alliance Internet of Things Working group, 2017.

[5] Charlie Miller, Chris Valasek, *A Survey of Remote Automotive Attack Surfaces*, 2015.

[6] Aastha Yadav, Gaurav Bose, Radhika Bhange, Karan Kapoor, N.Ch.S.N Iyengar, Ronnie D. Caytiles, *Security, Vulnerability and Protection of Vehicular On-board Diagnostics*, 2016.

[7] https://en.wikipedia.org/wiki/On-board_diagnostics.

[8] https://en.wikipedia.org/wiki/OBD-II_PIDs.

[9] https://en.wikipedia.org/wiki/CAN_bus.

[10] Charlie Miller, Chris Valasek, *CAN Message Injection*, OG Dynamite Edition , 2016.

[11] Charlie Miller, Chris Valasek, /textitAdventures in Automotive Networks and Control Units.

[12] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage University of California, San Diego Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno, University of Washington *Comprehensive Experimental Analyses of Automotive Attack Surfaces.*

[13] Stephanie Bayer, Thomas Enderle, Dennis Kengo Oka , Marko Wolf. *Security Crash Test Practical Security Evaluations of Automotive Onboard IT Components*, 2015.

[14] European Union Agency for Network and Information Security (ENISA), *Cyber Security and Resilience of smart cars*, 2016.

[15] Allen Lyons, California Air Resources Board , *On-Board Diagnostics (OBD) Program Overview*, 2015 .

[16] Marien Saarinen, http://www.autoexpress.co.uk/car-news/consumer-news/92304/car-hacking-study-shows-over-100-models-at-risk, *Car hacking: study shows over 100 models at risk*, 2015.

[17] *ISO 14229-1:2013: Road vehicles Unified diagnostic services (UDS) Part 1: Specification and requirements.* ISO, 2013.

[18] *ISO 13400-1:2011: Road vehicles Diagnostic communication over Internet Protocol (DoIP) Part 1: General information and use case definition.* ISO, 2011.

[19] Alexei Sintsov, *(pen)testing vehicles with CANToolz*, 2016.

[20] Philip Koopman. *A case study of toyota unintended acceleration and software safety.* Public seminar, September 2014.

[21] *Guidelines for the Use of the C Language in Critical Systems.* MISRA, 2004.

[22] Koen Claessen and John Hughes. *QuickCheck: A lightweight tool for random testing of haskell programs.* In Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming, ICFP 00, pages 268279. ACM, 2000.

[23] Bart Jacobs, Frank Piessens, *The VeriFast Program Verifier*, Department of Computer Science, Katholieke Universiteit Leuven, Belgium.

[24] Olaf Pfeiffer, Christian Keydel, *Scalable CAN security for CAN, CANopen and other protocols*, 2017.

[25] Job Noorman, Pieter Agten ,Wilfried Daniels ,Raoul Strackx Anthony Van Herrewege, Christophe Huygens, Bart Preneel, Frank Piessens ,Ingrid Verbauwhede, *Sancus: Low-cost trustworthy extensible networked devices with a zero-software Trusted Computing Base*, iMinds-DistriNet and iMinds-COSIC, KU Leuven.

[26] Jan Tobias Muhlberg, Frank Piessens, Jo Van Bulck, *VulCAN: E icient Component Authentication and So ware Isolation for Automotive Control Networks*, Imec-Distrinet KuLeuven, 2017.

[27] Andreea-Ina Radu, Flavio D. Garcia, *LeiA: A Lightweight Authentication Protocol forCAN*, School of Computer Science,University of Birmingham, UK.

[28] Stefan N urnberger, Christian Rossow, *vatiCAN, Vetted, Authenticated CAN Bus*, CISPA, Saarland University, Germany.

[29] ISO-26262-1, Road vehicles Functional Safety, ISO 26262-1, International Or- ganization for Standardization, Geneva, Switzerland, 2011.

[30] Cesar Bernardini, Muhammad Rizwan Asghar, Bruno Crispo, *Security and privacy in vehicular communications: Challenges and opportunities*, 2017.