Privacy and Security in Connected Vehicles Ecosystems

Marius POPA, Cristian TOMA, Cătălin BOJA, Alin ZAMFIROIU
Department of Economic Informatics and Cybernetics
Bucharest University of Economic Studies, Romania
marius.popa@ie.ase.ro, cristian.toma@ie.ase.ro, catalin.boja@ie.ase.ro,
alin.zamfiroiu@csie.ase.ro

Modern vehicles could not be figured out without Internet connections in order to provide customers a wide range of services in the vehicle: infotainment platforms, third-party support, onboard and online monitor and maintenance, business analytics for car fleets. Exposure of the vehicles to the Internet turns them into targets for viruses, worms, Trojans, DoS and lot of other threats for connected vehicle security. Beside the classic threats of the Internet exposure, other new threats are introduced by the Internet of Things (IoT) new technologies that are poor regulated or undefined yet from the security point of view. Also, the large variety of the IoT technologies not being standardized yet contribute to security issues in this area of the automotive industry. This paper provides an overview of the connected vehicle environment, considering the main components of such kind of system and the main security challenges to be considered for building reliable secure online systems for connected vehicles.

Keywords: *IoT, Connected Vehicle Security, Embedded Systems.*

1 Internet of Things Overview

Internet of Things (IoT) refers a tremendous variety of physical devices, vehicles, construction facilities and embedded systems having network connectivity features enabled to collect and exchange data between them directly or via a centralized information system. The Global Standards Initiative on Internet of Things (IoT-GSI) states: *"The Internet of Things (IoT) has been defined in Recommendation ITU-T Y.2060 (06/2012) as a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies."* [7]. ITU is the United Nations specialized agency for information and communication technologies – ICTs.

Many technology research companies estimates billions of devices to be connected into IoT infrastructures next year. Those large amounts of data generated by IoT represent a challenge for information systems to store, aggregate, index and process those data in order to provide more effectiveness during IoT usage processes.

Possible IoT applications include [8]:

- *Media*: application related to big data approach by using the IoT data for a better targeting of the consumers. Instead of the general traditional approach, the advertisers could target the marketing campaigns by using the consumers' mobile phones.
- *Environmental monitoring*: application using sensors to monitor quality of water, air and soil and their influence factors (e.g. waste management). Sensors could be deployed on large geographic areas such that the wildlife and its habitat are monitored. Also, implementations could be made triggering earthquakes and tsunami alarms for more effective emergency services.
- *Infrastructure management*: this kind of applications aims urban and rural infrastructures for monitoring the structural conditions and events that appear during operation. Collected data could be used to schedule maintenance activities, to ensure effective emergency services and lower costs for infrastructure operation.
- *Manufacturing*: large applications in industrial environments even there are proprietary monitoring systems. Thanks to network connectivity, IoT infrastructures could be deployed for network control,

DOI: 10.12948/issn14531305/21.4.2017.03

management of manufacturing equipment, asset and situation management, manufacturing process control ensuring integration of the existing systems with the external ones and offering o better flexibility of the manufacturing assets with the market demands.

- *Energy management*: application integrates specialized sensors with energy consuming devices to send useful data to energy supplier or to control them remotely. Data are used to balance the energy system and to ensure an efficient, effective, reliable and sustainable production and distribution of the electricity.
- *Medical and healthcare*: IoT devices used in application for health monitoring and emergency notification systems. Also, management related or medical care processes could be automated (e.g. smart beds, treatment administration, seniors' assistance).
- *Building and home automation*: such applications have as goals improved comfort, efficient operation, low utility costs by using an automatic centralized control building system, WiFi connected devices, remote monitoring. The remote access to such systems could be done wall-mounted terminals, mobile software, web interface or via cloud services.
- *Transportation*: applications have to consider the transportation system components: vehicle, infrastructure and the driver. Implementations include traffic control, smart parking, electronic billing systems, fleet management, vehicle control, safety and road assistance.
- *Metropolitan scale deployments*: they include smart city related applications for life quality improvement. These applications aim city service management and involve the habitants to contribute to high quality of life by using personal devices as data sources for efficient and effective services systems. The particularity of this kind of applications is the large urban scalability where they are implemented.
- *Consumer application*: they are those applications created for consumer use. They are related to those objects and services used by the consumers daily.

Enabling IoT is possible at large scale by using embedded systems with network capabilities. An embedded system represents a computer system included into a larger or mechanical system. They have specific tasks and particular features like low power consumption, small size, and low per-unit cost and could have no input/output peripherals.

Embedded systems could be based on microcontrollers or microprocessors. Microcontrollers are central processing units (CPUs) with integrated memory or peripheral interfaces.

Characteristics of the embedded systems could consider the following topics, [9]:

- *User interface:* very simple and single task embedded systems have no user interface at all or a simple one by using buttons, Light-Emitting Diodes (LEDs) or Liquid-Crystal Displays (LCDs) with a simple menu system. The sophisticated embedded systems use advance screen hardware with touch sensing features and complex menu systems. Also, embedded systems could be remotely addressed by serial or network interfaces, avoiding the need of a display.
- *Processor:* embedded systems use two types of processors: microprocessors (memory and peripherals are separated) and microcontrollers (built-in memory and on-chip peripherals). Embedded software is a proprietary one and it is not so easy to be installed by the end user. That is the result of CPU architecture variety used by embedded specialized hardware producers.
- *Peripherals:* they are used to address embedded system from their outside environment. Such communication interfaces could be, [9]:
    - *Serial Communication Interfaces (SCI):* Electronic Industries Association (EIA) standards RS-232, RS-422, RS-485, etc.
    - *Synchronous Serial Communication Interface*: Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI), Serial Servo Controller (SSC)

DOI: 10.12948/issn14531305/21.4.2017.03

and Enhanced Synchronous Serial Interface (ESSI).
- *Universal Serial Bus (USB).*
- *Multi Media Cards*: SD cards, Compact Flash, etc.
- *Networks*: Ethernet, LonWorks (local operating network), etc.
- *Fieldbuses*: Controller Area Network (CAN-Bus), Local Interconnect Network (LIN-Bus), Process Field Bus (PROFIBUS), etc.
- *Timers*: Phase-Locked Loop (PLL), Capture/Compare, etc
- *Discrete IO*: General Purpose Input/Output (GPIO)
- *Analog to Digital/Digital to Analog (ADC/DAC).*
- *Debugging*: Joint Test Action Group (JTAG), In-system programming (ISP), In-Circuit Serial Programming (ICSP), Background debug mode (BDM Port), BITP, and DB9 ports.

Embedded software architectures in common use have several types as [9] states:

- *Simple control loop:* embedded software has one single loop calls subroutines for hardware or software management.
- *Interrupt-controlled system:* embedded software implements handlers for events. Handlers call interrupts for controlling the embedded system.
- *Cooperative multitasking:* embedded software has an appropriate environment to add new tasks and run them by the multitasking system yielding the control periodically.
- *Preemptive multitasking or multi-threading:* embedded software uses a timer for switching between tasks or threads. In order to allow the focus on device functionality instead of operating system services, a Real-Time Operating System (RTOS) is used for large systems.
- *Microkernels and exokernels:* these represent the step-up of RTOS. Microkernel is the minimum-needed software to implement the operating system. Exokernel is a kind of operating system kernel developed by the Massachusetts Institute of Technology (MIT) Parallel and Distributed Operating Systems in order to minimize the abstractions of hardware resources to applications.
- *Monolithic kernels:* they are large kernels having sophisticated capabilities adapted to embedded environment. They provide an environment similar to a desktop operating system one. Hence, development productivity is increase, and more hardware is required. Also, they are less predictable and reliable due to complexity of the embedded software.
- *Additional software components:* there are embedded systems allowing layered software components for networking, storage and multimedia capabilities.

In the context of IoT, a connected vehicle is a vehicle having networking capabilities inside and well as outside. Usually, the Internet access is made via wireless network in order to optimize the vehicle operation and maintenance as well as the convenience and comfort of passengers.

A connected vehicle comes to consumer with concerns about data privacy and possibility of hacking.

2 Interfaces for Connected Vehicles

In vehicle field, the interface represents the interaction point between two connected hardware pieces having as goal data translation from a representation format into a different one. Data captures and interpretation are made by automatic tools based on software that implement standard or proprietary interface definitions for automotive subsystems.

Automotive subsystems communication is made by using a vehicle bus through messages are exchanged without a host computer. There is a communication infrastructure called vehicle bus, and automotive subsystems could be microcontrollers or hardware devices. Such kind of vehicle bus is Controller Area Network (CAN bus) [10].

The modern vehicles have dozens of embedded systems as electronic control units (ECUs), controlling one or more electrical systems. These systems communicate via CAN bus, provide sensor data and perform control of systems (actuators). The benefits of the communication infrastructure used by automotive embedded systems lead to safety, economy and software development for high-tech vehicles increasing the driver comfort and better maintenance.

Automotive ECUs are nodes within CAN bus architecture. They could be simple as I/O devices or complex as embedded systems with CAN, USB or Ethernet interfaces and complex embedded software.

ISO defines two specifications for CAN bus architecture:

- *High speed CAN:* ISO 11898-2 specification, figure 1, usually used in automotive and industrial applications.

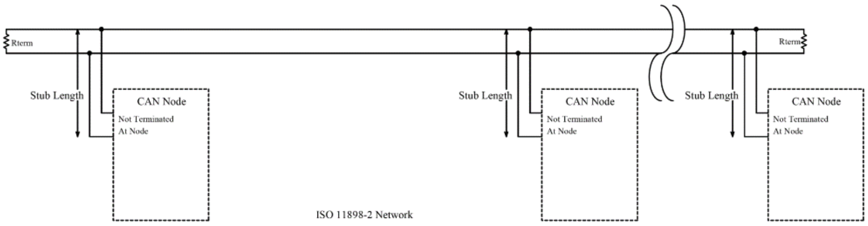

Fig. 1. High speed CAN [10]

- *Low speed or fault tolerant CAN:* ISO 11898-3 specification, figure 2, usually used where groups of nodes have to be connected together.

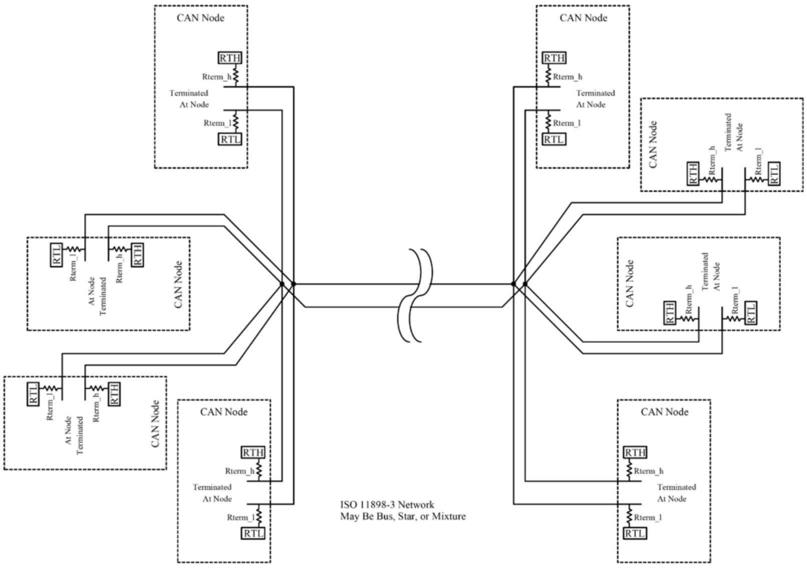

Fig. 2. Low speed or fault tolerant CAN [10]

At physical layer, there are de facto emerged standards for mechanical connection, having custom connectors and no complete physical layer specification by CAN bus specification. Figure 3 depicts D-subtype connector having 9 pins with the below pin-out [10]:

- *Pin 2:* CAN-Low (CAN−).
- *Pin 3:* GND (Ground).
- *Pin 7:* CAN-High (CAN+).
- *Pin 9:* CAN V+ (Power).

Fig. 3. D-subtype male connector [10]

There are many CAN bus – based protocols completing the two ISO specifications presented above. Each vehicle producer has adopted its own convenient standard.

For CAN bus environments, the application software need drivers or libraries in order to have access to CAN controller hardware and implement data exchange between CAN nodes. Such drivers are Can4linux (Linux), SocketCAN (Linux), CPC Development Kits for Linux and Windows, Kvaser etc.

CAN bus-related specifications are used in OBD-II vehicle diagnostics standard. OBD term means On-Board Diagnostics providing access to the status of the vehicle automotive subsystems. OBD-II interface is built for a signaling protocol as follows [11]:

- *SAE J1850 VPW:* Variable Pulse Width protocol at 10.4/41.6 Kbps, single wire.
- *SAE J1850 PWM:* Pulse Width Modulation protocol at 41.6 Kbps, two wire differential.
- *ISO 9141-2:* Asynchronous serial communication protocol at 10.4 Kbps. It is similar to RS-232.
- *ISO 14230 KWP2000:* Asynchronous serial communication protocol up to 10.4 Kbps, covering the application layer of OSI model in computer networking.
- *ISO15765-4/SAE J2480:* Protocol CAN-related at 250kbps or 500kbps.

The previous five protocols are widely used by automotive industry and biggest automotive manufacturers. The OBD-II interface requires a physical connector in order to get vehicle data for diagnostics. The physical connectors are emphasized by the figures 4 and 5.

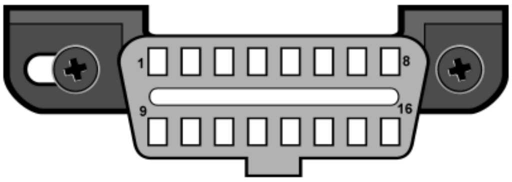

Fig. 4. Type A J1962 Vehicle Connector [11]

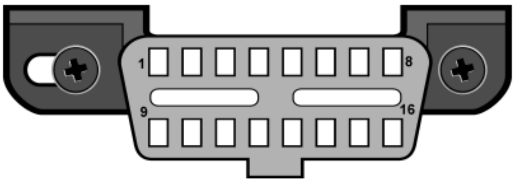

Fig. 5. Type B J1962 Vehicle Connector [11]

Depending on what pin has material metallic contact inside of it, the OBD-II vehicle connector is designed for a particular communication protocol. For each of the five communication protocols, the activated pins within the physical vehicle connector are [11]:

- *SAE J1850 VPW:* Pins 2, 4, 5, and 16.
- *SAE J1850 PWM:* Pins 2, 4, 5, 10, and 16.
- *ISO 9141-2:* Pins 4, 5, 7, 15, and 16.
- *ISO 14230 KWP2000:* Pins 4, 5, 7, 15, and 16 (as ISO 9141-2).
- *ISO15765-4/SAE J2480:* Pins 4, 5, 6, 14 and 16.

The activated pins by the metallic contact have the following meaning [11]:

- *Pin 2:* J1850 Bus+.
- *Pin 4:* Framework/Chassis Ground.
- *Pin 5:* Signal ground.
- *Pin 6:* CAN High (J-2284).

DOI: 10.12948/issn14531305/21.4.2017.03

- *Pin 7:* ISO 9141-2 K Line.
- *Pin 10:* J1850 Bus.
- *Pin 14:* CAN Lower (J-2284).
- *Pin 15:* ISO 9141-2 L Line.
- *Pin 16:* Power of Battery.

OBD-II interface is mainly used to control the vehicle engine functions and diagnose the engine problems. It has become more complex and sophisticated and currently it is able to monitor the almost engine functions, vehicle accessories and control network. Its goal is to provide useful data for government agencies to monitor vehicle emissions as background for environmental protection strategy development. By the monitor program, automotive industry and services have to make commitments to produce, keep, monitor, fix and update the vehicle parameters to meet the environment protection requirements.

Getting vehicles parameters requires an OBD-II physical connector (J1962 connector) installed at vehicle's board, a connection cable and software scan tool. More sophisticated vehicle scanning architectures based on OBD-II interface could be designed and deployed by using contactless OBD-II dongles (instead of cable connector) and data read from the vehicle are sent via a mobile gateway to a centralized storage and processing infrastructure. Such kind of contactless OBD-II dongles use wireless computer networks (figure 6) or Bluetooth standard (figure 7).

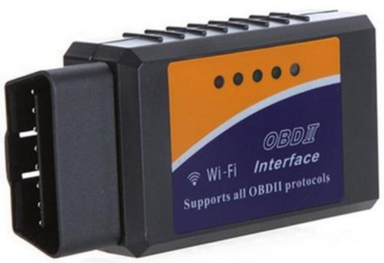

Fig. 6. WiFI OBD-II Dongle [13]

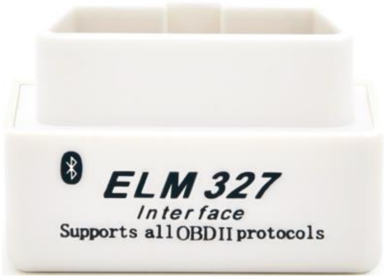

Fig. 7. Bluetooth OBD-II Dongle [13]

OBD-II interface provides the means to develop various applications having different complexity from the consumer level to highly sophisticated ones as [14]:

- *Hand-held scan tools*: There are simple tools to read/reset the codes and professional tools for advanced diagnosis, setting specific ECU parameters, access to other control units, real-time monitor.
- *Mobile device-based tools and analysis*: Mobile applications running on mobile phones or tablets providing access to OBD-II data via USB connectors, Bluetooth or WiFi connections. Sending data to Internet is possible.
- *PC-based scan tools and analysis*: There is software installed on computer which translates serial data from OBD-II connector into a visual representation and data storage.
- *Data loggers*: Recording OBD-II data over a time while the vehicle is in normal operation. Data are used later for analysis for diagnostics or tuning, by insurance companies in risk evaluation and price calibrations, and driver's behavior by the fleet operators.
- *Emission testing*: Verifications of vehicle's emissions for particular codes flagging troubles in compliance with emission standards.
- *Supplementary vehicle instrumentation*: Vehicle operation-related data could be provided by the manufacturer to the driver during normal vehicle operation.
- *Vehicle telematics*: Used for fleet tracking, monitor fuel efficiency, prevent unsafe driving, as well as for remote diagnostics and by pay-as-you-drive insurance.

OBD-II-related software could be implemented considering the OBD-II data reading infrastructure implemented by an API. Such

DOI: 10.12948/issn14531305/21.4.2017.03

kind of API could be found at [15] providing needed Java classes and functions (obd-java-api) to create a connection to ELM327 device and read OBD-II dongle data by sending commands to the device.

For instance, in order to get current revolutions-per-minute (RPM) data, obd-java-api contains OBD-II command to be instantiated [15]:

```
private RPMCommand command = null;
// Other code lines
command = new RPMCommand();
```

Instantiation will place the RPM code in the command to be sent to the OBD-II vehicle dongle. In case of RPM, the OBD-II command code is **"010C"** and it is executed by a thread launched after establishing the connection to the OBD-II device (eventually pairing if the device is a Bluetooth one).

The OBD-II RPM command is placed in package `com.github.pires.obd.commands.engine. RPMCommand`.

The following call reads the OBD-II response to the command for RPM by getting the data stream provided by the device as response, checking the errors the response stream, saving decoded data into a file buffer and applying RPM-related calculations [15]:

```
command.readResult(inputStream);
```

Getting the current RPM data is made by the next call, returning an integer value [15]:

```
command.getRPM()
```

Such APIs offer the opportunity to the developers to create OBD-II interface software for less or more complex vehicle diagnostics architectures.

3 Addressing Security in a Connected Vehicle Environment

A connected vehicle means a vehicle that has several dozens of microcontrollers with computing power over a dozen of personal computers, processing couple of dozen of gigabytes an hour, running applications having features implemented by dozens of millions of programming code. The traditional automotive digital technology has focused on operation aspects by monitor and optimization the internal function of the vehicle. Currently, the automotive digital technology focuses on connection with the outside environment of the vehicle and enhancement of the in-car experience, including access to the Internet. Hence, improved operation and maintenance of the vehicle are supported and the passengers benefit by better convenience and comfort.

Penetration of information and communications technology (ICT) instruments into the vehicle will change the business model in the automotive industry by added value brought by the latest digital technology opening adoption of new valuable services within the vehicle, entering of new software and telecommunication companies into the automotive industry, larger pool of data and shared mobility.

Despite all potential advantages offered by integration of ICT tools within a vehicle, there are some concerns about security, mainly aiming data privacy and hacking a connected vehicle.

McKinsey consultant defines vehicle data as *"Data generated by a vehicle and its occupants either when the car is moving or stationary, by itself or in communication with other vehicles (V2V) or infrastructure (V2I), in the 'use' phase of its lifecycle."* [6]. Car data users or contributors are identified by McKinsey and classified as follows [6]:

- *Driver and passengers*: Data are generated by mobile devices for use cases as telecommunications, audio applications, traffic information, and portable navigation.
- *Original equipment manufacturers and dealers*: Data usage and/or generation use cases: remote onboard diagnostic, warranty management.
- *Other cars*: Use cases for data usage and/or generation: rolling map network, safety systems, automatic cruise control.
- *Service providers*: Data used and/or generated by contents streaming, direct mobile payments, Pay-As-You-Drive (PAYD) insurance, reservations/concierge services.

DOI: 10.12948/issn14531305/21.4.2017.03

- *Retailers*: Vehicle data: In-car offerings targeted advertising, proximity/customers flow data analytics.
- *Mobility providers*: Use cases for vehicle data: e-hailing services, vehicle sharing, public transport hubs.
- *Home and workplace*: Data aimed for: remote appliances, IT systems operation, automated customer login from the car, self-recharging/refueling.
- *Authorities*: Use cases for vehicle data: emergency and breakdown calls, law enforcement, vehicle-data-based road maintenance.
- *Infrastructure*: Data used and/or generated by automated road toll/taxation system, average speed monitoring systems, traffic flow management, monitoring systems
- *High-tech suppliers*: Data aimed for: maps, targeted advertising, contents streaming.

Customers are aware about exposure of their private data (current location, address book, browser history etc.) to third party by using mobile applications. Data has become a currency to access valuable services provided by mobile applications and customers grant access to their data in exchange for benefits such as free application usage or free content.

Using data as currency and data connectivity could generate significant benefits for the customers as [6]:

- *Safety*: Real-time emergency calls, early and on-scene accident information, real-time road hazard warning.
- *Convenience*: Reduced breakdown risk and vehicle downtime by using on-board diagnostic and spare parts management at dealer/workshop, concierge services.
- *Time*: Optimized routing/navigation and traffic management system, networked parking and connected navigation.
- *Cost*: Usage of PAYD insurance, automated payment infrastructure, in-car purchasing or in-car advertising.

There are two sources of enormous amount of data generated and used by connected vehicles:

- *ECU*: Each vehicle contains dozens of microcontrollers together with complex software that generate large amount of data.
- *Connection to a telecommunication network*: It is required by the features expected to be on-board by the customers. Those features connect the vehicle with the connected society where the customers live in.

The network connection ensures the conveniences expected by the customers in their cars, but they come with security risks and concerns about the hacking of the connected vehicle. Also, software complexity of the microcontrollers, having dozens of millions of programming code lines introduces vulnerabilities.

Such critical operational features and safety systems are exposed to the outside environment of the vehicle when a network connection is used. For instance, health diagnostics, automatic braking or steering system makes a connected vehicle to be vulnerable to attacks. Access the vehicle data via OBD-II interface provides access to 3rd party developers to run applications inside the vehicle. Also, integration of smartphone technologies could lead a connected vehicle to be hacked.

In [5], connected vehicle security threats considering the physical location are presented:

- *ECU and in-vehicle network* – They run operating software on-board that could come with certain security vulnerabilities or backdoors. A different threat could be reverse engineering applied on ECU software by disassembly and possibility to reflash the ECU by an attacker with malicious firmware. In-vehicle network could also be used for reverse engineering purposes in order to establish certain patters of packet exchanges between different ECUs. Also, security threats are introduced by in-vehicle network design and communication protocols because they did not require security for non-connected vehicles. Opening the in-vehicle network to the Internet brings security vulnerability because the interface points must translate and interpret data packets between the two infrastructures: the in-

DOI: 10.12948/issn14531305/21.4.2017.03

vehicle network and the Internet infrastructure.

- *Mobile device as access point to Internet services* – Variety of physical handheld devices and operating systems introduces security threats for connected vehicles because it requires a bigger effort for security management of those. The operating system life-cycle raised the security threats because the software producers do not offer security updates for old versions of operating system. Also, the mobile eco-system does not impose installing the security updates for operating system and applications by the user. Even the mobile system is up-to-date, the operating system or application could be hacked or used in reverse engineering processes. Mobile applications are created and distributed by application stores for vehicle self-diagnostic. They could hide malicious software by opening o communication channel between the in-vehicle network and the Internet remote attacker.
- *Communication channel* – A wireless channel is required since the connection is made for a vehicle. Wifi, Bluetooth and GSM communications are used as channels with Internet infrastructure. Each technology has its own security challenges and requirements. By securing the communication channel, security vulnerabilities as man-in-the-middle and spoofing are decreased.

By making a vehicle to be connected to the Internet offers the possibility of a remote attacker to address some vehicle's functions where the write operations are allowed. For instance, reprogramming of certain ECU functions makes the vehicle to respond to remote commands. Hence, malicious code could land into internal operational systems of the vehicle and strange and uncontrolled behavior by the driver could be assigned to the vehicle.

Security elements to be considered for connected vehicle security solution architectures are provided by [4], figure 8.

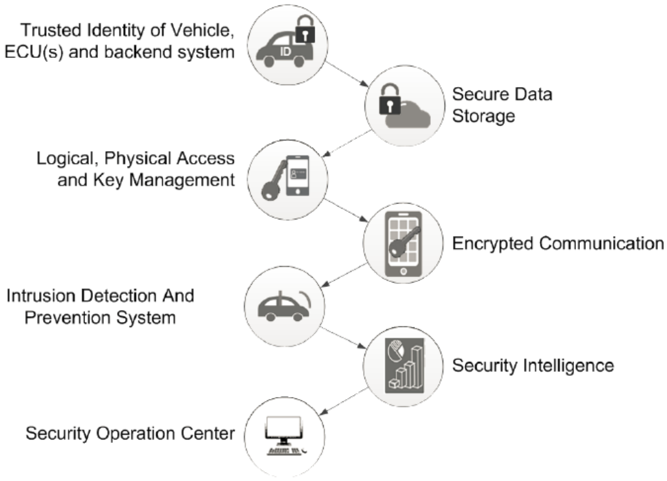
Trusted Identity of Vehicle, ECU(s) and backend system
Secure Data Storage
Logical, Physical Access and Key Management
Encrypted Communication
Intrusion Detection And Prevention System
Security Intelligence
Security Operation Center

Fig. 8. Connected Vehicle Security Elements Overview [4]

IBM Company and Giesecke & Devrient have proposed a security architecture requiring the following components [4]:

- *Trusted Identity of Vehicle, ECU(s) and backend systems* – It enables the right entity to have the proper access to the vehicle resources. A key management mechanism is necessary to identify the vehicle components, driver and backend systems, and what are the access rights to those. This security component must accomplish four basic functions: identification, access, service delivery and identity federation. Also, the identity system must be flexible enough for future requirements.
- *Secure Data Storage* – It is a mandatory security component in order to prevent unauthorized people to access the software code, vehicle and driver's personal data for accidental or intentional destruction, infection or corruption. As data storage security layers, data maintenance and disaster recovery procedures, and data encryption have to be considered.
- *Logical, Physical Access and Key Management* – Logical security is enforced by access control systems for data, software and processes. Physical security aims the access control to hardware or locations and could be regulated by logical security when the physical locations are managed by software systems for people authentication by specialized biometrics devices

DOI: 10.12948/issn14531305/21.4.2017.03

or identification cards. The management system must ensure a reliable cryptographic system within the connected vehicle ecosystem.

- *Encrypted Communication* – It is a secure component deals with data security exchanges between the car and the backend system. Because the main characteristic of a vehicle is its mobility, the encryption must deal with all challenges of wireless network security.
- *Intrusion Detection and Prevention System* – It can detect malicious activity or policy violations by monitoring the connected vehicle ecosystem. The system must be able to identify possible incidents, log and report them. Also, it must be updated according to the latest progresses and security policies defined and implemented within the connected vehicle systems.
- *Security Intelligence* – It is based on analytics systems by using large amount of real-time data collected from vehicles of different types and models, being in different geographic regions. Such anomaly patterns could be identified and re-acting measures could be deployed within the connected car infrastructure or outside the vehicle within transport or Internet infrastructure.
- *Security Operation Center* – It defines new rules and policies to be deployed to connected vehicle infrastructure as response to detected, analyzed and classified anomalies. That experience could be used for improved security mechanisms preventing attacks on connected vehicle infrastructure.

It is very important that a connected vehicle software bug once being discovered to be fixed very quick because the exploitation of that bug could have fatal consequences. This is the reason to involve the ICT companies in the automotive industry because they have the experience and resources to implement the quality assurance process in a shorter time than the automotive industry could do.

4 Conclusions

All objects exposed to the Internet must have properly protection by attackers. More than that, those objects must not allow transforming themselves into attack vectors to other Internet connected components.

Cybersecurity of the connected vehicles becomes a critical requirement of automotive and ICT industries because the current predictions estimate over 75% of the cars shipped in 2020 will be connected.

The above requirements have to be accomplished by connected vehicles as increasing and significant component of the IoT world.

Acknowledgment

Parts of this paper were communicated within International Conference “Current Economic Trends in Emerging and Developing Countries” TIMTED 2017, May 19 – 20, 2017, West University of Timişoara, Romania.

References

[1] C. Toma, M. Popa, C. Ciurea and C. Vinţe, “Secure Issues in IoT – Internet of Things Architecture for Sensors Data Processing”, in Proc. The 7th International Conference on Security for Information Technology and Communications, 2014, ASE Publishing House, Bucharest, pp. 163 – 172

[2] C. Toma and M. Popa, “IoT – Internet of Things Architecture for Context Aware Sensors Data Processing in Waste Management Solution”, in Proc. The 13th International Conference on Informatics in Economy (IE 2014), 2014, ASE Publishing House, Bucharest, pp. 26 – 32

[3] M. Popa and C. Cartas, “OBD2 IoT Device Proof of Concept for the Insurance Companies Connected Cars”, in Proc. The 15th International Conference on Informatics in Economy (IE 2016), 2016, ASE Publishing House, Bucharest, pp. 103 – 107

[4] M. Bongartz, H. Chen, V. Fricke, V. Gerstenberger, M. Koehn, A. Kohler and H. Scherzer, IT Security for the Connected Car, Intelligent mobility made by IBM and G&D, White Paper, Munich, March

2016
[5] T. Bécsi, S. Aradi, P. Gaspar, "Security issues and vulnerabilities in connected car systems, in Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, Budapest, pp. 477 – 482
[6] McKinsey&Company, Car data: paving the way to value-creating mobility – Perspectives on a new automotive business model, Advanced Industries, March 2016
[7] ITU, Internet of Things Global Standards Initiative, available online at http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx
[8] Wikipedia, Internet of things, available online at https://en.wikipedia.org/wiki/ Internet_of_things
[9] Wikipedia, Embedded System, available online at https://en.wikipedia.org/wiki/ Embedded_system
[10] Wikipedia, CAN bus, available online at https://en.wikipedia.org/wiki/CAN_bus
[11] Scan Tool Center, Which OBD2 protocol is supported by your vehicle?, available online at http://scantoolcenter.com/obd2-basic/which-obd-2-protocol-is-supported-by-your-vehicle/
[12] AutoTap, OBD-II, available online at http://www.obdii.com/
[13] Ali Express, https://www.aliexpress.com
[14] Wikipedia, On-board diagnostics, available online at https://en.wikipedia.org/wiki/On-board_diagnostics
[15] Paulo Pires, OBD-II Java API, available online at https://github.com/pires/obd-java-api/
[16] McKinsey Company, What's driving the connected car, available online at http://www.mckinsey.com/industries/automotive-and-assembly/our-insights/whats-driving-the-connected-car

Marius POPA has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2002. He holds a PhD diploma in Economic Cybernetics and Statistics. He joined the staff of Academy of Economic Studies, teaching assistant in 2002. Currently, he is Associate Professor in Economic Informatics field and branches within Department of Economic Informatics and Cybernetics at Faculty of Cybernetics, Statistics and Economic Informatics from Bucharest University of Economic Studies. He is the author and co-author of 9 books and over 140 articles in journals and proceedings of national and international conferences, symposiums, workshops in the fields of data quality, software quality, informatics security, collaborative information systems, IT project management, software engineering.

Cristian TOMA has graduated from the Faculty of Cybernetics, Statistics and Economic In-formatics, Economic Informatics specialization, within Academy of Economic Studies Bucha-rest in 2003. He has graduated from the BRIE master program in 2005 and PhD stage in 2008. In present, he is associate professor at Economic Informatics and Cybernetics Depart-ment and he is member in research structures such as ECO-INFOSOC. Since the beginning - 2005 - he is scientific secretary of IT&C Security Master Program from Bucharest University of Economic Studies and since 2006, he is in the editorial board of the SECITC – The Inter-national Conference on Security for Information Technology and Communications and JMEDS – Journal of Mobile, Embedded and Distributed Systems. His research areas are in: distributed and parallel computing, mobile applications, smart card programming, e-business and e-payment systems, network security, computer anti-viruses and viruses, secure web technologies and computational cryptography. He is teaching in Department of Economic Informatics and Cybernetics, and IT&C Security Master program. He has published 3 books and over 50 papers in indexed reviews and conferences proceedings.

DOI: 10.12948/issn14531305/21.4.2017.03

Cătălin BOJA is associate professor at the Economic Informatics and Cybernetics Department at the Academy of Economic Studies in Bucharest, Romania. In June 2004 he has graduated the Faculty of Cybernetics, Statistics and Economic Informatics at the Bucharest University of Economic Studies. He is a team member in various undergoing university research projects where he applied most of his project management knowledge. His work currently focuses on the analysis of mobile computing, information security and cryptography. He is currently holding a PhD degree on software optimization and on improvement of software applications performance.

Alin ZAMFIROIU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2009. In 2011 he has graduated the Economic Informatics Master program organized by the Bucharest University of Economic Studies and in 2014 he finished his PhD research in Economic Informatics at the Bucharest University of Economic Studies. Currently he works like a Senior Researcher at “National Institute for Research & Development in Informatics, Bucharest” and like an associate lector at Faculty of Cybernetics, Statistics and Economic Informatics at the Bucharest University of Economic Studies. He has published as author and co-author of journal articles and scientific presentations at conferences

DOI: 10.12948/issn14531305/21.4.2017.03