Global Journal of Researches in Engineering: G
Industrial Engineering
Volume 23 Issue 1 Version 1.0 Year 2023
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Layer-wise Security Challenges and a Secure Architectural Solution for Internet of Things at Physical, Network and Application Layers

By Sriranga Narasimha Gandhi Aryavalli & Hemantha Kumar
University of Mysore (UOM)

Abstract- In recent years, the Internet of Things has emerged as one of the most important technologies of the twenty-first century. We can now connect everyday objects to the internet via embedded devices such as kitchen appliances, cars, thermostats, and baby monitors, allowing for seamless communication between people, processes, and things. Because of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyper-connected world, digital systems can record, monitor, and adjust every interaction between connected things. The physical and digital worlds intersect and work together. By enabling connected cars, IoT is reinventing the automobile.

Keywords: internet of things (IoT), smart internet of things (SIoT); secure architecture design, secure design, secure engineering, unified secure architecture solution for IoT, connected cars.

GJRE-G Classification: DDC Code: 004.678 LCC Code: QA76.9.B45

Strictly as per the compliance and regulations of:

Layer-wise Security Challenges and a Secure Architectural Solution for Internet of Things at Physical, Network and Application Layers

Sriranga Narasimha Gandhi Aryavalli [α] & Hemantha Kumar [σ]

Abstract- In recent years, the Internet of Things has emerged as one of the most important technologies of the twenty-first century. We can now connect everyday objects to the internet via embedded devices such as kitchen appliances, cars, thermostats, and baby monitors, allowing for seamless communication between people, processes, and things. Because of low-cost computing, the cloud, big data, analytics, and mobile technologies, physical things can share and collect data with minimal human intervention. In this hyper-connected world, digital systems can record, monitor, and adjust every interaction between connected things. The physical and digital worlds intersect and work together. By enabling connected cars, IoT is reinventing the automobile. The global market for connected cars is expected to grow significantly in the coming years as connectivity innovations transform the automotive industry. However, as with any other device that connects to the internet, cyber criminals pose a threat to automotive security. Personal data leaks, threats to a vehicle's essential security and safety mechanisms, and, in extreme cases, full remote control of the vehicle can all result from security breaches. And, as the industry moves toward more self-driving vehicles, these risks are only going to grow due to increased reliance on applications, connectivity, and more complex and integrated electronic components. Failure to address these risks could have disastrous consequences for consumer trust, privacy, and brand reputation. Worse, customer safety is jeopardized.

In this paper, the author discusses Layer-wise Security Challenges, Attack Vectors, and Architectural Flaws in the Physical layer by taking an example of a device connected to Connected cars and proposes a secure architectural solution for the Internet of Things (IoT) that assists in delivery teams in securely designing/architecting resource-intensive smart Internet of Things (IoT)/Narrowband (NIoT) use cases earlier in the Life cycle by employing the Secure Design Shift Left approach.

Keywords: internet of things (IoT), smart internet of things (SIoT); secure architecture design, secure design, secure engineering, unified secure architecture solution for IoT, connected cars.

I. Introduction

The Internet of Things (IoT) is a network of physical objects-"things"-embedded with sensors, software, and other technologies for connecting and exchanging data with other devices and systems via the internet.

These gadgets range from common household items to sophisticated industrial tools. Experts predict that the number of connected IoT devices will increase to 10 billion by 2020 and 22 billion by 2025, from more than 7 billion today. One of the classic examples of IoT is Connected Cars.

Figure 1: Internet of Things – Connected Car

Car owners can use IoT to remotely operate their vehicles, such as preheating the car before the driver gets in it or summoning a car via phone. Cars will be able to book their service appointments, and IoT can enable device-to-device communication. The connected car enables automakers or dealers to flip the car ownership model on its head. Previously, manufacturers maintained a distance from individual buyers. The manufacturer's relationship with the vehicle essentially ended when it was delivered to the dealer. Automobile manufacturers and dealers can maintain a continuous relationship with their customers by using connected cars. Instead of selling cars, they can charge drivers usage fees and provide "transportation-as-a-service" with self-driving cars. IoT enables car manufacturers to continuously upgrade their vehicles with new software, which is a significant departure from the traditional model of car ownership in which vehicles depreciate in performance and value.

The advantages to consumers are numerous: connectivity provides drivers with everything from high-definition streaming media to Wi-Fi access, improved

Author α σ: Department of Studies in Computer Science, University of Mysore (UOM), Mysore, India. e-mail: gandhi.aryavalli@gmail.com

entertainment systems, and the ability to remotely control aspects of the vehicle, such as the locking/ unlocking and ignition mechanisms, via mobile phone applications. IoT use cases include smart transportation, smart farming, smart grids, smart lighting, and connected vehicles.

In Smart/Connected farming, the use of technology to monitor, analyze, manage, control, and ultimately improve key agricultural processes at all stages of the farming cycle: pre-production, production, and post-production is referred to as connected farming.

Figure 2: Internet of Things – Connected Tractor for Farming needs

It entails the communication of various devices, starting with sensors in the field and progressing to smartphones in farmers' hands. To practice connected farming, a farmer should have IoT ecosystems in the field, agricultural equipment, the cloud, and the office, allowing for a 360-degree view of the entire farming cycle. As data is collected using sensor-equipped devices, this concept is closely related to IoT in agriculture. Devices can tell how moist the land is, allowing farmers to decide whether to irrigate or check nitrogen levels in the soil, allowing workers to decide whether to add more fertilizer. Crop drone imagery can also be used to determine whether pesticides should be applied.

However, because connected vehicles are so adaptable, they also present certain security risks.

When one device is physically connected to another, hackers can use a variety of methods to hack the system. Regardless of who owns the data, all stakeholders - the car fleet, Original Equipment Manufacturer (OEM), or a third-party Telematics Service Provider - are accountable for the telematics data's security.

While cyber-security has emerged as a key focus area for the automotive industry, OEMs are also investigating the topic because they must assess their products' cyber-security vulnerabilities. OEMs have significant IT and OT operations that are vulnerable to cyber threats, and they frequently lack the necessary internal resources to address the problem. Whether an in-house or third-party TSP, channel partners, and the OEM are all equally responsible for securing telematics data and gradually advancing the automotive industry to the next level of technological advancement.

In Section II of this paper, the author discusses various threats, attack vectors, and security challenges that hackers may use to hack IoT devices at the physical layer [1], and in Section III, the author discusses a Secure Architectural solution to mitigate these threats/ design challenges so that the underlying physical systems are safe and secure enough.

II. Layer-wise Security Challenges of Internet of Things (IOT) – Physical and Network Layers

To discuss the Security Challenges [1] or Threat Vectors [1] of IoT Connected Devices at the Physical Layer, we must first examine the underlying Architecture. We will be looking closely when we see a connected vehicle and evaluating a Connected Vehicle.

There were ECUs in all vehicles, whether they were cars, buses, tractors, or four-wheelers. ECUs, or Electronic Control Units, are critical components of a vehicle. In a car, multiple ECUs operate various features and control numerous parameters.

Vehicles with multiple electronic control units are divided in terms of what tasks they perform. Engine Control Modules, Brake Control Modules, Transmission Control Modules, Telematic Control Modules, Suspension Control Modules, and other ECUs are examples.

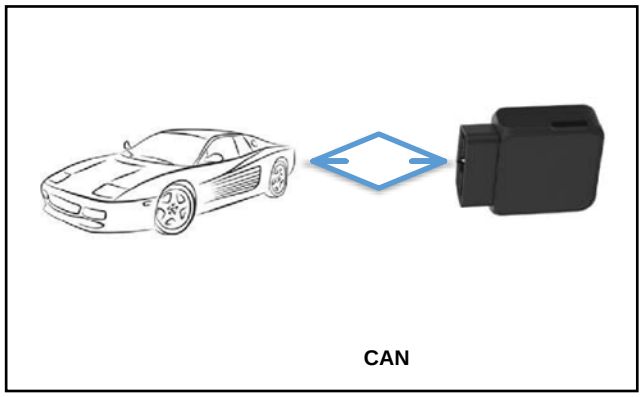

Figure 3: CAR ECU plugged with External Device via CAN

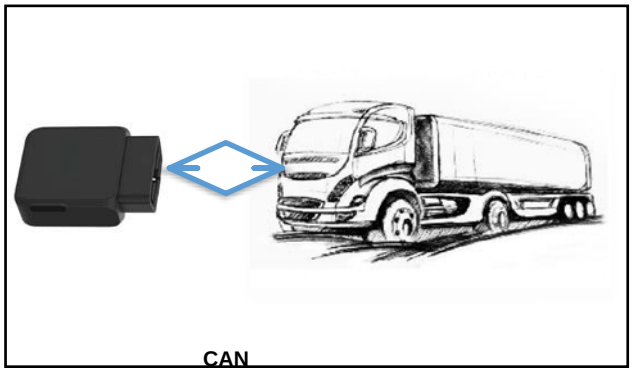

Figure 4: Heavy Vehicle ECU plugged with External Device via CAN

Simply put, an ECU is a device that controls all of the electronic features in a vehicle. This can include everything from fuel injection to maintaining a constant cabin temperature to controlling braking and suspension. Some vehicles have multiple ECUs that control different features, while others have a single ECU that controls everything.

Figure 5: ECU

Vehicles with multiple electronic control units are divided in terms of what tasks they perform. Engine Control Modules, Brake Control Modules, Transmission Control Modules, Telematic Control Modules, Suspension Control Modules, and other ECUs are examples.

An ECU is an electronic device with a memory filled with base numbers and parameters. With multiple IoT sensors around a vehicle feeding data to the ECU, it can efficiently manage and control the electronic systems by issuing orders to improve their output.

Consider how airbags [2] are deployed during an accident as an example of how an ECU controls something. The crash sensors are sensors located around the car that alert the ECU when a crash occurs. The ECU then measures the speed of the vehicle when it is involved in an accident and compares the data to determine whether or not the airbags should be deployed. If the data is sufficient, the ECU [3-4] will deploy the airbags. Take note that all of this happens in milliseconds.

The TCU collects telemetry data from the vehicle [5,6], such as position, speed, engine data, connectivity quality, and so on, by interfacing with various subsystems in the vehicle via data and control busses. It may also provide in-vehicle connectivity via Wifi and Bluetooth and, in certain markets, the eCall function. A TCU is made up of a satellite navigation (GNSS) unit that keeps track of the vehicle's latitude and longitude values; an external mobile communication interface (GSM, GPRS, Wi-Fi, WiMax, LTE, or 5G) that sends the tracked values to a centralized geographical information system (GIS) database server; an electronic processing unit; a microcontroller in some versions; a microprocessor or field programmable gate array (FPGA) that processes information and acts as an interface between the GPS; a mobile communication unit; and some memory for storing GPS values in mobile-free zones or intelligently storing information about the vehicle's sensor data.

TCU is linked to an external device for vehicle tracking. This device controls all of the vehicle's important features by reading and sending data. This external device reads data from the TCU's IOU (Input/output Unit) and correlates it using the SCU (Software/System Control Unit).

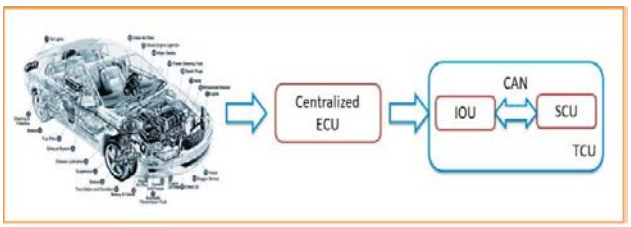

Figure 6: ECU'S Connectivity with TCU

The IOU receives all alerts/events from the ECU and correlates the data so that meaningful action can be taken. In the previous example, gathering all breakage-related information and correlating it to activate the airbags is the entire thing for which IOU will collect the dots. All of this data is sent to the SCU (Software Control Unit) via network connectivity.

Both IOU and SCU have traditionally been manufactured as separate components. Due to cybercrime, particularly via SCUs and IOU networks, all OEMs are producing SCUs and IOUs in a single unit known as TCU (Telemetry control unit), which is tightly coupled with both IOU and SCU.

Figure 7: SCU

Traditionally, OEMs connected the SCU unit to the IOU separately via CAN via Partners. As the threat vector grows, OEMs integrate this SCU unit inside the vehicle to prevent physical tampering. It is obvious that if the SCU is tampered with in any way, the entire vehicle can be controlled.

SCU is crucial, especially in connected vehicles. Having proper physical and information security controls that embed the SCU protects and secures the vehicle. The author discusses the various threat vectors that are possible for SCUs at the design level, which aids device hacking and thus controls the function of the entire connected vehicle in the following section.

III. Architectural Security Flaws, Threat Vectors in Respective Layers

In this section, the author discusses the potential threats for hacking the external device that is fitted/integrated with the OEM via CAN network to the ECU in light of the context discussed in the previous section.

a) Physical and Networking Layer – Security Challenges

Security Challenge/Threat Vector - 1: External Device Tampering

When it comes to SCU security, one of the threat vectors that comes to mind is device tampering. The hacker may tamper with the device and feed his/her special instructions to the device, causing the device to be controlled via his/her Command- and-Control. Device tampering is a very common security threat that the author must consider protecting a mechanism to safeguard architecturally/via Secure by Design.

2nd Security Challenge/Threat Vector: Replacing the Device/Stealing the Device

Replacing the SCU unit with a malicious SCU is another potential threat vector that could occur in real-world scenarios. A hacker may physically replace an algorithm-driven, C&C-controlled SCU with the device built into the vehicle and then take control of it. This is a serious security risk that can be mitigated by using secure design principles.

3rd Security Challenge/Threat Vector: Turning off the Device

In some cases, a hacker can disable the SCU device, preventing proper analytics from being fed to the server and, as a result, alarms from being triggered. This is most likely the threat that we see from traditional hackers. Switching off/unplugging the SCU may result in information not being shared with the IoT servers/ analytics platform, so we must consider this threat vector and find a meaningful solution to this challenge at the design level.

Resetting the Device is a security challenge/threat vector-4:

Resetting the device regularly after it has been hacked (or) sending malicious instructions to reset the device are common methods that the hacker can think of to escape the crime. OEMs and partners who build the SCU unit must consider how to solve this problem architecturally/at design levels.

5th Security Challenge/Threat Vector: Device Misconfiguration

There is a strong possibility that the device has been tampered with by maliciously configuring the device. Due to device misconfiguration, the OEM receives the incorrect feed being sent to the servers by overriding the actual instruction. One of the security threats that can be addressed architecturally is device misconfiguration.

Malicious command feeding to ECU is a security challenge/threat vector - 6.

Once the device has been tampered with by hackers, there is a strong possibility that the hacker will send malicious commands to the device to take control of the entire connected vehicle. Once this occurred, no one had control of the vehicle other than the hacker. Secure by Design must consider a kill chain to break this connection if the vehicle is already being controlled maliciously by the hacker.

Security Challenge/Threat Vector - 7: Response/ Communication Mechanism Tampering

When IOU receives a feed, IOU expects a response from the controlling IoT Server. There is an absolute risk that the connected car will be compromised if a man-in-the-middle who controls the SCU and IOU communication tampers with the packet response. Secure by design must address this challenge/threat vector on an architectural level.

Security Challenge/Threat Vector - 8: Network Choking

In some cases, hackers attempt to choke the network by sending unwanted requests/responses, resulting in the underlying network connection being choked, tainting the important information passing through the pipeline. The hacker gains control of the

vehicle for a few minutes and causes damage to the vehicle's safety. One of the threats that Secure by Design must address is network choking.

Security Challenge/Threat Vector - 9: No Authentication/ Authorisation of the device in the Network

Authentication and authorization are critical in securing the physical device that is connected to the ECU. This is an important mechanism for identifying the original device to the vehicle OEM. The critical aspect to consider when designing devices architecturally is secure authentication and authorization.

Security Challenge/Threat Vector - 10: Loosely coupled protocols and tunneling of the device

While the SCU is connected to the IOU via the network, the underlying network and the protocol used for connectivity must be secure enough. When it comes to the protocol being used, strong algorithms for traffic encryption and secure tunneling are critical. Loosely coupled protocol exposes information to the man-in-the-middle and allows hackers to easily control the connected device. It must be very important when selecting the protocol to be used when connecting IOU to SCU. To mitigate protocol-related threats in the architecture, Secure by Design principles must be strong enough while suggest the underlying protocol.

IV. Unified Secure Architecture Solution Phsical & Network Layer Security Controls

In the preceding section, the author discussed the top ten security threats or challenges that are capable of jeopardizing the physical security of the device being integrated into the OEM. Though the author considered demonstrating the security threats by using Connected Vehicles as an example, imagine it in a way that if we plug a sensor device into any infrastructure, the challenges remain.

Having said that, the author discusses the Security Principles that can be used as a foundation for designing/architecting the Internet of Things (IoT) sensor via Secure by Design Architecturally in this section.

Architectural Solution: Physically Secure Sensor Devices by Design -

Consider the scenario with Connected Vehicle once more. The hacker may physically tamper with the SCU by removing the cover and attempting to change the chips (or) short-circuit (or) de-solder (or) additional solder of pins to change the behaviour pattern of the SCU motherboard.

The hacker must first open the container to remove the motherboard. So the challenge before us is to figure out how to keep the kill chain (or) a security control in place so that the OEM or user of the vehicle receives the alarm proactively.

The author proposes two techniques that can be used by OEMs or partners to mitigate this challenge.

Secure Architecture/Secure by Design – Physical tampering of chipboards/sensors

Let's take a closer look at the SCU device by opening it up. SCU is made up of a motherboard and a few integrated chips.

Figure 8: Tampered SCU

The author suggests two designs to prevent tampering.

1st Design:

Install magnetic sensors between the body and the motherboard. Magnetic sensors will continuously generate flux while keeping the circuit closed. When the motherboard is tampered with or removed from the cover, the flux circuit opens, and an inbuilt mechanism (message, continuous beep, or any mechanism that suits the OEM) activates the OEM to respond immediately.

Second Design:

Keep the motherboard inserted into the container and a Compression Spring pressed between the motherboard and the container. This pressure must be fed into the device for use as a reference. When someone tampers with/removes the motherboard, the pressure is released, which sends alerts to the OEM for immediate action.

One of the designs mentioned above can be used by vendors to reduce the risk of physical tampering with devices.

Secure Architecture/Secure by Design – Device Stealing

There must be a design mechanism in place to feed the sensor Longitudinal and Latitude data into the sensor to receive a notification if the device being integrated is tampered with or stolen. There must be a mechanism in place to notify the OEM if there is a change beyond 100 meters of the Long-Lat. In this way, we can reduce the likelihood of the device being stolen.

Secure Architecture/Secure by Design – Replacing the device

There must be a mechanism in place during OEM integration to allow the SCU and ECU to exchange trusted secrets, with the secret keys being asymmetric. Anyone attempting to tamper with the device with a different device will have these sensor-trusted secrets tampered with, and there will be an immediate mechanism to notify the OEM that the device is being replaced/tampered with.

Secure Architecture/Secure by Design – Switching off/resetting/coring the device

The hacker can reset, switch off, or coring the device through various means. There must be a secure-by-design approach where the hacker cannot turn off/reset the device, including replacing the battery. OEMs must consider how best to remove the entire functionality of the device being reset/switched off within the quoted warranty period.

Secure Architecture/Secure by Design – Misconfiguration of the device

In most cases, when the hacker does not gain access to the device, he or she attempts to misconfigure it through various means. As part of the Secure by Design approach, OEMs must investigate the methods of a standard factory reset configuration, and any activities that touch the configuration must have the code built in such a way that it is intelligent enough to reset the entire configuration to its original form. This configuration must be encrypted and placed in the core, where only the system can command and reset.

Secure Architecture/Secure by Design – Man-in-the-Middle

There was a chance that the hacker could perform a man-in-the-middle attack while the IOU and SCU were communicating. A strong tunnelling mechanism must be established between IOU and SCU so that hackers cannot tamper with this connection. If this occurs, the system must be intelligent enough to reset the secure tunnel and establish it quickly. Strong encryption, whitelisted commands, whitelisted codes for both request and response, and pre-configured hashes known only to the whitelisted Requester and Responder will suffice for good security in secure tunnel communication. Strong Ciphers, hashes (independent packet (or) complete packet (with/without headers), Header hashes, and Body Hashes are a few types that OEMs can experiment with depending on their needs.

Secure Architecture/Secure by Design – Secure Communication

In general, all OEMs or Partners use the CAN protocol to communicate between the SCU and the IOU. The CAN [7, 8] bus's existing built-in security features are primarily intended to ensure reliable communication rather than cybersecurity; thus, they cannot protect the network from cyberattacks. As a result, cyberattacks on CAN [9,10] are expected to have far-reaching consequences. For example, an attack on an airbag [2] or ABS systems can jeopardize the driver's and passengers' safety.

It may eventually harm the reputation of the car manufacturer, with serious financial consequences such as recalls. Tampering with ECUs (for example, used-car odometers [5] is another example that could have serious consequences for consumers and manufacturers.

The lack of encryption in CAN is also concerning, as it has a significant impact on individual data privacy. CAN is a broadcast network by design, allowing nodes to capture messages as they travel through the network. An adversary can obtain the desired data because the broadcasted data is not encrypted. This may result in an invasion of privacy, especially since modern cars [10,11,12,13,14,15,16] are capable of acquiring personal information from the driver.

CAN attacks can be mitigated with network segmentation, encryption, authentication, and intrusion prevention systems. Several CAN vulnerabilities can be prevented by IPS with minimal overhead.

Hardware security modules (HSMs) integrate security functions directly into the main processors of ECUs. When used in conjunction with security software stacks, they prevent unauthorized access to in-vehicle communications and vehicle control. Security functions are encapsulated in hardware security modules, which are integrated chips designed specifically for security applications. Several of today's leading chip manufacturers, including Infineon, ST Microelectronics, Renesas, and NXP, are involved create HSMs suitable for use in vehicles These HSMs use their processor cores to provide all of the main IT security functions required for automotive use cases: a 128-bit AES hardware accelerator, a true random number generator (TRNG) to generate key material, hardware-protected storage of cryptographic keys, flash and debugging functions, and the HSM's processor cores.

Secure Architecture/Secure by Design – Tokenisation

Access Control Lists (ACLs) [18], Whitelisted Messages, and Tokenisation (Token life cycle - creation of tokens, expiry, safe shredding of tokens) are some methods for aiding in the analysis of communication between various actors. These are some of the security controls that must be implemented when designing secure physical devices.

V. Summary and Future Directions

As the geometric progression of IoT devices has increased, various threat vectors have emerged that allow hackers to gain an advantage over the system. Using Connected Vehicles as an example, the author discusses various threat vectors in the physical and network layers of IoT devices and has proposed the best ways to architect/design secure plug-and-play

systems using the Secure by Design approach. Although a strong crypto protocol stack combined with Secure Tunnelling will solve the vast majority of transit traffic security challenges, there are some performance and narrowband resource concerns to be addressed. To assist IoT in building a strong tunnel, research must be conducted to address scanning delays and in-line scans, as well as to improve an efficient authentication and authorization process.

VI. Conclusions

In this paper, the author delves deeper into the IoT security flaws, threat vectors, actors, and various security challenges that affect the majority of IoT sensors (or) sensor-enabled devices physically; at the physical layer. The author used connected vehicles as an example to demonstrate the flaws and dug deep into them, discovering threats, threat vectors, and architecture design flaws, and determining the best way to design the system architecturally using Secure by Design. The discussion in Section II of this paper focused more on the Layer wise security challenges for IoT devices in the physical and network layers, using a Connected Vehicle as an example, and did a complete deep dive on understanding the flaws of connected vehicles and identifying the layer-wise security challenges. In Section III, the author identified architecture flaws and suggested the top ten flaws, and in Section IV, he provides a holistic approach to mitigating these security challenges/threats architecturally through the Secure by Design approach. Section V of this paper includes a summary and future directions for the next generation of researchers, as well as references and a summary.

VII. Declarations

Availability of data and material: Not Applicable
Funding: No Funding Source
Acknowledgements: Not Applicable

References Références Referencias

1. Sriranga Narasimha Gandhi Aryavalli, Hemantha Kumar; Top 12 layer-wise security challenges and a secure architectural solution for Internet of Things, Special section VSI-iotl, Received 2 February 2022; https://doi.org/10.1016/j.compeleceng.2022.108487
2. Dürrwang J., Braun J., Rumez M., Kristen R. Security evaluation of an airbag-ECU by reusing threat modeling artifacts; Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence, CSCI; Las Vegas, NV, USA. 14–16 December 2017; pp. 37–43.
3. "ECU" is a Three Letter Answer for All the Innovative Features in Your Car: Know How the Story Unfolded. Embitel. [(accessed on 23 May 2018)]; 2017 Available online: https://www.embitel.com/blog/embedded-blog/automotive-control-units-development-innovations-mechanical-to-electronics.
4. Hira E. Automotive Electronic Control Unit (ECU) Market Size Share, 2022. Allied Market Research; 2017 online: https://www.alliedmarketresearch.com/automotive-electronic-control-unit-ecu-market.
5. Murvay P.S., Groza B. Security shortcomings and countermeasures for the SAE J1939 commercial vehicle bus protocol. IEEE Trans. Veh. Technol. 20 18;67:4325–4339. doi: 10.1109/TVT.2018.2795384.
6. Mukherjee S., Shirazi H., Ray I., Daily J., Gamble R. Practical DoS Attacks on Embedded Networks in Commercial Vehicles. Volume 10063 Springer International Publishing; Cham, Switzerland: 2016.
7. Groza B., Murvay S. Security solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks. IEEE Veh. Technol. Mag. 2018; 13:40– 47. doi: 10.1109/MVT.2017.2736344.
8. Hoppe T., Dittman J. Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy; Proceedings of the 2nd workshop on embedded systems security (WESS); Brussels, Belgium. 4 October 2007; pp. 1–6.
9. Taylor A., Japkowicz N., Leblanc S. Frequency-based anomaly detection for the automotive CAN bus; Proceedings of the 2015 World Congress on Industrial Control Systems Security (WCICSS); London, UK. 14–16 December 2015; pp. 45–49.
10. Hamada Y., Miyashita Y., Hata Y., Inoue M., Ueda H. Anomaly-based intrusion detection using the density estimation of reception cycle periods for in-vehicle networks. SAE Int. J. Transp. Cybersecurity. Priv. 2018;1:39–56. doi: 10.4271/11-01-01-0003.
11. Greenberg A. Hackers Remotely Kill a Jeep on the Highway. Wired.com. [(accessed on 10 September 2018)];2015 online: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.
12. Koscher K., Czeskis A., Roesner F., Patel S., Kohno T., Checkoway S., Kantor B., Anderson D., Shacham H., Savage S., et al. Experimental security analysis of a modern automobile; Proceedings of the IEEE Symposium on Security and Privacy; Oakland, CA, USA. 16–19 May 2010; pp. 447–462.
13. Palanca A., Evenchick E., Maggi F., Zanero S. A stealth, selective, a link-layer denial-of-service attack against automotive networks; Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment; Bonn, Germany. 6–7 July 2017; pp. 185–206.
14. Hoppe T., Kiltz S., Dittmann J. Security threats to automotive CAN networks practical examples and selected short-term countermeasures. Reliab. Eng. Syst. Saf. 2011;96:11–25. doi: 10.1016/j.ress.2010.06.026.

15. Nie S., Liu L., Du Y. Free-Fall: Hacking Tesla from wireless to CAN bus. BlackHat USA. 2017;2017:1–16.
16. Tencent Keen Security Lab. Experimental Security Assessment of BMW Cars: A Summary Report. Peerlyst Inc.; Shenzhen, China: 2018.
17. Mukherjee S., Shirazi H., Ray I., Daily J., Gamble R. Practical DoS Attacks on Embedded Networks in Commercial Vehicles. Volume 10063 Springer International Publishing; Cham, Switzerland: 2016.
18. Statista. Automotive Electronics Cost as A Percentage of Total Car Cost Worldwide from 1950 to 2030. Statista; London, UK: 2018.
19. Checkoway S., McCoy D., Kantor B., Anderson D., Shacham H., Savage S., Koscher K., Czeskis A., Roesner F., Kohno T. Comprehensive experimental analyses of automotive attack surfaces; Proceedings of the 20th USENIX conference on Security; San Diego, CA, USA. 20– 22 August 2014.
20. S. Chakrabarty et al., "Black sdn for the internet of things," in Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on. IEEE, 2015, pp. 190–198.
21. Z. Qin et al., "A software defined networking architecture for the internet-of-things," in Network Operations and Management Symposium (NOMS), 2014 IEEE. IEEE, 2014, pp. 1–9.
22. Y. Lu et al., "Sdtcp: Towards datacenter tcp congestion control with sdn for iot applications," Sensors, vol. 17, no. 1, p. 109, 2017.
23. R. K. Das, N. Ahmed, F. H. Pohrmen, A. K. Maji, and G. Saha, "6le-sdn: An edge-based software-defined network for internet of things," IEEE Internet of Things Journal, 2020.
24. A. Hesham et al., "A simplified network access control design and implementation for m2m communication using sdn," in Wireless Communications and Networking Conference Workshops (WCNCW), 2017 IEEE. IEEE, 2017, pp. 1–5.
25. P. K. Das et al., "Context-sensitive policy based security in internet of things," in Smart Computing (SMARTCOMP), 2016 IEEE International Conference on. IEEE, 2016, pp. 1–6.
26. A. A. Levy et al., "Beetle: Flexible communication for bluetooth low energy," in Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, ser. MobiSys '16. New York, NY, USA: ACM, 2016, pp. 111–122.
27. J. Hong et al., "Demo: Building comprehensible access control for the internet of things using beetle," in Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion, ser. MobiSys '16 Companion. ACM, 2016, pp. 102–102.
28. A. L. M. Neto et al., "Aot: Authentication and access control for the entire iot device life-cycle," in Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM. ACM, 2016, pp. 1–15.
29. C.-J. M. Liang et al., "Sift: building an internet of safe things," in Proceedings of the 14th International Conference on Information Processing in Sensor Networks. ACM, 2015, pp. 298–309.
30. A. A. Yavuz, "Eta: efficient and tiny and authentication for heterogeneous wireless systems," in Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks. ACM, 2013, pp. 67–72.