Cybersecurity Assessment of CAN Networks and Alternatives: A Comprehensive Survey on Recent Attacks and Resource Requirement

Pratiksha Ashok Kumar Pole
ID 40230412

Deep Bhavesh Gajiwala
ID 40231725

Rohan Yogeshkumar Modi
ID 40255454

Snehpreet Singh
ID 40254443

Saket Suman
ID 40225128

Meet Rakeshbhai Patel
ID 40239187

Devina Shah
ID 40238009

Simaran Kaur
ID 40241517

Karanjot Singh
ID 40220861

Abstract—**Automotive control networks, anchored by the Controller Area Network (CAN) and its variants, constitute the digital backbone of modern vehicles, overseeing a spectrum of critical functions. As the automotive industry embraces connectivity and automation, the security of in-vehicle control networks emerges as a paramount concern. Potential cyber-attacks on these networks can have far-reaching consequences, from compromising privacy to jeopardizing vehicle safety. This survey analyses an in-depth examination of recent attacks on automotive networks, encompassing CAN, Local Interconnect Network (LIN), and FlexRay, to identify enduring vulnerabilities and quantify the computational and communication resources required for potential. The insights gained from this analysis will inform the development of advanced security measures, enhancing the integrity of in-vehicle control networks and fortifying the safety and reliability of modern automobiles.**

***Index Terms*—Automotive Control Networks, Controller Area Network (CAN), Local Interconnect Network (LIN), FlexRay, Cybersecurity, Attack Vectors, Vulnerabilities, Threat Landscape, Security Measures, In Vehicle Control Networks.**

I. INTRODUCTION

Overview of Automotive Networks

Automotive networks are the intricate web of communication protocols that connect electronic control units (ECUs) within a vehicle. The Controller Area Network (CAN) serves as the backbone, facilitating real-time data exchange for critical functions, including powertrain control, safety systems, infotainment, and more. Alongside CAN, alternative protocols like Local Interconnect Network (LIN) and FlexRay cater to specific requirements. The central challenge lies in securing these networks against cyber threats, as vulnerabilities could lead to unauthorized access, data manipulation, or system compromise. This project delves into recent attacks, assessing remaining vulnerabilities, and gauging the computational and communication resources required for potential breaches, ultimately reinforcing the security of automotive control networks. The significance of security in automotive networks cannot be overstated, as it stands at the crossroads of safety, privacy, and the future of transportation. In an era marked by rapid technological advancement, modern vehicles are no longer merely mechanical marvels; they are complex digital eco-systems on wheels. These vehicles rely on intricate networks of electronic control units (ECUs) and sensors to manage everything from engine performance and safety systems to infotainment and connectivity. While this digital transformation promises greater convenience and efficiency, it also ushers in a new era of cybersecurity challenges. The integrity of in- vehicle control networks is a linchpin in ensuring the safety of occupants, the privacy of data, and the reliability of the automotive industry. This discussion explores the multifaceted significance of security in automotive networks, shedding light on the critical factors that underpin the imperative need for robust and comprehensive cybersecurity measures.

The vulnerability assessment of the CAN protocol reveals shortcomings in ensuring confidentiality, integrity, and availability. The protocol's lack of cryptographic methods compromises data confidentiality, while issues in integrity checks and the priority-based messaging system impact the integrity and availability of communicated data. The discussion extends to the expansion of the automotive attack surface, detailing both physical access attacks (via the OBD port, selective DoS attacks, and indirect access methods) and remote access threats (exploiting wireless interfaces, OTA updates, and V2V/V2I communications). The classification of attacks on in-vehicle network systems identifies multiple entry points susceptible to security breaches, including various ports and communication interfaces.

In a detailed examination of notable attacks, the content explores real-world examples such as the lock picking attack on keyless entry systems, TPMS exploitation, road infrastructure attacks, and specific vulnerabilities in the CAN bus system. The Jeep hack via cellular network serves as a landmark case, illustrating the potential dangers of remote attacks on interconnected vehicle systems. Other instances include manipulation via the OBD-II port, selective DoS attacks, and vulnerabilities in the SAE J1939 standard. The content underscores the critical need for enhanced security measures in the automotive industry, considering the evolving attack surfaces, potential consequences, and the increasing complexity of in-vehicle communication networks.

II. OVERVIEW OF CAN, LIN, AND FLEXRAY NETWORKS

Securing automotive networks is paramount due to the potential implications of cyberattacks. Threats include unauthorized access, data manipulation, and system compromise. Attack vectors may exploit software vulnerabilities, hardware weaknesses, or weak network segmentation. Given the criticality of vehicle functions, any security breach can lead to dire consequences, ranging from loss of privacy to physical harm.

Controller Area Network (CAN) and its alternatives, namely Local Interconnect Network (LIN) and FlexRay, play a pivotal role in the automotive and industrial automation industries due to their distinct characteristics and applications.

1) **Controller Area Network (CAN):** It is a widely used communication protocol known for its robustness and efficiency. Its importance and usage in various domains are as follows:

Automotive Industry: CAN is the backbone of automotive networks, facilitating real-time communication between electronic control units (ECUs) that control various vehicle functions. It ensures the seamless operation of engine control, powertrain management, safety systems, and more.

Industrial Automation: It is employed in industrial automation, connecting PLCs, sensors, and actuators. Its determinism and reliability make it suitable for time-sensitive industrial processes.

The Controller Area Network (CAN) is an essential part of vehicle communication systems. Initially developed by Bosch in the 1980s for automotive applications, CAN has since become a standard in various industrial control environments. The primary purpose of CAN is to allow multiple microcontrollers and devices within a vehicle to communicate with each other without requiring a central computer. This feature makes it highly effective in managing complex operations where multiple subsystems need to interact seamlessly.

The network's physical layer usually consists of two wires forming a twisted pair, which helps in reducing electromagnetic interference. This robust design enables the CAN network to function reliably in the harsh electrical environments of vehicles. to prevent faulty nodes from disrupting the entire network. In terms of security, while CAN provides robust data transmission, it was not designed with security features to prevent malicious attacks. This lack of inherent security has become a concern in modern automotive systems, where the threat of cyberattacks is rising cost and compatibility difficulties all focus on this point.

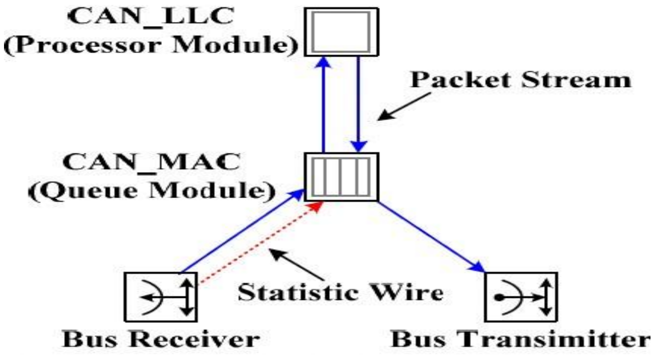

Fig. CAN NODE Architecture

2) **Local Interconnect Network (LIN):** LIN is an alternative to CAN with its own set of importance and applications:

Supplement to CAN: LIN is often used in conjunction with CAN in vehicles. It complements CAN by managing non-critical, low-speed functions like interior lighting, climate control, and infotainment systems. This ensures that CAN resources are reserved for critical tasks.

Energy Efficiency: LIN is designed with energy efficiency in mind, making it suitable for functions that must run continuously with minimal power consumption. The Local Interconnect Network (LIN) is a simpler, more cost-effective alternative CAN designed for less critical communication tasks within vehicles. Developed in the late 1990s, LIN is primarily used for managing simple actuators and sensors, such as mirror adjustments, seat positions, and rain sensors.

LIN operates as a single-master, multiple-slave network, where a central master unit controls communication with several slave nodes. This architecture simplifies the network design and reduces costs, making LIN an ideal choice for simpler and lower-speed applications. The communication speed in a LIN network is typically around 20 kbit/s, which is sufficient for the non-time-critical tasks it manages. A notable feature of LIN is its single-wire design, contrasting with the two-wire design of CAN.

Since LIN is used in less critical functions, the security risks are generally lower. However, as with CAN, the increasing connectivity of automotive systems raises the need for improved security measures in LIN networks as well.

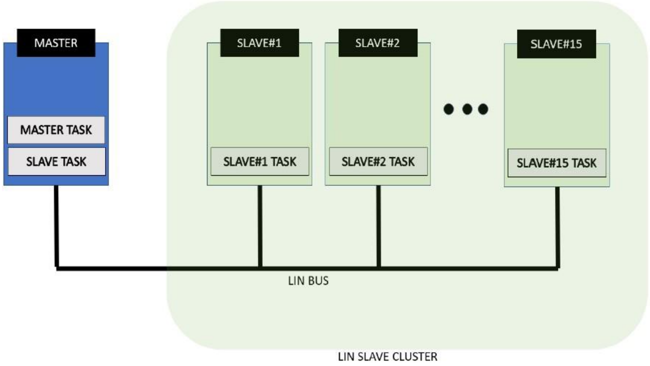

Fig. LIN architecture

3) **FlexRay:** FlexRay is a high-speed communication protocol that serves specific high-performance applications:

Advanced Driver Assistance Systems (ADAS): FlexRay is crucial for real-time, safety-critical functions in advanced driver assistance systems, like adaptive cruise control and lane-keeping assistance. It provides the necessary determinism and redundancy for these applications.

Redundancy and Fault-Tolerance: FlexRay's dual-channel design ensures redundancy and fault-tolerance, making it suitable for critical applications where system failure is not an option.

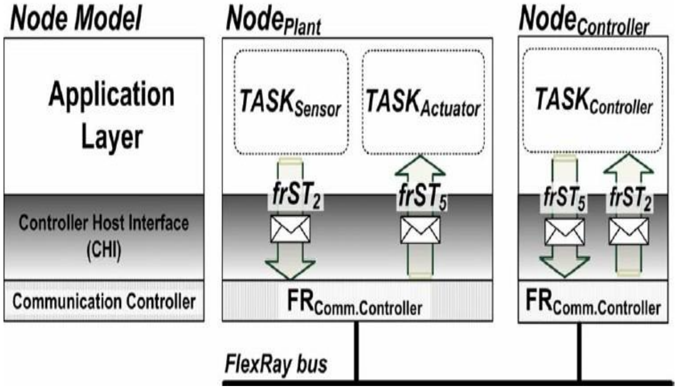

Fig. FlexRay Bus

FlexRay is the most advanced protocol among the three, designed to cater to the needs of complex and safety-critical systems in modern vehicles. Developed in the early 2000s by a consortium of automotive companies and suppliers, FlexRay addresses the limitations of CAN and LIN in handling high-speed and reliable communication required in advanced applications like x-by-wire systems.

III. Specific Attack Vectors and Methodologies

1. Vulnerability Assessment of the CAN Protocol

The Controller Area Network (CAN) protocol, a critical component in automotive communication systems, has been subjected to extensive vulnerability assessments due to its significant role in vehicle functionality and safety. A detailed analysis based on the principles of confidentiality, integrity, and availability reveals inherent weaknesses in the protocol, making it susceptible to various cyber-attacks.

- *Confidentiality Concerns*: Confidentiality, the principle of ensuring data accessibility only to authorized entities, is compromised in the CAN protocol due to its lack of inherent cryptographic methods. This omission allows intruders to access sensitive user data, leading to potential invasions of privacy and unauthorized data breaches.
- *Integrity Issues:* Integrity in data communication entails the assurance of data accuracy, completeness, and validity. The CAN bus utilizes a Cyclic Redundancy Check (CRC) for the verification of data integrity against transmission errors. However, this mechanism is insufficient in preventing data injections by malicious entities, leading to a breach of data integrity. The protocol's lack of comprehensive integrity checks results in its inability to sustain the integrity of communicated data.

- *Availability Vulnerabilities:* The principle of availability implies that authorized users should have consistent and reliable access to the system. In the CAN protocol, the priority-based messaging system can be exploited, wherein messages of the highest priority can dominate the network. This manipulation results in making the network inaccessible to lower-priority nodes, thereby violating the principle of availability.

2. Automotive Attack Surface Expansion

The evolution of automotive technology, especially the increase in embedded electronics, has led to a proportional rise in the attack surface for potential cyber threats. The integration of advanced electronic components and systems, such as sensors, actuators, control units, and communication systems, has transformed vehicles from closed to open systems. This transformation has introduced complexities in vehicle communication and expanded the potential for cyber-attacks, which can now be executed remotely without physical access to the vehicle.

- *Types of Attacks:* Automotive cyber-attacks can be categorized into physical access attacks and remote access attacks. Physical access attacks require direct interaction with the vehicle's network, typically through the On-Board Diagnostic (OBD) port or by installing a malicious node in the

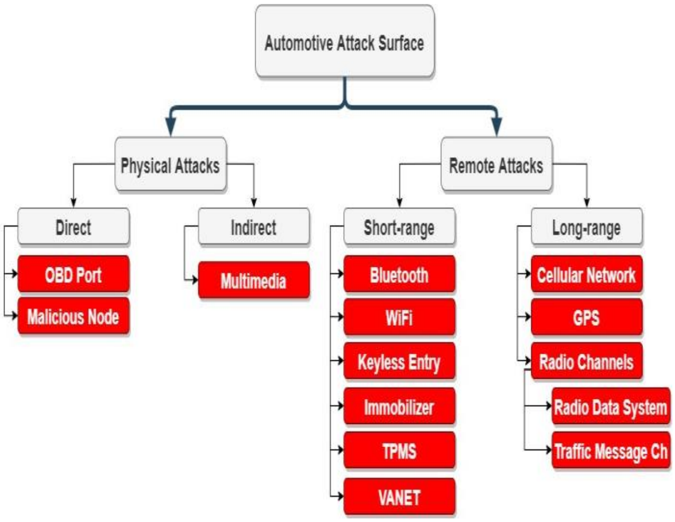

Fig. Automotive Attack Surface

vehicle's network. Remote access attacks are executed via wireless communication interfaces, such as Bluetooth, Wi-Fi, and cellular networks, allowing attackers to exploit vulnerabilities from a distance.

3. Notable Physical Access Attacks

Physical access attacks involve direct interaction with the vehicle's network systems, often through accessible ports or by installing unauthorized devices within the vehicle's network.

- On-Board Diagnostics (OBD) Port Attacks: The OBD port, being a direct gateway to the vehicle's network, is a primary target for attackers. By exploiting the OBD port, attackers can manipulate various vehicle modules, including critical systems like brake and engine control. Such attacks have demonstrated the ability to release brakes, prevent brake activation, manipulate instrument clusters, change engine parameters, and even disable the engine while the vehicle is in motion.
- Selective Denial-of-Service (DoS) Attacks: These attacks disrupt the network without necessitating full message transmission. They can be executed by overwriting specific bits in the transmitted data, thereby generating transmission errors, and exploiting the vulnerability of the CAN standard. Research in this area has focused on exploiting these vulnerabilities, leading to government alerts and increased awareness of the susceptibility of vehicles to such attacks.
- Indirect Physical Access Attacks: These attacks do not require direct access to the vehicle's network. For instance, hacking the IT system of a car service can provide indirect access to the CAN. Another method includes attacking via multimedia devices like CDs, USBs, or MP3 players. While these attacks may not directly breach the CAN, they can affect the driver by flashing warnings on the screen or playing alarm signals.

4. Remote Access Attacks

Remote access attacks represent a significant threat in modern vehicles due to the integration of various wireless interfaces necessary for communication with systems like anti-theft devices, tire pressure monitoring systems (TPMS), Bluetooth, and telematics units.

- Exploiting Wireless Interfaces: These interfaces, typically connected to the CAN via a gateway ECU, have been demonstrated as vulnerable points for hacking. Successful compromises of these systems can lead to unauthorized control over the vehicle, including unlocking doors and manipulating vehicle functions remotely.
- OTA Software Update Vulnerabilities: Over-the-Air (OTA) updates, while convenient and cost-effective for software delivery, present another attack surface. Hackers can potentially intercept these updates to infiltrate the vehicle's communication network, leading to ransomware attacks or other forms of cyber sabotage.
- V2V and V2I Communications: The advent of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, which are integral to vehicular ad hoc networks (VANETs), introduces new vulnerabilities. These systems, designed for traffic optimization and collision avoidance, can be compromised by spoofed messages, resulting in disruptions to in-vehicle communication networks.

5. Classification of Attacks on In-Vehicle Network System

The in-vehicle network system is prone to various attacks, with the following being the most significant entry points for attackers:

- *OBD-II Port:* This port, used for monitoring vehicle diagnostics, is a critical vulnerability as attackers can easily collect diagnostic data, gaining access to the in-vehicle network and deploying malicious programs.
- *USB and Charging Ports:* These ports are susceptible to severe security threats, such as the installation of malicious codes and reprogramming of the controller processor, enabling hackers to control critical vehicle systems like braking and engine control.
- *TPMS, LiDAR, and Keyless Entry Ports:* These systems can be exploited for eavesdropping attacks, signal jamming, and interception of key signals, leaving the vehicle network open to unauthorized access and manipulation.
- *Bus Network Ports:* The lack of communication protection in CAN protocols allows attackers to send fake frames to each node, leading to unintended vehicle behavior.
- *Vehicular Communication Ports:* Enabled with technologies like Bluetooth, Wi-Fi, DSRC, and cellular networks, these ports are vulnerable to various attacks, including jamming and eavesdropping, potentially allowing attackers to gain full access to vehicles.

IV. ATTACKS ON IN-VEHICLE NETWORK SYSTEM

A. **Detailed Discussion of Notable Attacks with Examples**

1. Lock Picking Attack on Keyless Entry Systems

The lock picking attack exploits the keyless entry system's vulnerability, typically used for car doors and garage openers. The attack involves a man-in-the-middle approach using a device that captures the key fob's transmitted signal while sending a jamming signal on the same frequency. When the key fob user retries, the attacker's device captures the second code transmission, using the first code to unlock the door while retaining the second code for future unauthorized access. This technique has been demonstrated on various car brands and garage door openers, with devices like Rolljam and OpenSesame specifically designed for these types of attacks. Despite their widespread availability and potential misuse, sales of these devices have continued unabated, prompting concerns and calls for enhanced security measures from manufacturers.

2. TPMS Exploitation

The Tire Pressure Monitoring System (TPMS) is a well-documented target for exploitation, lacking essential security safeguards. TPMS, which continuously monitors tire pressure and communicates with the vehicle's Electronic Control Unit (ECU), can be attacked passively or actively. Passive attacks allow attackers to track vehicle movement by capturing TPMS signals, while active attacks involve wirelessly injecting spoofed signals to deceive the ECU. These attacks can result in displaying false tire pressure measurements, posing significant risks to vehicle safety. Countermeasures include using hardware pairings and encrypted signal transmissions to mitigate these vulnerabilities.

3. Road Infrastructure Attacks

The increasing connectivity of vehicles has brought road infrastructure components into the scope of potential cyber threats. Vehicle-to-infrastructure (V2I) connectivity, encompassing elements like smart traffic lights and road signs, presents new attack vectors. One notable incident involved the compromise of networked road signs in multiple states, displaying messages indicating they were hacked. Although initially perceived as pranks, such attacks have serious implications, especially during emergencies. Mitigating these attacks requires stringent password management and secure design of sensors for V2V and V2I communications.

4. CAN Bus Specific Attacks

The classical CAN and CAN FD buses are susceptible to a range of attacks once access is gained:

- *CAN Sniffing:* Attackers can passively monitor and analyze data on the CAN bus, using devices like the CAN-do board, to manipulate and generate similar messages.
- *CAN Fuzzing:* This involves sending random CAN data frames to observe changes in vehicle behavior, like speed alterations, during frame injections.
- *Frame Falsifying and Injection Attacks:* Attackers modify CAN message payloads or inject data at abnormal rates to simulate events, leading to fake events that can drastically affect vehicle behavior.
- *DoS Attacks:* These attacks use the highest priority CAN frames to monopolize the bus, preventing other nodes from accessing it, thus disrupting sensor parts and overall vehicle functionality.

5. ECU Impersonation

Attackers can impersonate an ECU by analyzing traffic on the CAN bus and simulating the behavior of an ECU, including its CAN ID, payload range, and transmission rate. Sophisticated spoofing ECU attacks can exploit the error-handling mechanism of the CAN bus protocol, causing legitimate ECUs to be disconnected and dropping all CAN bus communication.

6. Jeep Hack via Cellular Network

The Jeep hack, a landmark event in automotive cybersecurity, demonstrated the vulnerability of in-car networks to remote attacks. In 2015, security researchers Charlie Miller and Chris Valasek exploited the cellular network interface of a Jeep Cherokee to gain remote control over the vehicle.

Methodology and Impact:

The attack was initiated through the Uconnect system, which is connected to the cellular network and controls entertainment and navigation and is connected to the CAN bus. By exploiting a vulnerability in the Uconnect system, the attackers remotely accessed Jeep's CAN network.

Once they gained access, Miller and Valasek were able to control various functions of the Jeep, including the air conditioning, radio, windshield wipers, transmission, and ultimately the braking system—all while the vehicle was in motion. This attack highlighted the potential dangers of interconnected vehicle systems and the need for robust security protocols in vehicle-to-network interfaces.

7. Replay Attack

In LCAP - Lightweight Controller Area Protocol has been discussed. It differs from CAN as the main purpose of the LCAP is to keep messages authenticity and integrity for all messages traveling over the CAN bus. However, it is highlighted that the LCAP has a vulnerability related to replay attacks, specifically in channel requests. The attackers can exploit this by periodically replaying channel request messages, leading to a Denial of Service (DoS) attack, blocking normal data messages. A malicious node can record messages on the bus, identify the channel request message through trial and error, and replay it periodically. Controlling messages take priority over data messages so periodically sending the control messages for a replay attack will block the network pipeline to keep sending controlling messages, and data messages don't get passed through.

The attack is quite severe as it affects controls in an automotive system. A delay or failure of those could cause major issues in a vehicle. The detection of this attack is also quite difficult but can be prevented by adding a nonce to the messages. A nonce or a timestamp will prevent a replay attack.

The simulation of this attack is conducted using the Vector CAN simulation, a well-known automotive network simulator. The proposed solution includes using nonce in messages to prevent replay attacks and improve system security.

Another form of replay attack was implemented in [4]. One is to replay the entire message and the other one is to replay only part of the frame excluding the identifier. In the first attack, the entire message is stored by a node and then transmitted again when the CAN bus is idle. In this attack all nodes will receive faulty data. The only way to detect this attack is when the ID is compared, and the original source node detects receiving a message with the same ID as theirs. They can create an error frame when a replay attack is detected.

In the second method, partial frames are replayed, and ID frames are replaced. This attack is harder to detect as the source uses its own ID. Since it's a replay attack the way to prevent this is also through adding timestamps or nounces.

CAN does not offer security services such as encryption or data frame authentication. This means that eavesdropping and replay attacks in CAN are possible.

8. Attack through a Malicious App

Other than replay attacks, a common attack researched was through a diagnostic malicious App. When a malicious app is used for attack, it enables the attacker to perform a long-range wireless attack where the attacker or attacking equipment need not be in close range of the vehicle to perform the attack. The user will install a self-diagnostic app to monitor status information after installing an OBD2 scan tool on the vehicle and then pairing it with his/her smartphone by Bluetooth. When the driver installs on his/her smartphone the malicious self-diagnostic app distributed by an attacker, the attacker can launch the actual attack. The attacker can obtain status information of the vehicle from the malicious self-diagnostic app and use it to inject malicious data into the in-vehicle network. As per this research vulnerabilities of in-vehicle CAN are weak access control, no encryption, and no authentication.

After the malicious app is installed on the victim's device, it transmits data frames of the in-vehicle CAN to the attacker's server using the smartphone's mobile communication network. This would force control of an ECU to the in-vehicle CAN via the malicious app. The target vehicle would have a physical malfunction caused by the abnormal control data that was transmitted from the attacker's server.

Another prominent attack with malicious apps is discussed. Apps for vehicles can be easily forged/repackaged and redistributed. Using a vehicle diagnostic device, the attacker can get a CAN data frame that can drive a specific ECU mounted in the target vehicle. Furthermore, the attacker can download a vehicle application that is distributed/sold in the app market and repackage it in a desired form. The diagnostic app will provide many services to the victim while they are driving and can use this service to start an attack.

9. Bus off Attack

Apart from attacks such as DOS or Replay attacks, there are other attacks that CAN is vulnerable to. One of those attacks is a bus-off attack, which can cause a victim ECU to disconnect itself from the CAN bus and, subsequently, an attacker can masquerade as that ECU. A limitation of the bus-off attack is that it requires the attacker to achieve tight synchronization between the transmission of the victim and the attacker's injected message.

The bus-off state is a state of error in a CAN controller in which the node is disconnected from the bus communications which means it can neither transmit nor acknowledge frames. When a node recovers from bus-off, it resets its counter and starts from the initial error-active state.

The attack's success depended on meeting a few conditions: matching message IDs with the victim and synchronization with the victim message. The attack is achieved by identifying a unique message that precedes the victim. However, if no unique preceded ID exists, fabricating and injecting unique preceded IDs to interfere with the victim's transmission was proposed. It can still be detected by an IDS. Alternatively, the schedule-based bus-off attack doesn't rely on unique preceded IDs for synchronization. It targets instances of the victim message facing blocking or interference, regardless of the preceding message's ID. The attacker gains knowledge about message sets on the CAN bus before launching the attack, allowing them to identify opportunities for successful synchronization.

10. Manipulation via OBD-II Port

Koscher et al.'s Research:

In a significant study, researchers Karl Koscher and his team at the University of Washington and the University of California, San Diego, demonstrated how the OBD-II port could be used to manipulate critical vehicle control systems.

Techniques and Consequences

The researchers were able to directly connect to the vehicle's CAN network through the OBD-II port. By sending malicious CAN messages, they could control critical functions like the brakes and engine.

Notably, they showed it was possible to disable the brakes or falsely engage them, turn off the engine, and interfere with other crucial systems. This research brought to light the vulnerabilities inherent in the OBD-II port, which is typically used for vehicle diagnostics and is easily accessible.

11. Selective Denial-of-Service (DoS) Attack

Palanca et al.'s Contribution:

A team led by Palanca, et al., demonstrated a selective Denial-of-Service attack on an unmodified car, emphasizing the vulnerabilities in the CAN standard.

Execution and Implications

The attack involved overloading the CAN network with high-priority messages, effectively blocking other legitimate messages from being processed. This type of attack can render certain vehicle functions inoperable or cause erratic behavior in the vehicle's electronic systems.

This form of attack was particularly concerning because it could be executed without requiring significant modification to the vehicle, demonstrating that even standard, unaltered vehicles are at risk.

12. SAE J1939 Standard Attacks

Mukherjee et al. and Murvay and Groza's Findings:

Researchers Mukherjee et al. and Murvay and Groza focused on the vulnerabilities within the SAE J1939 standard, which is commonly used in heavy-duty commercial vehicles.

DoS Attacks and Protocol Vulnerabilities:

Mukherjee and his team demonstrated the feasibility of DoS attacks on vehicles using the SAE J1939 standard. By sending excessive request messages or manipulating false requests, they could overload the recipient ECUs, causing disruptions in vehicle operations.

Murvay and Groza highlighted both specific protocol vulnerabilities and general weaknesses in the CAN bus system. Their work emphasized how the architecture of the CAN bus, without proper safeguards, could be exploited to compromise vehicle functions or safety.

*B. **Computational and Communication Resources Needed by Attackers***

The complexity of attacks on the Controller Area Network (CAN) and its variants in modern vehicles requires diverse computational and communication resources. These resources range from sophisticated software for data analysis to specialized hardware for network interfacing. This review consolidates insights from various academic sources to understand the resources attackers utilize in executing these cyberattacks.

Data Collection and Analysis

- CAN Bus Data Capture: Attackers capture data from a vehicle's CAN network, often via the On-Board Diagnostics (OBD) port, using tools like Vehicle Spy simulation test software. This process demands precision to avoid packet loss and maintain data integrity. Large datasets are created for deep analysis, requiring significant storage and processing capabilities.
- Anomaly Detection Algorithms: Post data collection, sophisticated algorithms analyze anomalies in the CAN ID's message stream, a process that requires substantial computational power. These algorithms discern between normal vehicle operation changes and deliberate tampering, making them crucial in attack execution.

Exploiting Connectivity Interfaces

- Hacking Connectivity Systems: Modern vehicles are equipped with a plethora of connectivity interfaces such as Bluetooth, Wi-Fi, and cellular networks. Exploiting these systems requires specialized software capable of intercepting and manipulating wireless communications. Such tools need to be adaptable to different technologies and robust enough to breach security protocols.

- Diagnostic Port Vulnerabilities: The OBD-2 port, used for diagnostics, is a frequent target for cyberattacks. Attackers use specialized hardware to interact with this port, leading to potentially severe safety compromises. The hardware employed here is often compact yet capable of powerful interfacing with the vehicle's internal systems.

- Targeting Vehicle Sensors and Actuators: Physical availability attacks, like signal jamming, are directed at crucial vehicle sensors and actuators. This method requires hardware that can effectively disrupt or manipulate sensor data, such as devices capable of eavesdropping on TPMS signals.

- Signal Jamming and Relay Attacks: Systems like LiDAR and keyless entry are susceptible to signal jamming and relay attacks. These attacks necessitate sophisticated hardware and software to capture and relay signals, enabling unauthorized access to vehicles.

*C. **Dependencies on Physical Access or Specific Conditions for Successful Attacks***

1. Physical Access to the Vehicle: Many attacks on the CAN network require physical access to the vehicle. This can range from a mechanic to a family member inserting a malicious component into the car's internal network, usually through the OBD-II port. The impact of these attacks can be profound, allowing control over critical vehicle functions.

2. Direct Access through OBD Port and Malicious Nodes: Direct access attacks are often executed through the OBD port, which provides comprehensive access to the vehicle's network. Such attacks can control critical vehicle systems, including braking and engine functions. Malicious nodes, once connected, can disrupt the network by intercepting or sending deceptive messages.

3. Denial-of-Service (DoS) Attacks: DoS attacks, particularly on commercial vehicles, involve overloading the network with excessive request messages or manipulating network connections. These attacks require an understanding of specific network protocols and the ability to generate substantial traffic to overload the system.

4. *Indirect Physical Access:* Indirect physical access attacks involve inserting a physical object into the car, without direct network access. Checkoway et al. developed an indirect access attack model by hacking the car service's IT system and accessing the CAN via a computer. Another indirect attack involved using multimedia devices like CDs, USBs, or MP3 players. Although these attacks may not directly breach the CAN, they can disrupt the driver by displaying warnings and playing alarm signals, emphasizing the importance of securing both direct and indirect access points to the CAN bus network.

V. COUNTERMEASURES AND MITIGATION STRATEGIES

A. ***Existing security measures and their effectiveness:***

In the last decade, researchers have explored a wide range of malware defense solutions for computer and mobile systems. Those solutions can be categorized into signature- based, behavior-based, heuristic-based, cloud-based, and ma- chine learning-based techniques. In this section, we present a detailed review of the main factors of applying these defense systems to protect intelligent vehicles against malware. These factors include the used approach, the used data analysis method, the targeted operating system, the detection time and the detection response, the data source, the main advantages, and disadvantages of each defense system. Figure 1 shows the taxonomy dimensions distributed into six classes. We also briefly describe these classes below.

1. *Techniques*: We classify the existing malware detection techniques into five categories, i.e., signature-based malware detection techniques, behavior-based malware detection techniques, heuristic-based malware detection techniques, cloud- based malware detection techniques, and machine learning- based malware detection techniques. Each of these techniques has certain advantages and disadvantages, we discuss the benefits and drawbacks of each technique.

2. *Analysis Methods:* The whole detection process is ac-accomplished with static, dynamic and hybrid analysis methods. The description of each method is presented below.
Static Analysis: It is a malware analysis method that analyzes an executable code without actually executing the code itself. In static analysis, the low-level information from codes is extracted by disassembling the codes by using any disassembler tools. The main advantage of this method is revealing the code structure of the program without executing it. However, this method may fail in analyzing unknown malware. It may also fail to detect malware that employs obfuscation and evasion techniques in its code.

Dynamic Analysis: It is a malware analysis method that entails running the malware and monitoring its behavior, interactions with the host system, and its impacts on the host environment. The infected files in this method are analyzed in a simulated environment such as an emulator, virtual machine, and sandbox in order to make the environment invisible to the malware. Although this method is efficient in detecting malware, nevertheless, it may fail to detect malware that uses obfuscation code and evasion techniques.

Hybrid Analysis: It is a malware analysis method that combines both dynamic and static analysis. It examines almost all of the static features of any malware code then combines them with other behavioral features to better the overall analysis process. Despite this method can overcome the limitations of both static and dynamic analysis methods. However, it may result in a rise in the execution time's total overhead.

3. *Target Operating System (OS).* It refers to the operating system analyzed by the system. It can be LINUX, Windows, or Android.

4. *Detection Time.* It refers to the time between the analyzed event and the detection itself. It can be real-time (online) detection, which enables an automatic response such as blocking the attacker and killing the malware process, or non-real-time (offline) detection.

5. *Detection Response.* The relevant outcome of the system, which can be a passive response which is an event notification such as printing an alert message, or an active response which is an automatic reaction such as blocking the attacker or killing the malware process.

6. *Data Source*. It refers to the source of the input data analyzed by the system. It can be hosting logs which are data from the operating system and system applications or application logs which are data directly generated by applications, or network traffic which are data generated by the network layer.

Detection Techniques:

Malware detection system taxonomy Signature-Based Malware Detection :

The process of signature-based malware detection, prominently used in commercial antivirus tools, comprises two main stages. Initially, a unique signature is crafted for each malware, derived from a mix of manual and automated analysis of data from networks and user devices. Then, devices store these signatures to identify malware in files or data streams. This technique, which disassembles and analyzes malware binary codes, is simple, fast, and safe, especially for intelligent vehicles, excelling in detecting known malware but falling short in identifying new, unknown threats due to its vulnerability to evasion. In contrast, innovative malware detection methods focus on digital footprints in various log files and employ different analysis techniques, such as static analysis, function call graph similarity, and API call sequences. These approaches are highly accurate for known malware but are limited in detecting unknown malware and not suitable for real-time applications, such as in intelligent cars.

Additionally, researchers have explored various sophisticated techniques for malware detection, primarily targeting the Windows operating system. These methods include analyzing program bit files without code execution (static analysis) and using unique identifiers like control flow graph signatures and byte sequences of executable files.

Behavior-Based Malware Detection :

The way of behaving based malware discovery strategy inspects how a program acts to decide whether it is malevolent. It does this in a safe climate like a virtual machine, without depending on outside frameworks, in any event, for new, never-seen-before malware. Many utilize this strategy to counter malware, taking a gander at expected ways of behaving and utilizing information signs on different working frameworks. In spite of its high location rate, it has disadvantages like significant expenses and intricacy, making it unacceptable for savvy vehicles. While it succeeds in distinguishing new malware, it battles with arranging all ways of behaving precisely, prompting possible bogus up-sides or negatives. Contrasted with signature-based recognition, it is harder to execute and asset serious for in-vehicle gadgets, presenting difficulties for long haul use in vehicles.

Heuristic-Based Malware Detection :

The heuristic-based malware discovery procedure evaluates program records for dubious attributes or reenacts program execution to recognize possible malignant exercises. This technique, known for its intricacy, draws on previous encounters and utilizes information mining, rule-based frameworks, and AI to learn program qualities. Generally utilized in antivirus programming, it can distinguish different known and obscure malware, including zero-day dangers. In any case, it battles with distinguishing most new and refined malware and is powerless to innovative code jumbling and avoidance methods. Analysts have proposed static investigation techniques, similar to control stream charts, and dynamic strategies utilizing DLLs or Programming interface call organizations. While powerful for known malware, these methodologies are perplexing, have high misleading positive rates, and are not appropriate for continuous recognition in that frame of mind because of their tedious nature. In spite of its solidarity in distinguishing obscure malware, the heuristic-based method is really difficult, and asset escalated contrasted with signature-based and con- duct based strategies. It may not be great for asset obliged in-vehicle gadgets, given its intricacy and possible outdated nature after some time.

Cloud-Based Malware Detection :

Distributed computing has acquired fame for its helpful access, on-request capacity, and cost-viability. As of late, it has been utilized in malware discovery through the Cloud- based method, utilizing location specialists on cloud servers. This technique permits clients to submit documents for investigation and get writes about their malware status. While it improves recognition execution with broad information bases and figuring assets, it faces disadvantages, for example, dependence on a steady and quick web association, powerlessness for continuous record observing, and weakness to jumbling and avoidance methods.

Scientists have investigated cloud-based strategies for malware examination, utilizing static investigation with highlights like document content and relations, as well as powerful investigation through framework call checking. Nonetheless, these strategies cause significant expenses, above, and time delays, making them unsatisfactory for continuous discovery, particularly in keen vehicles. Notwithstanding the benefits of speedy access and refreshed establishments, the cloud-based approach is evaluated by the requirement for a solid web association and defenselessness to cutting edge avoidance strategies, raising worries about its wellbeing in savvy vehicles. The rise of high velocity 5G innovation might work on its reasonability in this specific circumstance.

Machine Learning-Based Malware Detection :

AI has for quite some time been a foundation in the mission to recognize malware, with calculations like Guileless Bayes, Bayesian organization, strategic relapse, and others offering novel qualities. Every calculation's viability relies on factors like information dispersion and element relationships. Profound Learning, a branch-off of fake brain organizations, has arisen as an amazing asset, particularly in applications, for example, picture handling, voice control, and all the more so as of late, malware identification. Nevertheless, its power- lessness to jump and avoidance, combined with the time- concentrated course of developing secret layers, highlights the requirement for smart application.

Network Segmentation

Carrying out network division by partitioning the CAN organize into subnetworks is a primary safety effort, giving command over access and restricting the expected spread of assaults. This normal methodology in business vehicles includes a door Electronic Control Unit (ECU) directing interconnections between subnetworks. Nonetheless, weaknesses emerge in the event that the door ECU is compromised, as exhibited in certain hacking situations.

Encryption

The CAN convention, lacking implicit encryption, opens its correspondence to potential listening in by enemies, making the execution of a lightweight encryption framework critical. While business programming-based encryption strategies and exclusive methods by producers exist, reports recommend weaknesses in monetarily accessible vehicle encryption frameworks. Challenges in secure Could encryption at any point incorporate the restricted information field, tended to by sending various CAN outlines for a solitary message, though not great for high-traffic organizations. Moreover, the computational imperatives of ECUs require dynamic key trade to forestall static key split the difference over a vehicle's life expectancy. Nevertheless, dynamic key trade presents execution hardships, computational costs, and idleness issues for asset compelled ECUs, making it unacceptable for security basic continuous frameworks.

Secure Boot

Secure Boot is not an automotive industry exclusive idea, in fact, it's supported by most BIOS in home PCs. It is a Unified Extensible Firmware Interface (UEFI) mechanism which allows only software with valid signatures to be booted in the machine it is working on. The manufacturer loads some databases of keys and signatures in the device in manufacturing time. The firmware would then be signed and have its signature checked before booting. Secure booting is an extraordinarily strong security mechanism that stops malicious firmware from being booted, but previously there have been vulnerabilities in specific implementations that allowed for bypasses on some occasions, which makes secure booting not a silver bullet.

Secure Access Service

The scientists dug into the diagnostics validation component while figuring out a 2010 Toyota Prius and a 2010 Portage Break, uncovering a Security Access system implanted inside the Bound together Diagnostics Administration (UDS) characterized in ISO 14229-1. UDS fills in as a standard diagnostics correspondence convention for Electronic Control Units (ECUs) in the auto domain, offering different administrations to gather data about a vehicle's usefulness and state. The Security Access administration utilizes a Test Reaction convention, where the analyzer demands a "seed" from the ECU, which, upon age, is sent back. Both the analyzer and ECU share a cryptographically safe capability and a key to make a reaction. Notwithstanding, the free standard needs points of interest on the capability, keys, or seed age. Outstandingly, weaknesses were recognized, for example, unsurprising seeds and lacking reaction length, empowering assailants to utilize beast power or replay assaults. The specialists effectively separated keys through figuring out, uncovering shortcomings in the security of these auto frameworks.

Secure Onboard Communications

To address the confirmation deficiencies in the Regulator Region Organization (CAN), the AUTOSAR people group proposed the Solid Locally available Correspondences (SecOC) module, which presents CAN approach verification by adding marks to in-vehicle correspondences. SecOC works with both even and unbalanced cryptography, expecting that vital administration and trade are as of now settled. The method includes annexing a mark, alongside a newness an incentive for uniqueness, to the safeguarded information unit (PDU) in the CAN outline. In balanced mode, for example, the shipper computes a Message Validation Code (Macintosh) over the info information and newness esteem utilizing a common key, annexing it to the message. The recipient confirms the newness esteem and recalculates the Macintosh, tolerating the message if right. Nonetheless, squeezing this framework into a normal CAN approach raises difficulties, for example, shortening the 128-bit Macintosh into a 27-piece esteem, possibly powerless to beast force assaults. Another methodology, the Fitting and Secure Key Foundation (PnS), offers a minimal expense symmetric key foundation convention for CAN, using an actual property in the transport to characterize a key between two gadgets at whatever point they communicate a piece all the while, upgrading security in the organization.

Firewall Gateway

Improving in-vehicle correspondence security can include conducting a firewall system inside network doors. In the event that message confirmation codes (Macintosh) or advanced marks are used for verification and approval between Electronic Control Units (ECUs), firewall rules can be gotten from the approvals in each ECU's declaration. At the point when Macintoshes or computerized marks are missing, firewall rules can be separately characterized in light of vehicular subnet approvals, permitting just messages from legitimate and valid ECUs to go through and be sent on the in-vehicle transport framework. Another methodology is to limit the entrance level of various organization types to explicit pieces of the transport framework, forestalling fewer basic organizations, similar to LIN or MOST, from sending messages to higher wellbeing pertinent frameworks, for example, CAN or FlexRay.

Honeypots

Getting in-vehicle correspondence frameworks is vital because of the weakness presented by remote correspondence entryways. Assailants can take advantage of remote admittance to send off digital goes after straightforwardly on the in-vehicle organization, which controls basic vehicle activities. Understanding aggressor's conduct is critical to creating successful security arrangements. Honeypots, intended to show up as weak targets, function as instruments for avoidance and early discovery of pernicious assaults. In the car area, sensible honeypots can be executed to draw in genuine aggressors without disrupting typical vehicle activity. These honeypots gather information as the vehicle travels through unambiguous regions, recording data influencing the in-vehicle organization. Examination of this information distinguishes assault conduct, situations, and orders, helping with the improvement of vigorous safety efforts to shield in-vehicle correspondence frameworks in later plans and executions.

VI. CONCLUSION

Vulnerability to Evasion Techniques: A significant issue with existing malware detection methods is their susceptibility to evasion. Modern malware uses various obfuscation techniques, like throttling execution across multiple Electronic Control Units (ECUs) or leveraging multi-core processors to spread activities and avoid detection. These methods can make malware difficult to analyze and, consequently, evade existing detection systems. This vulnerability is particularly concerning in the context of intelligent vehicles, where passenger safety is paramount.

Challenges in Detecting New Malware: The current approaches struggle to identify new, sophisticated malware forms, posing a risk to driver and passenger safety in intelligent vehicles. Most traditional detection methods also face challenges in staying updated over a vehicle's lifespan, making them impractical due to the excessive costs and logistical complexities involved in updating millions of vehicles regularly. In contrast, cloud-based approaches, which update configurations regularly in the cloud, are seen as more viable, especially with the emergence of high-speed 5G technology, offering a more adaptable and up-to-date defense against malware.

The survey highlights the vulnerabilities in the Controller Area Network (CAN) protocol, emphasizing the need for enhanced security measures to address confidentiality, integrity, and availability concerns. As the automotive industry continues to evolve, understanding these cybersecurity challenges is crucial for developing robust solutions and exploring alternative technologies that offer improved resilience against cyber threats in vehicle communication networks.

REFERENCES

[1] Murvay, P., & Groza, B. (2020). Efficient Physical layer key agreement for FlexRay networks. *IEEE Transactions on Vehicular Technology*, *69*(9), 9767–9780. https://doi.org/10.1109/tvt.2020.3002616

[2] Jeong, W., Choi, E., Song, H., Cho, M., & Choi, J. (2022). Adaptive Controller Area Network Intrusion Detection System considering temperature variations. *IEEE Transactions on Information Forensics and Security*, *17*, 3925–3933. https://doi.org/10.1109/tifs.2022.3217389

[3] Tanksale, V. (2020). Controller Area Network Security Requirements. *Controller Area Network Security Requirements*. https://doi.org/10.1109/csci51800.2020.00034

[4] Chandwani, A., Dey, S., & Mallik, A. (2020). Cybersecurity of Onboard Charging Systems for Electric Vehicles—Review, Challenges and Countermeasures. *IEEE Access*, *8*, 226982–226998. https://doi.org/10.1109/access.2020.3045367

[5] Lin, C., & Sangiovanni-Vincentelli, A. (2012). Cyber-Security for the Controller Area Network (CAN) Communication Protocol. *Cyber-Security for the Controller Area Network CAN Communication Protocol*. https://doi.org/10.1109/cybersecurity.2012.7

[6] Taylor, A., Leblanc, S., & Japkowicz, N. (2016). Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks. *Anomaly Detection in Automobile Control Network Data with Long Short-Term Memory Networks*. https://doi.org/10.1109/dsaa.2016.20

[7] Rathore, R. S., Hewage, C., Kaiwartya, O., & Lloret, J. (2022c). In-Vehicle Communication Cyber Security: challenges and solutions. *Sensors*, *22*(17), 6679. https://doi.org/10.3390/s22176679

[8] Zhang, H., Pan, Y., Lu, Z., Wang, J., & Liu, Z. (2021). A Cyber Security Evaluation Framework for In-Vehicle Electrical Control Units. *IEEE Access*, *9*, 149690–149706. https://doi.org/10.1109/access.2021.3124565

[9] Scalas, M., & Giacinto, G. (2019). Automotive Cybersecurity: Foundations for Next-Generation Vehicles. *Automotive Cybersecurity Foundations for Next-Generation Vehicles*. https://doi.org/10.1109/ictcs.2019.8923077 [10] A. Sui and G. Muehl "Security for Autonomous Vehicle Networks", 13 Novembre,2020

[11] Bozdal, M., Samie, M., Aslam, S., & Jennions, I. K. (2020b). Evaluation of CAN bus security challenges. *Sensors*, *20*(8), 2364. https://doi.org/10.3390/s20082364

[12] Fakhfakh, F., Tounsi, M., & Mosbah, M. (2021). Cybersecurity attacks on CAN bus-based vehicles: a review and open challenges. *Library Hi Tech*, *40*(5), 1179–1203. https://doi.org/10.1108/lht-01-2021-0013

[13] Brewer, J. N., & Dimitoglou, G. (2019). Evaluation of Attack Vectors and Risks in Automobiles and Road Infrastructure. *Evaluation of Attack Vectors and Risks in Automobiles and Road Infrastructure*. https://doi.org/10.1109/csci49370.2019.00021

[14] Takahashi, J., Aragane, Y., Miyazawa, T., Fuji, H., Yamashita, H., Hayakawa, K., Ukai, S., & Hayakawa, H. (2017). Automotive attacks and countermeasures on LIN-Bus. *Journal of Information Processing*, *25*(0), 220–228. https://doi.org/10.2197/ipsjjip.25.220

[15] Sommer, F., Dürrwang, J., & Kriesten, R. (2019). Survey and classification of automotive security attacks. *Information*, *10*(4), 148. https://doi.org/10.3390/info10040148

[16] Zhang, H., Pan, Y., Lu, Z., Wang, J., & Liu, Z. (2021b). A Cyber Security Evaluation Framework for In-Vehicle Electrical Control Units. *IEEE Access*, *9*, 149690–149706. https://doi.org/10.1109/access.2021.3124565

[17] Bozdal, M., Samie, M., & Jennions, I. K. (2018). A Survey on CAN Bus Protocol: Attacks, Challenges, and Potential Solutions. *A Survey on CAN Bus Protocol- Attacks, Challenges, and Potential Solutions*. https://doi.org/10.1109/iccecome.2018.8658720

[18] Aliwa, E., Rana, O., Perera, C., & Burnap, P. (2021). Cyberattacks and countermeasures for In-Vehicle networks. *ACM Computing Surveys*, *54*(1), 1–37. https://doi.org/10.1145/3431233

[19] Elkhail, A. A., Refat, R. U. D., Habre, R., Hafeez, A., Bacha, A., & Malik, H. (2021). Vehicle Security: A survey of security issues and vulnerabilities, malware attacks and defenses. *IEEE Access*, *9*, 162401–162437. https://doi.org/10.1109/access.2021.3130495

[20] Hubaux, J., Capkun, S., & Luo, J. (2004b). The security and privacy of smart vehicles. *IEEE Security & Privacy*, *2*(3), 49–55. https://doi.org/10.1109/msp.2004.26

[21] Rathore, R. S., Hewage, C., Kaiwartya, O., & Lloret, J. (2022d). In-Vehicle Communication Cyber Security: challenges and solutions. *Sensors*, *22*(17), 6679. https://doi.org/10.3390/s22176679

[22] Hubaux, J., Capkun, S., & Luo, J. (2004c). The security and privacy of smart vehicles. *IEEE Security & Privacy*, *2*(3), 49–55. https://doi.org/10.1109/msp.2004.26

[23] Zhang, H., Xu, M., Zhang, X., & Liu, Z. (2020). CANSEC: a practical In-Vehicle Controller area network security evaluation tool. *Sensors*, *20*(17), 4900. https://doi.org/10.3390/s20174900

[24] Choi, W., Joo, K., Jo, H. J., Park, M. C., & Lee, D. H. (2018). VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. *IEEE Transactions on Information Forensics and Security*, *13*(8), 2114–2129. https://doi.org/10.1109/tifs.2018.2812149

[25] Bozdal, M., Samie, M., Aslam, S., & Jennions, I. K. (2020). Evaluation of CAN bus security challenges. *Sensors*, *20*(8), 2364. https://doi.org/10.3390/s20082364

[26] Xu, R., Joshi, J., & Li, C. (2019b). CryptoNN: Training Neural Networks over Encrypted Data. *CryptoNN Training Neural Networks Over Encrypted Data"*. https://doi.org/10.1109/icdcs.2019.00121

[28] "ProceedingsofIEEE19"

[29] Wang, Q., Yan, Z., Lu, X., Wang, Z., Qin, Z., & Ren, K. (2016b). Real-time and Spatio-temporal Crowd-sourced Social Network Data Publishing with Differential Privacy. *IEEE*

Transactions on Dependable and Secure Computing, 1. https://doi.org/10.1109/tdsc.2016.2599873

[30] Amirtahmasebi, K., & Jalalinia, S. R. (2010). *Vehicular Networks – security, vulnerabilities, and countermeasures*. https://odr.chalmers.se/bitstream/20.500.12380/123778/1/123778.pdf

[31] Engoulou, R. G., Bellaïche, M., Pierre, S., & Quintero, A. (2014). VANET security surveys. *Computer Communications*, *44*, 1–13. https://doi.org/10.1016/j.comcom.2014.02.020

[32] Zhang, H., Shi, Y., Wang, J., & Chen, H. (2018). A new Delay-Compensation Scheme for networked control systems in controller area networks. *IEEE Transactions on Industrial Electronics*, *65*(9), 7239–7247. https://doi.org/10.1109/tie.2018.2795574

[33] Park, I., & Sunwoo, M. (2011). FlexRay Network Parameter Optimization Method for Automotive Applications. *IEEE Transactions on Industrial Electronics*, *58*(4), 1449–1459. https://doi.org/10.1109/tie.2010.2049713

[34] Kim, J. H., Seo, S., Hai, N. T., Cheon, B. M., Lee, Y. S., & Jeon, J. W. (2015). Gateway framework for In-Vehicle networks based on CAN, FlexRay, and Ethernet. *IEEE Transactions on Vehicular Technology*, *64*(10), 4472–4486. https://doi.org/10.1109/tvt.2014.2371470

[35] Kelkar, S., & Kamal, R. (2014). Implementation of data reduction technique in Adaptive Fault Diagnosis Algorithm for Controller Area Network. *Implementation of Data Reduction Technique in Adaptive Fault Diagnosis Algorithm for Controller Area Network*. https://doi.org/10.1109/cscita.2014.6839252

[36] Oberti, F., Sánchez, E., Savino, A., Parisi, F., Brero, M., & Di Carlo, S. (2022). LIN-MM: Multiplexed Message Authentication Code for Local Interconnect Network message authentication in road vehicles. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2206.02602

[37] Kim, B., & Park, K. (2009). Probabilistic delay model of dynamic message frame in flexray protocol. *IEEE Transactions on Consumer Electronics*, *55*(1), 77–82. https://doi.org/10.1109/tce.2009.4814417

[38] Sami, M., Ibarra, M., Esparza, A. C., Al-Jufout, S., Aliasgari, M., & Mozumdar, M. (2020). Rapid, Multi-vehicle and Feed-forward Neural Network based Intrusion Detection System for Controller Area Network Bus. *Rapid Multi-vehicle and Feed-forward Neural Network Based Intrusion Detection System for Controller Area Network Bus*. https://doi.org/10.1109/igessc50231.2020.9285088

[39] Yu, Z., Liu, Y., Xie, G., Li, R., Liu, S., & Yang, L. T. (2023). TCE-IDS: Time interval Conditional entropy- based Intrusion Detection System for automotive controller area networks. *IEEE Transactions on Industrial Informatics*, *19*(2), 1185–1195. https://doi.org/10.1109/tii.2022.3202539

[40] Sunny, J., Sankaran, S., & Saraswat, V. (2020). A Hybrid Approach for Fast Anomaly Detection in Controller Area Networks. *A Hybrid Approach for Fast Anomaly Detection in Controller Area Networks*. https://doi.org/10.1109/ants50601.2020.9342791

[41] Wu, Y., Fu, L., Xu, Y., Ma, F., & Lu, Y. (2018). Controller Area Network Modeling and Its Application in Cyber-Physical Power System Co-Simulation. *Controller Area Network Modeling and Its Application in Cyber-Physical Power System Co-Simulation*. https://doi.org/10.23919/chicc.2018.8483930

[42] Kang, S., Seong, J., & Lee, M. (2018). Controller area network with flexible data rate transmitter design with low electromagnetic emission. *IEEE Transactions on Vehicular Technology*, *67*(8), 7290–7298. https://doi.org/10.1109/tvt.2018.2832659

[43] Othman, H., Aji, Y., Fakhreddin, F., & Al-Ali, A. R. (2006). Controller Area Networks: Evolution and Applications. *Controller Area Networks Evolution and Applications*. https://doi.org/10.1109/ictta.2006.1684909

[44] Ahn, B., Park, B., Ki, Y., Jeong, G., Ahn, H., & Kim, D. (2006). Development of a Controller Area Network Interface Unit and Its Application to a Fuel Cell Hybrid Electric Vehicle. *Development of a Controller Area Network Interface Unit and Its Application to a Fuel Cell Hybrid Electric Vehicle*. https://doi.org/10.1109/sice.2006.315774

[45] Novák, J. (2009). Flexible approach to the Controller Area Networks test and evaluation. *Flexible Approach to the Controller Area Networks Test and Evaluation*. https://doi.org/10.1109/idaacs.2009.5343029

[46] Prasad, B., Gao, R., Jing-Jou, T., & Jhen, H. C. (2022). LIN Bus Based Touchpad System for Smart Vehicle Cabin. *LIN Bus Based Touchpad System for Smart Vehicle Cabin*. https://doi.org/10.1109/icasi55125.2022.9774481

[47] Fax, J., & Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, *49*(9), 1465–1476. https://doi.org/10.1109/tac.2004.834433

[48] Woo, S., Jo, H. J., & Lee, D. H. (2014). A practical wireless attack on the connected car and security protocol for In-Vehicle CAN. *IEEE Transactions on Intelligent Transportation Systems*, 1–14. https://doi.org/10.1109/tits.2014.2351612

[49] Lee, Y., Woo, S., Lee, J., Song, Y., Moon, H., & Lee, D. H. (2019). Enhanced Android App-Repackaging attack on In-Vehicle Network. *Wireless Communications and Mobile Computing*, *2019*, 1–13. https://doi.org/10.1155/2019/5650245

[50] Hounsinou, S., Stidd, M., Ezeobi, U., Olufowobi, H., Nasri, M., & Bloom, G. (2021). Vulnerability of Controller Area Network to Schedule-Based Attacks. *Vulnerability of Controller Area Network to Schedule-Based Attacks*. https://doi.org/10.1109/rtss52674.2021.00051

[51] Thirumavalavasethurayar, P., & Ravi, T. (2021). Implementation of Replay Attack in Controller Area Network Bus using Universal Verification Methodology. *Implementation of Replay Attack in Controller Area Network Bus Using Universal Verification Methodology*. https://doi.org/10.1109/icais50930.2021.9395871

[52] Noureldeen, P., Azer, M. A., Refaat, A., & Alam, M. F. (2017). Replay attack on lightweight CAN authentication protocol. *Replay Attack on Lightweight CAN Authentication Protocol*. https://doi.org/10.1109/icces.2017.8275376

GOOGLE DRIVE
https://drive.google.com/drive/folders/1nMuopkeCDkhJPtdQKh9G_Bnz07LM5w3b?usp=sharing

GIT HUB
https://github.com/Deep6776/INSE6120-Fall2023-Project-Group10

Roles and Responsibilities

Team Member	Roles and Responsibility
Pratiksha (*40230412)*	-Overview of Automotive Networks -Brief discussion on the significance of security in automotive networks and Introduction to CAN -Importance and usage of Controller Area Network (CAN) and its alternatives (LIN, FlexRay)
Simaran (*40241517*) Snehpreet (*40254443*)	- Overview of CAN, LIN, and FlexRay networks - Comparison of functionalities and use cases - Security concerns and challenges in each network type
Deep (40231725), Devina (*40238009*), Saket (*40225128*)	-Specific attack vectors and methodologies (e.g., DoS attacks, spoofing, message manipulation) -Detailed discussion of notable attacks with examples -Summary of recent academic literature on attacks - Feasibility of attacks on these networks - Computational and communication resources needed by attackers - Dependencies on physical access or specific conditions for successful attacks
Rohan (*40255454*), Meet (*40239187*) Karanjot (*40220861*)	-Existing security measures and their effectiveness -Latest advancements in securing automotive networks - Recommendations and potential future directions for enhanced security
Pratiksha (*40230412*), Deep (40231725)	-Conclusion -Key takeaways from the surveyed literature