1. ***Existing security measures and their effectiveness***

**SECURE BOOT**

Secure Boot is not an exclusive concept limited to the automotive industry; on the contrary, it is widely supported by the majority of BIOS systems found in home personal computers. Serving as a Unified Extensible Firmware Interface (UEFI) mechanism, it effectively confines the booting process to only permit the execution of software equipped with valid and authorized signatures within the machine it operates on. During the manufacturing phase, the device is endowed with a set of databases containing keys and signatures by the manufacturer. Consequently, prior to initiating the booting sequence, the firmware undergoes the essential steps of being signed and subsequently having its signature verified. By implementing secure booting, a highly robust security measure is established, effectively thwarting any attempts to initiate the booting of malicious firmware. Nevertheless, it is worth noting that certain vulnerabilities have been identified in specific implementations, which have, on certain occasions, enabled the circumvention of secure booting, thereby undermining its status as an infallible solution.

**SECURE ACCESS SERVICE**

In the research conducted by Chris Valasek and Charlie Miller, they encountered a diagnostics authentication mechanism during the process of reverse engineering a 2010 Toyota Prius and a 2010 Ford Escape [MV13]. This particular mechanism is known as Security Access and is a component of the Unified Diagnostics Service (UDS) as defined in ISO 14229-1[Sta13]. The UDS serves as a standardized diagnostics communication protocol for ECUs within the automotive environment. It outlines a range of services that enable the retrieval of information regarding the functionality and state of the vehicle. One of these services is the Security Access. This service operates as a straightforward Challenge-Response protocol. Initially, the tester sends a message to the ECU requesting a "seed", which is then randomly generated by the ECU and sent back to the tester. Both the tester and the ECU are required to possess a cryptographically secure function and a shared key. By applying the key and the seed to the function, a response is generated that is difficult for an attacker to guess. The tester then sends this generated response to the ECU, which verifies if it is the correct response. If it is, the tester is granted the ability to carry out the relevant diagnostic actions. Unfortunately, the standard itself lacks specific information regarding the function and keys, as well as how the seed should be generated. Consequently, this leaves ample room for potential errors in the implementation of the Security Access service. Different ECUs have different implementations, leading to various issues. For instance, in the case of the 2010 Ford Escape, the Parking Aid Module ECU consistently generated the same seed. This behavior, combined with the fact that the response is only 24 bits long, allows an attacker to easily brute force or replay the response from a legitimate Security Access session. As for the other modules in the Ford Escape, the ECUs would always generate different seeds. The researchers managed to reverse engineer the Ford Integrated Diagnostic Tool software and successfully extract the keys. In the case of the Toyota Prius, most diagnostic actions do not require authorization through the Security Access. For the ECUs that do require it, they generate different seeds after every 10 failed authentication attempts or whenever the car is started. This makes it challenging to conduct an online brute force attack. However, through reverse engineering a Toyota tool called the Toyota Calibration Update Wizard, the researchers discovered that the function used to generate the response only utilizes two out of the four bytes in the seed. This significantly reduces the entropy of the entire system. In conclusion, the researchers were ultimately able to retrieve three keys.

**Secure Onboard Communications**

In an effort to address the absence of authentication in the Controller Area Network (CAN), the AUTOSAR community has put forth a strategy to directly combat the injection of malicious traffic into the CAN bus. The Secure Onboard Communications (SecOC) module is responsible for introducing CAN frame authentication within the automotive environment by incorporating signatures into the in-vehicle communications [SECb]. This module operates with both symmetrical and asymmetrical cryptography. In order to authenticate the frame, SecOC assumes that key management and exchange have already been attended to. The technique involves simply appending a signature to the protected data unit (PDU) within the CAN frame. To ensure that replay attacks are rendered unfeasible, this strategy necessitates the inclusion of a freshness value, which maintains a certain level of uniqueness for the signatures. For instance, in symmetrical mode, for a device to send a message to another device, the sender calculates a message authentication code (MAC) over the input data and the freshness value, utilizing a shared key, and subsequently appends the calculated signature along with the freshness value. The receiver then verifies the correctness of the freshness value and recalculates the MAC based on the received message. If the recalculated MAC is correct, the receiver accepts the message. Despite its simplicity, the implementation of this system in CAN poses the challenge of accommodating all these elements within the regular CAN frame. With the regular CAN frame, it becomes necessary to truncate the 128-bit MAC into a 27-bit value, which leaves it susceptible to brute force attacks. There are two proposed methods for maintaining the freshness of signatures: using either time stamps or monotonically increasing counters. When employing counters, each device must store the counter values for each type of message, requiring a separate counter for every unique message ID in the CAN. It is worth noting that SecOC does not specify the procedures for key management and synchronization of freshness values, which could potentially lead to issues, as desynchronization may occur even in the absence of an adversary [SECa].

The task of establishing secure key exchange between two devices in a bus environment remains a significant challenge within the Controller Area Network (CAN) protocol. To address this challenge, the Plug and Secure Key Establishment (PnS) protocol has been proposed as an elegant and cost-effective solution. The PnS protocol leverages a physical property of the bus to define a symmetric key between the two devices [ML15]. Specifically, when two devices simultaneously transmit a bit, the dominant bit (zero) overwrites the other and becomes the bit observed by the other nodes on the bus.

Consider a scenario where two devices, namely Alice and Bob, share the same CAN bus and aim to establish a shared secret value. Following the PnS key establishment protocol, both Alice and Bob generate a random string of bits. For each bit in the string, they append the inverse of that bit immediately after it. For instance, if the secret bit string is "0 1 0", it becomes "01 10 01" after the manipulation. Subsequently, both parties transmit their generated strings simultaneously. Due to the aforementioned physical property, the bits will mix and overwrite each other during transmission. Alice and Bob then observe which bit tuples contain a recessive bit (1) and discard those tuples from the final secret. Finally, Alice and Bob revert back to the original secret, excluding the bits that were removed in the previous step. Consequently, both devices possess a secret bit string and are aware that the other device possesses the same secret, albeit with opposite bits.

At this level of abstraction, a malicious device can only discern that the keys are the opposite of each other. Despite its simplicity and cost-effectiveness, the PnS key establishment method lacks an implementation model that aligns with the CAN standard. The protocol itself remains reliable, even considering its inherent randomness, which may lead to some scenarios where certain keys are smaller than others. However, the protocol relies on the physical property to ensure the confidentiality of the initially transmitted values. This reliance on a physical property exposes the protocol to potential side channel attacks that exploit differences in bit timing or variations in voltages. Such attacks could potentially recover entire keys. Nonetheless, it's important to mention that these concerns are rather insignificant since the protocol recommends using a fresh random key for every occurrence. Additionally, attacks of this nature require direct physical access to the CAN bus, indicating that the physical security has already been compromised and most security mechanisms are unable to safeguard against a physical attacker.

**Firewall Gateway**

Another potential solution for enhancing the security of in-vehicle communication is the implementation of a firewall mechanism within network gateways. This firewall mechanism can play a crucial role in ensuring the authentication and authorization between Electronic Control Units (ECUs) by utilizing message authentication codes (MAC) or digital signatures. By employing these authentication methods, the firewall rules can be derived from the authorizations specified in each ECU's certificate. Conversely, in the absence of MACs or digital signatures between ECUs, the firewall rules can be individually defined based on the vehicular subnet authorizations. Consequently, the firewall rules will only permit the transmission of messages from valid and authentic ECUs, effectively safeguarding the in-vehicle bus system. Moreover, a potential approach to defining firewall rules is to restrict the access level of different types of networks to various parts of the bus system. For instance, it is imperative to ensure that ECUs from less critical networks like LIN or MOST are unable to transmit messages into higher safety-relevant and more critical bus systems such as CAN or FlexRay. As previously stated, one more technique to boost communication and system security is the integration of Intrusion Detection Systems (IDS). By incorporating an IDS into a central gateway, the exchanged information can be thoroughly inspected and analyzed to detect any potential security-critical anomalies. In the event of communication misbehavior, the IDS can respond appropriately by initiating countermeasures that range from issuing simple intrusion warning messages to even deactivating the misbehaving ECU..

**Honeypots**

The wireless communication gateway of the vehicle will serve as an entry point that is susceptible to attacks from malicious individuals. The inclusion of wireless access to a vehicle provides attackers with the opportunity to initiate cyber attacks that directly target the network within the vehicle. Given that the most crucial and significant operations performed by a vehicle are controlled by the in-vehicle network, it becomes imperative to prioritize the security of the in-vehicle communication system and ensure that it remains unaffected by such attacks. An effective approach towards achieving this objective involves acquiring a comprehensive understanding of the behavior, techniques, and strategies employed by attackers, as this knowledge can aid in the development of highly efficient security measures for in-vehicle communication systems. In this regard, the utilization of a honeypot can prove to be advantageous in terms of gathering vital information pertaining to attackers. Honeypots operate on the fundamental principle of presenting themselves as vulnerable and easily accessible targets in order to entice attackers, thereby enabling the analysis of their behavior and techniques. These honeypots function as tools for preventing and detecting malicious attacks by examining the unauthorized and harmful activities carried out by attackers within the system. In the context of desktop IT systems, a honeypot typically consists of a regular computer that runs specialized software designed to mimic valuable and easily exploitable targets, thereby attracting potential attackers. Once an attacker successfully breaches the system, the administrator gains valuable insights into the techniques and behavior employed by the attacker, facilitating the development of specific security measures to safeguard the system and prevent future attacks. By making certain adaptations, honeypots can also be implemented within the automotive domain to identify potential threats targeting in-vehicle networks. In the automotive domain, a honeypot must exhibit a high level of realism in order to effectively attract genuine attackers, while also being isolated from the main functionality of the system to prevent any interference with normal vehicle operations and to ensure the safety of passengers. During the data collection phase, the vehicle equipped with the honeypot will traverse a predetermined area, recording all data transmitted to its gateway and thereby affecting the in-vehicle network. Subsequently, through the analysis and processing of the collected data, patterns of attack behavior and potential attack scenarios can be identified. Additionally, reverse engineering techniques can be employed to investigate the commands used in attacks and determine their impact on the system. The practical knowledge gained from these investigations will prove invaluable in enhancing the security of in-vehicle communication systems through the implementation of future designs and strategies.

**Latest advancements in securing automotive networks**

**Quantum Key Distribution (QKD)**

Quantum Key Generation and Quantum Key Distribution are two fundamental technologies that form the basis of quantum-safe security. These technologies, alongside the continuous development and utilization of quantum-safe algorithms, will serve as the underlying infrastructure for the future generation of secure V2X communication networks.

The security of any cryptographic system relies heavily on the security of its keys. These keys play a vital role in ensuring the confidentiality and integrity of data in motion. Furthermore, they facilitate authentication, non-repudiation, and access control for all parties involved in the data exchange process. In order for keys to be truly secure, they must possess unique characteristics, such as being genuinely random, and must be stored and distributed in a secure manner.

Quantum Key Distribution (QKD) is commonly employed to safeguard data exchange within the back-end infrastructure, which is often a prime target for malicious attackers due to the substantial amount of sensitive data that can be compromised in a single breach.

The combination of Quantum Random Number Generation (QRNG) and Quantum Key Distribution (QKD) guarantees the generation of keys with high entropy and ensures their secure distribution across the network. By encrypting vehicular communications at such a high level, they become impervious to current attacks as well as "hack now, crack later" attacks. Furthermore, these encryption techniques can be seamlessly integrated with quantum-resistant algorithms (QRAs) without requiring significant modifications to software or hardware components, once such algorithms become available.**Quantum-**

**Safe Security for Automotive Ecosystems**

From the perspective of ensuring security within the automotive ecosystem, it is crucial to prioritize the implementation of quantum-safe cryptographic solutions in order to future-proof vehicle systems, vehicle-to-vehicle networks, vehicle-to-infrastructure networks, and back-end systems. This is essential considering the potential threats that may arise in the future.

To address this concern, IDQ offers a comprehensive range of quantum-safe security solutions that are specifically tailored to safeguard data in motion across the vehicle-to-everything (V2X) ecosystems against both existing and emerging threats. These solutions are meticulously designed to provide robust protection and ensure the integrity of data transmission within the automotive domain. By leveraging quantum-safe cryptographic techniques, IDQ's security solutions offer a high level of resilience against potential attacks and vulnerabilities, making them a reliable choice for securing the sensitive information shared within V2X ecosystems.

By adopting IDQ's quantum-safe security solutions, automotive stakeholders can not only mitigate the risks associated with data breaches and unauthorized access but also establish a solid foundation for the future of connected vehicles. As the automotive industry continues to evolve and incorporate advanced technologies, it becomes imperative to proactively address security concerns and deploy solutions that can withstand the test of time. IDQ's quantum-safe security solutions fulfill this requirement by effectively countering the potential threats posed by quantum computing, ensuring the continuity and reliability of secure communication within V2X ecosystems.

In conclusion, the incorporation of quantum-safe cryptographic solutions is of utmost importance in safeguarding the automotive ecosystem. By prioritizing the implementation of such solutions, automotive stakeholders can establish a secure framework that enables seamless and trustworthy communication between vehicle systems, vehicle-to-vehicle networks, vehicle-to-infrastructure networks, and back-end systems. IDQ's range of quantum-safe security solutions offers a reliable and future-proofed approach to securing data in motion within V2X ecosystems, providing the necessary protection against both current and emerging threats.

**Quantum Random Number Generation (QRNG)**

Strong key generation plays a pivotal role in ensuring that a third party cannot make educated guesses or deduce the security key. Consequently, the utilization of truly random numbers assumes utmost importance. Quantum Random Number Generation (QRNG) effectively serves the purpose by instantaneously fortifying existing cryptographic mechanisms and guaranteeing the resilience of novel quantum resistant algorithms.

The instantaneous fortification of the encryption keys safeguards communication with external applications and hardware, as well as secures communication within the cloud.

IDQ's Quantis QRNG chip seamlessly integrates into automotive HSMs, making it an ideal choice. It offers a transparent and robust mechanism for generating entropy, accompanied by a security proof. Moreover, it is AEC-Q100 certified and incorporates NIST 800-90A/B/C compliant DRBG post-processing.

The features and benefits of the Quantis QRNG chip are as follows:

1. Provably random, enabling the generation of quantum enhanced keys with maximum value.

2. Cost-effective, energy-efficient, and resilient to environmental factors.

3. Instantaneous activation, facilitating prompt utilization.

**Multi-Layered Security Framework**

The in-vehicle architecture of modern advanced vehicles is characterized by its complexity and sophistication. These vehicles are equipped with a wide range of highly sensitive sensors, electronic devices, computer systems, and other technologies. In order to ensure the security of these sophisticated systems, it is necessary to develop a coordinated and integrated cybersecurity framework that can effectively design solutions and minimize security risks. For the protection of in-vehicle networks from cyber threats, it is imperative to adopt a multi-layered approach. This is because hackers can infiltrate vehicles through both cyber means and physical access. The accompanying figure depicts the multi-layered security framework for in-vehicle networks, along with the corresponding automotive protocols. It is important to note that this framework is based on a thorough analysis of existing literature and represents a significant contribution to the field. Each layer of the framework is designed to address a specific type of security threat in the context of in-vehicle communication networks. The development of this multi-layered security framework is grounded in a comprehensive understanding of existing approaches and informed by critical analysis. It is evident that the current literature lacks a cohesive and integrated multi-layer security framework for in-vehicle networks. Within this framework, various security issues related to in-vehicle networks are addressed in specific layers, including ECU-boot level security, ECU to ECU communication, domains/sub-domains communication, application software updates or versioning issues, gateway security controller, and security issues related to vehicle-to-outside services. It is important to emphasize that each layer of the proposed multi-layered security framework serves a specific function and addresses particular security threats. This is why the framework is organized into six layers, starting from ECU-boot level security and progressing to in-vehicle network-to-vehicle outside services security issues. The Secure Gateway/Domain Controller layer focuses on the security of the gateway, while the secure external communication layer prioritizes the protection of communication channels. In terms of the functional roles of automotive protocols, they all play a role in all layers due to the complex nature of modern in-vehicular systems. However, a careful analysis of the effectiveness of each automotive protocol in providing functionality to specific domains/sub-domains of the in-vehicular system enables us to determine which protocol is most suitable for each layer. Therefore, when developing the framework, we selected protocols based on their effectiveness and suitability for each layer.

**A diagram of a computer network

Description automatically generated with medium confidence**

The Control Platform Layer is designed to enhance the security of the platform. It serves as the central hub for securing the in-vehicle network of smart intelligent vehicles against malicious threats. The focus of the control layer is to provide security solutions for protecting the firmware of the Electronic Control Units (ECUs), enhancing security during the booting process, and ensuring the security of the hardware modules (HM). In this mechanism, the Original Equipment Manufacturer (OEM) trusted server and the Trusted Platform Module (TPM) chip at the ECU play a crucial role in maintaining a secure connection. This connection allows for the exchange of encrypted firmware update image files and data, thereby enhancing security during the booting process and for the ECU firmware. It is essential to frequently update the ECU firmware to counter the various attacks prevalent in the current advanced technological automotive era. Furthermore, the authenticity of the ECU firmware update can be verified through mechanisms such as signing or secure flashing.

The Secure ECU to ECU Communication in In-Vehicle Network layer ensures the delivery of messages with integrity proof between ECUs. Hackers are targeting a significant amount of sensitive information that is generated by the ECUs. The security mechanism in place should guarantee the reliability, confidentiality, and integrity of this sensitive information. The modern in-vehicle network should incorporate a hardware security module (HSM) chip that functions as a security controller. This security controller consists of three main modules. The first module is responsible for authenticating the ECU's identification and verifying its state. The second module handles the encryption and decryption of the messages exchanged between ECUs. The third module controls the secure flash storage. All messages from the ECUs are directed to the HSM chip, which serves as the security controller. The security controller analyzes the messages and identifies the destination ECU. Based on the state of the destination ECU and the security properties of the message, it determines whether the message should be forwarded to the destination ECU or not. Reliability and Privacy of Communication between Domains/Sub-Domains in In-Vehicle Network This layer ensures the reliability and privacy of communication between domains/sub domains in the vehicle. A vehicle domains/sub-domain reflects the grouping of functions and systems with respect to specified areas, for example telematics domain, infotainment domain, chassis domain, powertrain domain, sensor domain, and body domain, etc. This layer protects the domains/subdomains by using four security mechanisms as follows: # Message authentication system: Cryptographic certificate is used to ensure an authenticated sender and message integrity. This certificate is added to all messages in the network. # Encryption: Messages are encrypted inside the vehicles to guard against loss of data and identity theft since messages are distributed with several different ECUs. # Detection of intrusion: The corrective mechanism should use the cryptographic accelerators and security subsystems integrated with the microcontroller to guard against threats. # Validation at ECU level: In the vehicle network, ECU’s validity is checked first at the start of the engine and afterwards at specified time intervals

All application software should ensure the qualities of reliability and authenticity. Hackers exploit the software download/update features to launch attacks. When security vulnerabilities are detected, the secure mechanism promptly updates the vehicle software. Typically, the application software is developed by a third-party manufacturer, and a verified software file image is sent to the trusted server of the original equipment manufacturer (OEM) for uploading. The software update manager module, located at the OEM trusted server, notifies the vehicle owner of the availability of an updated software version. Upon confirmation from the vehicle owner, the updated software version is installed in a secure in-vehicle network storage area. Throughout this entire process, the security of the channels is of utmost concern. Various techniques, such as hash functions, digital signatures, and blockchain technologies, exist to enhance channel security. Furthermore, modern advanced microcontrollers offer features like real-time integrity testing to verify the trustworthiness of the encryption mechanism.

The layer of secure gateway/domain controller ensures the security of the gateway/domain controller. It is imperative for modern critical gateways to possess advanced security mechanisms like advanced key management schemes/firewalls and intrusion detection schemes. The primary responsibility of the gateway is to maintain the secure configuration of the system. The gateway implements context-aware routing, whereby it verifies the validity of messages and permits the transfer of all valid messages to the corresponding destination through a series of complex controls. Automotive manufacturers are actively working on designing robust and secure gateway controller modules. The secure hardware extension (SHE) chip, which is part of the secure gateway controller module, stores the cryptographic credentials and protects them from hackers. The cryptographic credentials inside the SHE chip are managed by the public key infrastructures (PKI) of various communication service providers.

This layer ensures secure communication from the in-vehicle network to various services outside the vehicle. It provides authentication and message validation functionality to protect message integrity and safeguard communication channels from data manipulation and theft. As a result of these features, vehicle-to-everything communication, telematics, and other related functions are secured. All communications from the in-vehicle communication stack to the roadside unit (RSU) must be established through a trusted communication authority (TCA) to ensure authentication and message validation. Certificates used during communication between the in-vehicle communication stack and TCA, as well as between TCA and RSU, should be regularly changed to maintain the integrity of the sender. Additionally, the integrity of the message content can be protected using digital signatures.

**Recommendations and potential future directions for enhanced security**

Fully autonomous driving, which has the capability to operate on any type of road without any human intervention, has been regarded as the goal of autonomous vehicles that would profoundly transform the automotive industry and enhance the transportation experience in our daily lives. Recently, the achievements of self-driving systems utilizing deep learning algorithms have unveiled a practical and highly promising path towards attaining this ultimate objective. In essence, such a system comprises a well-trained machine learning model and numerous sophisticated sensors. The trained model acts as the cognitive center for the vehicle, enabling it to independently perceive, interpret, and make sound driving decisions. While it is captivating and convenient to entrust full control to the vehicle itself, there is also the possibility of catastrophic incidents if the self-driving system malfunctions. Furthermore, updating the self-driving system necessitates the acquisition of new training data from the vehicle, which could potentially expose sensitive information about daily routines and other private details. This section initially presents an overview and analysis of the novel and severe threats that may arise in future generations of autonomous vehicles, specifically fully automated self-driving vehicles. Subsequently, it offers potential defense strategies to enhance the safety of fully automated self-driving vehicles. A. New Security Threats The following section outlines the new security threats that exist in fully automated self-driving vehicles.

1) Errors in the trained model: Fully autonomous self-driving vehicles heavily rely on a perception system based on deep learning models to accurately identify objects and navigate independently. However, the presence of algorithmic flaws or model errors can lead to misclassification of objects within the perception system, resulting in potentially fatal car accidents. A recent incident involving an Uber self-driving vehicle serves as an example, where the vehicle misidentified a pedestrian as something else and failed to apply the brakes in a timely manner, leading to a collision. Thus, the primary concern in the context of fully automated self-driving vehicles is the existence of errors in the implemented deep learning model. Ensuring the robustness and absence of bugs in the trained model for object identification and classification is of utmost importance in a safety-critical system like this.

2) Adversarial Instances: In addition to the errors present in the trained deep learning models, misclassification can also occur due to deliberately crafted adversarial inputs. This new form of threat surpasses inherent model bugs and errors, as it allows external attackers to manipulate the behavior of self-driving vehicles by introducing adversarial inputs. A recent study demonstrated this by showing how an attacker can deceive a self-driving vehicle by strategically placing misleading signs on the roadside, causing the trained deep learning model to misclassify the signs and drive recklessly. Consequently, the presence of adversarial instances poses a severe threat, with potentially life-threatening consequences if left unaddressed. Furthermore, in the case of blackbox attack models where no information about the target deep learning model's parameters is required, the threat of adversarial instances becomes even more pronounced for self-driving vehicles. In this scenario, an attacker could train an adversarial network to generate more advanced adversarial instances, making the defense against such instances more challenging.

3) The privacy of model training is a crucial aspect when it comes to developing accurate deep learning models for self-driving vehicles. To achieve this accuracy, a substantial dataset of road images or real driving videos is required as learning inputs. Thus, it is crucial to persistently collect learning inputs from autonomous vehicles to improve the strength and precision of the deep learning model in real-life situations. However, the current infrastructure for model training is predominantly centralized, meaning that the input data from autonomous vehicles is transferred to a central server in a transparent manner. Given that the contributed learning dataset is closely intertwined with individuals' daily lives, it has the potential to expose sensitive information such as routines and locations. Furthermore, recent research has demonstrated that the trained deep learning models can inadvertently disclose sensitive information about the data contributors. Specifically, an attacker can exploit the model's ability to memorize unique or rare sequences within the learning inputs and extract valuable information from the trained models. Hence, thorough investigation is necessary to devise strategies for data collection in model training while effectively preventing privacy breaches prior to the deployment of self-driving vehicles in real-world scenarios.

4) The execution of models in self-driving systems presents its fair share of threats. To accomplish lightweight and efficient deep learning, groundbreaking hardware designs such as TPU, GPU, ASIC, and FPGA are incorporated into autonomous vehicles. Existing systems typically incorporate a central operating component within the self-driving vehicle to facilitate seamless control of the hardware, without any interference. However, like in-vehicle systems, this central operating component is susceptible to malicious attacks, such as malware injections, which compromises the reliability of model executions. Accordingly, it is imperative to consider this operating system as an untrusted environment, with a wide attack surface that can be easily exploited by attackers. Therefore, it is crucial to implement advanced defense mechanisms to ensure the integrity of model executions in self-driving vehicles.

Defense Strategies

We briefly discuss the strategies for defending against the security threats in fully automated self-driving vehicles.

1. Enhancement of DNN Robustness: To enhance the robustness and mitigate errors of deep learning models, a comprehensive testing approach can be employed for the trained models.

The current testing methods for trained models in the context of self-driving vehicles primarily involve either:

1) assessing and analyzing recognition errors within a newly inputted learning dataset, or

2) conducting real driving tests on the road and paying attention to disengagements, which refer to incidents where the self-driving vehicle fails to make decisions.

In the future, there is a potential for the testing procedures of deep learning models to become more automated. Upon detecting an error, the system can automatically retrain the model to improve its accuracy. Additionally, the testing can be extended to real-time, allowing for continuous monitoring of errors during model execution, and automatic patching to further enhance driving safety.

2) Defense against Adversarial Examples: To confront adversarial examples that can manipulate the behavior of deep learning models, there are several possible defense strategies. One approach is to preprocess or filter the input data in order to detect and eliminate adversarial examples prior to execution. For instance, standard blurring techniques such as Gaussian blur can be utilized to enable the trained model to evade adversarial examples. Another useful defense strategy involves generating adversarial examples or detecting potential adversarial examples using data mining methods. Later on, the design might be retrained by employing these generated adversarial examples to enhance its resilience. Moreover, efforts can be made to enhance the interpretability of the underlying deep learning models. This would enable close monitoring of the model execution procedures, facilitating timely detection of incorrect driving decisions and prevention of fatal accidents.

3) Preservation of Data Privacy: To ensure the privacy of contributed training data, one viable strategy is to leverage the emerging federated learning architecture for training and updating the deep learning model. With this privacy-enhanced architecture, the sensitive inputs used for model training remain within the self-driving vehicles, with only model parameter updates being transmitted to the server for convergence and updating of the model. Consequently, the private training data from all self-driving vehicles can be safeguarded during the model training and updating process. Specific configurations can be implemented to minimize memorization during training in order to prevent data leakage. In particular, one potential defense strategy against memorization involves carefully adding chosen noise to each gradient update during learning, thereby ensuring that the trained models maintain differential privacy. This approach effectively conceals the occurrence of certain private information within the trained models, thwarting any attempts by attackers to extract such information through model memorization.