

# Enhancing Secure Communication Protocol in ADAS system

MSc Research Project  
Masters in Cybersecurity

SHANMUGASUNDAR RAMESH

Student ID: X22171649

School of Computing  
National College of Ireland

Supervisor: MICHAEL PANTRIDGE

MSc Project Submission Sheet

School of Computing

<b>Student Name:</b>	Shanmugasundar Ramesh		
<b>Student ID:</b>	22171649		
<b>Programme:</b>	Masters in Cybersecurity	<b>Year:</b>	2023
<b>Module:</b>	MSc Research Project		
<b>Supervisor:</b>	Michael Pantridge		
<b>Submission Due Date:</b>	14/12/2023,31/01/2024		
<b>Project Title:</b>	ENHANCING SECURE COMMUNICATIONS PROTOCOL IN ADAS (ADVANCED DRIVING ASSISTANCE SYSTEM) SYSTEM		
<b>Word Count:</b>	<b>7303 Page Count 30</b>		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

<b>Signature:</b>	SHANMUGASUNDAR RAMESH
<b>Date:</b>	31/01/2024

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# **ENHANCING SECURE COMMUNICATIONS PROTOCOL IN ADAS (ADVANCED DRIVING ASSISTANCE SYSTEM) SYSTEM**

## **Abstract**

The objective of this study is to improve the secure communications protocol in Advanced Driver Assistance Systems (ADAS) by confronting the difficulties caused by the sensitive and constantly changing character of information in autonomous driving scenarios. The suggested method combines blockchain technology to create an encrypted proof of authentication, Advanced Encryption Standard in Galois/Counter Mode (AES-GCM) for reliable information encryption, and Multi Input Data Merging (MIDM) for comprehensive information synthesizing. A key component in integrating various data streams produced by sensors, cameras, and other ADAS components is the Multi Input Data Merging (MIDM) approach. Using the intelligent integration of various information, MIDM provides a comprehensive and instantaneous comprehension of the vehicle's environment. By offering a dynamic and adaptive encryption solution suitable to the various properties of numerous kinds of information inside the ADAS environment, the incorporation of AES-GCM as the encryption mechanism improves the security of data transmission. Moreover, a decentralized and impenetrable structure for authentication evidence is introduced by the use of blockchain technology. The blockchain provides an immutable and transparent ledger by securely recording every transaction or authentication event. This offers a distributed consensus technique for verifying the validity of activities inside the ADAS system in addition to ensuring the integrity of authentication procedures. In order to handle the challenges of data integration, encryption, and authentication, this study suggests an all-encompassing and flexible security architecture for ADAS. The goal of the beneficial combination of blockchain, AES-GCM, and MIDM technology is to substantially improve the secure communications protocol, consequently enhancing the effectiveness and security of autonomous driving systems.

**Keywords:** Advanced Driver Assistance Systems (ADAS), Advanced Encryption Standard (AES), Galois/Counter Mode (GCM), Encryption, Decryption, Autonomous Driving System

## 1. Introduction

The automobile industry has undergone a fundamental change with the introduction of Autonomous Driving Systems (ADS), which has resulted in a new age when vehicles are outfitted with cutting-edge technologies to drive themselves [1]. The core of this revolution is the Advanced Driving Assistance Systems, an extensive range of cameras, sensors, and communication units that allow automobiles to sense their environment while making judgments in actual time. Strong and secure protocols for communication are essential for ADAS systems as the automobile industry develops [2].

Communication that is secure is not only a technological difficulty in the framework of ADAS it is also a vital component in guaranteeing the resilience, safety, and dependability of autonomous cars. In addition to enabling efficient data interchange between different components, the communication protocols used in ADAS must ensure the integrity and security of the information that is being communicated [3]. This is particularly pertinent since malicious assaults or illegal access to private information might have serious repercussions and jeopardize the safety of passengers as well as other road users [4].

Despite the fact that ADAS systems have advanced significantly in the past few decades, the current communication channels might not be entirely capable of handling the changing threat situation [5]. Conventional protocols, including Controller Area Network (CAN), lacked strong security features since they were created with an emphasis on speed and real-time performance [6]. With autonomous cars becoming more interconnected and dependent on outside networks for information and updates, the shortcomings of existing communication protocols are becoming more noticeable [7].

Improving the security associated with communication protocols in ADAS systems is vital to overcome these issues [8]. To prevent future cyber-attacks, this requires implementing modern techniques for encryption, authentication methods, and detection systems for intrusions into practice [9]. In order to maintain the confidentiality and integrity of sensitive data pertaining to navigation, sensor components, and vehicle control orders, the improved secure communication

protocol must be able to protect information while it is in use as well as while it is being sent [10]. Access restrictions, digital certificate management, and key exchange should all be handled by a complete framework in conjunction with the incorporation of reliable communication protocols in ADAS systems [11]. By limiting access to and interaction with the ADAS network to only authorize organizations, this comprehensive method reduces the possibility of illegal manipulation or interference [12]. Enhancing secure communication protocols is a task that must be handled with caution as the automobile industry accelerates towards a future ruled by autonomous cars [13]. It is a sensitive undertaking to strike a balance between the constantly changing demands of ADAS systems and the security standards. Creative solutions are needed to ensure that system responsiveness is not compromised [14].

The key contributions of the study are given as follows,

- The suggested study presents a comprehensive and flexible security architecture designed to meet the specific challenges presented by ADAS. The research addresses the complex problems related to information integration, encryption, and authentication in autonomous vehicle circumstances by combining blockchain, AES-GCM, and MIDM technologies.
- The addition of MIDM remains out as a major addition. In order to integrate various information streams from sensors and ADAS components, MIDM is essential. This method provides a thorough and real-time awareness of the environment around the vehicle, enabling improved decision-making processes that are essential for the effectiveness and security of self-driving vehicles.
- AES-GCM integration advances research through providing a flexible and dynamic encryption solution. Since AES-GCM adapts encryption methods to the unique properties of various kinds of information throughout the ADAS environment, it improves the security of the transmission of information.
- The use of blockchain technology provides a decentralized, impenetrable structure for authenticating proof. Utilizing secure blockchain recording of authentication events, the study provides an irreversible and visible ledger.

## **2. Related Works**

A thorough investigation of several components of automotive systems security and measures against novel challenges is included in the literature study. The security issues raised by vehicles becoming more and more linked are covered first, particularly in consideration of the trends toward connected and self-driving vehicles. The necessity of security countermeasures is emphasized, with a focus on advanced simulated verification employing model-based simulation techniques, and Intrusion Detection Systems (IDS). The evaluation also covers linked autonomous driving, with a particular emphasis on security vulnerabilities in existing in-car networks such as the Controller Area Network (CAN). In order to address the issue of power consumption and possible FPGA deployment, a lightweight multi-attack quantized machine learning algorithm for CAN intrusion detection is presented. The third section discusses safety-critical assaults with particular reference to security issues with Autonomous Driver Assistance Systems (ADAS). In order to prevent indistinct command assaults, a unique technique combining camera perspectives and a suggested security system exploiting multimodal fusion utilizing neural networks are explored. An autonomous solution for information interchange between the ADAS and In-Vehicle Infotainment domains is presented in the concluding portion of the literature study. It emphasizes the significance of appropriate communication protocols and acknowledges potential constraints associated with certain Interface Definition Languages. The suggested communication management software component is found to require security concerns, which highlights the necessity for a conversation about the confidentiality and integrity of information in the automobile system.

## **2.1 Comparative Evaluation of Automotive Systems Security Methods**

A growing number of sensors are being used in cars to accurately perceive their surroundings because of developments like linked and self-driving vehicles. Vehicles communicate with Others Road Users using wireless connections, such as cellular ones, in order to facilitate autonomous driving features. The probability of cyberattacks is additionally enhanced by this increased connectedness. Original Equipment Manufacturers are required to integrate security interventions, such as IDS, in accordance with the United Nations Economic Commission for Europe regulatory authority. In order to confirm the efficacy of these safety precautions for ADAS features, factors like vehicle behavior, network connectivity, and environmental conditions must be considered. Electrical and electronic designs may be virtually verified earlier before the corresponding

hardware is fully created as a result of the implementation of model-based simulation techniques [15].

As automotive networking increases, novel innovations like ADAS and linked self-driving vehicles become possible, enhancing the dependability and safety of automobiles of the future. The growing accessibility to in-car features undermines vital features that depend upon antiquated in-car networks, such as the CAN, which lacks a verification or security system. Due to their capacity to generalize to novel vectors, security detection and protection techniques particularly those based on machine learning models have demonstrated optimistic findings regarding the identification of several techniques for attack in CAN. However, the majority of installations need significantly more power-intensive specialized computer units, such as GPUs, to do line-rate identification [16].

Many Electronics Controlling Units, which enable different efficient driving features like the ADAS, are installed in contemporary automobiles. The commonly utilized CAN protocol allows these ECUs to communicate with one another. However, CAN is open to assaults since it does not have any security mechanisms. Researchers have studied the use of machine learning IDS for CAN in order to prevent this. Significant identification errors are still present in the majority of IDSs in use today. In order to reduce the detection mistakes of machine learning-based systems for intrusion detection, the study suggests a novel filtering-based IDS. FIDS makes utilization of blacklists and whitelists produced from CAN databases [17].

## **2.2 Defense Mechanisms against Inaudible Threats in Autonomous Vehicles**

As a result of ADAS improving driving, a growing proportion of cars are becoming partially autonomous. Nevertheless, the cars are vulnerable to critical to safety errors and assaults due to the ADAS's growing complexity and connectedness. This study examines an ADAS's resistance to safety-critical assaults, which aim to compromise the control systems at key moments in various driving situations and result in collisions. Based on experimental data, we offer Context-Aware assaults that can cause hazards with an 83.4 percent effectiveness rate, more than ninety percent of which happen without any notice. These findings demonstrate how the ADAS is intolerant of assaults that compromise security, and they emphasize how crucial it is for individuals driving or autonomous recovery systems to respond quickly to prevent disasters [18].



An increasing number of people are becoming concerned about harmful assaults against self-driving cars. With the development of conversations in self-driving vehicle structures, in especially, assaults using indistinct speech instructions present a serious risk. It is still unclear how to effectively safeguard against these indistinct assaults. Previous studies explore the use of multidimensional fusion centered on deep learning for protection, but they do not take model ambiguity in reliability into account [19].

Language control mechanisms are being used more frequently as means of human-vehicle connection in vehicles as a result of current advances in self-driving technology. The technology is going to be accessible through ADAS and allows users to operate the vehicle with voice commands. Previous research has demonstrated how susceptible Alexa, Cortana, and Siri are to indistinct command assaults. This might be expanded to include ADAS in practical applications, and because of microphone irregularities, it is challenging to identify such an invisible directive risk. With the use of camera views, the research attempts to provide a more workable defense against inaudible command attacks in situations when ADAS are equipped with several sensors to sense their surroundings [20].

## **2.3 Evaluation and Considerations of Proposed Automotive Communication Solutions**

Automobile networks are intricate and made up of many parts that are created concurrently by several firms, depend on various technologies, and have various user experiences and security criteria. This results in redundant hardware assets and numerous implementations of identical capabilities across several vehicle sectors. It would turn out to be easier to design software and reduce the expense of vehicle components if different domains could communicate with one another. However, achieving inter-domain connection is not a simple task; it declares for choosing the appropriate communication protocol in addition to taking into account the wide range of application scenarios, the particular sector expertise of automotive designers, the diversity of software and platforms requirements, privacy and security concerns, etc. An automated method that has been verified for sharing information over SOME/IP across the In-Vehicle Information technology and ADAS domains is presented in this paper. The core of the produced solution leverages the most widely used connected definitions language for particular domains: FIDL, AIDL, and ARXML [21].

The division of the automobile systems into many domains is the outcome of the advancement of automotive software products. The ADAS domain provides strong and dependable software and hardware systems that control vehicle behavior, resulting in high computational capacity and safety. Conversely, In-Vehicle Information devices handle communication with the operator and offer a selection of entertaining and informational content. These domains are usually distinct from one another and have no shared resources. However, IVI's resource requirements might be significantly decreased, and driver-vehicle engagement could potentially be effectively enhanced through communication between the ADAS and IVI domains. For the purpose of to link the Android apps running on IVI with the SWCs that are currently in place on ADAS, this study provides an interaction management software element that is developed and put into operation on the ADAS end [22].

## **2.4 Blockchain in Autonomous vehicles**

New issues have emerged with the introduction of connected automobiles (CV) and the developing idea of CV as a Service (CVaaS), especially in terms of guaranteeing reliable and secure data transfer between cars in transportation networks. This study discusses how vehicular networks are evolving into interconnected and autonomous cars, highlighting how immediate sharing of data may lead to better experiences and less traffic. Nevertheless, communications integrity is threatened by the increase of malevolent users in the World of Vehicles. In order to address these issues, the research suggests using a blockchain architecture as a practical way to improve the security of smart sensors in automated vehicles while lowering the possibility of compromise by skilled hackers but the researchers also suggest that the blockchain architecture will not be sufficient as there is vulnerability in the blockchain architecture where there is integration of other methods to the block chain could be more secure. The suggested approach is validated by the research against a range of security criteria, such as spoofing user requests, device compromise, probabilistic authentication situations, and modifications to user ratings that have been saved. A 79% success rate when compared to current methods is noteworthy, confirming the effectiveness of the blockchain-based solution in resolving security issues in connected car environments [23].

## **3. Proposed MIDM-AES-GCM Framework**

The technique takes a multipronged approach to improving ADAS's secure communications protocol. The study combines the use of AES-GCM for strong data encryption, MIDM for thorough data synthesis, and blockchain technology to provide a safe proof of authenticity. MIDM provides a comprehensive picture of the vehicle's surroundings by assimilating various data streams from sensors and components in an intelligent manner. Ensuring safe data transfer, AES-GCM provides dynamic and adaptive encryption that may be adapted to a variety of data types. A decentralized, tamper-resistant authentication mechanism is introduced by blockchain. The proposed methodology aims to address challenges in data integration, encryption, and authentication, presenting a flexible and comprehensive security architecture for ADAS. The synergistic combination of blockchain, AES-GCM, and MIDM is designed to significantly enhance the secure communications protocol, contributing to the effectiveness and security of autonomous driving systems. It is depicted in Figure 1.

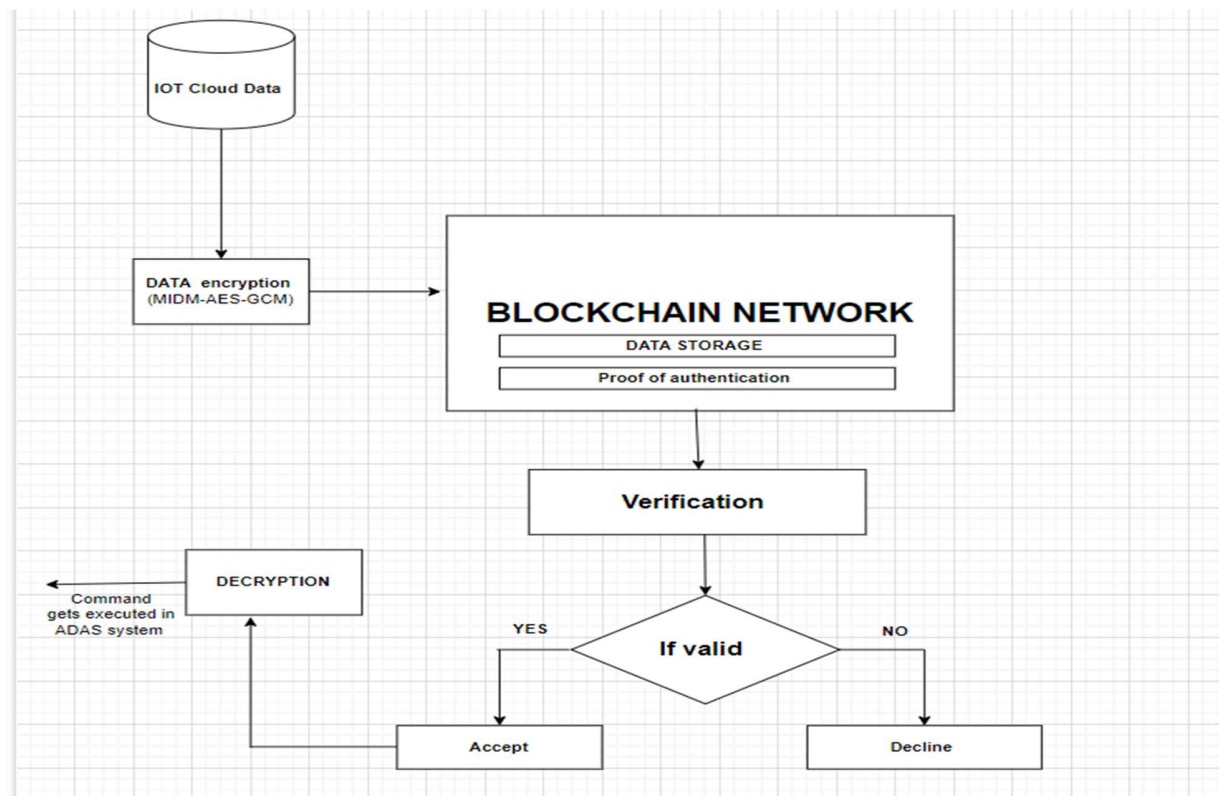


Figure 1: Proposed Methodology

### 3.1 Data Collection

The study utilized an Internet of Things (IoT) real-time data collecting methodology, including sensors and networked devices into the self-driving cars. In real-world operating settings, this IoT platform made it easier to continuously and instantaneously record dynamic information, such as camera feeds, GPS locations, and sensor readings. Protocols for communication in ADAS in real-world operating settings.

## **3.2 Employing MIDM-AES-GCM for Data Encryption**

Particularly significant in complicated systems like ADAS, MIDM in combination with AES-GCM creates a stable and flexible foundation for data encryption. By effectively integrating various data streams from sensors and other components in the ADAS environment, MIDM plays a crucial role. This method enables a thorough and instantaneous synthesis of data, generating a full comprehension of the environment around the vehicle. Through the integration of data from many sources, including as cameras and sensors, MIDM supports well-informed decision-making procedures that are essential to the effectiveness and safety of autonomous driving. The foundation for an encryption method that can adjust to the changing characteristics of information in ADAS is laid by this all-encompassing approach to data merging.

The integrity of the data encryption procedure is further improved by the incorporation of AES-GCM. AES-GCM is a symmetric key method that offers authentication and encryption. In the ADAS system, this guarantees the integrity and secrecy of data flows. Because AES-GCM is adaptive, encryption settings may be customized in accordance with the unique properties of various data kinds. This dynamic encryption approach is in complete alignment with the adaptability needed in autonomous driving situations, where a variety of data, from communication signals to sensor readings, necessitates a customized and strong security framework. When combined, MIDM and AES-GCM create a potent synergy that provides an all-encompassing and flexible solution for data encryption in ADAS, meeting the particular requirements of secure communications in autonomous driving systems.

### **3.2.1 Multi Input Data Merging (MIDM)**

A complex data integration method called MIDM is intended to bring together disparate data streams from different sources into a coherent and cohesive dataset. When it comes to ADAS, MIDM is essential for combining data from many sensors, cameras, radar, and other devices. ADAS

is able to create a full, real-time picture of the environment around the car because to this thorough merging process, which helps users make better decisions. By taking into account the distinctive qualities of every data stream, MIDM goes beyond basic aggregation and assures that the final dataset is loaded with context and complete. This method is especially useful for ADAS applications, where the safety and effectiveness of autonomous driving systems depend on the proper interpretation of a complicated environment.

The capacity of MIDM to adjust to the dynamic nature of information produced by various sensors and components accounts for much of its efficacy. Through the ingenious fusion of disparate inputs, MIDM helps ADAS to surmount obstacles associated with inconsistent and heterogeneous data. This improves the system's awareness of situations and makes it easier for data to be easily incorporated into processes that come after, such data encryption and algorithms for making decisions. The function of MIDM is becoming more and more important as ADAS technologies advance in order to fully use the many data streams that contribute to making autonomous driving possible.

### **3.2.2 AES-GCM Encryption**

The role of AES-GCM encryption is paramount in enhancing the secure communications protocol within the complex framework of ADAS in autonomous driving. AES-GCM is a robust and widely adopted symmetric encryption algorithm that combines the Galois/Counter Mode for encryption and Counter Mode for authentication, providing a high level of security and efficiency. Its application in ADAS serves several crucial purposes to fortify the communication protocols [23].

AES-GCM ensures the confidentiality of sensitive information exchanged within the ADAS network. In an environment where data streams from various sensors and components converge, the encryption of these diverse data types, such as GPS coordinates, sensor readings, and camera feeds, is imperative. AES-GCM utilizes a symmetric key approach, efficiently encrypting and decrypting the data, safeguarding it from unauthorized access and ensuring that only authorized entities can decipher the information. AES-GCM contributes to the integrity and authenticity of the data transmitted across the ADAS network. By employing cryptographic hash functions and authentication tags, the algorithm verifies that the data has not been tampered with

during transmission. This is crucial for ADAS, where the accuracy and reliability of the information are essential for making real-time decisions, especially in the context of autonomous vehicles navigating dynamic environments [25].

### **AES-GCM Algorithm**

*Initialization*

*CounterBlock = GCM CreateCounterBlock (Nonce)*

*Keystream = AES Encrypt (Key, CounterBlock)*

*Encrypt the plaintext*

*Ciphertext = Plaintext XOR Keystream*

*Authentication Tag generation*

*Tag = GCM Generate Tag (Keystream, AAD, Ciphertext)*

*return (Ciphertext, Tag)*

*Function GCM CreateCounterBlock(Nonce)*

*Create a 256-bit counter block from Nonce*

*CounterBlock = Nonce || 0<sup>31</sup> ||*

*return CounterBlock*

*function AES Encrypt (Key, Block)*

*Use AES block cipher to encrypt Block with Key*

*return AES Encryption Function (Key, Block)*

*function GCM GenerateTag (Keystream, AAD, Ciphertext)*

*Galois Field Multiplication for Authentication Tag*

*GHash = GCM GenerateGHASH (Keystream, AAD, Ciphertext)*

*Finalize Tag*

*Tag = GHash XOR (len(AAD) || len(Ciphertext))*

*return Tag*

*function GCM GenerateGHASH (Keystream, AAD, Ciphertext)*

*Galois Field Multiplication*

*GHash = 0<sup>BLOCK SIZE</sup>*

*for each block in AAD*

```

GHash = GCM GaloisFieldMultiply(GHash XOR block, Keystream)
for each block in Ciphertext:
GHash = GCM GaloisFieldMultiply(GHash XOR block, Keystream)
return GHash

function GCM GaloisFieldMultiply (a, b)
Galois Field Multiplication
result = 0
for i in 0 to (BLOCK SIZE - 1)
if b bit at position i is 1
result ^= a # XOR operation
if a bit at position (BLOCK_SIZE - 1) is 1:
a = (a << 1) XOR 0x87 Reduction step
else
a = a << 1
return result

```

### 3.3 Blockchain Network for Proof of Authentication

The encrypted IoT data is transferred to the blockchain network. A Blockchain Network designed for Proof of Authentication serves as a secure and transparent framework for verifying the authenticity of various entities and transactions within a decentralized system. In this paradigm, each authentication event is securely recorded as a block on the blockchain, creating an immutable ledger that can be audited and verified by all participants in the network. This system is particularly valuable in contexts where establishing trust and ensuring the legitimacy of actions or information is paramount [25].

In such a network, each block contains cryptographic hashes that uniquely represent the authentication data, ensuring the integrity of the recorded information. The decentralized and distributed nature of the blockchain ensures that no single entity has control over the entire authentication process, preventing manipulation or fraud. Authentication events, whether related to user access, device interactions, or data transactions, are timestamped and cryptographically linked to previous blocks, creating an unbroken chain of trust [26].

The consensus mechanism, often employed in blockchain networks, plays a pivotal role in establishing agreement among participants regarding the validity of authentication events. Proof of Authentication relies on consensus algorithms like Proof of or PoS to ensure that only legitimate transactions are added to the blockchain. This not only enhances security but also fosters a transparent and accountable environment where all network participants can rely on the authenticity of recorded events. In summary, a Blockchain Network for Proof of Authentication provides a robust foundation for building secure and trustworthy systems, offering a decentralized and tamper-resistant mechanism for verifying the authenticity of various activities within a network.

### **3.3.1 Web Application and Web3 Connectivity**

The integration of a web application with Web3 technology forms a crucial bridge between traditional web development and blockchain interactions [28]. Web3.js, a JavaScript library, is often employed in the development stack to enable seamless communication between a decentralized application (DApp) and the Ethereum blockchain. The web application's front end, often built with frameworks like React.js, Angular, or Vue.js, interacts with the Ethereum blockchain through the Web3.js library. This library allows developers to access and manipulate smart contracts, query blockchain data, and execute transactions, all within the web application's user interface. Web3.js integration allows developers to design dynamic, intuitive user interfaces that retrieve and present real-time data from the blockchain, improving the user experience in general. Moreover, events released by smart contracts can cause updates to be made to the web application, guaranteeing that users are informed about blockchain activities and transactions in a timely and accurate manner.

Tools such as Metamask are typically used to enable account connectivity in web applications within the context of blockchain interactions [29]. With the help of the browser extension wallet Metamask, users may safely manage their Ethereum accounts. Users may easily authenticate themselves, sign transactions, and engage with decentralized applications when Metamask is integrated into a web application. To securely manage private keys without exposing them to the web browser, users can connect their Metamask wallet to the web application. With this connection, security is improved and sensitive operations like sending encrypted IoT data or carrying out smart contract functions are subject to the express consent of the user via Metamask. Web3.js for blockchain connectivity and Metamask for account management together provide a



strong basis on which to construct safe and decentralized online apps that take advantage of blockchain technology.

### **3.3.2 Smart Contract Deployment on Ganache**

Developers take use of the powerful features of Ganache, a personal blockchain for Ethereum development, when they implement smart contracts on the Ganache platform with the Solidity programming language [30]. Writing the smart contract code in Solidity, a language designed specifically for Ethereum smart contracts, and then converting it into bytecode is the usual procedure. Ganache acts as a local Ethereum blockchain, providing a regulated and effective environment for testing and deploying contracts. Developers can interact with the local blockchain by deploying smart contracts using the Ganache command line or user interface. The deployment procedure enables comprehensive testing and debugging prior to the smart contract being made available on the Ethereum mainnet. Developers now have a robust and user-friendly framework for creating, testing, and implementing smart contracts in Ethereum-based decentralized apps thanks to the combination of Solidity and Ganache.

## **4. Design Specification**

A strong and flexible architecture for protecting communications is formed when MIDM and AES-GCM are joined, especially when considering ADAS. The way that MIDM-AES-GCM works is that it uses MIDM to combine various data inputs from cameras, sensors, and other parts of the ADAS environment. A comprehensive and coherent dataset that accurately depicts the current condition of the autonomous vehicle's surrounds is ensured by this merging procedure. The model preserves the integrity of combined information by adjusting to the dynamic nature of data produced by many sources.

AES-GCM is used in the encryption stage to protect the combined dataset. As a block cypher, AES-GCM ensures data integrity and secrecy by offering both encryption and authentication. To ensure the integrity of the data during transmission, the encryption procedure entails splitting the information into blocks, encrypting each block with a unique key, and creating authentication tags. The AES-GCM adaptive encryption settings enable customization according to the data's properties, hence optimizing the encryption process for different kinds of information in the combined dataset.

For secure communication in ADAS, MIDM-AES-GCM's combined capabilities offers a strong option. While AES-GCM protects the security and integrity of the combined information during transmission, MIDM makes sure the data is intelligently merged while taking into account the uniqueness of each input source. This paradigm is especially helpful for ADAS, since the safe and effective functioning of autonomous cars depends heavily on the integration of various data kinds and the requirement for secure, real-time communication.

## **5. Implementation**

JavaScript libraries' cryptographic features are utilized in the development of the AES-GCM encryption technique in a React.js application. Easily include the AES-GCM encryption process into a React.js application with a dependable cryptographic library, such the Web Crypto API or a third-party library like CryptoJS. Usually, the encryption technique is activated when a user interacts with the program or when it is a part of secure communication procedures. React.js is a JavaScript user interface package that makes it easier to integrate encryption features into the application's components, guaranteeing a safe and smooth user experience. To reduce possible security risks, care should be given when handling cryptographic keys and adhering to safe coding best practices. The implementation complies with the React.js guidelines, enabling the development of reliable and safe applications that support encrypted communication.

In the Windows environment, the Ganache platform offers a convenient and built-in solution for blockchain development and testing. Ganache is a personal blockchain for Ethereum development that provides a local test network, making it easier for developers to deploy and test smart contracts. By providing a built-in blockchain platform, Ganache streamlines the development and testing process for Windows users, enabling them to experiment with smart contracts and decentralized systems efficiently and securely within a local environment.

A multitiered strategy is used in the installation of IoT data flow and security measures. In the beginning, IoT sensor data is gathered and sent to the IoT server, which serves as the main hub for data aggregation. The server transmits the data to a web application in an effective manner by means of the IoT API. AES-GCM is used to encrypt the data before to transmission in order to guarantee the confidentiality and integrity of the information. After the data has been encrypted, it

is easily moved to a blockchain network and stored there in immutable blocks, guaranteeing an auditable and unchangeable record.

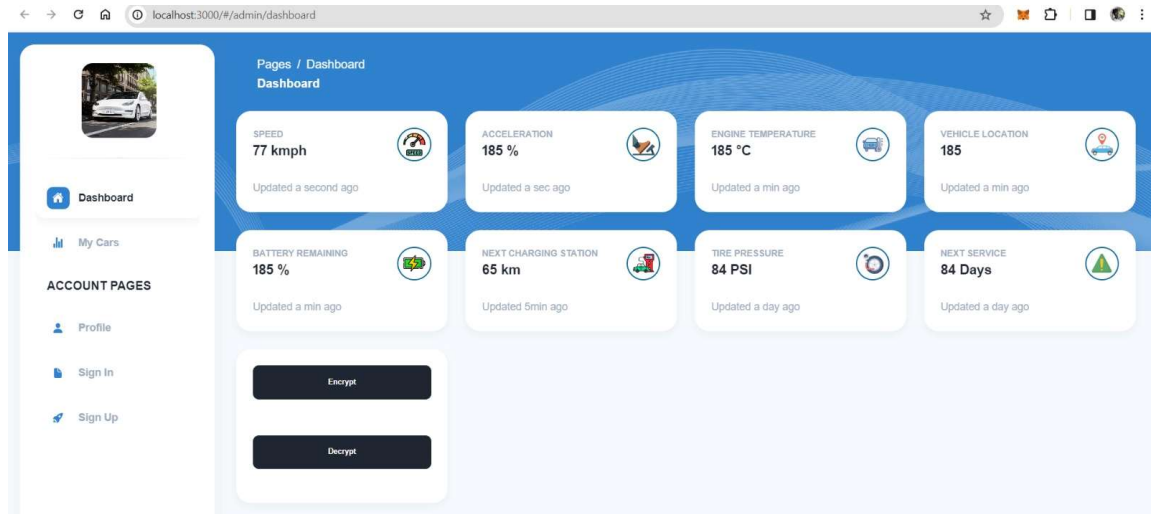
The web application and the blockchain network are connected via the incorporation of web3, a JavaScript package designed to enable communication with Ethereum-based blockchains. Through this connectivity, a decentralized and transparent data exchange is facilitated by the web application's ability to communicate with and obtain data from the blockchain network. Furthermore, MetaMask is a browser plugin that serves as a bridge between the blockchain and the browser, enabling users to safely manage their blockchain accounts and engage with decentralized apps (dApps) straight from their browsers.

The Ganache platform is used with the Solidity programming language to create smart contracts that control the actions of the blockchain network. Without requiring communication with the Ethereum mainnet, developers may install and test smart contracts using Ganache, a local blockchain environment for development and testing. The smart contract code, which establishes the guidelines and reasoning for data exchange inside the blockchain network, is written and assembled using the Solidity programme. With IoT sensor data encrypted, stored in a blockchain, and intelligently managed via smart contracts, this all-encompassing implementation guarantees a secure, decentralized, and interoperable system, offering a strong basis for decentralized applications within the IoT ecosystem.

## **6. Evaluation**

### **6.1 IoT Sensor Data**

An illustration of the integration of IoT sensor data gathered from self-driving cars may be found in Figure 2. The figure offers a thorough depiction of the complex network of sensors built into driverless cars, highlighting the many data streams produced by these Internet of Things gadgets. The graphic illustrates how data is captured and sent in a fluid manner, shedding light on the mutually beneficial interaction between the cars and their sensory infrastructure. Figure 2 depicts the diverse range of sensor data collecting equipment, ranging from lidar and radar sensors to cameras and additional environmental monitoring apparatuses. Understanding the intricacy and diversity of the data ecology necessary for the operation and decision-making skills of autonomous cars in a variety of contexts is made easier with the help of this illustration.



**Figure 2: User Dashboard**

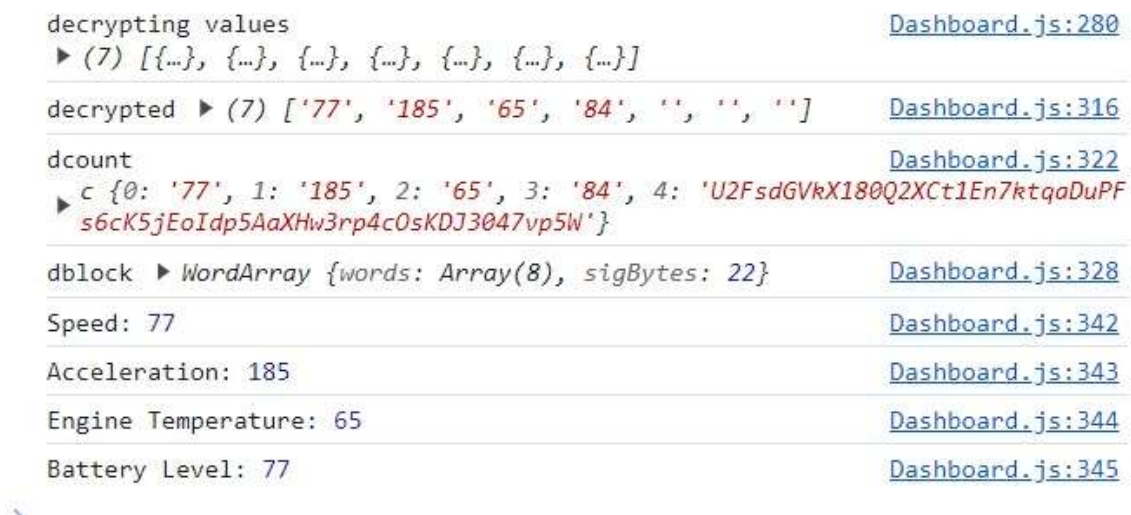
## 6.2 Sensor Data Encryption and Decryption

AES-GCM encryption of sensor data is illustrated in Figure 3. This figure gives a clear picture of the cryptographic procedure used to protect the sensor data that was gathered from many sources. The initialization step, data encryption using the AES method, the creation of GCM authentication tags, and the ultimate transmission of the encrypted data are all highlighted in the diagram of the AES-GCM encryption. This graphic illustrates how crucial it is to use strong encryption methods to protect sensitive sensor data, guaranteeing authenticity, secrecy, and integrity during the data transfer procedure. Understanding the security mechanisms put in place to safeguard the integrity of IoT sensor data in the autonomous cars and other applications where data privacy is a concern is made easier with the help of Figure 3.



**Figure 3: Encryption**

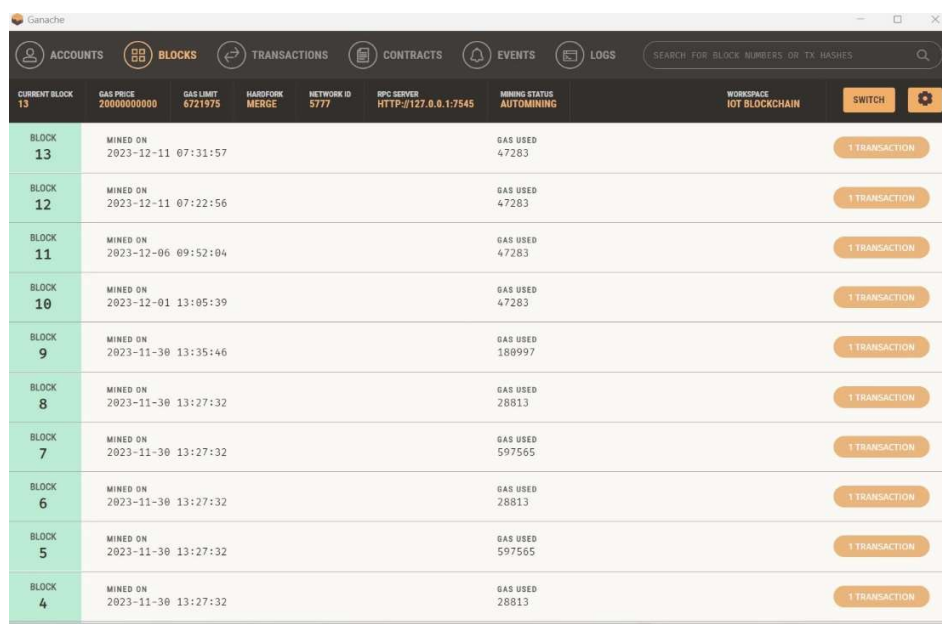
The decryption step after AES-GCM data encryption is illustrated in Figure 4, which shows the process of decrypting sensor data. The sequence of events that take place during the decryption process is depicted in the diagram. These phases include receiving the encrypted data, initialization, AES decryption, and GCM authentication tag verification. Figure 4 offers a thorough grasp of how encrypted sensor data is safely converted back into its original form by depicting the decryption operation. In applications like autonomous cars where protecting sensitive data is crucial, this visualization highlights the crucial role that decryption plays in guaranteeing authorized access to the sensor data and enhancing overall security and privacy of information.



**Figure 4: Decryption**

## 6.3 Blockchain for Data Storage

Figure 5 provides a detailed visual representation of the decentralized and secure architecture for storing sensor data through the implementation of a blockchain. The blockchain's linked blocks, each holding time-stamped and encrypted sensor data, are depicted in the diagram. Blocks are arranged sequentially and connected by cryptographic hashes, which highlights the transparency and immutability of the data being stored. This graphic illustrates how consensus processes preserve data integrity and highlights the decentralized character of blockchain technology. The distributed ledger structure, nodes, and cryptographic signatures highlight how resilient the blockchain is against tampering and guarantee reliable sensor data storage. Understanding the fundamentals and benefits of using blockchain technology for data storage, especially in situations like the IoT where data security and integrity are crucial factors, is made easier with the help of Figure 5.



The screenshot shows the Ganache application window. The top navigation bar includes icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below this, a status bar displays various metrics: CURRENT BLOCK 13, GAS PRICE 2000000000, GAS LIMIT 6721975, HARDFORK MERGE, NETWORK ID 5777, RPC SERVER HTTP://127.0.0.1:7545, MINING STATUS AUTOMINING, and WORKSPACE IOT BLOCKCHAIN. The main area is a table of blocks, with the first column highlighted in green. Each row represents a block with its number, mined-on timestamp, gas used, and a button to view transactions.

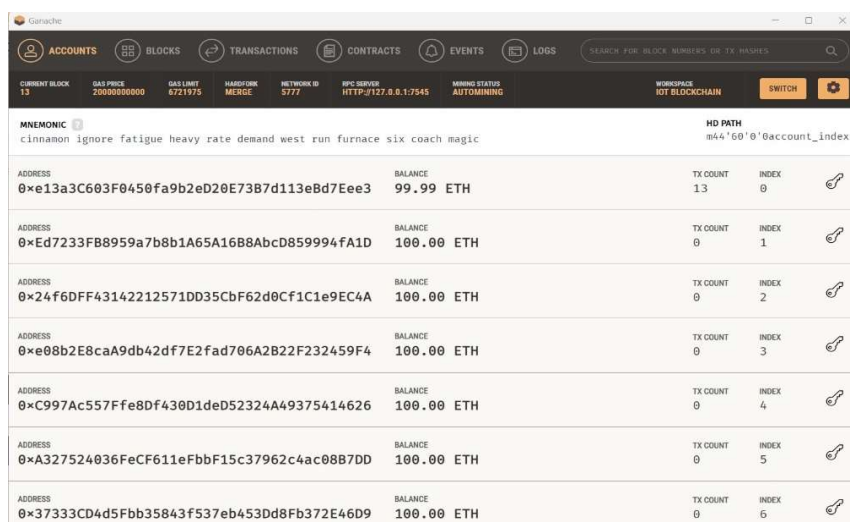
BLOCK	MINED ON	GAS USED	
BLOCK 13	2023-12-11 07:31:57	47283	1 TRANSACTION
BLOCK 12	2023-12-11 07:22:56	47283	1 TRANSACTION
BLOCK 11	2023-12-06 09:52:04	47283	1 TRANSACTION
BLOCK 10	2023-12-01 13:05:39	47283	1 TRANSACTION
BLOCK 9	2023-11-30 13:35:46	180997	1 TRANSACTION
BLOCK 8	2023-11-30 13:27:32	28813	1 TRANSACTION
BLOCK 7	2023-11-30 13:27:32	597565	1 TRANSACTION
BLOCK 6	2023-11-30 13:27:32	28813	1 TRANSACTION
BLOCK 5	2023-11-30 13:27:32	597565	1 TRANSACTION
BLOCK 4	2023-11-30 13:27:32	28813	1 TRANSACTION

Figure 5: Blockchain for Data Storage

## 6.4 Ethereum User Development Account

The Ethereum User Development Account is visually depicted in Figure 6, which also offers an example of the account's components. Important components like the Ethereum address,

smart contracts, and related transactions inside the Ethereum blockchain are probably included in the graphic. The purpose of this visual aid is to provide a clear understanding of the function that user development accounts play in the Ethereum ecosystem by demonstrating the deployment, execution, and interaction of smart contracts. Furthermore, by highlighting the independence and power individuals have over their development accounts, the visualization can draw attention to Ethereum's decentralized structure. A more comprehensive knowledge of blockchain-based application development and the use of smart contracts in the Ethereum network may be gained by consulting Figure 6, which is a useful reference for understanding the Ethereum development environment.



The screenshot shows the Gethache interface with a top navigation bar containing icons for ACCOUNTS, BLOCKS, TRANSACTIONS, CONTRACTS, EVENTS, and LOGS. Below the navigation bar, there is a status bar with various metrics like CURRENT BLOCK, GAS PRICE, GAS LIMIT, HARDFORK, NETWORK ID, RPC SERVER, and MINING STATUS. The main content area displays a list of accounts with columns for ADDRESS, BALANCE, TX COUNT, and INDEX. The first account has a balance of 99.99 ETH and 13 transactions. The other accounts have a balance of 100.00 ETH and 0 transactions.

ADDRESS	BALANCE	TX COUNT	INDEX
0xe13a3C603F0450fa9b2eD20E73B7d113e8d7Eee3	99.99 ETH	13	0
0xEd7233F88959a7b8b1A65A16B8AbcD859994fA1D	100.00 ETH	0	1
0x24f6DFF43142212571DD35CbF62d0Cf1C1e9EC4A	100.00 ETH	0	2
0xe08b2E8caA9db42df7E2fad706A2B22F232459F4	100.00 ETH	0	3
0xC997Ac557Ffe8Df430D1deD52324A49375414626	100.00 ETH	0	4
0xA327524036FeCF611eFbbF15c37962c4ac08B7DD	100.00 ETH	0	5
0x37333CD4d5Fbb35843f537eb453Dd8Fb372E46D9	100.00 ETH	0	6

**Figure 6: Ethereum User Development Account**

## 6.5 Smart Contract

Figure 7 provides a thorough overview of smart contracts self-executing, programmable contracts in a blockchain framework by visualizing their design and functioning. The contract code, terms, and the decentralized network that makes it possible for them to be executed are probably all depicted in the diagram. The beginning of a smart contract via a transaction, the network nodes verification procedure, and the ensuing carrying out of predetermined activities are illustrations for visual cues. This illustration helps users comprehend the automated and decentralized characteristics of smart contracts and highlights how they enable tamper-proof, transparent, and trustless transactions on a variety of blockchain systems. The specifics of smart

contract deployment and execution are well illustrated in Figure 7.1, 7.2, 7.3, 7.4 which aids in a better understanding of their importance in decentralized systems and blockchain-based applications.

```
3_sensordata_migrations.js
=====

Replacing 'SensorData'
-----
> transaction hash: 0xc017bf7b202958058960e8648b215adcb9896e6a3a67d3e8091c25107bbb371c
> Blocks: 0       Seconds: 0
> contract address: 0xcd2611DA66Bcd8485EEDc0798BFfEAe58C455c3A
> block number: 5
> block timestamp: 1701350852
> account: 0xe13a3C603F0450fa9b2eD20E73B7d113eBd7Eee3
> balance: 99.995382198972027932
> gas used: 597565 (0x91e3d)
> gas price: 3.031984417 gwei
> value sent: 0 ETH
> total cost: 0.001811807768144605 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.001811807768144605 ETH
```

Figure 7.1

```
2_deploy_migration.js
=====

Replacing 'Auth'
-----
> transaction hash: 0x27105b8bb98fd1184241bbcb5efae7a92190fcc17a77007d0a910a31390f91d7
> Blocks: 0       Seconds: 0
> contract address: 0x2f3E162046CeEf2435F612C146f8Fcd425Ab26F9
> block number: 3
> block timestamp: 1701350852
> account: 0xe13a3C603F0450fa9b2eD20E73B7d113eBd7Eee3
> balance: 99.997283535603534045
> gas used: 541752 (0x84438)
> gas price: 3.178366198 gwei
> value sent: 0 ETH
> total cost: 0.001721886244498896 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.001721886244498896 ETH
```

Figure 7.2

```
Starting migrations...
=====
> Network name: 'development'
> Network id: 5777
> Block gas limit: 6721975 (0x6691b7)

1_initial_migration.js
=====

Replacing 'Migrations'
-----
> transaction hash: 0x63051a2235ccaceb6fa379a87685f6d338a2651ee2968eab1d5955ff2112eed6
> Blocks: 0       Seconds: 0
> contract address: 0x9E3a4b794dF20ebd0C40a6A8E5da31D60cbB801A
> block number: 1
> block timestamp: 1701350851
> account: 0xe13a3C603F0450fa9b2eD20E73B7d113eBd7Eee3
> balance: 99.99915573025
> gas used: 250154 (0x3d12a)
> gas price: 3.375 gwei
> value sent: 0 ETH
> total cost: 0.00084426975 ETH

> Saving migration to chain.
> Saving artifacts
-----
> Total cost: 0.00084426975 ETH
```



**Figure 7.3**

```
4_vehicle_migrations.js
=====

Replacing 'VehicleData'
-----
> transaction hash:    0x1ab983bd6acde40741e1f6f30563c593841e8f61e41f216d7c583f546c0f4cdf
> Blocks: 0           Seconds: 0
> contract address:   0xEf825563aaa37d84C8E30A3166445694c54afEe1
> block number:       7
> block timestamp:    1701350852
> account:            0xe13a3C603F0450fa9b2eD20E73B7d113eBd7Eee3
> balance:            99.993552625169544023
> gas used:           597565 (0x91e3d)
> gas price:          2.918157168 gwei
> value sent:         0 ETH
> total cost:         0.00174378858809592 ETH

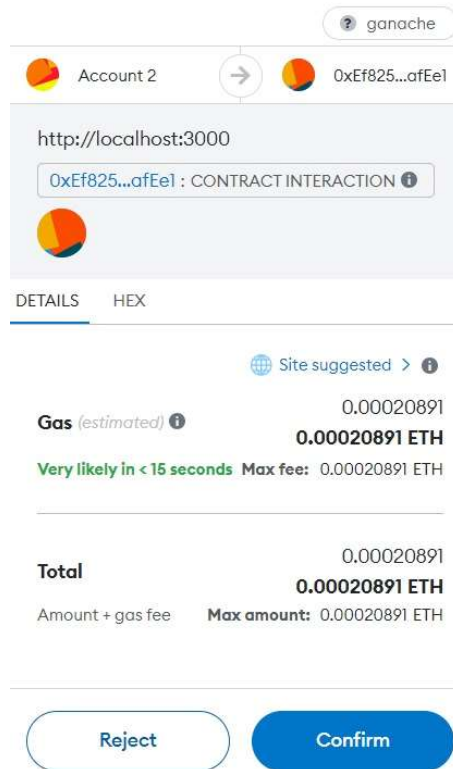
> Saving migration to chain.
> Saving artifacts
-----
> Total cost:         0.00174378858809592 ETH

Summary
=====
> Total deployments:  4
> Final cost:         0.006121752350739421 ETH
```

**Figure 7.4: Smart Contract**

## 6.5 Transaction Confirmation

Figure 8 provides a thorough visual explanation of the confirmation process for transactions, showing how they are put to the blockchain and confirmed. The illustration most likely illustrates the sequential processes that go into verifying a transaction: submitting it, letting it spread throughout the network, having it validated by consensus techniques like proof-of-work or proof-of-stake, and finally adding the validated transaction to a fresh block on the blockchain. In order to create a safe and transparent transaction ledger, visual signals may emphasize nodes coming to a consensus, cryptographic hashing, and the chronological linking of blocks. With an emphasis on the security and immutability of the recorded transactions, this visualization is a useful tool for comprehending the decentralized and consensus-driven nature of transaction confirmation in blockchain networks. A more comprehensive understanding of the tenets of blockchain technology and its function in safe, transparent, and trustless transaction processing is facilitated by Figure 8.



**Figure 8: Transaction Confirmation**

## 7. Conclusion and Future Work

This study has offered a fresh and all-encompassing method for improving the ADAS secure communications protocol. A groundbreaking attempt to address the complex challenges posed by the dynamic and sensitive nature of data in autonomous driving environments is represented by the integration of blockchain technology for secure proof of authentication, AES-GCM for dynamic and adaptive data encryption, and MIDM for intelligent data synthesis. The harmonious fusion of these technologies results in an all-encompassing security framework for ADAS, guaranteeing the authenticity, secrecy, and integrity of communication operations that are vital to the effectiveness and safety of autonomous cars. There are several opportunities for additional investigation and improvement of the suggested security architecture. Above all, it will be crucial to thoroughly test and validate the upgraded secure communications protocol in ADAS systems in real-world scenarios. To further strengthen the system's resistance to new attacks, investigating the integration of artificial intelligence and machine learning for anomaly detection inside the secure communications protocol is recommended. Future research in this area can guarantee the safety

and security of autonomous driving systems in an ever-changing technical environment by pushing the envelope of secure communications in ADAS.

## References

- [1] M. M. Antony and R. Whenish, ‘Advanced driver assistance systems (ADAS)’, in *Automotive Embedded Systems: Key Technologies, Innovations, and Applications*, Springer, 2021, pp. 165–181.
- [2] G. De-Las-Heras, J. Sanchez-Soriano, and E. Puertas, ‘Advanced driver assistance systems (ADAS) based on machine learning techniques for the detection and transcription of variable message signs on roads’, *Sensors*, vol. 21, no. 17, p. 5866, 2021.
- [3] Y. Xu, Z. Ye, and C. Wang, ‘Modeling commercial vehicle drivers’ acceptance of advanced driving assistance system (ADAS)’, *Journal of intelligent and connected vehicles*, vol. 4, no. 3, pp. 125–135, 2021.
- [4] S. Rath, C. Selvan, R. Suriya, C. Rakshith, and G. Nageshwaran, ‘IoT Based Vehicle Authentication Study of ADAS and Futurescope of IoV’, EasyChair, 2022.
- [5] A. Ledezma, V. Zamora, Ó. Sipele, M. P. Sesmero, and A. Sanchis, ‘Implementing a gaze tracking algorithm for improving advanced driver assistance systems’, *Electronics*, vol. 10, no. 12, p. 1480, 2021.
- [6] Y. Li, M. Liu, C. Cao, and J. Li, ‘Communication-Traffic-Assisted Mining and Exploitation of Buffer Overflow Vulnerabilities in ADASs’, *Future Internet*, vol. 15, no. 5, p. 185, 2023.
- [7] R. Bogdan *et al.*, ‘Optimization of AUTOSAR communication stack in the context of advanced driver assistance systems’, *Sensors*, vol. 21, no. 13, p. 4561, 2021.
- [8] S. Sinha and R. West, ‘Towards an integrated vehicle management system in DriveOS’, *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 20, no. 5s, pp. 1–24, 2021.
- [9] C. M. Fourie and H. C. Myburgh, ‘An intra-vehicular wireless multimedia sensor network for smartphone-based low-cost advanced driver-assistance systems’, *Sensors*, vol. 22, no. 8, p. 3026, 2022.

- [10] L.-A. Tran, T.-D. Do, D.-C. Park, and M.-H. Le, ‘Robustness Enhancement of Object Detection in Advanced Driver Assistance Systems (ADAS)’, *arXiv preprint arXiv:2105.01580*, 2021.
- [11] R. Bera, ‘Smart Automotive System With CV2X-Based Ad Hoc Communication’, *Cloud and IoT-Based Vehicular Ad Hoc Networks*, pp. 293–323, 2021.
- [12] L.-A. Tran, T.-D. Do, D.-C. Park, and M.-H. Le, ‘Enhancement of robustness in object detection module for advanced driver assistance systems’, in *2021 International Conference on System Science and Engineering (ICSSE)*, IEEE, 2021, pp. 158–163.
- [13] S. D. McLean, E. A. Juul Hansen, P. Pop, and S. S. Craciunas, ‘Configuring ADAS platforms for automotive applications using metaheuristics’, *Frontiers in Robotics and AI*, vol. 8, p. 762227, 2022.
- [14] A. Dakić, M. Hofer, B. Rainer, S. Zelenbaba, L. Bernadó, and T. Zemen, ‘Real-time vehicular wireless system-level simulation’, *IEEE Access*, vol. 9, pp. 23202–23217, 2021.
- [15] R. Broux, E. Lisova, and S. Mubeen, ‘Communication patterns in automotive systems’, tech. rep., March, 2022.
- [16] S. Khandelwal and S. Shreejith, ‘A lightweight multi-attack CAN intrusion detection system on hybrid FPGAs’, in *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*, IEEE, 2022, pp. 425–429.
- [17] H. Borhan, R. Radulescu, M. Lammert, C. Zhang, K. Kelly, and A. Vahidi, ‘Advancing Platooning with ADAS (Advanced Driver-Assistance Systems) Control Integration and Assessment’, Cummins, 2022.
- [18] X. Zhou *et al.*, ‘Strategic safety-critical attacks against an advanced driver assistance system’, in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, IEEE, 2022, pp. 79–87.

- [19] J. Guan, L. Pan, C. Wang, S. Yu, L. Gao, and X. Zheng, ‘Trustworthy Sensor Fusion against Inaudible Command Attacks in Advanced Driver-Assistance Systems’, *IEEE Internet of Things Journal*, 2023.
- [20] J. Guan, X. Zheng, C. Wang, Y. Zhou, and A. Jolfaei, ‘Robust sensor fusion algorithms against voice command attacks in autonomous vehicles’, in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2021, pp. 895–902.
- [21] P. Dixit and P. C. Kumar, ‘V2X Communication for Message Transmission and Warning Detection’.
- [22] H. Abaza *et al.*, ‘RDMA-Based Deterministic Communication Architecture for Autonomous Driving’, in *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, IEEE, 2023, pp. 137–146.
- [23] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, ‘A Blockchain Framework for Securing Connected and Autonomous Vehicles’, *Sensors*, vol. 19, no. 14, Art. no. 14, Jan. 2019, doi: 10.3390/s19143165.
- [24] PhD Scholar, CSE Department, NIT Silchar, Silchar, India., P. K. Das\*, N. Sinha, Professor, Electrical Department, NIT Silchar, Silchar, India., A. B., and Professor, CSE Department, NIT Karnataka, Surathkal, Mangalore, India., ‘Data Privacy Preservation using Aes-Gcm Encryption in Heroku Cloud’, *IJRTE*, vol. 8, no. 3, pp. 7544–7548, Sep. 2019, doi: 10.35940/ijrte.C6131.098319.
- [25] K. Kim, S. Choi, H. Kwon, H. Kim, Z. Liu, and H. Seo, ‘PAGE—Practical AES-GCM Encryption for Low-End Microcontrollers’, *Applied Sciences*, vol. 10, no. 9, p. 3131, Apr. 2020, doi: 10.3390/app10093131.
- [26] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, ‘A decentralized lightweight blockchain-based authentication mechanism for IoT systems’, *Cluster Comput*, vol. 23, no. 3, pp. 2067–2087, Sep. 2020, doi: 10.1007/s10586-020-03058-6.

- [27] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C.-M. Chen, ‘An Improved Blockchain-Based Authentication Protocol for IoT Network Management’, *Security and Communication Networks*, vol. 2020, pp. 1–16, Oct. 2020, doi: 10.1155/2020/8836214.
- [28] ‘Manteniment’. Accessed: Dec. 15, 2023. [Online]. Available: <https://upcommons.upc.edu/bitstream/handle/2117/379908/171754.pdf?sequence=1>
- [29] M. Šipek, M. Žagar, B. Mihaljević, and N. Drašković, ‘Application of Blockchain Technology for Educational Platform’, in *Human Interaction, Emerging Technologies and Future Systems V*, vol. 319, T. Ahram and R. Taiar, Eds., in Lecture Notes in Networks and Systems, vol. 319. , Cham: Springer International Publishing, 2022, pp. 1283–1287. doi: 10.1007/978-3-030-85540-6\_165.
- [30] G. Mathur, ‘GANACHE: A Robust Framework for Efficient and Secure Storage of Data on Private Ethereum Blockchains’, In Review, preprint, Oct. 2023. doi: 10.21203/rs.3.rs-3495549/v1.

**Video Link:** <https://youtu.be/4w-EtEfdx84>