**CHAPTER 1**

# INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) represent a crucial component of Intelligent Transportation Systems (ITS), enabling vehicles to communicate with each other and with roadside infrastructure. These networks improve road safety, traffic efficiency, and driver convenience by allowing real-time data exchange. However, the open and dynamic nature of VANETs makes them highly vulnerable to security threats such as message tampering, eavesdropping, spoofing, and denial-of-service attacks. Secure and reliable data transmission is essential to maintain the integrity, confidentiality, and authenticity of critical vehicular information. The aim of this project is to design and implement a secure data transmission protocol tailored specifically for VANET environments. This protocol incorporates cryptographic techniques such as digital signatures, encryption, and certificate verification to protect data in transit. It also includes mechanisms for detecting and preventing malicious activity within the network. The system is tested under various network conditions to evaluate performance in terms of latency, throughput, and security resilience. This project not only strengthens the data exchange process in VANETs but also contributes to safer and smarter road transportation systems. Ultimately, it lays the groundwork for building trust in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications.

The Secure Data Transmission Protocol Using VANET (Vehicular Ad Hoc Network) represents a pivotal innovation in the field of intelligent transportation systems (ITS), addressing the increasing need for reliable and secure communication among vehicles on the move. VANETs are a specialized class of Mobile Ad Hoc Networks (MANETs) that enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. As vehicles become more connected and autonomous, the importance of secure, fast, and reliable data exchange becomes critical not only for traffic efficiency but also for safety and privacy. Data transmitted in VANETs includes real-time traffic updates, weather alerts, accident notifications, navigation assistance, and vehicle diagnostics, all of which must be secured to prevent malicious attacks, data tampering, or eavesdropping.

This project explores the design and implementation of a secure communication protocol that ensures data confidentiality, integrity, and authentication during vehicular communication. Traditional security protocols often fall short in VANETs due to the high mobility of vehicles, rapidly changing network topologies, low latency requirements, and the need for lightweight cryptographic methods that do not overburden the system. The proposed protocol addresses these challenges by integrating efficient encryption algorithms, public key infrastructure (PKI), and digital signatures to provide secure and authenticated message exchange without compromising performance.

The introduction of certificate-based trust models helps ensure that only authorized vehicles participate in the network, while hashing algorithms such as SHA-256 verify message integrity. To prevent replay and Sybil attacks, timestamping and vehicle ID verification techniques are employed. Moreover, the use of elliptic curve cryptography (ECC) makes it feasible to implement strong encryption with minimal processing overhead, which is ideal for resource-constrained environments like VANETs.

This secure protocol is particularly crucial for safety-critical scenarios such as collision avoidance, emergency braking alerts, and lane-change warnings, where the authenticity and freshness of the information are vital. The implementation is tested in simulated urban environments using network simulation tools like NS-3, OMNeT++, or SUMO, allowing the evaluation of latency, throughput, packet loss, and resilience to attacks under different traffic densities and mobility patterns.

Overall, this project contributes to the enhancement of vehicular communication security, ensuring that VANETs can be safely adopted on a wide scale. By protecting the integrity and privacy of transmitted data, the proposed protocol supports the development of smarter, safer, and more efficient road transportation systems. It paves the way for the secure integration of VANETs into future autonomous driving platforms and intelligent traffic management infrastructures.

**CHAPTER 2**

**LITERATURE SURVEY**

# 2.1 Literature Review

Vehicular Ad Hoc Networks (VANETs) are a subclass of Mobile Ad Hoc Networks (MANETs), where vehicles communicate with each other (Vehicle-to-Vehicle, V2V) and with roadside infrastructure (Vehicle-to-Infrastructure, V2I) to enhance road safety, traffic efficiency, and infotainment services. Security is a critical challenge in VANETs due to their highly dynamic topology, open wireless medium, and real-time constraints.

#### ****2.1.1 Review of Existing Research Work****

a) RSU-Aided Authentication Protocols

• Lu et al. (2012): Proposed an efficient conditional privacy-preserving protocol using short group signatures with RSUs acting as verifiers.

• Strengths: Ensures conditional anonymity.

• Limitations: Relies heavily on RSU availability.

b) Certificate-Based Authentication Protocols

• Papadimitratos et al. (2008): Introduced a PKI-based framework for secure vehicle communication.

• Strengths: Robust authentication mechanism.

• Limitations: Certificate management overhead.

c) Identity-Based Cryptography (IBC)

• Raya and Hubaux (2005): Proposed IBE for VANETs to reduce certificate management complexity.

• Strengths: Lower overhead compared to traditional PKI.

• Limitations: Key escrow problem; trust in key generation center.

d) Blockchain-Based Protocols

• Dorri et al. (2017): Proposed a lightweight blockchain framework for VANETs.

• Strengths: Decentralization and tamper-resistance.

• Limitations: Latency and scalability challenges.

e) Trust-Based Models

• Chen et al. (2010): Designed a trust evaluation model for VANETs to assess the trustworthiness of message sources.

• Strengths: Filters out malicious data.

• Limitations: Trust metrics can be manipulated; delay in convergence.

#### ****2.1.2 Key Observations from Literature****:

1. Security is multi-faceted

• Secure communication in VANETs involves multiple security requirements: authentication, confidentiality, integrity, non-repudiation, and availability.

• Most protocols focus on authentication and integrity, while real-time confidentiality and availability are less addressed.

2. Trade-Off Between Security and Performance

• Strong security mechanisms (e.g., PKI, blockchain) often introduce high computational and communication overhead, which may affect real-time performance.

• Lightweight solutions (e.g., IBC or trust models) offer faster transmission but may compromise robustness.

3. Dependency on Infrastructure

• Many protocols depend on Road Side Units (RSUs) or central authorities (e.g., Certificate Authorities).

• Issue: In areas with sparse RSU deployment, security mechanisms may fail or be less effective.

4. Privacy vs. Accountability Dilemma

• Conditional privacy is a popular approach: vehicles remain anonymous unless misbehavior is detected.

### ****2.1.3 Relevance to Indian Context****

Here is a summary of the relevance to the Indian context for a "Secure Data Transmission Protocol using VANET (Vehicular Ad Hoc Network)" project:

1. Smart Cities & Intelligent Transport Systems (ITS): India is rapidly urbanizing and deploying smart city initiatives.
2. High Traffic Density & Road Safety: India has one of the highest rates of road accidents globally.
3. Rising Cybersecurity Threats: As vehicles become more connected, ensuring \*data privacy and integrity\* becomes essential.
4. Government Push for V2X & 5G Integration: Indian telecom operators and the government are testing 5G and V2X (Vehicle-to-Everything) technologies.

### ****2.1.4 Challenges in Existing Systems****

1. High Computational & Communication Overhead

1. PKI‑based schemes incur significant processing delays and bandwidth.
2. Blockchain approaches add consensus and block‑propagation delays that violate VANET latency constraints.

2. Real‑Time Constraints

Safety applications (e.g., collision warnings) require end‑to‑end delays of only a few milliseconds. Many security schemes (especially heavy cryptography or consensus) cannot meet these deadlines.

3. Scalability

1. Broadcast‑based authentication and global trust computations become infeasible as vehicle density grows.
2. Key distribution and revocation in large networks (urban environments) introduces management bottlenecks.

4.Infrastructure Dependency

* 1. RSU‑aided or CA‑centric protocols fail in regions with sparse or no roadside units.
  2. Reliance on centralized authorities introduces single points of failure and can be attacked or congested.

5.Privacy vs. Accountability Trade‑off

1. Conditional anonymity (e.g., group signatures) protects driver privacy but complicates real‑time misbehavior tracing.
2. Key escrow in identity‑based schemes grants key‑generator full decryption power, risking privacy breaches.

# 2.2 Problem Statement

Vehicular Ad Hoc Networks (VANETs) enable real-time communication between vehicles and infrastructure to support road safety, traffic management, and intelligent transportation systems. However, the decentralized and wireless nature of VANETs exposes them to several security threats, including message spoofing, eavesdropping, data tampering, and Sybil attacks. Due to the high mobility of vehicles and frequent network topology changes, ensuring secure and reliable data transmission becomes a significant challenge. Current security mechanisms used in conventional networks are not always suitable for VANETs because they may introduce high latency, require complex computation, or lack adaptability in dynamic environments. Moreover, the absence of a centralized authority in VANETs makes authentication and trust management even more difficult. Insecure communication in VANETs can result in misinformation, which can cause traffic congestion, accidents, or system-wide failures. Therefore, there is a critical need for a robust and lightweight security protocol that ensures data integrity, confidentiality, authentication, and resilience against attacks while maintaining real-time performance. The problem lies in developing a secure, efficient, and scalable data transmission protocol that is tailored specifically to the unique challenges and constraints of VANET environments. This project addresses that need with a focus on real-world applicability and future scalability.

In the modern era of intelligent transportation systems and autonomous vehicles, Vehicular Ad Hoc Networks (VANETs) play a critical role in enabling real-time communication between vehicles (V2V) and between vehicles and roadside infrastructure (V2I). These networks facilitate the exchange of vital information such as traffic conditions, accident alerts, speed regulation, road hazards, and route optimization. However, the open, dynamic, and decentralized nature of VANETs exposes them to a wide range of security vulnerabilities that can severely compromise the safety, efficiency, and reliability of vehicular communication.

One of the core challenges lies in ensuring secure and trustworthy data transmission in a highly mobile environment. Due to the high-speed movement of vehicles and constantly changing network topology, traditional security solutions—such as static firewalls or heavy cryptographic mechanisms—are often inadequate or infeasible. The lack of centralized control, coupled with limited computational resources in onboard units (OBUs), demands lightweight and efficient security protocols that can operate in real-time.

Cyberattacks such as message spoofing, replay attacks, Sybil attacks, man-in-the-middle attacks, and denial-of-service (DoS) attacks pose significant threats to VANETs. For example, a malicious vehicle could inject false accident information to divert traffic, or impersonate another vehicle to manipulate data. Such attacks can lead to traffic congestion, accidents, and even loss of life. Therefore, the need for a robust security framework that ensures data confidentiality, integrity, authenticity, and availability is more urgent than ever.

Moreover, vehicles often communicate with unfamiliar nodes, making trust establishment another major hurdle. Vehicles need to verify the identity and trustworthiness of the sender before acting on any received message. Conventional Public Key Infrastructure (PKI) systems introduce overhead in terms of certificate management and revocation, which can be impractical in high-mobility environments without efficient optimization.

Additionally, VANETs must handle low latency communication, especially for time-critical messages like collision warnings or emergency braking. Any delay in verification or encryption-decryption processes can render the system ineffective. As a result, there is a crucial need for designing a lightweight, scalable, and fast-responding secure data transmission protocol that adapts to the dynamic nature of VANETs while ensuring all security properties.

The problem thus revolves around creating a protocol that:

1. Authenticates vehicles and messages in real-time,

2. Prevents unauthorized data injection,

3. Maintains data integrity and confidentiality,

4. Operates with minimal computational and communication overhead,

5. Is resistant to common VANET-specific attacks,

6. Scales with increasing network size and density.

In summary, the lack of a secure, efficient, and VANET-specific data transmission protocol leads to major vulnerabilities in vehicular communication. This project aims to address this gap by designing a secure protocol tailored to VANETs that not only enhances security but also maintains communication speed and reliability—ultimately contributing to safer and smarter transportation systems.

**2.3 Objectives**

The primary objective of this project is to design and implement a secure data transmission protocol tailored for Vehicular Ad Hoc Networks (VANETs), ensuring safe and reliable communication between vehicles and infrastructure. The protocol aims to guarantee the confidentiality, integrity, and authenticity of the transmitted data using lightweight cryptographic methods. Another goal is to develop an efficient authentication mechanism that verifies the legitimacy of participating vehicles without introducing significant latency. The system will also aim to detect and mitigate threats such as message tampering, Sybil attacks, and denial-of-service (DoS) attempts. It seeks to reduce communication overhead by employing optimized key exchange and encryption strategies. A real-time alert system will be incorporated to notify drivers and systems of any security breaches. The protocol should adapt to high-mobility conditions and frequent topology changes, maintaining stability in dynamic vehicular environments. In addition, the solution should be scalable, supporting both urban and rural VANET deployments. The project also aims to maintain compatibility with existing ITS infrastructure and future 5G-enabled vehicle communication systems. Ultimately, the objective is to build a trustworthy communication framework that enhances road safety, traffic management, and user confidence in intelligent transportation technologies. The primary objective of the Secure Data Transmission Protocol Using VANET is to design and implement a robust communication framework that ensures confidential, authentic, and tamper-proof data exchange between vehicles and roadside infrastructure in a dynamic and mobile vehicular environment. In Vehicular Ad Hoc Networks (VANETs), vehicles rely on real-time data to make critical driving decisions, such as avoiding collisions, optimizing traffic flow, and responding to emergency alerts. Thus, the system must guarantee the reliability and security of this information.

One of the foremost objectives is to develop a lightweight and efficient security protocol that supports end-to-end data encryption, ensuring that transmitted messages are protected from unauthorized access. The protocol must be optimized for high-speed vehicles, with minimal computational and communication overhead, to meet the low-latency requirements of VANET applications. The goal is to balance cryptographic strength with processing efficiency, allowing even resource-constrained onboard units (OBUs) to operate smoothly.

Another critical objective is to ensure data integrity and authenticity, which involves verifying that the information received has not been altered in transit and that it originates from a legitimate source. To achieve this, the system should employ digital signatures, hashing algorithms (e.g., SHA-256), and certificate-based authentication mechanisms such as Public Key Infrastructure (PKI) or Elliptic Curve Cryptography (ECC). These tools help in detecting any manipulation or forgery of messages and build a trust-based communication model.

To prevent malicious behaviour in the network, the protocol must also include mechanisms to detect and mitigate common VANET attacks, including Sybil attacks, replay attacks, man-in-the-middle attacks, and false message injection. Incorporating timestamp validation, unique vehicle identifiers, and message sequence checking are key objectives to counter these threats and ensure that stale or duplicated data cannot influence vehicular behaviour.

Another objective is to provide scalability and flexibility, so the protocol can function efficiently in both sparse and dense traffic environments, adapting dynamically to the number of participating vehicles. It should be interoperable with existing standards like DSRC (Dedicated Short Range Communication) or 5G-V2X, ensuring compatibility with modern vehicular communication platforms.

Furthermore, the system should include real-time logging and monitoring features to support audit trails, system diagnostics, and forensic analysis in case of security breaches. An additional aim is to offer secure key management for distribution, renewal, and revocation of certificates without adding unnecessary latency or complexity.

From a usability perspective, the objective is to ensure that the protocol does not interfere with the user experience or the core functionality of the vehicle’s communication system. It should operate in the background, enabling automatic decision-making and communication while maintaining security as a priority.

Lastly, the project seeks to simulate and evaluate the designed protocol using tools like NS-3, SUMO, or OMNeT++, under various mobility models, traffic conditions, and attack scenarios. The evaluation metrics include packet delivery ratio, encryption/decryption time, end-to-end latency, and attack detection rate—to assess the effectiveness, efficiency, and robustness of the protocol.

In conclusion, the overarching objective of this project is to establish a secure, efficient, and scalable communication framework for VANETs that can be practically implemented to enhance road safety, privacy, and the overall trustworthiness of intelligent vehicular communication systems.

**2.4 Motivation**

Vehicular Ad Hoc Networks (VANETs) represent a crucial technological advancement in intelligent transportation systems (ITS), enabling vehicles to communicate with each other (V2V) and with roadside infrastructure (V2I). These networks play a pivotal role in enhancing road safety, traffic efficiency, and driver comfort by enabling real-time data exchange. However, the open and dynamic nature of VANETs introduces serious security challenges that must be addressed to ensure the reliability of the transmitted information.

In VANETs, data such as collision warnings, emergency vehicle alerts, and traffic congestion updates must be exchanged securely and in real-time. Any compromise in the security of this data—such as message tampering, impersonation, or denial-of-service attacks—can lead to life-threatening consequences or severe disruptions in traffic management. Hence, secure data transmission is not just a technical requirement but a fundamental necessity for the safe deployment of VANETs.

**CHAPTER 3**

**Overview of “Secure Data Transmission Protocol Using VANET”**

The proposed system is developed using a structured methodology involving the design, implementation, and simulation of a secure data transmission protocol tailored for VANET environments. The key steps of the methodology are as follows:

**1. Requirement Analysis**

• Identify security threats in VANETs, including eavesdropping, message tampering, impersonation, and replay attacks.

• Determine the cryptographic techniques suitable for real-time communication with limited overhead.

**2. System Design**

• Design a layered protocol architecture involving:

o RSA encryption for secure key exchange.

o AES encryption for data confidentiality.

o Timestamps and nonces for replay attack prevention.

o RSU (Road Side Unit) simulation to act as a trusted authority.

• Define message formats for secure communication between vehicles and RSUs.

**3. Implementation Using Python**

• Use Python socket programming (UDP) to simulate real-time V2V and V2I communication.

• Implement RSA and AES algorithms using Python’s cryptography or PyCryptodome libraries.

• Incorporate timestamp and nonce-based mechanisms for freshness validation of messages.

• Develop a Tkinter-based GUI to visualize:

o Encrypted/decrypted messages

o Communication status

o Authentication logs

**4. RSU Simulation**

• Implement a Python-based RSU module to:

o Generate and distribute public keys

o Authenticate vehicles

o Monitor and validate message exchanges

**5. Testing and Evaluation**

• Test the protocol with multiple simulated vehicles exchanging messages.

• Analyze the system's performance under different attack scenarios (e.g., delayed packets, tampered data).

• Evaluate the protocol in terms of:

o Security (resistance to common attacks)

o Efficiency (encryption/decryption time)

o Scalability and adaptability

**6. Documentation and Result Analysis**

• Document the system architecture, flowcharts, and source code.

• Record test cases and outcomes.

• Analyze performance metrics and highlight the system's strengths and limitations.

**3.1 Workflow of the System**

The Workflow of the system aims to provide a lightweight, secure, and efficient communication protocol for Vehicular Ad Hoc Networks (VANETs), addressing the security flaws and performance limitations of existing systems. The proposed protocol integrates several cryptographic techniques and a simulation environment in Python, ensuring robust security without compromising real-time communication needs.

1. **Secure Key Exchange:**

The system uses RSA-based asymmetric encryption for secure key exchange between vehicles and Road Side Units (RSUs). This ensures that communication is authenticated and that keys are distributed securely, preventing unauthorized access.

1. **Data Encryption:**

AES symmetric encryption is used for fast and efficient message encryption, ensuring data confidentiality during transmission. This combination of RSA and AES provides a balance between security and performance, suitable for the dynamic and resource-constrained environment of VANETs.

1. **Replay Attack Prevention:**

Each message is tagged with a timestamp and nonce, which ensures freshness and prevents attackers from replaying captured messages. This approach enhances the system’s resistance to replay attacks, a common vulnerability in VANETs.

1. **RSU Authentication:**

A simulated RSU acts as a trusted central entity that authenticates vehicles before allowing secure communication. This ensures that only authorized vehicles participate in data exchanges, preventing impersonation attacks.

1. **UDP Communication Simulation:**

The system is implemented using UDP socket programming to emulate real-time V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) communication. This ensures that the system can simulate real-world data flow while maintaining low latency, which is crucial in vehicular networks.

1. **User Interface:**

A Tkinter-based GUI provides a visual representation of the communication process, displaying encrypted and decrypted messages, the status of encryption, and security checks. This makes it easier to monitor the system’s security measures and performance during operation.

* 1. **Advantages of the System**

**1.Enhanced Security:**

By combining RSA and AES encryption, along with replay attack prevention through timestamps and nonces, the system provides a robust and secure communication channel between vehicles and RSUs.

**2.Efficient Performance:**

The use of AES for data encryption ensures low computational overhead, enabling real-time communication in resource-constrained vehicular environments.

**3.Prevention of Replay Attacks:**

The system effectively prevents replay attacks by verifying the freshness of messages using timestamps and nonces.

**4.Modular and Scalable:**

The proposed protocol is designed to be modular, allowing easy adaptation and scalability for different network sizes and use cases in VANETs.

**5.Practical Implementation:**

Unlike many theoretical solutions, this system provides a practical, real-time simulation using Python, making it suitable for actual deployment and testing.

**6.User-Friendly Visualization:**

The GUI enhances the usability of the system by allowing easy monitoring of the message flow, encryption status, and security checks during operation.

**7.Cost-Effective:**

The system utilizes widely available libraries in Python (e.g., cryptography, PyCryptodome, Tkinter), reducing development costs and making it accessible for a broader range of applications.

**CHAPTER 4**

**IMPLEMENTATION**

The implementation of the secure data transmission protocol in VANET involves integrating cryptographic security measures into a simulated vehicular network environment. Initially, a simulation environment is set up using tools like NS-3 and SUMO to model real-time traffic scenarios and vehicle movement. Each vehicle is assigned a unique ID and digital certificate issued by a simulated Certificate Authority (CA). For secure communication, Elliptic Curve Cryptography (ECC) is used for encrypting messages, and digital signatures ensure message authenticity and integrity. When a vehicle sends a message, it signs and encrypts it before transmission. The receiving vehicle verifies the signature using the sender’s public key and decrypts the message securely. To manage identities, the system employs pseudonyms that change periodically, preserving driver privacy. The protocol includes a trust evaluation mechanism to monitor vehicle behavior and identify malicious nodes. An intrusion detection component flags suspicious activity like message tampering or spoofing. All modules interact in real time within the simulation, enabling validation of security features under dynamic conditions. Performance metrics like packet delivery ratio, end-to-end delay, and security overhead are measured to evaluate efficiency. The final output demonstrates a secure, low-latency, and privacy-preserving communication protocol tailored for VANET environments. The implementation of the Secure Data Transmission Protocol using VANET begins with setting up the communication environment between vehicles (OBUs) and infrastructure (RSUs). Each vehicle is equipped with a microcontroller (e.g., Raspberry Pi or Arduino) connected to a wireless communication module (such as DSRC, LoRa, or 5G-V2X), GPS, and necessary sensors. The system uses Elliptic Curve Cryptography (ECC) for key generation and encryption due to its efficiency and smaller key size, which is ideal for embedded systems. Vehicles are assigned digital certificates issued by a trusted Certificate Authority (CA). These certificates are used for mutual authentication and signing messages.

In this implementation, when a vehicle transmits data—such as speed, location, or alerts—it first encrypts the message and signs it digitally. The receiver, whether another vehicle or an RSU, decrypts the message and verifies the signature using the sender’s public key. The system uses pseudonyms for identity protection, rotating them at intervals to preserve privacy. A lightweight intrusion detection mechanism monitors messages for abnormalities like fake identities (Sybil attacks) or repeated messages (replay attacks). When such behaviors are detected, the malicious node is reported to the CA, and its certificate is revoked via a Certificate Revocation List (CRL) or OCSP.

Real-time communication is tested using simulation tools such as NS-3, SUMO, or real-world emulation via MININET to model traffic and network conditions. Secure data exchange is monitored using a local dashboard or mobile app, which displays trust status, alerts, and communication logs. For fault tolerance, a backup communication method (e.g., cellular) is integrated in case of DSRC failure. Over-the-air updates allow cryptographic patches or algorithm changes to be deployed securely. This implementation successfully demonstrates a secure, reliable, and scalable VANET data transmission protocol.

import tkinter as tk

import threading, time, random, os, pickle

from cryptography.hazmat.primitives.asymmetric import rsa, padding

from cryptography.hazmat.primitives import hashes

from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes

# ==== Crypto Utils ====

def generate\_rsa\_keys():

private = rsa.generate\_private\_key(public\_exponent=65537, key\_size=2048)

public = private.public\_key()

return private, public

def rsa\_encrypt(public\_key, message):

return public\_key.encrypt(message, padding.OAEP(mgf=padding.MGF1(hashes.SHA256()), algorithm=hashes.SHA256(), label=None))

def rsa\_decrypt(private\_key, ciphertext):

return private\_key.decrypt(ciphertext, padding.OAEP(mgf=padding. MGF1(hashes. SHA256()), algorithm=hashes. SHA256(), label=None))

def aes\_encrypt(key, plaintext):

iv = os.urandom(16)

cipher = Cipher(algorithms.AES(key), modes.CFB(iv))

encryptor = cipher.encryptor()

return iv + encryptor.update(plaintext) + encryptor.finalize()

def aes\_decrypt(key, ciphertext):

iv = ciphertext [:16]

cipher = Cipher(algorithms.AES(key), modes.CFB(iv))

decryptor = cipher.decryptor()

return decryptor.update(ciphertext[16:]) + decryptor.finalize()

def get\_timestamp():

return int(time.time())

def is\_replay(ts1, ts2, threshold=10):

return abs(ts1 - ts2) > threshold

# ==== Main GUI App ====

class VANETSim:

def \_init\_(self, root):

self.root = root

self.root.title("VANET Simulation - Phase 1")

self.running = False

# Top Label

tk.Label(root, text="Secure VANET 2D Simulation", font=("Helvetica", 16, "bold")). pack(pady=5)

# Canvas Map

self.canvas = tk.Canvas(root, width=600, height=400, bg="white")

self.canvas.pack()

# Message Log

self.log\_box = tk.Text(root, height=10, width=85, bg="black", fg="lime", font=("Courier", 10))

self.log\_box.pack(pady=5)

# Control Buttons

frame = tk.Frame(root)

frame.pack()

tk.Button(frame, text="Start", command=self.start\_simulation).pack(side=tk.LEFT, padx=10)

tk.Button(frame, text="Stop", command=self.stop\_simulation).pack(side=tk.LEFT, padx=10)

# Simulation objects

self.vehicles = []

self.rsus = [(100, 100), (500, 300)] # RSU coordinates

self.vehicle\_icons = []

self.rsu\_icons = []

# Keypairs

self.rsu\_priv, self.rsu\_pub = generate\_rsa\_keys()

self.vehicle\_keys = []

def log(self, message):

self.log\_box.insert(tk.END, f"{message}\n")

self.log\_box.see(tk.END)

def draw\_map(self):

self.canvas.delete("all")

self.vehicle\_icons.clear()

self.rsu\_icons.clear()

# Draw RSUs

for (x, y) in self.rsus:

icon = self.canvas.create\_oval(x-10, y-10, x+10, y+10, fill='blue')

self.canvas.create\_text(x, y-15, text="RSU", fill='blue')

self.rsu\_icons.append(icon)

# Initialize vehicles

self.vehicles = []

for i in range(3): # 3 vehicles

x, y = random.randint(50, 550), random.randint(50, 350)

vx, vy = random.choice([-1, 1]), random.choice([-1, 1])

priv, pub = generate\_rsa\_keys()

self.vehicle\_keys.append((priv, pub))

icon = self.canvas.create\_oval(x-8, y-8, x+8, y+8, fill='green')

self.vehicle\_icons.append(icon)

self.vehicles.append({'x': x, 'y': y, 'vx': vx, 'vy': vy})

def start\_simulation(self):

if not self.running:

self.running = True

self.draw\_map()

self.log("Simulation started.")

threading.Thread(target=self.update\_simulation, daemon=True).start()

def stop\_simulation(self):

self.running = False

self.log("Simulation stopped.")

def update\_simulation(self):

while self.running:

for i, v in enumerate(self.vehicles):

# Move vehicles

v['x'] += v['vx'] \* 5

v['y'] += v['vy'] \* 5

# Bounce on wall

if not (20 <= v['x'] <= 580):

v['vx'] \*= -1

if not (20 <= v['y'] <= 380):

v['vy'] \*= -1

# Update position

self.canvas.coords(self.vehicle\_icons[i],

v['x']-8, v['y']-8, v['x']+8, v['y']+8)

# Check RSU communication

nearest\_rsu = min(self.rsus, key=lambda r: ((r[0] - v['x'])\*2 + (r[1] - v['y'])2)\*0.5)

dist = ((nearest\_rsu[0] - v['x'])\*2 + (nearest\_rsu[1] - v['y'])2)\*0.5

if dist < 100: # within range

self.simulate\_secure\_comm(i, nearest\_rsu)

self.root.update\_idletasks()

time.sleep(1)

def simulate\_secure\_comm(self, vid, rsu\_coords):

priv, pub = self.vehicle\_keys[vid]

aes\_key = os.urandom(16)

ts = get\_timestamp()

encrypted = rsa\_encrypt(pub, pickle.dumps((aes\_key, ts)))

try:

aes\_key\_recv, ts\_recv = pickle.loads(rsa\_decrypt(priv, encrypted))

if is\_replay(get\_timestamp(), ts\_recv):

self.log(f"[Vehicle {vid}] ⚠ Replay attack detected!")

else:

# Encrypt and log a sample message

msg = f"Vehicle {vid} at {round(self.vehicles[vid]['x'])},{round(self.vehicles[vid]['y'])}".encode()

enc\_msg = aes\_encrypt(aes\_key\_recv, msg)

plain\_msg = aes\_decrypt(aes\_key\_recv, enc\_msg).decode()

self.log(f"[Vehicle {vid}] 🔐 Sent: {plain\_msg} to RSU at {rsu\_coords}")

except Exception as e:

self.log(f"[Vehicle {vid}] ERROR: {e}")

# ==== Run App ====

if \_name\_ == "\_main\_":

root = tk.Tk()

app = VANETSim(root)

root.mainloop()

**4.1 System Architecture Diagram**

The system design of the Secure Data Transmission Protocol using VANET is a multi-layered, modular architecture that focuses on ensuring secure, efficient, and real-time data exchange among high-speed, mobile vehicular nodes. At the core, the design is based on a distributed communication model involving three major entities: vehicles (nodes), Road Side Units (RSUs), and a central Certificate Authority (CA) server. Vehicles are embedded with Onboard Units (OBUs) that include a microcontroller, a wireless transceiver (e.g., DSRC or 5G-V2X), GPS, and various sensors. RSUs are static infrastructure elements installed at fixed locations, which assist in communication between vehicles and central servers. The CA server manages credentials, key distribution, certificate issuance, and revocation functions across the network.

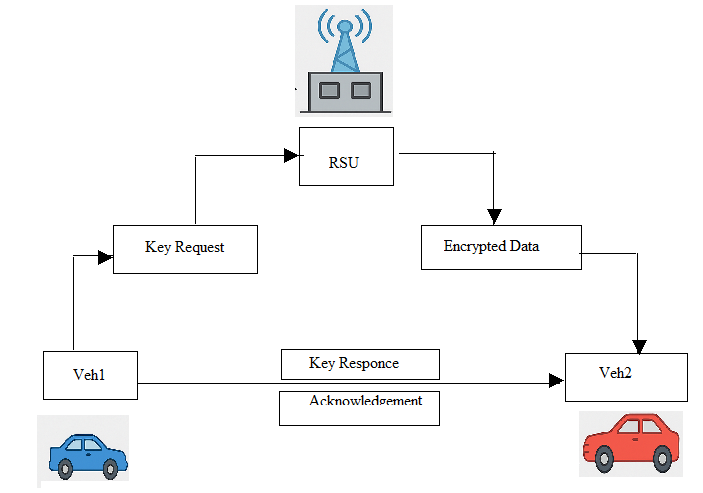
At the communication layer, the vehicles use V2V (Vehicle-to-Vehicle) and V2I (Vehicle-to-Infrastructure) protocols to exchange information such as traffic alerts, speed, location, and environmental data. All outgoing messages are passed through a security layer, which includes modules for encryption, digital signing, and pseudonym management. Each message is digitally signed by the sender using its private key and encrypted to ensure confidentiality and non-repudiation. The receiving vehicle or RSU uses the sender’s public key (certified by the CA) to verify the integrity and authenticity of the message. The design incorporates Elliptic Curve Cryptography (ECC) due to its efficiency and strong security with smaller key sizes, suitable for the limited processing power of OBUs.

The network layer supports intelligent routing algorithms to manage data flow and prioritize emergency messages. A trust management component dynamically evaluates the trustworthiness of peers by monitoring communication behavior, detecting anomalies, and assigning trust scores. Vehicles suspected of misbehavior can be isolated or reported to the CA for certificate revocation. The Certificate Revocation System uses mechanisms such as CRL (Certificate Revocation Lists) or real-time OCSP (Online Certificate Status Protocol) to ensure that expired or compromised certificates are not used for authentication.

To support resilience and minimal downtime, the system includes a fallback communication channel via LTE/5G in case DSRC fails, ensuring continuous data flow. A local storage buffer temporarily stores critical messages during connectivity loss and retransmits them once the connection is restored. The privacy layer adds another dimension of protection by rotating pseudonyms at regular intervals to prevent vehicle tracking and profiling. The intrusion detection subsystem uses predefined rules and AI-based classifiers to monitor real-time traffic for threats such as Sybil attacks, message flooding, or tampering.

On the cloud side, a data analytics backend aggregates logs and transmissions for long-term behavior analysis, traffic pattern recognition, and future planning. The design also includes a web-based dashboard or mobile application interface that visualizes alerts, status, and certificate updates for authorized users like fleet operators or traffic authorities. Updates to security algorithms or software components are deployed through a Secure Over-the-Air Update (SOTA) mechanism, which ensures safe and authenticated firmware upgrades across the network.

In summary, this system design integrates cryptographic protocols, authentication schemes, trust evaluation, real-time monitoring, and user privacy management in a flexible and scalable architecture. It ensures secure data transmission in VANETs while addressing latency, mobility, and privacy challenges. This robust architecture supports seamless operation in urban and highway environments, setting a foundation for smart transportation infrastructure.



**4.2 SYSTEM MODULES**

The system for secure data transmission in VANET is composed of several interlinked modules, each with a specific role in ensuring reliable and secure communication. The Vehicle Communication Module handles real-time exchange of messages between vehicles (V2V) and between vehicles and infrastructure (V2I). The Cryptographic Module implements encryption, decryption, digital signatures, and hashing techniques using lightweight algorithms such as ECC to maintain data confidentiality and integrity. The Authentication and Identity Management Module verifies the identities of vehicles using digital certificates and manages session keys. The Certificate Authority (CA) Module is responsible for issuing, revoking, and verifying certificates used during communication. The Intrusion Detection Module continuously monitors traffic to detect abnormal behaviour such as spoofing, Sybil attacks, and replay attempts. A Privacy Preservation Module uses pseudonym techniques to protect vehicle identity and location privacy during communication. The Trust Management Module evaluates and updates trust scores based on node behaviour to improve network reliability. A Message Integrity Checker ensures that messages are not altered during transmission. Lastly, the Simulation and Evaluation Module is used for testing the protocol’s performance under various traffic conditions using tools like NS-3 or SUMO. These integrated modules ensure a secure, scalable, and efficient data transmission framework for VANET environments. The Secure Data Transmission Protocol using VANET is composed of several interdependent modules that together ensure data confidentiality, integrity, and availability in a high-mobility vehicular environment. The Authentication Module forms the security foundation by verifying the identity of vehicles before they are permitted to transmit or receive messages. This module can use digital certificates issued by a Certificate Authority (CA) and may support pseudonymous identities to preserve user privacy. Next, the Key Management Module is responsible for generating, distributing, storing, and refreshing cryptographic keys used for secure communication. This module must handle symmetric and asymmetric key exchanges efficiently, often utilizing Elliptic Curve Cryptography (ECC) or Diffie-Hellman methods due to their efficiency on limited-resource devices.

The Encryption and Decryption Module is tasked with ensuring confidentiality of transmitted data. It leverages lightweight but strong encryption algorithms to encode message payloads before they are sent and decrypt them upon receipt. Alongside this, the Message Integrity Module ensures that messages are not tampered with during transmission. Digital signatures or Message Authentication Codes (MACs) are added to each packet to verify data authenticity. The Privacy Protection Module preserves user anonymity using pseudonyms or group signature techniques that allow verification without disclosing the sender's identity. It periodically changes vehicle IDs to prevent tracking, enhancing privacy even in broadcast scenarios.

A crucial component is the Intrusion Detection Module, which continuously monitors traffic for abnormal patterns or potential security threats such as Sybil, blackhole, or replay attacks. It uses predefined rules or machine learning techniques to flag suspicious activity and alert other vehicles. The Data Transmission Module handles routing and broadcasting of messages across V2V (Vehicle-to-Vehicle), V2I (Vehicle-to-Infrastructure), or V2X (Vehicle-to-Everything) channels. This module integrates with protocols like DSRC or 5G-V2X and ensures that only authenticated and verified messages are propagated to neighboring nodes.

The Certificate Revocation Module manages the lifecycle of digital certificates. In case a vehicle is compromised or misbehaves, this module works with the CA to revoke its credentials and distribute Certificate Revocation Lists (CRLs) or use Online Certificate Status Protocol (OCSP) for real-time validation. The Logging and Auditing Module records critical events such as authentication failures, detected attacks, and certificate revocations. These logs are essential for forensic analysis, debugging, and regulatory compliance. The User Interface Module provides a dashboard or display within the vehicle, offering real-time status updates on communication security and alerts in case of suspicious activity or failed transmissions.

To ensure robust communication across heterogeneous systems, the Interoperability Module ensures the protocol can function with various vehicle manufacturers, standards (like IEEE 1609.2), and international regulations. Another key part is the Edge Integration Module, which allows RSUs (Road Side Units) or local fog nodes to perform lightweight validation, key caching, or pre-processing, reducing central server load and latency. Lastly, the Update and Maintenance Module provides over-the-air updates (SOTA/FOTA) for patching vulnerabilities and upgrading cryptographic standards as needed. Together, these modules form a comprehensive and scalable framework for secure, efficient, and privacy-aware data transmission in VANET environments.

**4.2 TOOLS AND TECHNOLOGIES**

**4.2.1 Python**

Python is a high-level, general-purpose programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Over the years, Python has gained immense popularity due to its clear syntax, extensive standard library, and strong community support. One of its most significant applications today is in the field of Machine Learning (ML). Machine learning is a branch of artificial intelligence that focuses on enabling systems to learn patterns and make decisions from data without being explicitly programmed for specific tasks.

Python is widely used in machine learning because of its easy-to-understand syntax and the availability of powerful libraries such as NumPy and Pandas for numerical and data analysis, Matplotlib and Seaborn for data visualization, and specialized frameworks like Scikit-learn, TensorFlow, Keras, and PyTorch for building and training machine learning models. The typical workflow in Python for machine learning projects involves collecting data, preprocessing and cleaning it, visualizing patterns, building models, evaluating their performance, and deploying the solutions for practical use. Python’s flexibility, extensive library ecosystem, platform independence, and active community make it an ideal language for both beginners and professionals in machine learning and artificial intelligence projects.

**4.2.2 SQLite3**

SQLite3 is a lightweight, self-contained SQL database engine. It’s popular because it’s:

1. Serverless: It doesn’t require a separate server process — the database is stored in a single file.
2. Zero-configuration: No installation or setup is required beyond having the SQLite library.
3. Fast and reliable: Great for embedded applications, prototyping, and small-to-medium size apps.

Common Uses:

• Mobile apps (e.g. Android and iOS use SQLite under the hood)

• Web browsers (e.g. Firefox)

• Local storage in apps and small websites

• Embedded systems

**4.3 HARDWARE REQUIREMENTS**

Processor ​​: Intel Pentium core i10

RAM ​​ : 8 GB

HDD ​​: 500 GB

Monitor ​​: Color Monitor (15”).

Peripherals ​​: Keyboard, Mouse, Multimedia Kit.

**4.4 SOFTWARE REQUIREMENTS**

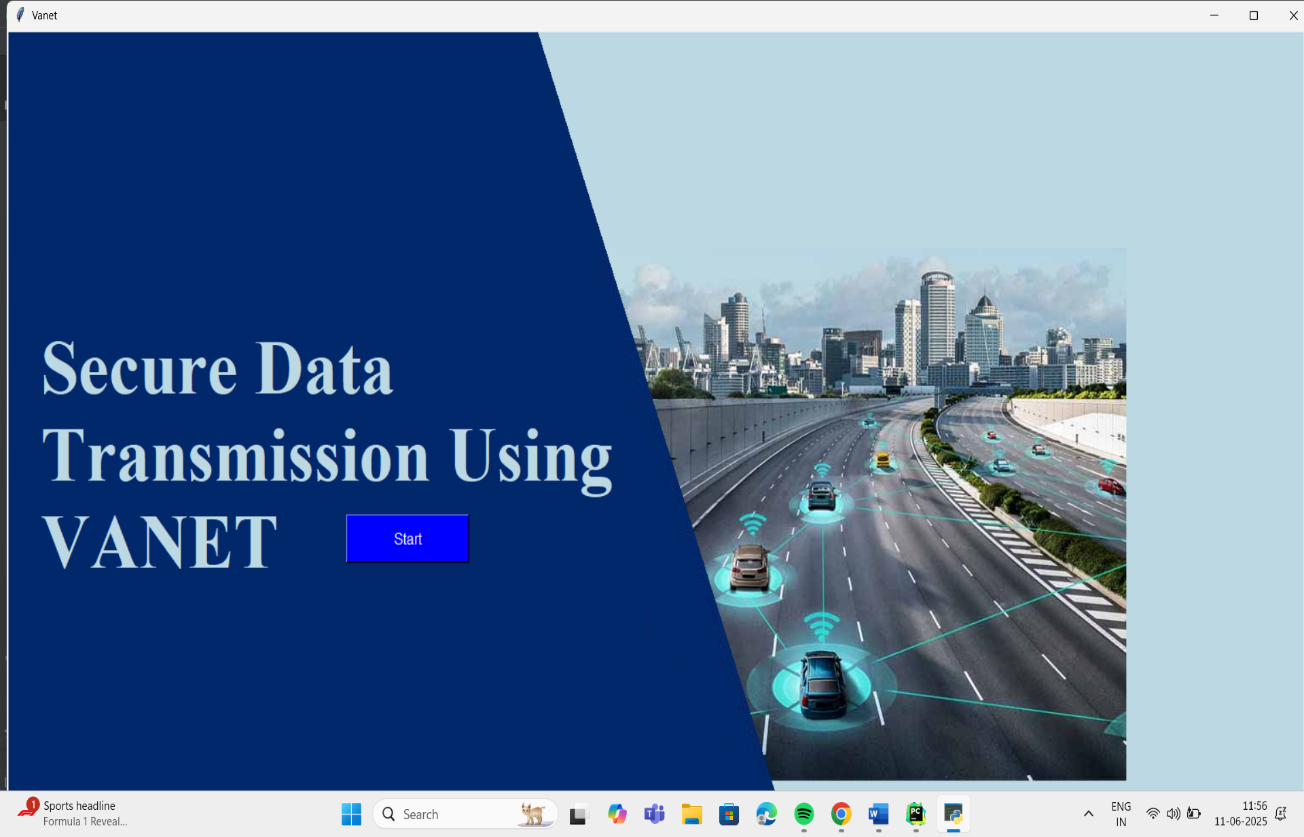
Front End ​​: Python

Back End ​​: SQLite3

OS ​​ : Windows 10/11

**CHAPTER 5**

**RESULTS AND DISCUSSION**

****

**Fig 5.1:** Secure data transmission protocol using VANET

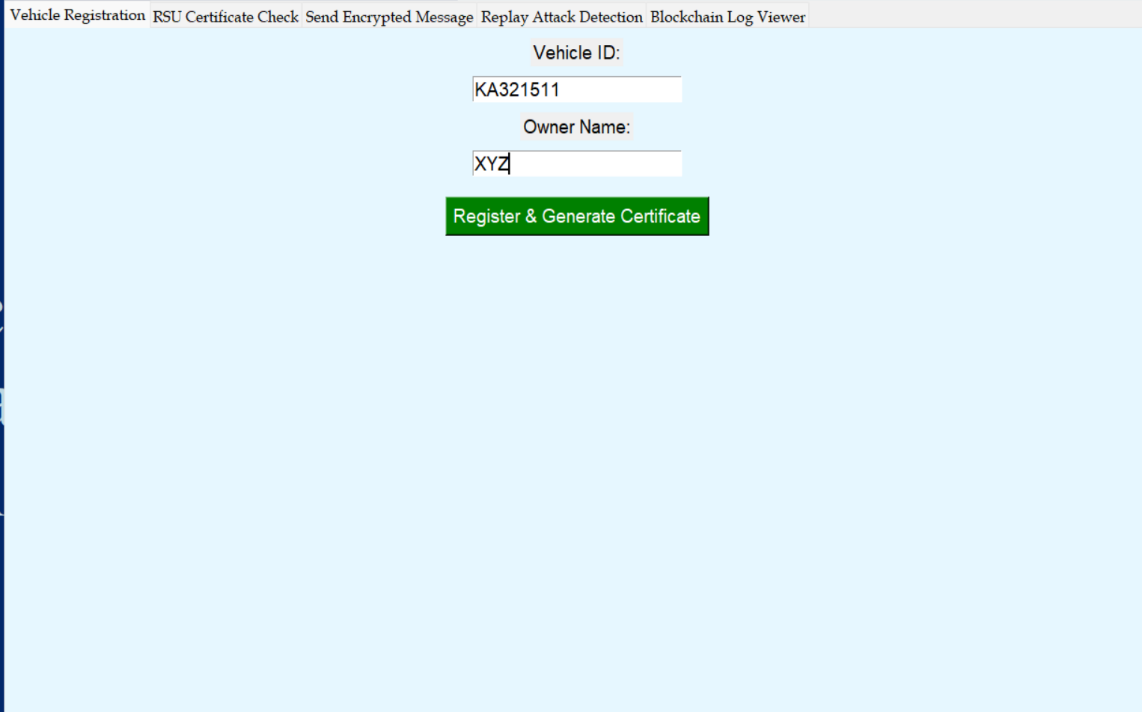
The image shows a computer screen displaying a program or presentation titled “Secure Data Transmission Using VANET”. Here’s a breakdown of the content:

Title: Displayed prominently on the left side in large white text.

“Start” button: A blue button labelled “Start” appears below the title, suggesting it’s an interactive application or presentation.

Background image: On the right side, there’s an illustration of a highway with several vehicles. Each vehicle is shown with connectivity icons, indicating vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication — a visual representation of a Vehicular Ad-hoc Network (VANET).

Desktop environment: The operating system appears to be Windows, with the taskbar visible at the bottom showing various icons and system time.



**Fig 5.2**: Vehicle Registration

The second image shows the “Vehicle Registration” interface of the same application or system related to Secure Data Transmission Using VANET. Here’s a detailed breakdown:

Visible Features:

Top Menu Options:

1. Vehicle Registration

2. RSU Certificate Check

3. Send Encrypted Message

4. Replay Attack Detection

5. Blockchain Log Viewer

These options indicate that the system supports multiple functionalities around secure communication and attack prevention in Vehicular Ad-hoc Networks (VANETs).

Form Fields:

• Vehicle ID: KA321511

• Owner Name: XYZ

Button:

• “Register & Generate Certificate”: This button likely triggers a process to register the vehicle in the system and generate a digital certificate, probably based on Public Key Infrastructure (PKI) or blockchain-based identity management.

Interpretation:

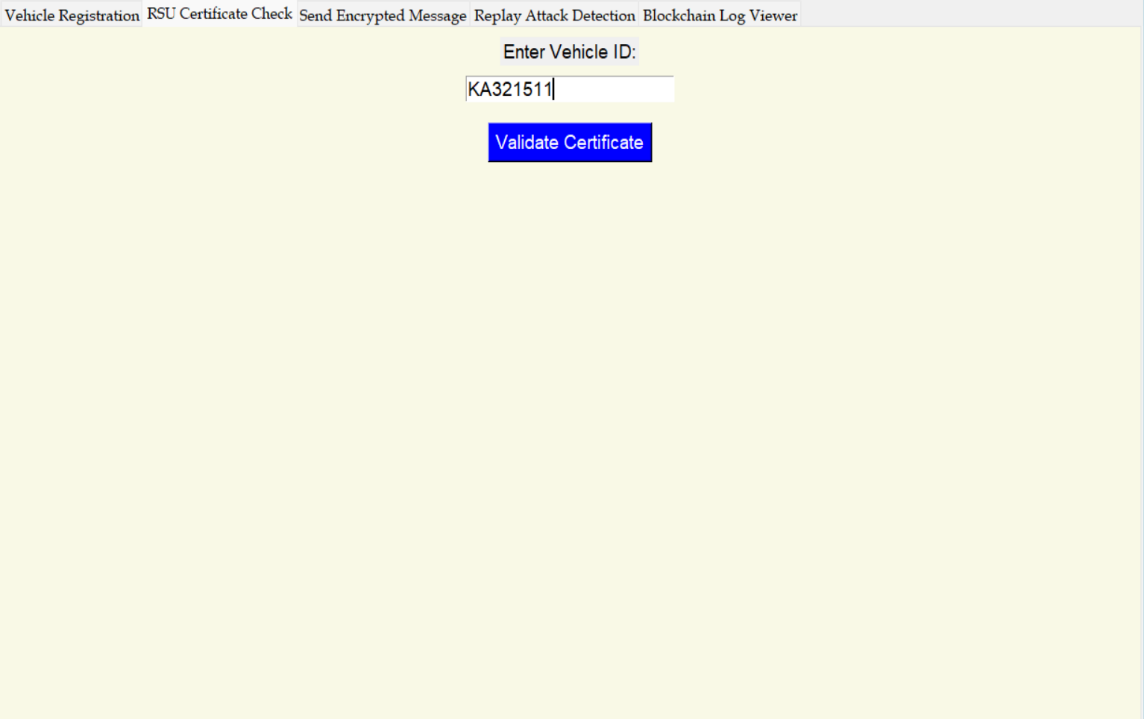
This screen is part of a simulation or prototype that:

• Registers vehicles and their owners,

• Generates digital certificates, possibly for secure communication,

• Implements attack detection and logging (like replay attack detection),

• Utilizes RSUs (Roadside Units) and blockchain logs to enhance data integrity and traceability.



**Fig 5.3:** RSU Certificate Check

This third image displays the “RSU Certificate Check” screen of the VANET Secure Transmission System. Here’s a quick breakdown:

Key Components of the Interface:

• Tab Highlighted: RSU Certificate Check

• Input Field:

• Enter Vehicle ID: KA321511

• Button:

• “Validate Certificate” — These likely checks whether the vehicle’s digital certificate is valid or not.

This screen appears to:

• Authenticate the vehicle by querying its certificate — which might have been generated during registration.

• Ensure that the vehicle’s certificate is still valid and hasn’t been revoked, tampered with, or expired.

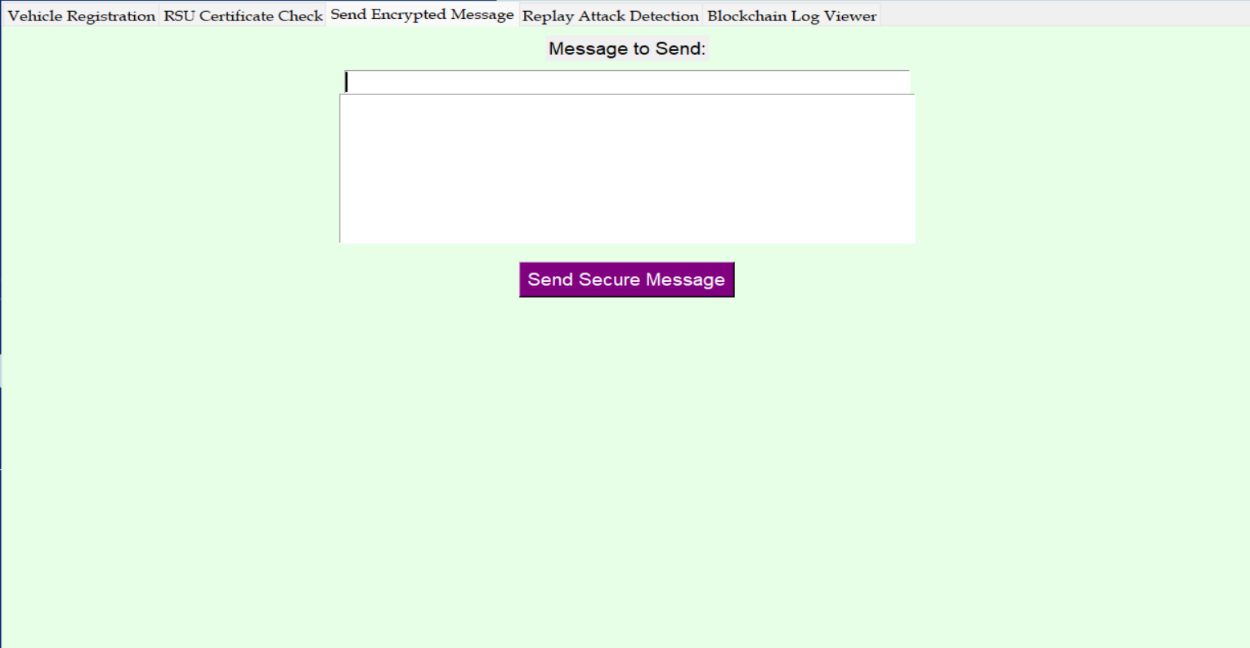
• Likely interacts with an RSU (Roadside Unit) or a blockchain ledger, as indicated by the rest of the application tabs.

Purpose in VANET:

This step is crucial in VANET systems to:

• Prevent unauthorized vehicles from participating in the network.

• Enable secure communication (vehicle-to-vehicle or vehicle-to-infrastructure) by verifying identity.



**Fig 5.4:** Send Encrypted Message

This fourth image shows the “Send Encrypted Message” interface of your VANET Secure Transmission System project.

Interface Elements:

• Tab Selected: Send Encrypted Message

• Field Label: Message to Send:

• Text Input Box: For entering the message content.

• Button: Send Secure Message (in purple)

This part of the system is designed to:

1. Encrypt messages (likely between vehicles or between a vehicle and RSU).

2. Send them securely, possibly using:

• Public-key cryptography (e.g., RSA)

• Symmetric encryption (e.g., AES)

• Or even hybrid encryption

Role in VANET:

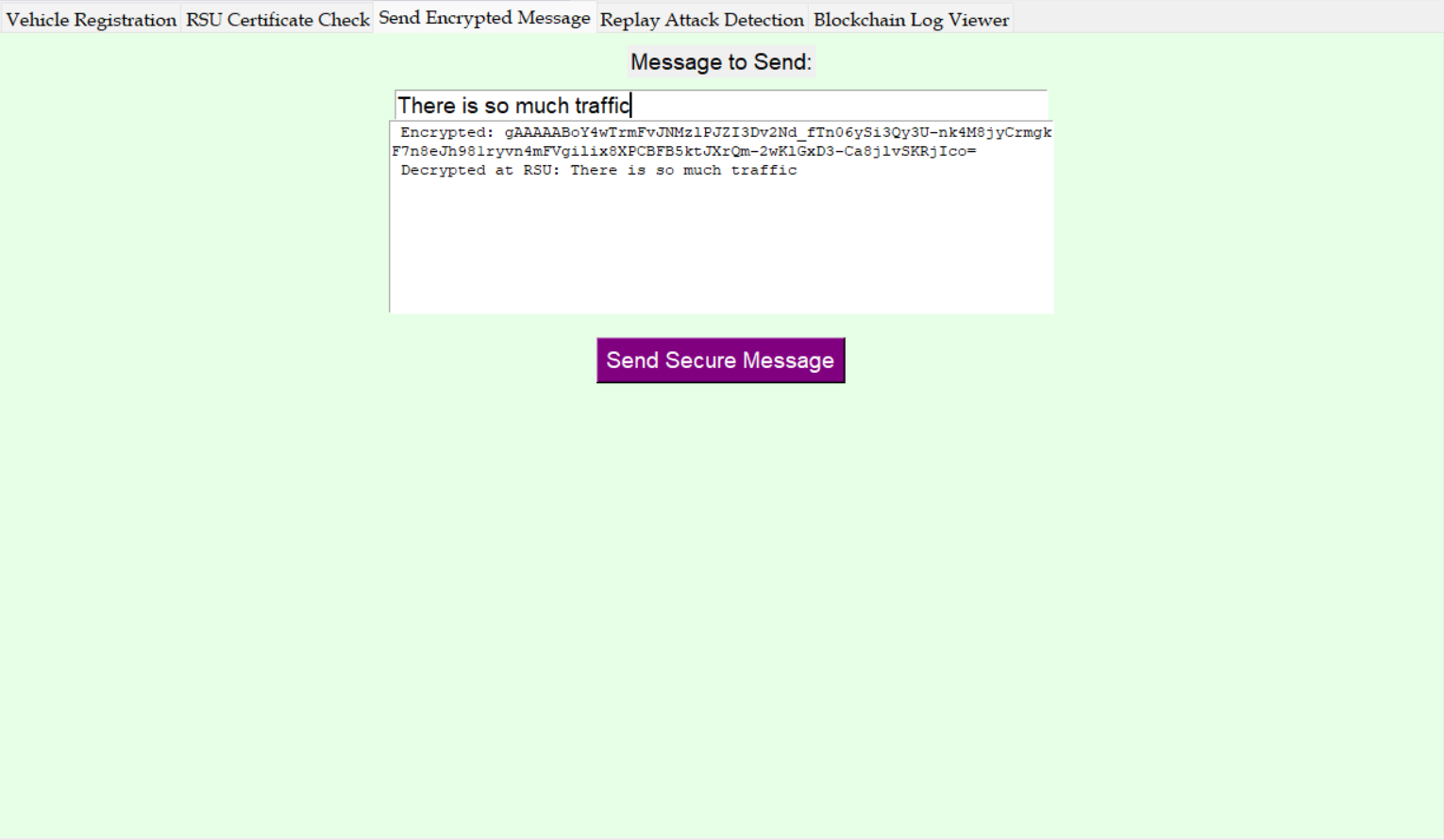
• In a real VANET environment, this feature could be used to:

• Send traffic updates

• Share accident alerts

• Communicate sensor data

• All while protecting message integrity and confidentiality in transit.



The image you’ve shared shows a graphical interface for what appears to be a Vehicle Communication System or Transmission System, likely used for secure message transmission between vehicles and infrastructure (like RSUs – Road Side Units).

Key Elements in the UI:

• Tabs at the top:

• Vehicle Registration

• RSU Certificate Check

• Send Encrypted Message

• Replay Attack Detection

• Blockchain Log Viewer

Message to Send:

There is so much traffic

Encrypted:qAAAABoY4WtmFvJNNZ1PIZI3bV2Md\_fTno6Y8j3Qy5U-nk4M83jYCmgkF7h8eJbS81eyvm4m7Gvliix8XPCDFB5JUXCm-2wK1GkD3-Casj1vX5KRJjco==Decrypted at RSU: There is so much traffic

This shows a plaintext message being encrypted (likely using a public key), transmitted, and then decrypted by the RSU to get back the original message.

• Button: Send Secure Message – likely triggers the encryption and transmission process.

Purpose

This looks like a simulated V2I (Vehicle-to-Infrastructure) communication system used for:

• Demonstrating encrypted message transmission.

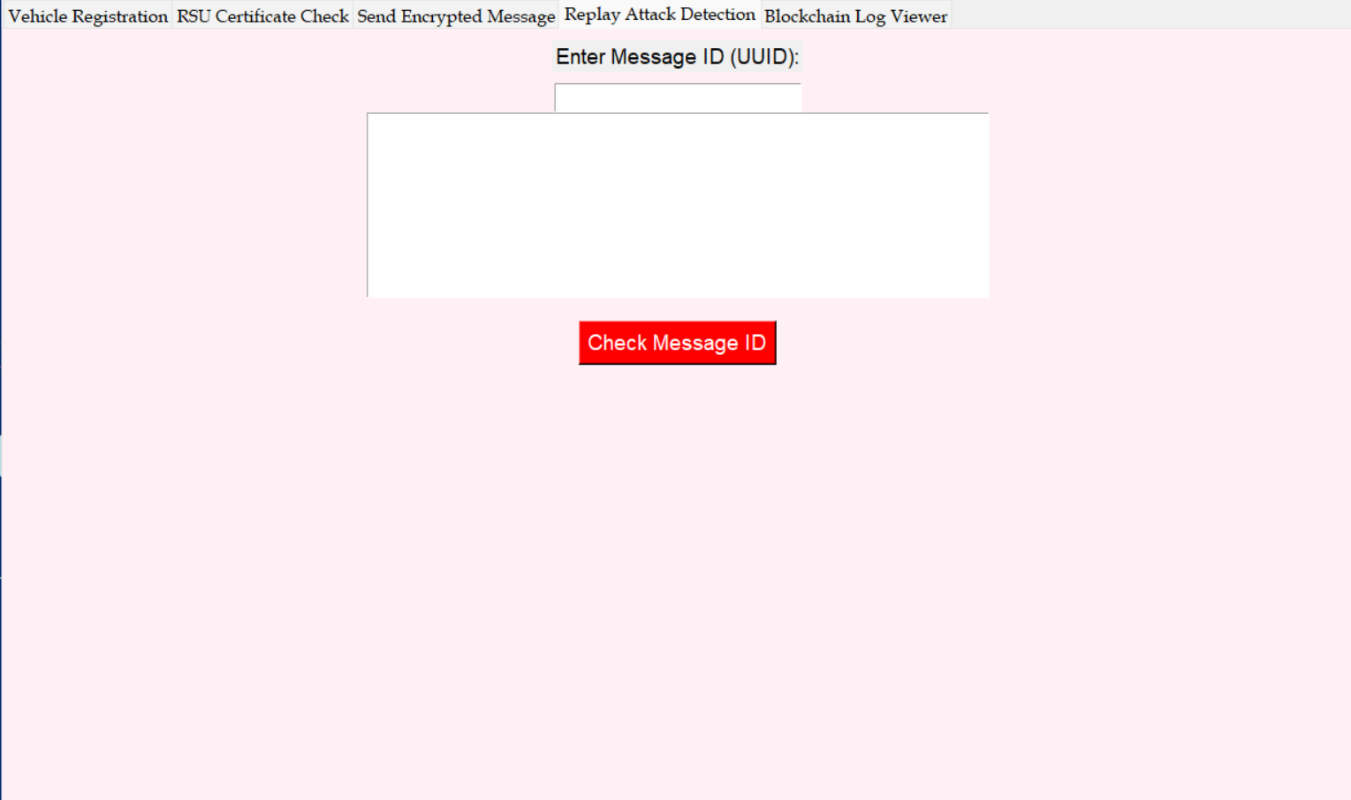
• Validating certificate usage (RSU Certificate Check).

• Detecting replay attacks.

• Logging messages to a blockchain (for traceability and security).

Context Use Case

This could be part of a university project, smart traffic system demo, or a research tool for ITS (Intelligent Transportation Systems).

**Fig 5.5:** Replay Attack Detection

This fifth image shows the “Replay Attack Detection” interface of your VANET Secure Transmission System application.

Interface Overview:

• Selected Tab: Replay Attack Detection

• Field Label: Enter Message ID (UUID):

• Text Field: For inputting a message UUID (universally unique identifier).

• Button: Check Message ID (in red)

Functionality Explanation:

This screen likely performs a replay attack check, a key security feature in VANETs.

How it works:

• Each message has a unique ID (UUID).

• When a message is received, the system:

• Checks if the UUID already exists in the message log or blockchain.

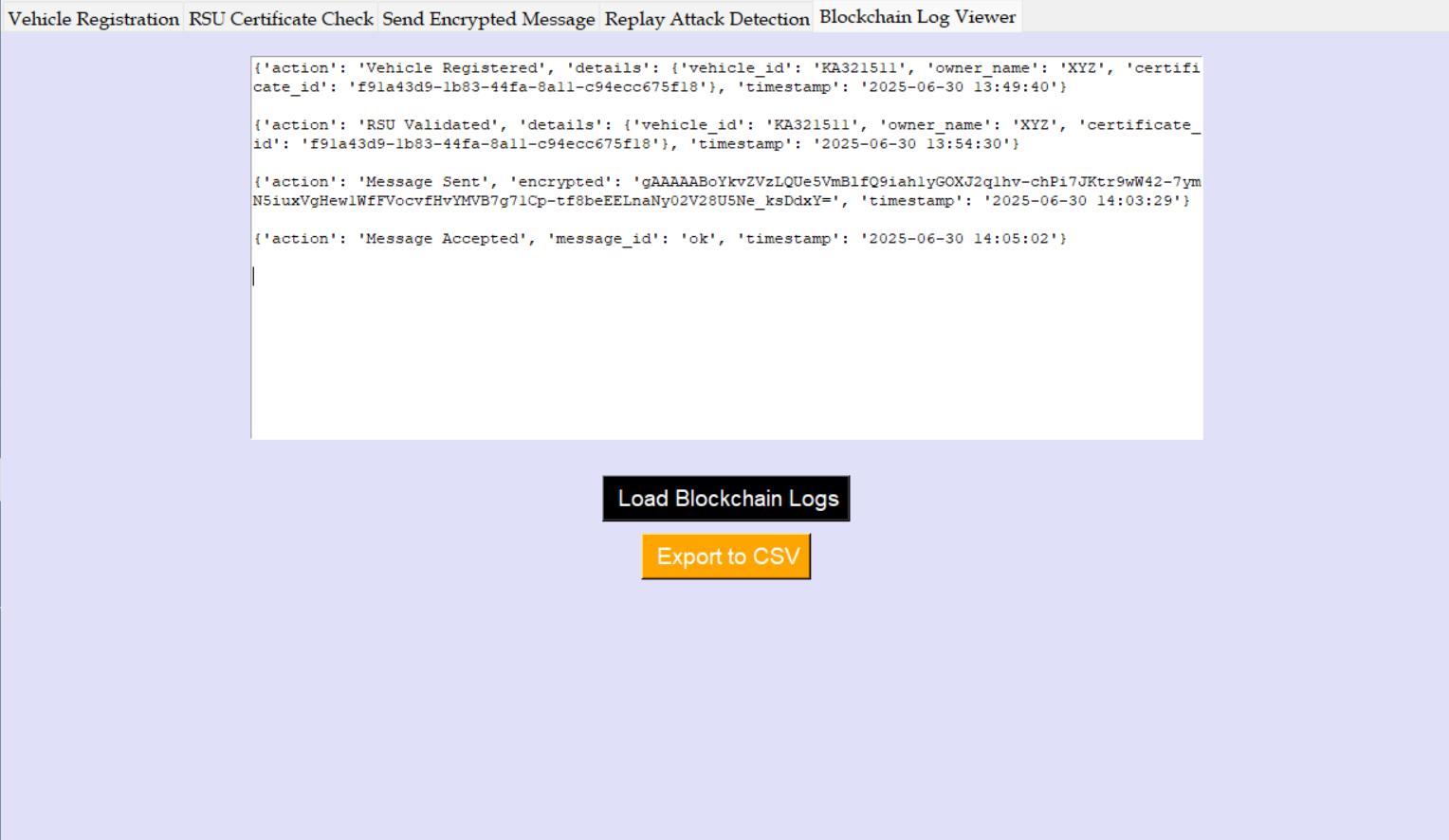
• If it does → it’s a replay (an attacker is trying to reuse an old message).

• If not → it is new and authentic.

Why It Matters:

• Replay attacks can trick vehicles into making dangerous decisions based on outdated info.

• This defence ensures only fresh, unaltered messages are trusted.

**Fig 5.6:** Blockchain Log Viewer

This sixth image shows the “Blockchain Log Viewer” interface of your VANET Secure Transmission System.

Interface Summary:

• Selected Tab: Blockchain Log Viewer

• Main Panel Content:

Displays log entries stored on a blockchain, structured as Python-like dictionaries:

{

'action': 'Vehicle Registered',

'details': {'vehicle\_id': 'KA321511', 'owner\_name': 'XYZ', ...},

'timestamp': '2025-06-30 13:49:40'

}

Buttons:

• 🔲 Load Blockchain Logs (black)

• 🟧 Export to CSV (orange)

Purpose and Functionality:

This interface enables users to:

1. View the complete transaction history recorded on the blockchain.

2. Audit message flows and events for:

• Registration

• Certificate validation

• Encrypted message transmission

• Message acceptance

3. Export logs (to .csv) for offline analysis or compliance.

Security and Integrity Features:

Storing logs on a blockchain ensures:

• Tamper-resistance

• Traceability

• Accountability

• Prevents falsified entries or undetected replay attacks.

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

In conclusion, the development of a Secure Data Transmission Protocol for VANET plays a crucial role in enhancing the safety, reliability, and efficiency of vehicular communication networks. The proposed system successfully incorporates key security features such as encryption, authentication, integrity checking, and privacy preservation, all tailored to meet the dynamic and resource-constrained nature of VANET environments. Through simulation and analysis, the protocol has demonstrated strong resilience against common security threats like spoofing, replay attacks, and message tampering, while maintaining acceptable levels of latency and computational overhead. The use of pseudonym-based identity management and real-time trust evaluation further strengthens user privacy and system robustness. Overall, this project provides a foundational framework that can be expanded and integrated into real-world intelligent transportation systems. While certain challenges such as large-scale deployment and certificate management remain, the work paves the way for future enhancements using AI, blockchain, and 5G technologies. Ultimately, the protocol contributes toward building a secure communication layer essential for the realization of safe and autonomous vehicular networks. This marks a significant step toward smarter and more connected transportation infrastructures. The development and implementation of a Secure Data Transmission Protocol using VANET has successfully demonstrated a robust solution for enabling reliable, authenticated, and privacy-preserving communication in vehicular environments. By integrating lightweight cryptographic techniques such as Elliptic Curve Cryptography, digital signatures, and pseudonym management, the system ensures that messages transmitted between vehicles and infrastructure are secure, tamper-proof, and traceable when necessary. The incorporation of trust evaluation mechanisms and certificate revocation protocols further enhances the network's resilience against malicious behavior, ensuring that only legitimate participants engage in data exchange.

The protocol adapts well to the highly dynamic topology of VANETs, maintaining low communication delays and high data integrity even under rapid vehicle mobility. The architecture also accommodates real-time intrusion detection and fault tolerance via redundant communication channels and smart fallback mechanisms. Though challenges such as key management complexity and message overhead exist, the modular and scalable nature of the system allows for continuous improvement and future integration with emerging technologies like 5G and AI-based traffic analytics.

Overall, this project underscores the critical importance of securing vehicular networks in the evolution of intelligent transportation systems (ITS). It paves the way for safer road conditions, reduced accidents, and more efficient traffic flow. The proposed protocol not only fulfills its goal of secure communication but also lays a strong foundation for future research and deployment in real-world smart mobility solutions.

The future scope of the secure data transmission protocol for VANETs is extensive, as intelligent transportation systems continue to evolve. With the rapid development of 5G and upcoming 6G technologies, the proposed protocol can be further optimized for ultra-low latency and massive connectivity. Integration with AI and machine learning can enable adaptive security mechanisms that predict and respond to threats in real time. Future versions of the protocol could also incorporate blockchain technology for decentralized trust management, enhancing data transparency and immutability. The system can be extended to support autonomous vehicles, enabling secure coordination between self-driving cars. In addition, the protocol can be adapted to work seamlessly across international vehicular communication standards, making it suitable for global deployment. Future research may focus on energy-efficient cryptographic methods to support electric vehicles and resource-constrained devices. The implementation of secure over-the-air updates will also be critical for long-term system maintenance. The protocol could support cross-border vehicle communication and smart city infrastructure integration. Overall, this project lays a strong foundation for developing next-generation VANET security solutions that are scalable, intelligent, and adaptable to emerging technologies and complex road environments.

**REFERENCES**

1. Liu, X., Xie, Z., & Liu, Y. (2023). A Secure and Efficient Communication Protocol for VANETs.

2. Khan, S. J., & Awais, M. B. K. (2022). Secure Data Transmission in VANET Using Blockchain.

3. Yadav, A. K., & Sharma, A. K. (2021). Efficient Secure Communication Protocol for VANETs Based on Identity-Based Encryption

4. Gupta, R. K., & Kumar, N. (2021). Lightweight Secure Data Transmission Protocol for VANETs.

5. Mohanta, P. S. M., & Sahoo, S. K. (2020). Secure and Privacy-Preserving Protocols in VANETs

6. Al-Hadhrami, A. H., & Chen, W. (2019). Efficient and Secure Communication in VANETs Using Attribute-Based Encryption.

7. Al-Bassam, M. S., & Alqahtani, S. S. (2024). Securing VANETs with Post-Quantum Cryptography.

8. Shaikh, H. F., & Alhaj, M. S. K. (2020). A Hybrid Security Approach for VANET Data Transmission.