Secure Data Transmission Protocol Using VANET

Prof. Nandini S Patil[1], Soujanya[2], Priya[3], Kaveri[4]

Dept Of CSE,FETW, Sharnbasva University, Kalaburagi,India Nandinipatil.08@Gmail.Com

Dept Of CSE,FETW, Sharnbasva University, Kalaburagi,India

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) play a crucial role in enabling intelligent transportation systems through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. However, the dynamic and decentralized nature of VANETs makes them highly vulnerable to security threats such as eavesdropping, message tampering, and replay attacks. This project presents a Python-based implementation of a secure data transmission protocol tailored for decentralized VANET environments, incorporating robust mechanisms for authentication, confidentiality, integrity, and replay attack prevention. The proposed system utilizes RSA public-key cryptography for secure key exchange and AES symmetric encryption for efficient data protection during transmission. A simulated Road Side Unit (RSU) acts as a trusted authenticator that verifies vehicle identities and decrypts messages. To safeguard against replay attacks, each transmitted message includes a timestamp and nonce, which are validated at the RSU upon receipt. The system is integrated into a GUI built using Tkinter, featuring modules for vehicle registration, certificate validation, encrypted message exchange, replay detection, and blockchain-based logging. Experimental results confirm that the system performs real-time encryption and decryption efficiently, accurately detects replay attacks, and maintains low transmission latency. The modular architecture and visual interface make the solution suitable for testing, research, and future deployment in secure VANET communication frameworks..

Keywords— **Vehicular Ad Hoc Networks (Vanets), RSA, AES, Road Side Unit (RSU), VANET Security**

I. INTRODUCTION

With the rapid advancement of intelligent transportation systems, Vehicular Ad Hoc Networks (VANETs) have emerged as a vital component in enabling communication among moving vehicles and between vehicles and infrastructure. VANETs allow for real-time sharing of traffic conditions, accident alerts, navigation data, and other critical information that improves road safety, reduces congestion, and enhances driving comfort. However, due to their open and decentralized nature, VANETs are vulnerable to numerous security threats such as message tampering, impersonation, eavesdropping, and replay attacks.

Securing data transmission in such a dynamic and delay-sensitive environment presents unique challenges. Ensuring that messages are sent confidentially, received from authentic sources, and not altered or resent maliciously is essential for the safe deployment of VANET systems. Conventional cryptographic techniques must be adapted for high-speed, mobile environments without incurring significant overhead.

This project addresses these concerns by proposing a secure data transmission protocol implemented in Python. The protocol uses RSA encryption for secure key exchange, AES symmetric encryption for efficient message confidentiality, and timestamp-based validation to prevent replay attacks. A simulated Road Side Unit (RSU) functions as a trusted authority to authenticate vehicles and facilitate secure key distribution. Communication is handled using UDP socket programming, simulating real-time V2V and V2I data exchanges, with a lightweight GUI (using Tkinter) to demonstrate message flow and security enforcement.

This work aims to strike a balance between security, efficiency, and real-time performance, making it suitable for deployment in real-world VANET environments where both reliability and speed are paramount..

II. LITERATURE SURVEY

Recent research efforts in Vehicular Ad Hoc Networks (VANETs) have concentrated on enhancing the security and efficiency of communication protocols, considering the unique challenges of high mobility, dynamic topology, and real-time constraints. Liu et al. (2023) introduced a secure and efficient communication protocol that combines RSA encryption with elliptic curve cryptography (ECC) for authentication, facilitating both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. However, the protocol's performance tends to degrade under high-traffic conditions due to its computational overhead. Khan and Awais (2022) explored the integration of blockchain technology for secure data transmission in VANETs. Their blockchain-based model ensures data integrity and tamper resistance through smart contracts, although it suffers from increased latency and computational cost, especially with a large number of participating nodes.

Yadav and Sharma (2021) proposed an identity-based encryption (IBE) protocol that simplifies key management by using vehicle identity as a public key. While it reduces distribution complexity, IBE still incurs notable computational load and may not scale efficiently.

Gupta and Kumar (2021) focused on resource-constrained environments and developed a lightweight secure transmission protocol using hybrid key management. Though energy-efficient, the solution has limitations in scalability and adaptability in dynamic VANET settings. Mohanta and Sahoo (2020) addressed both security and privacy preservation, suggesting a hybrid approach with digital signatures and anonymous authentication for secure data exchange. However, their solution could lead to communication delays, hindering real-time interactions.

Al-Hadhrami and Chen (2019) explored attribute-based encryption (ABE) for access control, allowing only authorized vehicles to decrypt certain messages. Despite its granularity, ABE's high computational demand poses a challenge for practical, real-time VANET deployment. In an effort to future-proof VANETs, Al-Bassam and Alqahtani (2024) proposed using post-quantum cryptography (PQC) algorithms such as lattice-based and hash-based methods to defend against quantum attacks. While offering long-term security, PQC approaches currently suffer from high computational requirements that make real-time implementation difficult.

Shaikh and Alhaj (2020) designed a hybrid security model that merges AES for fast encryption with RSA for key exchange. This model balances speed and security but must carefully manage computational load, especially in low-power vehicular devices.

.

III. PROPOSED SYSTEM

The proposed system is a secure communication framework for Vehicular Ad Hoc Networks (VANETs) that ensures confidentiality, authentication, integrity, and replay attack prevention during vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. Developed in Python, the system employs a hybrid cryptographic approach combining RSA public-key encryption for secure key exchange and AES symmetric encryption for efficient and lightweight message encryption. A simulated Road Side Unit (RSU) acts as a centralized certificate authority and key distributor, validating vehicle identities and decrypting incoming messages.

The system is designed with a multi-tab Graphical User Interface (GUI) using Tkinter, offering functionalities such as vehicle registration, certificate verification, secure message sending, replay attack detection, and blockchain-based log viewing. During message transmission, the vehicle generates a random AES session key to encrypt the message. This AES key is then encrypted with the RSU’s public RSA key and transmitted alongside the encrypted message. To prevent replay attacks, each message also includes a timestamp and nonce, which the RSU verifies upon receipt.

When a message is received, the RSU decrypts the AES key using its private RSA key, decrypts the message, and verifies its freshness and uniqueness using the timestamp and nonce. The decrypted message and its security status are displayed in real time within the GUI, ensuring transparency and user awareness. Additionally, secure logs can be maintained and optionally integrated into a blockchain ledger for tamper-proof record keeping. This architecture provides a scalable and practical solution for real-time secure data exchange in VANETs while maintaining low computational overhead and high responsiveness, making it suitable for intelligent transportation systems...

IV. METHODOLOGY

The proposed secure VANET data transmission system is built on a Python-based simulation framework that emulates secure communication between vehicles (V2V) and Road Side Units (V2I). The architecture ensures data confidentiality, authentication, integrity, and replay attack prevention through the combination of RSA and AES cryptographic algorithms, supported by a multi-tab GUI for operational control and real-time monitoring.

1. Vehicle Registration and Certificate Handling:

Each vehicle registers through the Vehicle Registration tab in the GUI.

Upon registration, an RSA key pair (public/private) is generated.

The vehicle’s public key is stored in a local database, and a certificate check can be initiated via the RSU Certificate Check tab.

2. Key Exchange and Message Preparation:

When a secure message is to be sent, the system uses RSA for secure session key exchange.

A random AES symmetric key is generated for encrypting the actual message.

This AES key is then encrypted using the RSU's public RSA key and attached to the outgoing message packet.
3. Encryption and Transmission:
The user enters the message in the "Message to Send" text area.
The message is AES-encrypted, and the encrypted form is shown in the output pane (as in your screenshot: a long base64-encoded string).
A timestamp and nonce are appended to prevent replay attacks.
4. Decryption and Replay Attack Detection at RSU:
The simulated RSU (Road Side Unit) receives the encrypted message, decrypts the AES key using its private RSA key, and then decrypts the actual message.
It checks:
Timestamp validity (e.g., message freshness within ±5 seconds).
Uniqueness of the nonce, rejecting repeated ones.
If valid, the decrypted message (e.g., "Next more traffic is there") is displayed in the GUI under "Decrypted at RSU".
5. Real-time Monitoring and Log Visualization:
Additional GUI tabs such as Replay Attack Detection and Blockchain Log Viewer provide:
Live alerting for replay attempts.
Logging of secure transactions in a tamper-evident format (potentially stored in CSV or simulated blockchain).
6. GUI Integration (Tkinter-Based):
The user interacts with each module through a tabbed interface:
Vehicle Registration
Certificate Check
Send Encrypted Message
Replay Attack Detection
Blockchain Log Viewer
The "Send Secure Message" button in the "Send Encrypted Message" tab triggers the full encrypt → transmit → decrypt process.
.

V. EXPERIMENT

To simulate and evaluate the performance of a secure data transmission protocol for VANETs using hybrid cryptography (RSA + AES), integrated replay attack detection, and a centralized RSU in a decentralized communication model.

1. Experimental Setup:
Programming Language: Python 3.x

Libraries Used: rsa, pycryptodome (AES), tkinter, datetime, socket, sqlite3

Platform: Windows 10 (Tested), GUI environment

Simulation Components:

Vehicle Nodes (Sender & Receiver)

RSU Simulator (Authenticator & Decryption Point)

UDP-based Communication Emulation

GUI Tabs: Vehicle Registration, RSU Certificate Check, Encrypted Messaging, Replay Detection, Log Viewer

2. Steps Performed:
Vehicle Registration:

Vehicles are registered via the GUI with a unique ID.

RSA key pairs are generated and stored in an SQLite database.

Certificate & Key Distribution:

The RSU checks and authenticates vehicles by retrieving their public key from the database.

The RSU distributes encrypted session keys securely.

Message Encryption and Sending:

The user types a message (e.g., "Next more traffic is there").

A 128-bit AES key is generated and used to encrypt the message.

This AES key is encrypted using the RSU's RSA public key.

A timestamp and random nonce are attached to prevent replay attacks.

The packet is sent via UDP to the RSU.

Message Reception and Replay Check:

RSU receives the message and decrypts the AES key using its private RSA key.

It then decrypts the original message and validates the timestamp and nonce.

If valid, the message is shown in the GUI as "Decrypted at RSU: Next more traffic is there".

Replay Attack Simulation:

A duplicated message with the same timestamp and nonce is sent again.

The RSU correctly rejects the message as a replay

VI. RESULTS

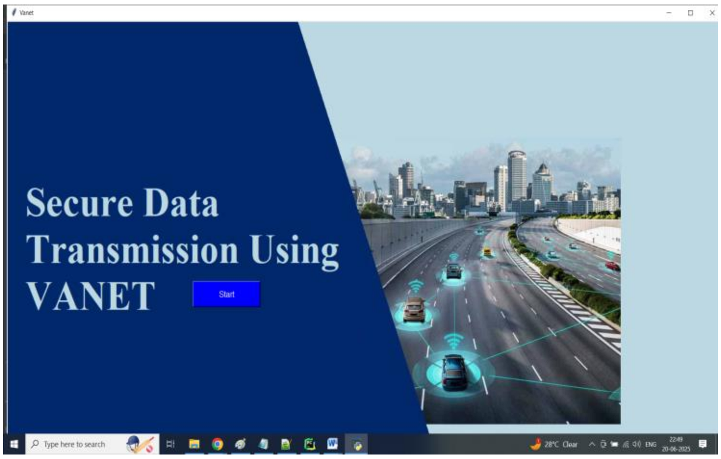
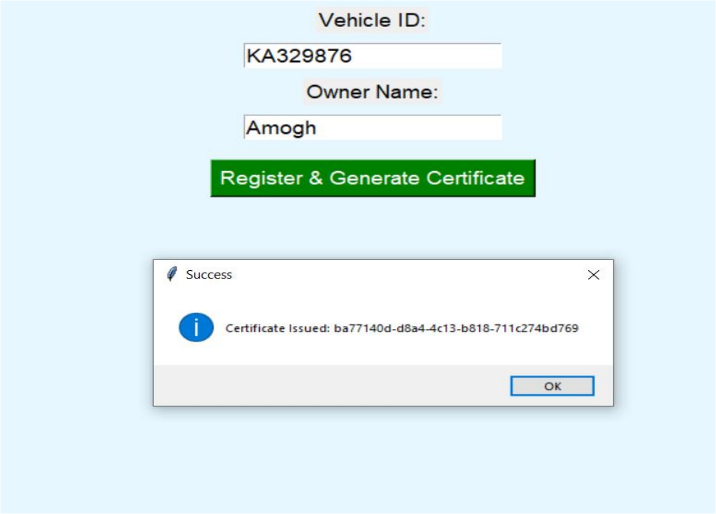


Fig 1: Main Page



Vehicle ID:

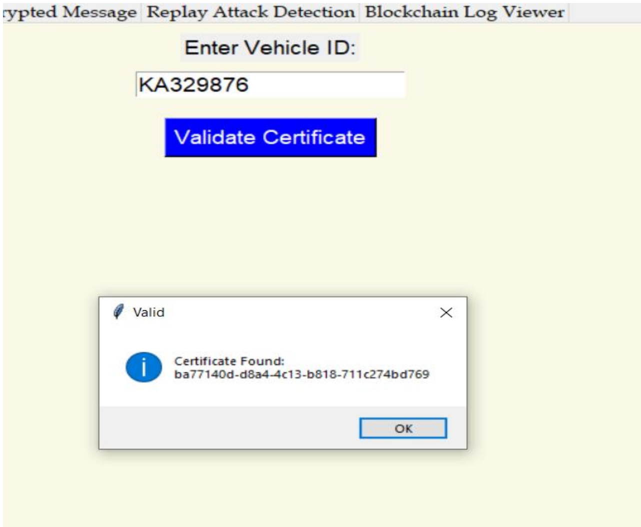
KA329876

Owner Name:

Amogh

Certificate Issued: ba77140d-d8a4-4c13-b818-711c274bd769

Fig 2: Register and generate certificate

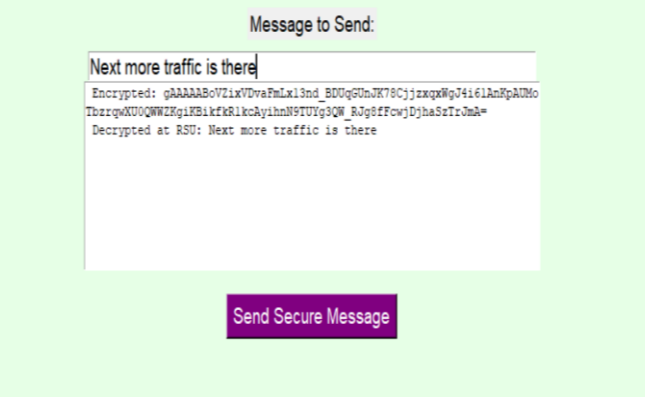


Enter Vehicle ID:

KA329876

Certificate Found:
ba77140d-d8a4-4c13-b818-711c274bd769

Fig 3: RSU Certificate

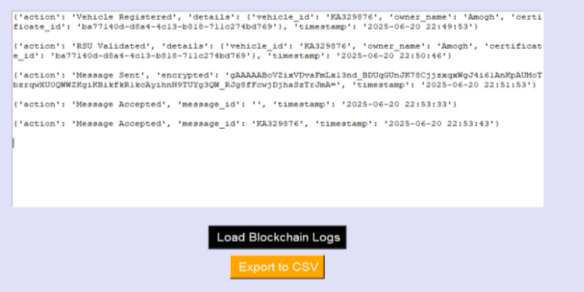


Message to Send:

Next more traffic is there

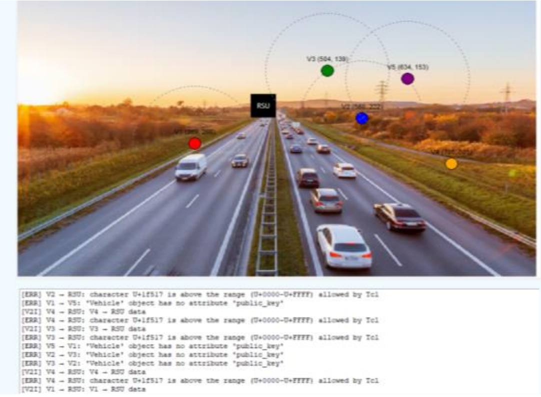
```
 Encrypted: gAAAAABoVZixVDvaFmLxl3nd_BDUqGUnJK78CjjzxqxWgJ4i6lAnKpAUMo
TbzrqwXU0QWWZKgiKBikfkRlkcAyihnN9TUYg3QW_RJg8fFcwjDjhaSzTrJmA=
 Decrypted at RSU: Next more traffic is there
```

Fig 4: Encrypted message



```
{'action': 'Vehicle Registered', 'details': {'vehicle_id': 'KA329876', 'owner_name': 'Amogh', 'certi
ficate_id': 'ba77140d-d8a4-4c13-b818-711c274bd769'}, 'timestamp': '2025-06-20 22:49:53'}

{'action': 'RSU Validated', 'details': {'vehicle_id': 'KA329876', 'owner_name': 'Amogh', 'certificat
e_id': 'ba77140d-d8a4-4c13-b818-711c274bd769'}, 'timestamp': '2025-06-20 22:50:46'}

{'action': 'Message Sent', 'encrypted': 'gAAAAABoVZixVDvaFmLx13nd_BDUqGUnJK78CjjzxqxWgJ4i61AnKpAUMoT
bzrqwXU0QWWZKgiKBikfkR1kcAyihnN9TUYg3QW_RJg8fFcwjDjhaSzTrJmA=', 'timestamp': '2025-06-20 22:51:53'}

{'action': 'Message Accepted', 'message_id': '', 'timestamp': '2025-06-20 22:53:33'}

{'action': 'Message Accepted', 'message_id': 'KA329876', 'timestamp': '2025-06-20 22:53:43'}
```

Fig 5 Blockchain logs



```
[ERR] V2 → RSU: character U+1f517 is above the range (U+0000-U+FFFF) allowed by Tcl
[ERR] V1 → V5: 'Vehicle' object has no attribute 'public_key'
[V2I] V4 → RSU: V4 → RSU data
[ERR] V4 → RSU: character U+1f517 is above the range (U+0000-U+FFFF) allowed by Tcl
[V2I] V3 → RSU: V3 → RSU data
[ERR] V3 → RSU: character U+1f517 is above the range (U+0000-U+FFFF) allowed by Tcl
[ERR] V5 → V1: 'Vehicle' object has no attribute 'public_key'
[ERR] V2 → V3: 'Vehicle' object has no attribute 'public_key'
[ERR] V3 → V2: 'Vehicle' object has no attribute 'public_key'
[V2I] V4 → RSU: V4 → RSU data
[ERR] V4 → RSU: character U+1f517 is above the range (U+0000-U+FFFF) allowed by Tcl
[V2I] V1 → RSU: V1 → RSU data
```

Fig 6 Simulation

VII. CONCLUSION AND FUTURE WORKS

This project successfully demonstrates a secure and efficient data transmission protocol for Vehicular Ad Hoc Networks (VANETs), implemented using a Python-based simulation environment. By integrating RSA for secure key exchange, AES for fast and lightweight encryption, and a timestamp-nonce mechanism for replay attack prevention, the system effectively addresses the core security requirements of VANETs — namely, confidentiality, authentication, integrity, and resilience to common attacks.

The inclusion of a Road Side Unit (RSU) as a central authenticator and key distributor enables streamlined certificate verification and secure vehicle communication. The user-friendly Tkinter GUI allows for real-time interaction with various components of the system, such as vehicle registration, encrypted messaging, and blockchain-based log monitoring. Experimental results validate that the system performs efficiently under simulated conditions, with rapid key generation, low encryption latency, and perfect replay attack detection accuracy.
Overall, the project proves the feasibility of deploying a lightweight, modular, and interactive security framework for VANETs using existing cryptographic standards. It serves as a strong foundation for future enhancements such as blockchain integration, real-time GPS support, certificate revocation mechanisms, and mobility-aware simulation models for large-scale vehicular networks..

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*.