Secure Routing Protocol Simulation for VANET Using Cryptographic Techniques

Internship Assignment Submission
Name: Mohd Abubakar
Email: mohdabubakar477@gmail.com
Date: 10th May 2025

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are critical components of intelligent transportation systems. Ensuring secure communication between vehicles is vital due to the decentralized and mobile nature of VANETs, which makes them susceptible to various attacks such as message tampering, impersonation, and spoofing.

This project aims to implement a **secure routing protocol** using **cryptographic techniques**, specifically **digital signatures** and **hash functions**, to protect message integrity and authenticity in VANETs.

2. Objectives

- Simulate vehicle-to-vehicle communication in a lightweight Python environment.
- Use **RSA digital signatures** to authenticate messages.
- Apply **SHA256 hash functions** to verify data integrity.
- Simulate and detect common VANET-specific attacks (e.g., tampering, spoofing).
- Measure performance metrics such as **hash generation time**.
- Provide visualization of results for analysis.

3. Technologies Used

- **Language:** Python 3
- **Libraries:** `cryptography`, `hashlib`, `pandas`, `matplotlib`
- **Cryptographic Algorithms:**
  - Hash: SHA256 (with salted hashing)
  - Signature: RSA (2048-bit keys)

---

4. Project Files

File	Description
`vehicles.py`	Defines the Vehicle class with movement, hashing, and digital signing capabilities
`simulations.py`	Runs the main simulation loop, simulates communication and attacks
`plot.py`	Generates a time-series plot of SHA256 hash generation time
`results/hash_times.csv`	CSV log of hash computation times per message
`README.md`	Documentation with instructions and descriptions

---

5. Security Protocol Design

Each vehicle generates a message containing:

- Vehicle ID
- Speed
- Position

Before sending:

- The message is hashed using SHA256 with a random salt.
- The message is digitally signed using the sender's RSA private key.

On receiving:

- The receiver uses the sender's **public key** to verify the **signature**.
- If verification fails, the message is rejected as tampered or spoofed.

---

6. Attack Scenarios Simulated

6.1 Tampering Attack

- At simulation step 10, the speed in the message is maliciously modified.
- The receiver fails to verify the signature, correctly detecting the attack.

6.2 Impersonation (Spoofing) Attack

- At step 15, a third-party vehicle signs a message pretending to be another.
- Since the public key does not match, verification fails.

---

7. Results and Analysis

✅ Signature Verification

- Under normal conditions, message signatures are validated correctly.
- Attacks are successfully detected due to signature mismatches.

📊 Hash Timing Analysis

A performance graph (`hash_plot.png`) was generated, showing the time required to compute the SHA256 hash at each message step. Results show:

- Hash times are consistently low (milliseconds), proving suitability for real-time communication.
- Minor variations may occur due to system randomness and salt usage.

---

8. Conclusion

This Python-based VANET security simulation demonstrates that **digital signatures and hash functions are highly effective** in securing vehicular communications. Despite not using NS-2/NS-3 or OMNeT++, the implementation successfully:

- Detects message tampering and spoofing
- Maintains a lightweight and readable structure
- Allows visualization and analysis of cryptographic performance

---

9. Future Improvements

- Add simulation of **packet loss, delay, and delivery ratio**
- Use a **key distribution mechanism or PKI**
- Extend to **multi-hop routing scenarios**
- Integrate **SUMO** or traffic traces for mobility realism
