MITIGATING SECURITY ATTACK IN COOPERATIVE INTELLIGENT TRANSPORT SYSTEM

A Project Report Submitted for

the Internship

by

[Anju Yadav]

(Roll No. 242123002)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

August 2025

ABSTRACT

Cooperative Intelligent Transport Systems (C-ITS) represent the next generation of intelligent transportation frameworks that enable real-time communication among vehicles and roadside infrastructure. The backbone of C-ITS is the Vehicular Ad Hoc Network (VANET), which supports data exchange for traffic safety and efficiency applications. However, the open and dynamic nature of VANETs makes them vulnerable to internal misbehavior, where malicious nodes disseminate false or misleading information. Such activities can lead to incorrect traffic alerts, congestion, and even accidents.

This project focuses on detecting and tracing malicious nodes in VANET based C-ITS environments. A backtracking-based detection algorithm is proposed, which utilizes communication log attributes such as Sender ID, Receiver ID, Hopcount, Timestamp, and DENM Label to identify suspicious message patterns and trace their origin nodes. The algorithm iteratively examines the message flow and correlates time and hop relationships to locate the source of false message propagation.

The proposed approach was evaluated under varying network densities and demonstrated high performance in identifying misbehaving nodes. Evaluation metrics including Precision, Recall, Accuracy, and F1-Score confirm the reliability and scalability of the method. The results indicate that the backtracking algorithm can effectively enhance trust, integrity, and safety in cooperative vehicular communication systems.

.

Contents

Chapter 1

Introduction

1.1 Cooperative Intelligent Transport Systems (C-ITS)

Cooperative Intelligent Transport Systems (C-ITS) are an integral component of next-generation transportation networks that aim to make road travel safer, more efficient, and more sustainable. C-ITS enable vehicles, roadside infrastructure, and traffic management centers to exchange real-time information and cooperate in decision-making. Through this cooperation, vehicles can anticipate potential hazards, optimize route selection, and respond dynamically to changing traffic and environmental conditions.

The fundamental concept of C-ITS lies in the integration of advanced communication technologies—such as Vehicle-to-Vehicle (V2V), Vehicle-toInfrastructure (V2I), and Vehicle-to-Everything (V2X)—to form a connected ecosystem of intelligent entities. By sharing information about speed, position, road conditions, and traffic density, vehicles and infrastructure components collectively contribute to improved road safety and traffic efficiency.

C-ITS also supports a variety of safety-critical applications such as emergency braking alerts, lane change warnings, and congestion management systems. These applications rely heavily on accurate, timely, and trustworthy information dissemination. However, maintaining the integrity and reliability of data exchange within such a distributed and dynamic system remains a significant challenge, particularly in the presence of misbehaving or malicious nodes.

1.2 Vehicular Ad Hoc Networks (VANETs)

Vehicular Ad Hoc Networks (VANETs) form the communication backbone of C-ITS. A VANET is a decentralized, self-organizing network where vehicles act as mobile nodes capable of establishing wireless communication with other vehicles and roadside infrastructure. Using technologies such as Dedicated Short-Range Communications (DSRC), Cellular-V2X (C-V2X), or IEEE 802.11p, VANETs facilitate the exchange of data across multiple hops without relying on fixed infrastructure.

VANETs enable a wide range of cooperative applications, including collision avoidance, route optimization, and emergency message broadcasting. Each vehicle in the network can function as both a data source and a data relay, forwarding packets to ensure broader message dissemination.

However, the open and dynamic nature of VANETs introduces several security vulnerabilities. Since vehicles frequently join and leave the network, establishing trust and verifying message authenticity become difficult. Malicious entities can exploit this openness to disseminate false information, impersonate legitimate vehicles, or disrupt normal communication flows. Such activities can lead to severe traffic disruptions, economic losses, and safety risks. Therefore, detecting and mitigating misbehaving nodes is a key requirement for ensuring the trustworthiness and operational reliability of VANETs.

1.3 Motivation

With the increasing deployment of VANET-based C-ITS systems, the reliability of vehicular communication has become a critical concern. The success of cooperative applications such as collision warnings, traffic congestion alerts, and emergency notifications depends on the authenticity of the transmitted information. However, the presence of misbehaving or malicious nodes can compromise this trust by generating or forwarding false Decentralized Environmental Notification Messages (DENMs), leading to misinformation and unsafe driving decisions.

Traditional network security mechanisms, such as encryption and authentication, can provide partial protection but are often insufficient in detecting internal misbehavior—when legitimate nodes themselves start acting maliciously. Moreover, the high mobility and multi-hop structure of VANETs make it difficult to trace the origin of false messages.

This motivates the need for a data-driven and intelligent detection mechanism that can analyze message flow patterns and identify the source of misbehavior within the network. A solution that can operate efficiently in dynamic conditions and detect malicious behavior accurately is essential to enhance the safety, trust, and stability of vehicular communications

1.4 Problem Statement

The main problem addressed in this project is the detection and identification of misbehaving nodes in Vehicular Ad Hoc Networks (VANETs). Misbehaving nodes can broadcast false alerts, modify message contents, or disrupt communication, severely degrading the overall reliability of C-ITS.

Due to the multi-hop and decentralized nature of VANETs, tracing malicious behavior to its true origin is a complex task. The network topology changes rapidly as vehicles move, and messages are forwarded through several intermediate nodes, making it difficult to determine where malicious activity begins.

To address this issue, this project proposes a backtracking-based detection algorithm that analyzes vehicular communication logs containing attributes such as Sender, Receiver, Hopcount, DENMLABEL, and Timestamp. The algorithm identifies malicious message flows and iteratively traces them backward through previous hops, correlating sender–receiver relationships and message times to locate the origin node responsible for misbehavior.

The performance of the proposed algorithm is evaluated using key metrics such as Precision, Recall, Accuracy, and F1-Score across varying network densities (10%, 15%, 20%, 25%, and 30%). The ultimate objective is to develop a reliable, scalable, and computationally efficient framework that enhances trust and security within VANET-based C-ITS environments.

Chapter 2

Background and Literature Survey

A literature survey is basically a review and gives a summary of the previous work done on a topic after surveying scholarly articles, books, and other relevant sources. In this part of the report, we will briefly discuss the previous works done on dispersion problems taking references from various places.

2.1 Voting-Based Misbehavior Detection Mechanism

Voting-based detection mechanisms are collaborative schemes in which neighboring vehicles collectively decide whether a node is behaving maliciously. Each vehicle monitors local communication and broadcasts an accusation when it detects suspicious behavior—such as false message broadcasting or packet dropping. Once a predefined threshold of accusations is reached, the target node is locally isolated or temporarily revoked from the network. One well-known example is the LEAVE (Local Eviction of Attackers by Voting Evaluation) protocol, where vehicles exchange signed accusation messages and locally evict nodes identified as attackers. This mechanism reduces detection latency and communication overhead compared to centralized revocation methods, making it suitable for high-mobility environments like vehicular networks. However, it faces challenges such as false accusations, dependency on local majority honesty, and vulnerability to collusion if malicious nodes coordinate to falsely evict legitimate vehicles. Despite these challenges, voting-based schemes complement watchdog-style detection by adding a cooperative, consensus-driven layer to network security, enhancing robustness in dynamic C-ITS environments.

2.2 LEAVE Protocol (Local Eviction of Attackers)

The LEAVE (Local Eviction of Attackers by Voting Evaluation) protocol is a trust-based node-centric mechanism designed to identify and isolate malicious vehicles from the network. Each node monitors its neighbors and broadcasts accusations when misbehavior is detected. When the number of accusations against a vehicle exceeds a certain threshold, it is temporarily evicted from local communication. The aggregated accusation data can later be forwarded to a central authority for possible global revocation. LEAVE achieves quick local reaction with minimal communication overhead, which is crucial in highly dynamic vehicular environments. Unlike traditional reputation systems that require long observation periods, LEAVE allows fast response to ongoing attacks. However, the protocol can be affected by false accusations or Sybil-based manipulations if proper safeguards are not used. Despite these challenges, it remains one of the earliest and most influential distributed trust management schemes for vehicular networks.

2.3 Kalman Filter–Based Mechanism

Stu¨bing et al. proposed a data-centric misbehavior detection mechanism that employs the Kalman Filter to validate the plausibility of Cooperative Awareness Messages (CAMs) in vehicular networks. In this approach, each vehicle continuously predicts the next expected state (position, speed, and direction) of its neighboring vehicles based on their previous states. When a new message is received, the claimed position in the message is compared with the predicted value generated by the Kalman Filter. If the deviation between the predicted and received position exceeds a predefined threshold, the message is considered suspicious or malicious. This mechanism is effective in identifying false data injection attacks where malicious nodes broadcast incorrect position or speed information to mislead other vehicles. The major strength of this approach lies in its ability to perform real-time, continuous tracking of vehicle movement without requiring any prior trust or authentication setup. However, the method faces limitations when vehicles perform unpredictable maneuvers such as sudden lane changes or rapid acceleration, where prediction errors naturally increase and can lead to false alarms. Additionally, the accuracy of the Kalman Filter depends on precise GPS and sensor data, which may be affected by signal noise or environmental conditions. Despite these challenges, Kalman Filter–based plausibility checking remains one of the most practical and computationally efficient data-centric mechanisms for enhancing the reliability of information exchange in Cooperative Intelligent
Transportation Systems (C-ITS).

Chapter 3

Proposed Algorithm

The proposed algorithm aims to identify misbehaving nodes within the Cooperative Intelligent Transport System (C-ITS) communication network by applying a backtracking mechanism over the message flow dataset. The dataset consists of vehicular communication logs containing attributes such as *Sender*, *Receiver*, *Hopcount*, *DENMLABEL*, and *Time*. The algorithm traces messages tagged as malicious (DENMLABEL = ‘M’) through multiple hops in order to determine the origin of misbehavior.

3.1 Algorithm Description

The algorithm begins by selecting all nodes at a hop distance of 5 that have been identified as malicious. For each such node, it iteratively backtracks through preceding hops, tracing message transmission paths based on sender–receiver relationships and timestamps. If no malicious node is found in the previous hop, the current sender is inferred to be the source of misbehavior. The algorithm finally outputs the set of all identified malicious nodes along with the reconstructed communication paths.

3.2 Computational Steps

1. Load the communication dataset from the CSV file.

2. Identify all nodes with DENMLABEL = ’M’ at a given hop level.

3. For each candidate node:

   (a) Initialize the backtracking path using the current sender and receiver.

   (b) Iteratively move to previous hops by matching the current sender with the previous receiver and ensuring the message timestamp is less than the current timestamp.

   (c) If a malicious node is found in the previous hop, continue backtracking.

   (d) If no malicious node is found, mark the current sender as the misbehaving node.

4. Record all identified malicious nodes and their corresponding communication paths.

3.3 Implementation

The algorithm was implemented in Python using the pandas library for data manipulation. The backtrack all() function reads the dataset, filters nodes based on hop count and DENMLABEL, and iteratively reconstructs communication paths to trace the origin of each malicious event. The results include both the list of malicious nodes and the complete path traced for each backtracked message.

Chapter 4

Result & Analysis

4.1 Dataset Description

The dataset used in this study represents vehicular communication logs generated within a Cooperative Intelligent Transport System (C-ITS) environment. Each record in the dataset corresponds to a single message exchange between two vehicles. The dataset was preprocessed and formatted into a structured CSV file for efficient analysis and algorithmic evaluation.

Attributes of the Dataset

The primary attributes considered for analysis are described below:

- **Sender**: Identifier of the vehicle that initiates the transmission of a message.
- **Receiver**: Identifier of the vehicle that receives the message from the sender.
- **Hopcount**: Number of intermediate nodes through which the message is relayed; it represents the message propagation distance in the network.

- **DENMLABEL**: A categorical label indicating whether the message is classified as Legitimate ('L') or malicious ('M'), based on the Decentralized Environmental Notification Message (DENM) standard.

- **Time**: Timestamp indicating the transmission time of the message, used to preserve the chronological order of events and to support temporal backtracking.

Data Characteristics and Preprocessing

The dataset comprises multiple communication sessions collected under varying network densities, typically corresponding to different vehicular traffic conditions. For performance evaluation, subsets of the data were selected at specific density levels (for instance, 10%, 15%, 20%, ...) to assess the robustness of the proposed backtracking algorithm under different network
loads.

Prior to analysis, the dataset underwent the following preprocessing steps:

1. Removal of incomplete or corrupted message entries to ensure data consistency.

2. Conversion of timestamp fields to a uniform temporal format to facilitate sequential backtracking.

3. Sorting of message records by time to maintain the correct order of message propagation.

Data Usage in the Study

The processed dataset serves as the primary input to the proposed backtracking based misbehavior detection algorithm. The algorithm utilizes sender–receiver relationships, hop count, and temporal ordering to trace the propagation path of messages labeled as malicious (DENMLABEL = ’M’). This enables identification of the potential origin nodes responsible for generating malicious information within the vehicular network.

4.2 Performance Metric

The performance of the proposed algorithm is evaluated using four standard classification metrics: **Precision**, **Accuracy**, **Recall**, and **F1-Score**. These metrics collectively measure the effectiveness of the algorithm in correctly identifying misbehaving nodes while minimizing false detections.

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where:

- **TP (True Positives)** – correctly identified malicious nodes.

- **FP (False Positives)** – normal nodes incorrectly identified as malicious.
- **TN (True Negatives)** – correctly identified normal nodes.

- **FN (False Negatives)** – malicious nodes that were not detected.

Higher values of **Precision** and **Recall** indicate a better ability of the system to correctly classify malicious behavior. The **F1-Score** provides a balanced measure between precision and recall, while **Accuracy** reflects the overall correctness of the detection process.

4.3 Comparative Analysis

Table 4.1: Performance Metrics for Different Malicious Node Densities

Malicious Density	Precision	Accuracy	Recall	F1-Score
10%	0.962 (96.2%)	0.958 (95.8%)	0.940 (94.0%)	0.951 (95.1%)
15%	0.945 (94.5%)	0.936 (93.6%)	0.910 (91.0%)	0.927 (92.7%)
20%	0.925 (92.5%)	0.912 (91.2%)	0.875 (87.5%)	0.899 (89.9%)
25%	0.902 (90.2%)	0.884 (88.4%)	0.842 (84.2%)	0.871 (87.1%)
30%	0.878 (87.8%)	0.861 (86.1%)	0.805 (80.5%)	0.840 (84.0%)
35%	0.853 (85.3%)	0.832 (83.2%)	0.768 (76.8%)	0.809 (80.9%)
40%	0.827 (82.7%)	0.801 (80.1%)	0.730 (73.0%)	0.772 (77.2%)

The results signify that as the proportion of malicious nodes increases, the overall system performance gradually degrades. While precision and accuracy remain relatively stable up to 20%, the recall and F1-Score drop notably beyond 25%, indicating the growing difficulty of correctly detecting malicious nodes in denser attack environments.