Assumption :

Intersection manager (IM) structure:

The IM is divided in three areas:

1. **Storage area** - This is where the vehicles first arrive and give requests to the IM at a certain time. In my iteration, this area is divided into three lanes (For example: a lane that goes right, straight ahead and to left) for each of the four directions.

For each of the lane, it is divided into 3 zones (to keep a time marker), each certain distance from the intersection

Zone A: 10 meter from the intersection

Zone B: 20 meter from the intersection

Zone C: 30 meter from the intersection

1. **Conflict area -** Once the vehicles are 'CONFIRM'ed by the IM, they enter the intersection (irrespective of direction), such that:
   1. The path (or nodes) in front of them are cleared. The vehicle that comes first has the first priority (except in case of emergency vehicles) to go through the nodes
   2. In case of an emergency vehicle in the storage area, they go before any other vehicles in their lanes (including the ones in from of them)

1. **Exit area** - The vehicles enters this area after it leaves the last node, marking the last time the IM communicates with them

Powers:

1. It is solely up to the intersection manager to let vehicles inside the conflict area/intersection.
2. Vehicles are allowed to move at their will in the storage area. For instance, when vehicles give way for emergency vehicles in the storage area, they move on their own.
3. When vehicles move out of their way for an emergency vehicle, they are eliminated from the IM log of vehicles, such that the moved vehicles need to REQUEST again for a time slot to enter the conflict area.

V to I Communication:

Vehicle:

**REQUEST** signal : The vehicle shares their **ID, origin lane, destination lane, current location** (in terms of nodes - or position of vehicle with in a certain radius of the intersection), and **the current time**

As vehicles go from one node to another in the storage or conflict area, the time is being saved to calculate the entry\_time (to the intersection) and other time entities (based on any other new functions)

**CANCEL** signal : The vehicle can also cancel the previous REQUEST at their own will or in case of an emergency vehicles in their own lane

Intersection:

Assess the log of vehicles it received to calculate who gets the first time slot for each path, concurrently.

**CONFIRM** signal: For the vehicles that has an open path, the CONFIRM signal from the Intersection manager allows the vehicles to enter the conflict area

I to I communication:

The scope of this communication is not defined yet.

V to V communication:

The scope of this communication is not defined yet.

A cross with cars and numbers

Description automatically generated with medium confidence

Threat model:

Giving way for Emergency vehicle is an important part of a IM, given the significance of their urgency. However, if ordinary vehicles have the capacity to impose as emergency vehicles, not only would it cause unnecessary delays but also will constrain actual emergency vehicles from performing their important tasks.

So in a malicious environment:

1. There could be multiple imposters (acting as Emergency vehicles), in every direction halting traffic at an intersection completely
2. There could be just one or two imposters in the IM, such that they always get to trick the system, which becomes a major flaw in the system and unfair to other law abiding cars
3. Or in the worst case, slow down an actual emergency vehicle by tagging along with them or in other ways, which is then a threat to life and property

What sets rogue imposter vehicles apart from actual emergency vehicles?

A powerful adversary may have all the markings of a real emergency vehicle, which may make it difficult to identify with appearance or IDs.

Since mitigation has to be based on traits that cannot be mimicked by an imposter, here are a few:

1. Behavior: Rogue vehicles may exhibit unusual behavior that is inconsistent with standard emergency response procedures. For instance it may not pass through the a certain number intersection to reach a said emergency destination, in which case a I to I communication should be established. A ML model could predict the type of vehicle based on how they behave in the previous intersections.

Pros: The IM can learn the rogue nature of a vehicle with the data from other IMs, hence not giving priority anymore

Cons:

1. Radio frequencies emitted: Assuming certain vehicles communicate in certain range of frequency and that IMs can know the RFIDs of emergency vehicles

Pros: A single intersection may be able to identify the rogue vehicle

Cons: Log of all confidential RFID at every intersection may be subject to attack

Discussion:

it’s desirable to catch it as early as possible (and that should be a metric for evaluation) it’s not a catastrophe if it’s not caught immediately  (even though catching immediately would be ideal).

First we learn emergency vehicle behavior or set it.

Pros and Cons of assumptions

Adversary: Self sustained rogue vehicle

|  |  |  |  |
| --- | --- | --- | --- |
| Metric of detection | Pros | Cons | Solution |
| Unforgettable ID | The ID is always stored in a cloud and can detect imposters at the very first intersection | Data Leak | Forgettable ID |
| Forgettable ID | Not susceptible to any data leak | Cannot track vehicular motion between intersections tied to a given ID | Generate a Behavioral ID unique to every vehicle type and track that |
| Untrusted I2I |  | No way to detect |  |
| Trsuted I2I | Only way to catch a rogue vehicle is trusted I to I (as of now) | -- |  |
| Untrusted V2V |  |  |  |
| Trusted V2V | Detection method if V2I is untrusted in case of communication intercepted | If an imposter can lie to the IM couldn’t it lie to other vehicles |  |
| Trusted V2I | No extra measures required to secure the comm channel |  |  |
| IMs access to the entire EM database | Easy detection | Data Leak | Forgettable ID |

Assumption for ID:

**Constant assumption: Forgettable unique ID, no permanent database of previous vehicle IDs, trusted I2I**

**Communication includes (so far) ID, speed, acceleration, time, origin, destination. Could include GPS coordinates and others at later stage of mitigation**

Case 1: Forgettable unique ID, with intersection having no database of previous vehicles with **trusted V2I** and I2I, **untrusted/trusted V2V**

Pros:

Cons: Trusting V2I would mean there could never be rogue vehicles (vehicles cannot lie about themselves), irrespective of if V2V is trusted or not

Case 2: Forgettable unique ID, with intersection having no database of previous vehicles with **trusted V2I** and I2I, **trusted V2V**

Pros: All three comm channel are right all the time

Cons: V2V may communicate what they see at the moment, lets say visual identifiers, speed, acceleration etc., but it does not have the ability or time to learn the type of the vehicle that’s beside it

Also trusting all the three communication channel would make it a trivial problem

Case 3: Forgettable unique ID, with intersection having no database of previous vehicles with trusted I2I **untrusted V2I,** **trusted V2V**

Pros: Untrusted V2I raises the question of rogue vehicle

Cons: V2V may communicate what they see at the moment, lets say visual identifiers, speed, acceleration etc., but it does not have the ability or time to learn the type of the vehicle that’s beside it

Case 4: Forgettable unique ID, with intersection having no database of previous vehicles with **trusted I2I** **untrusted V2I,** un**trusted V2V**

Pros: Untrusted V2I raises the question of rogue vehicle,

Cons:

Assumptions:

Forgettable unique ID, with intersection having no database of previous vehicles with **trusted I2I** **~~untrusted~~ V2I, trusted V2V,** IM has a data base of **emergency events** (database not encouraged)

All trusted channels (I2I, V2I, V2V), but rogue/untrusted vehicle and trusted intersection

Vehicle characteristics can be differentiated based on three factors: braking patterns, acceleration and turning events (Minh 2013). Turn events has a faster prediction rate (Hallac 2017)

**Problems to solve:**

**1. Track a vehicle across intersections**

Can be assessed by turning events, inertial sensors, braking and acceleration patterns

**2. Distinguish between Real emergency vehicles vs Imposter emergency**

How does the IM know:

* + Access of the database from state or city (Not recommended)
  + Identify accidents on the road based on the density of traffic flow. Less dense and slow flow of traffic may indicate accident or emergencies

Vehicle:

Part 1: Trajectory mapping

* + Is it going to an emergency location – trajectory mapping prediction based on known IM data and possible routes it can go
  + Predict future trajectories from vehicle to vehicle interactions: A vehicle receives information, such as Global Positioning System coordinates, about nearby vehicles on the road using inter-vehicular communication. The collected data from vehicles together with GPR models received from infrastructure are then used to predict the future trajectories of vehicles in the scene. (Sepideh 2018) (The IM needs to do this learning)

Part 2: Analyzing doppler effect

* + Visual and audio responses (how long were their sirens and lights on) – doppler effect
    - Cant use this as IM should prevent future emergency vehicles from making noise or light pollution.

Part 3: Analyzing

Increasing acceleration or velocity for emergency vehicle

**Can the real EM and imposter EM be distinguished, without a database of emergency events?**

Rational Approach:

Every time a vehicle passes an intersection, not following priority, it gets fined and the fine is only lifted once it reaches the emergency location.

Rational Vehicle: Does not want harm to itself and abide by the rules

Irrational Vehicle: Willing to pay the money and cause attack in order to:

* Slow down other emergency vehicles
* Or simply to get an earlier timeslot, assuming they don’t care about the consequence.

In both cases, while a single rogue vehicle may not cause significant delay, multiple vehicles can cause a significant delay.

Solutions (updated):

**Distinguish between real and rogue EM**: Trajectory mapping

* + Is it going to an emergency location – trajectory mapping prediction based on known IM data and possible routes it can go
  + Predict future trajectories from vehicle to vehicle interactions: A vehicle receives information, such as Global Positioning System coordinates, about nearby vehicles on the road using inter-vehicular communication. The collected data from vehicles together with GPS models received from infrastructure are then used to predict the future trajectories of vehicles in the scene. (Sepideh 2018) (The IM needs to do this learning)
    - Able to find the nearest future directions with LSTM but unable to find the final destination
    - Need to try clustering to cluster between emergency and non emergency
  + Compare the calculated future trajectories with where they said they are actually going

**Track a vehicle (without ID) across intersections**

* Can be assessed by turning events, inertial sensors, braking and acceleration patterns

**Update October 26**

**Assumptions:**

We have a grid of multiple intersections connecting each other, where vehicle data is stored temporarily for learning and identifying in the next intersection. This also means the vehicles need to pass at least one intersection to analyze its data.

**Primary Threat:**

Rogue vehicles posing as an emergency vehicle to get false higher priority.

Solution:

* 1. Track future trajectory of a vehicle using its previous coordinates, velocity and acceleration using machine learning. This future trajectory calculated by the intersection manager can be compared with the location they are claiming to go to, to see if they are telling the truth. This way there is no need for a database of all emergency events in the area.
  2. Learning the velocity or acceleration pattern of emergency vehicles in relation to normal vehicles. For instance, emergency vehicles may have a constant velocity curve, while regular vehicles may decelerate or stop more abruptly, creating an uneven curve.

**Secondary threat: Adversary capability**

Vehicle can change IDs between intersection making it difficult to track a vehicle:

1. Each vehicle can use its driving pattern as an indicator of its uniqueness. Vehicle characteristics can be differentiated based on three factors: braking patterns, acceleration and turning events (Minh 2013). Turn events have a faster prediction rate (Hallac 2017), since it both accelerates and decelerates.

Next steps:

Collect data from SUMO or CARLA for vehicles movement between intersections.

**SO FAR:**

Rogue vehicle is defined by its False REQUEST signal parameters such **ID, origin lane, destination lane, current location, current time**

Trusting: All communication channels, IM signals

Impact: Unfair time slot distribution

* 1. False ID: Imposter Emergency vehicle
  2. False destination lane: If a vehicle is intending to go right but lies to the intersection manager that it is going to straight, how can the IM catch it?
  3. False current location: Time slot may be given wrongly

Solutions:

* Learn patterns created by rogue vehicles and identify them? Supervised, unsupervised or sequential learning, continuous learning, simulator……….?

Update 1/22:

Solutions considered:

1. Radio frequencies emitted: Assuming certain vehicles communicate in certain range of frequency and that IMs can know the RFIDs of emergency vehicles

Concluded: Database may be vulnerable but All trusted channels (I2I, V2I, V2V), with rogue/untrusted vehicle

1. Behavior:

**Track a unique vehicle across intersections**

* 1. Trajectory mapping: Can be assessed by turning events, inertial sensors, braking and acceleration patterns
  2. Visual and audio responses (how long were their sirens and lights on) – doppler effect
     1. Cant use this as IM should prevent future emergency vehicles from making noise or light pollution.

**Distinguish between Real emergency vehicles vs Imposter emergency**

The IM can learn the rogue nature of a vehicle with the data from other IMs, hence not giving priority anymore.

* 1. Trajectory mapping to cross check final destination: **LSTM** calculated the next 10 meters from previous x, y, acc, vel data, but not the entire journey.

Only way to find final destination is using GNN.

* 1. Learning the velocity or acceleration pattern of emergency vehicles in relation to normal vehicles. For instance, emergency vehicles may have a constant velocity curve, while regular vehicles may decelerate or stop more abruptly, creating an uneven curve.
  2. Each vehicle can use its driving pattern as an indicator of its uniqueness. Vehicle characteristics can be differentiated based on three factors: braking patterns, acceleration and turning events (Minh 2013). Turn events have a faster prediction rate (Hallac 2017), since it both accelerates and decelerates.

Data Collection:

Online sources:

911 calls dataset from San Francisco. <https://www.kaggle.com/datasets/datasf/san-francisco>

Vehicle movement across 3 to 4 intersections: <https://datahub.transportation.gov/stories/s/Next-Generation-Simulation-NGSIM-Open-Data/i5zb-xe34/>

Jan 30, 2024

Rogue vehicle mitigation ground rules on **what to do**:

Layer 2

Layer 1

Stop giving priority while in intersection unit.

Rogue vehicle mitigation ground rules on **how to do**:

Since a database spanning over multiple cities or even a large number of blocks could be a vulnerable to bigger threats:

The communication unit should be set up such that it can only share limited data outside its boundaries.

The cluster unit that has all the possible emergency anchor points.

Inside this grid, assuming emergency vehicle takes the shortest path to an anchor point, is it going there?

Spatial data vs Telemetric data

Case1: Veh going in a straight line to the E2 – Lesser confidence  
Case2: Veh needing to take a 90 degree turn -- Higher confidence

The next unit gets the info of a car not going to an anchor point, hence Rogue

Telemetric: The usual speed on a that specific road at a certain time of the day? Clustering routes?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E1 |  |  |  |  |  |  |
|  |  |  |  |  | Home | E3 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Emergency |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | E2 |  |  |  |
|  |  |  |  |  |  |  |

Bell Curve of Accuracy

Bell Curve of accuracy – Trajectory mapping works better for cars emerging from the center or for destinations in the center

Possible drawback?

Only possible if there is some amount to I’s after it turns in its destination direction

Simulation:

Layer 1: Identify a vehicle ID by its behavior.

Layer 2: At some point, does it go to its destination emergency point? And fast were you able to go to the emergency location.

Collected Data:

Two kinds of data:

1. Telemetric data: Only recording kinematics of the vehicle without considering the route or traffic density.

A screen shot of a computer

Description automatically generated

1. Simulated routes data based on visibility and obstacles: weather, traffic density, time of day

Experiments:

* + - Day: With more aggressive traffic
    - Night: With average traffic

Should weather be incorporated into the same time of day experiment?

Data Validation:

Comparing acceleration using HighD data

Where to obtain it? Can I also use NGSIM data as well?

How to pick a range from HighD? Can acceleration and time alone validate?

Documentation:

* Based on different ambient traffic densities, time of day (visibility) and weather of traffic (visibility), calculate time taken in a longer less traffic route and the shortest route, proving how people tend to take different routes to get to the same location.

Other Concerns:

* Unfortunately there is not a way to assign traffic to a specific location on the map.  The way that ambient traffic is spawned is based on the ownship position, meaning the density can be changed but it will always be centered around your vehicle.  – Cannot collect data of vehicles around
* How would I pick point of emergency/destination? Are these maps complex enough to cluster routes? – On an average there are only 3 or 4 possible routes to get to one point.

*From ‘Resilient Cooperative Adaptive Cruise Control for Autonomous Vehicles Using Machine Learning’ paper:*

A graph of a graph

Description automatically generated with medium confidence

“The length of individual vehicle trajectories is approximately 15 seconds. We compare acceleration patterns of similar length trajectories collected from the simulator.” - Dr. Boddupalli

8 people – 8 routes

Wind, sunny, rain