# Secure Smart Cars and Vehicular Networks

...

Project for Network and Computer Security

**Group** 18:
**Name**: João Franco **Number**: 75219
**Name**: Fábio Almeida **Number**: 76959

## Introduction

Road crashes rank as the 9th leading cause of death in the world.

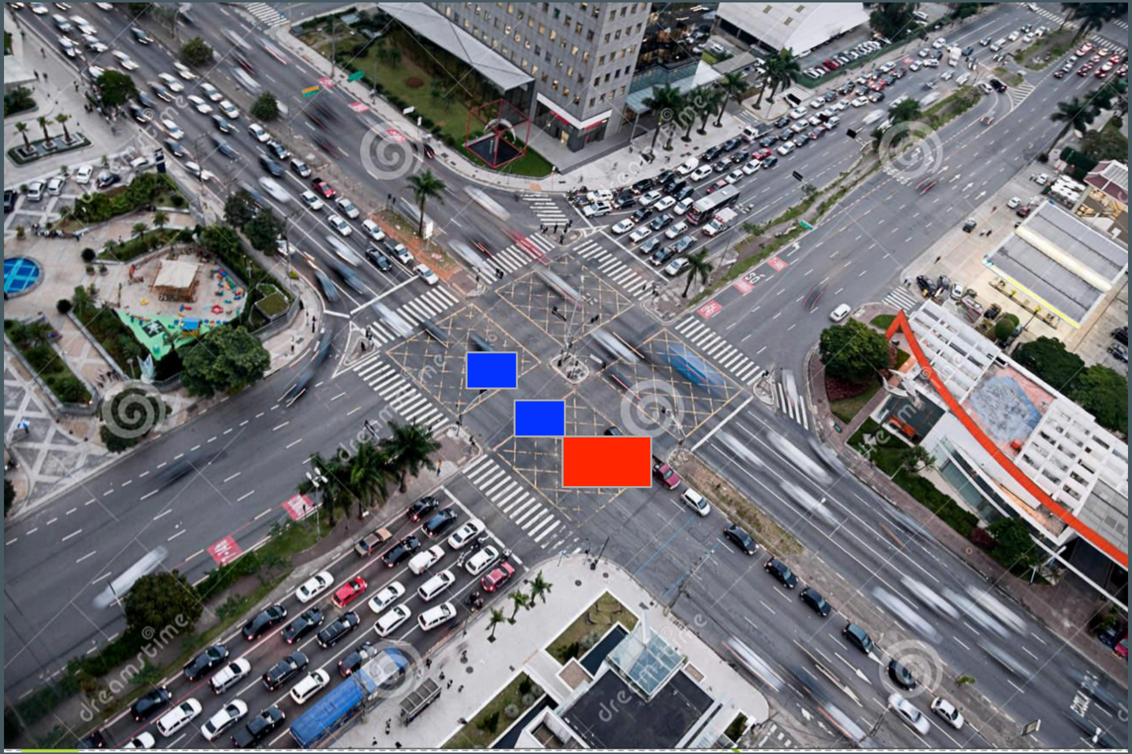
Improving road safety is urgent!!

Automated driving is already being tested and we believe access for the masses is just around the corner.

One essential component for this to work are VANETs, Vehicle Ad Hoc NETworks.

In this presentation we will explain our solution on making VANETs secure, more specifically for beaconing messages..

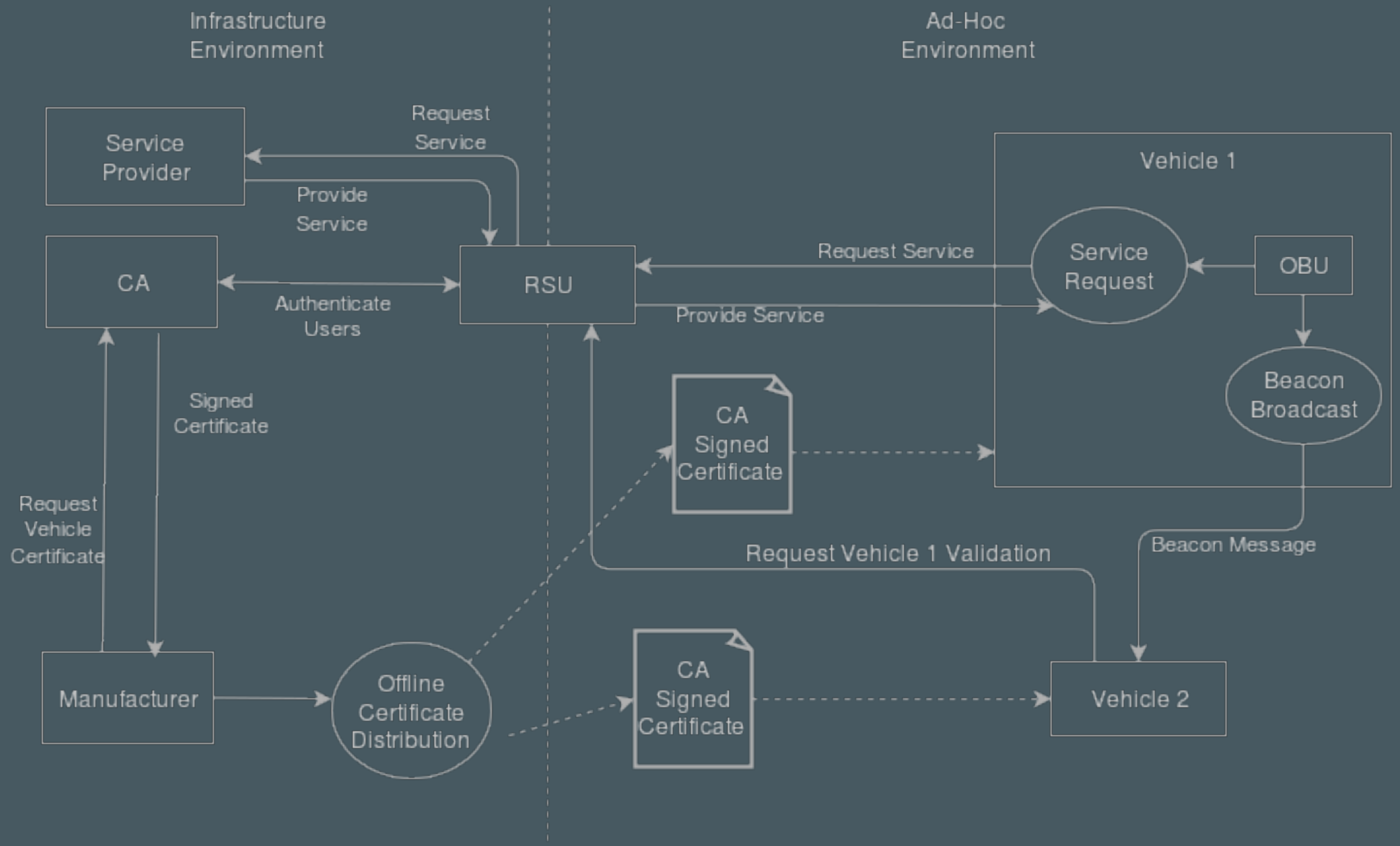
# VANETs



## Requirements

- Entity Identification
- Message Sender authentication
- Message Integrity
- Non-Repudiation of the Sender
- Freshness
- Data Trust
- Availability
- Privacy

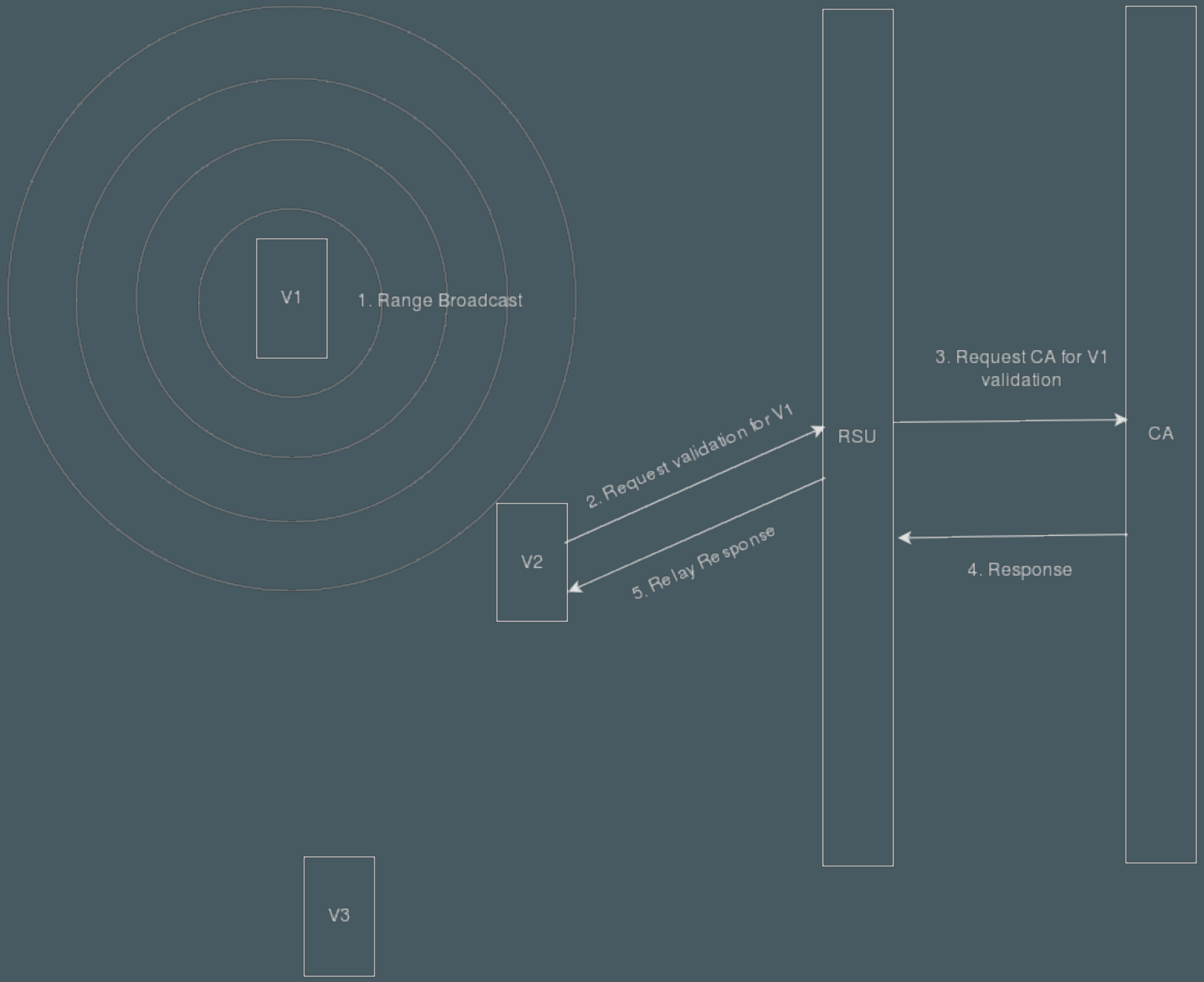
# System Architecture Overview



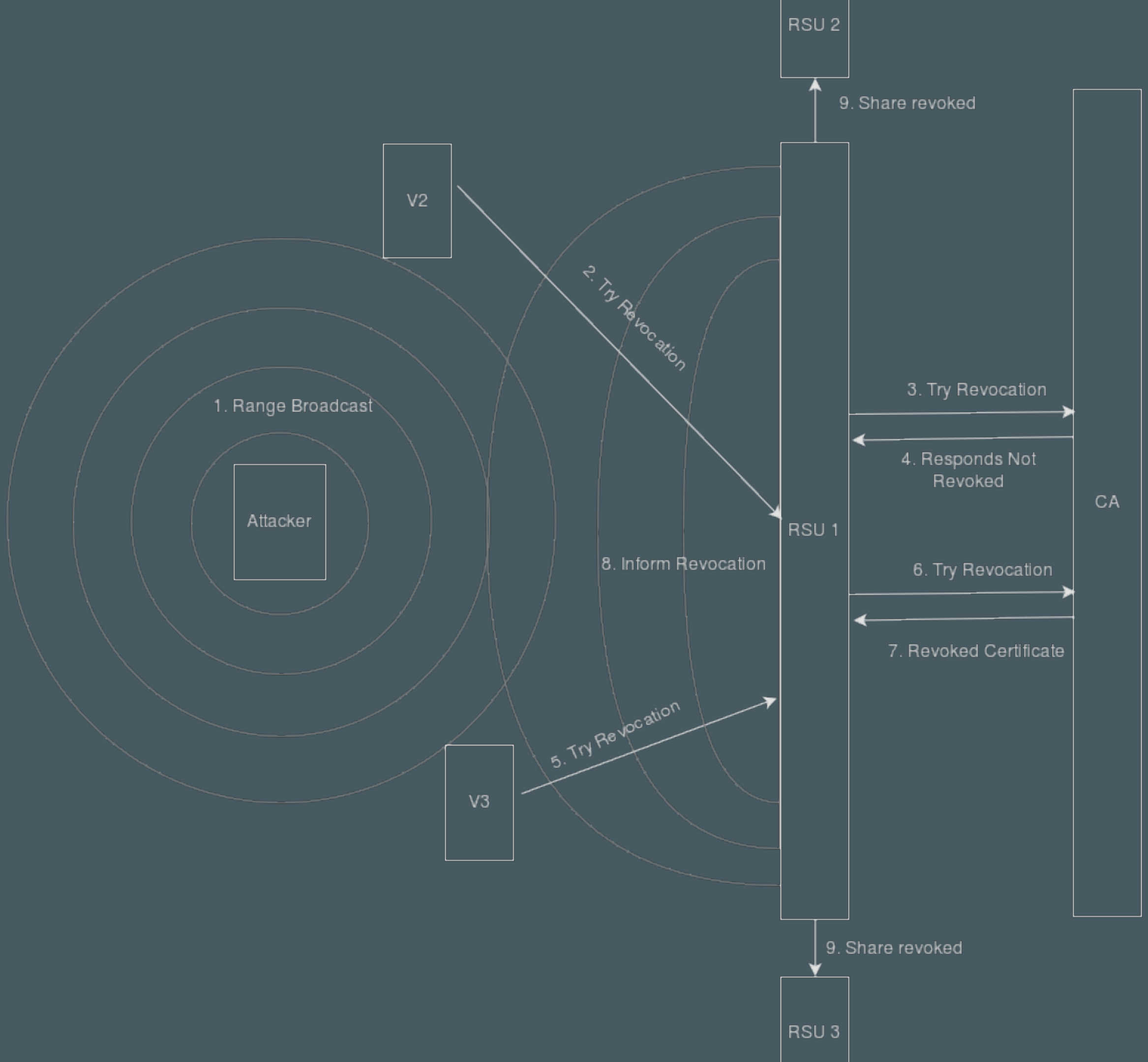
## Keys / Certificate Distribution

- CA public key certificate is distributed offline to vehicle manufacturers and consequently to the vehicles
- CA signs the certificate of every vehicle and RSU

## Beacon Protocol



# Revoke Certificate Protocol



## Possible Improvements

**Pseudonym Certificates:**

- used in short periods of time or short sequence of messages

**Certificate Omission:**

- for performance improvements

## Demos:

Since there wasn’t time to present them we included a few videos to try and demonstrate normal execution and a few attack attempts, along with a few notes to help make sense of the logs shown in the videos.

In all the demos we present a window displaying vehicles positions. The idea was to add vehicles through it and display vehicle beaconing ranges and other information. However due due lack of time we couldn't add all the features we wanted and it’s not fully functional nor tested. So unexpected behaviors may ensue.

## Demo: Default beaconing default_run.ogv

This is the normal case where every vehicle is behaving normally.

First the CA, the simulated Ad-Hoc environment and the RSUs are launched. After them 3 vehicles are launched and it’s possible to see them exchanging beacon messages.

The vehicles “complain” when they have another in the vicinity and stop when a collision is likely. However our prediction model is not completely accurate...

## Demo: Bad CA Certificates bad_certificates.ogv

We start again by launching the CA, the simulated Ad-Hoc environment and the RSUs.

The attacker (terminal below) contains a certificate signed by another entity other than the CA recognized by all the other vehicles and consequently they report this vehicle.

The CA (terminal in the top middle) revokes the certificate (for demo purposes the revocation threshold is after 2 attempts, from individual vehicles). The vehicles themselves never actually realize he’s revoked since they first check if it's signed by the CA.

The attacker also complains from invalid CA signatures on beacons since other vehicles don't use his wrong CA and he also tries to revoke them, however the RSU detects his invalid attempt and refuses his request since his certificate is invalid.

## Demo: Late Timestamps replay_messages.ogv

We start again by launching the CA, the simulated Ad-Hoc environment and the RSUs.

It's possible to see the attacker (vehicle terminal on the right) generating messages with “late” timestamps ( these are actually gathered messages replayed again ).

On the left the normal vehicle ignores these messages since they aren’t fresh.

## Demo: Wrong Message Signatures

tampered_signatures.ogv

This is a simple case where the attacker (terminal on the right) generates bad signatures on his messages.

These signatures are detected by the vehicle on the left and the message is ignored.

## Demo: False Positions data_trust.ogv

In this example it's possible to see the attacker (terminal below) generating false positions.

Both vehicles immediately report his certificate and the CA (top left) revokes his certificate (again after 2 attempts for demo purposes)