1. Problematyka. Cel i zakres projektu - //Bartosz Goss
2. Idea sieci VANET, opis założeń tej sieci - //Mateusz Lesiak
3. Dokumentacja techniczna - //Krystian Broniarek, Sylwia Mieszkowska
    a. wymagania funkcjonalne
    b. wymaganie niefunkcjonalne
    c. diagram klas
    d. architektura aplikacji, diagram pakietów, komponentów itp
4. User documentation - //Zbigniew Nowacki
5. Atak Bogus
    a. Teoretyczny opis idei ataku
    b. Dotychczasowe rozwiązania
    c. Opis stworzonego rozwiązania, diagramy, itp
6. Atak Sybil
    a. Teoretyczny opis idei ataku
    b. Dotychczasowe rozwiązania
    c. Opis stworzonego rozwiązania, diagramy, itp
7. Wnioski - //Adam Troszczyński

Vehicular Ad-hoc Network

Pracownia problemowa

Implementacja sieci:

1. Krystian Broniarek, 210
2. Zbigniew Nowacki, 210284
3. Filip Florczyk, 210175
4. Bartosz Goss, 207230
5. Mateusz Lesiak, 210248
6. Sylwia Mieszkowska, 210276
7. Adam Troszczyński 210342

Algorytm sprawdzania Sybil:

1. Dominik Ciesielski, 210155
2. Jędrzej Hasiura
3. Maciej Socha

Algorytm sprawdzania Bogus:

1. Kacper Prądzyński
2. Bartosz Kacperski 210210
3. Bartłomiej Szewczyk 210334

1. Aim and scope of the project

An ad hoc network is a wireless network with a decentralized structure, where every connected mobile device can be either a client or an access point. Transferring data doesn't require the existence of any network infrastructure because the communication between nodes in the network happens directly. This means that there is no need for any additional nodes to control the traffic. Ad-hoc nodes detect other nodes and connect with each other sharing information.

MANETs (mobile ad hoc networks) consist of mobile/semi mobile nodes. If the mobile nodes are vehicles then this type of network is called VANET (vehicular ad hoc network).

Modern vehicles are increasingly utilising benefits of being connected to the internet which is crucial for deploying solutions useful for both drivers and car manufacturers alike. VANET emerged as one of the solutions to connect vehicles into a network. VANET makes each cooperating vehicle a wireless router or a node allowing spontaneous creation of a wireless network for vehicle-to-vehicle (V2V) information exchange. This allows connecting every vehicle in range to a RoadSide Unit and creating a large scale network. VANET not only helps in traffic monitoring and safety but could also be used to increase traffic efficiency.
Nevertheless it is not immune to various kinds of attacks. Due to the growing popularity of such networks the problem of security is growing constantly, especially because a successful attack on a VANET can damage its infrastructure or expose the users to the risk of losing time, money, or even life. Even only one infected car may pose a danger to surrounding vehicles.

Before the concept of wirelessly connected vehicles was introduced, the automotive industry did not devote much time and resources to securing this aspect of their products. In the past, intruders required physical access to the vehicle to tamper with the software installed on its computer. Nowadays the connected vehicle is surrounded by many access points which utilize either LTE or Wi-Fi to connect to the internet. Because of that it is crucial that the user is updating the software installed in his vehicle to patch bugs or security holes, although it is worth noticing that a security patch, due to human error, might cause more vulnerabilities to the software.

The research of VANET security dates back to middle 2000s and the number of publications regarding this topic has skyrocketed around 2007. There are various attacks that can be performed in a VANET and they have been categorized by researchers into 5 classes:

1. **ID information** - a class of attacks utilizing stolen or forged identities to create disarray among connected devices. The most notable being a *sybil attack* - malicious node using many forged identities to declare that it's in fact multiple vehicles. The node then proceeds to broadcast information signed with those IDs to make the claims of for example a traffic jam more convincing.
2. **Sending false/altered information** - attacks targeted at the information carried by messages exchanged between vehicles. A great example of this type of attack is a *bogus information attack*, which distributes forged information to affect the decision of other nodes in the network. An attacker may distribute a message about an accident to redirect traffic somewhere else.
3. **Dropping, delaying or sending packages** - tampering with messages that are already sent, most notably refusing to forward critical information, as is the case in a *black hole* attack - the attacker either soaks the packets received by their node and doesn't redirect them, or redirects them to a wrong node.
4. **Listening and collecting information** - attacks that intercept information for the attacker's benefit. The leading example is an *eavesdropping attack* - the attacker monitors the traffic in a wireless network, gaining access to protected information that should be disclosed only to authorized users in the network. Collecting location data of a target vehicle could lead to identity theft or other misuse of knowledge about someone's daily routines.
5. **Tampering with VANETs technology and infrastructure** - attempts to destroy or corrupt VANET infrastructure. One of those attacks is a *GPS spoofing* attack - deploying a satellite imitating device that produces a signal stronger than the actual satellite system. Nodes reading fabricated coordinates position themselves in wrong locations.

VANET security remains an open issue but researchers are coming with more and more solutions to those attacks. Even though the research doesn't tackle estimating the potential benefits the attacker would gain, the possibility that an attacker would do it purely from malicious intents is enough to justify the time and resources devoted to securing the network. The aim of this project is to create a simulation of a VANET - a sandbox allowing deploying and monitoring virtual legitimate nodes and malicious ones performing attacks on the network. We have chosen to implement nodes performing bogus information and sybil attacks. The other, most important, goal is implementing our own countermeasure algorithms to neutralize the threat posed by said malicious nodes and measuring their effectiveness.

2. Idea of network and description of the assumptions

Although VANET is not a new topic, it continues to provide new challenges and problems. The main idea of this network is to help groups of vehicles to communicate with each other with information about the road and environment. Information such as speed hike or limit, sharp turn, road conditions, police on road etc. Also they share information about themselves such as vehicle speed, direction, route etc. Managing all this information in one app, can lead us e.g. to less corked cities.

VANET is responsible for communication between moving vehicles in an environment. This network treats any vehicle as a node. After that, nodes connect themselves if they are in their limited range. And if they are connected, they exchange information about events. There are three types of nodes:

1. **RoadSide Unit (RSU)**
2. **OnBoard Unit (OBU)**
3. **Application Unit (AU)**

Let's describe this nodes:

- **RoadSide Unit – RSU** – represents the constantly available nodes on the route such as road events. Usually the RSU hosts an application which provides service to other nodes. The application possibly will reside in the RSU or in the OBU.
- **OnBoard Unit – OBU** – this applies to mobile nodes (such as vehicles) equipped with an electronic interface that connects wirelessly to other nodes. The OBU is a peer device connected to RSU.
- **Application Unit – AU** – this is a device which is capable of within the vehicle which uses the applications provided by the provider. AU us communication capabilities of OBU. Usually AU is connected to OBU wireless by it is common to find it connected through a wire.

Characteristics of the Ad-Hoc network:

- **Mobility** – in VANET network there are nodes connected to each other only for a couple of seconds. In such a small amount of time and such highly inhomogeneous environments, they need to send all important information about the situation on the road.
- **Security** - VANET is and open network in which any node is allowed to connect to it. There is no certain security mechanism to ensure that node is reliable. This

identification is really difficult. What is more, untrustworthy nodes can send incorrect data to other nodes which is unacceptable.

- **Scalability** – VANET is easily scalable. All you need to do to increase the performance is to make your application provided with more IT resources. For large scale VANET's, number of nodes participating in a network can increase to up to millions. This of course leads to large amount of data.
- **Energy conservation** – The only thing that VANET application has to worry about is their application. One application for all vehicles instead of millions of small applications is really easier to develop and update.
- **Self-organization** – This network is self-organization because nodes can join it without asking permission. Network can increase completely without any human intervention.

3. Technical Documentation

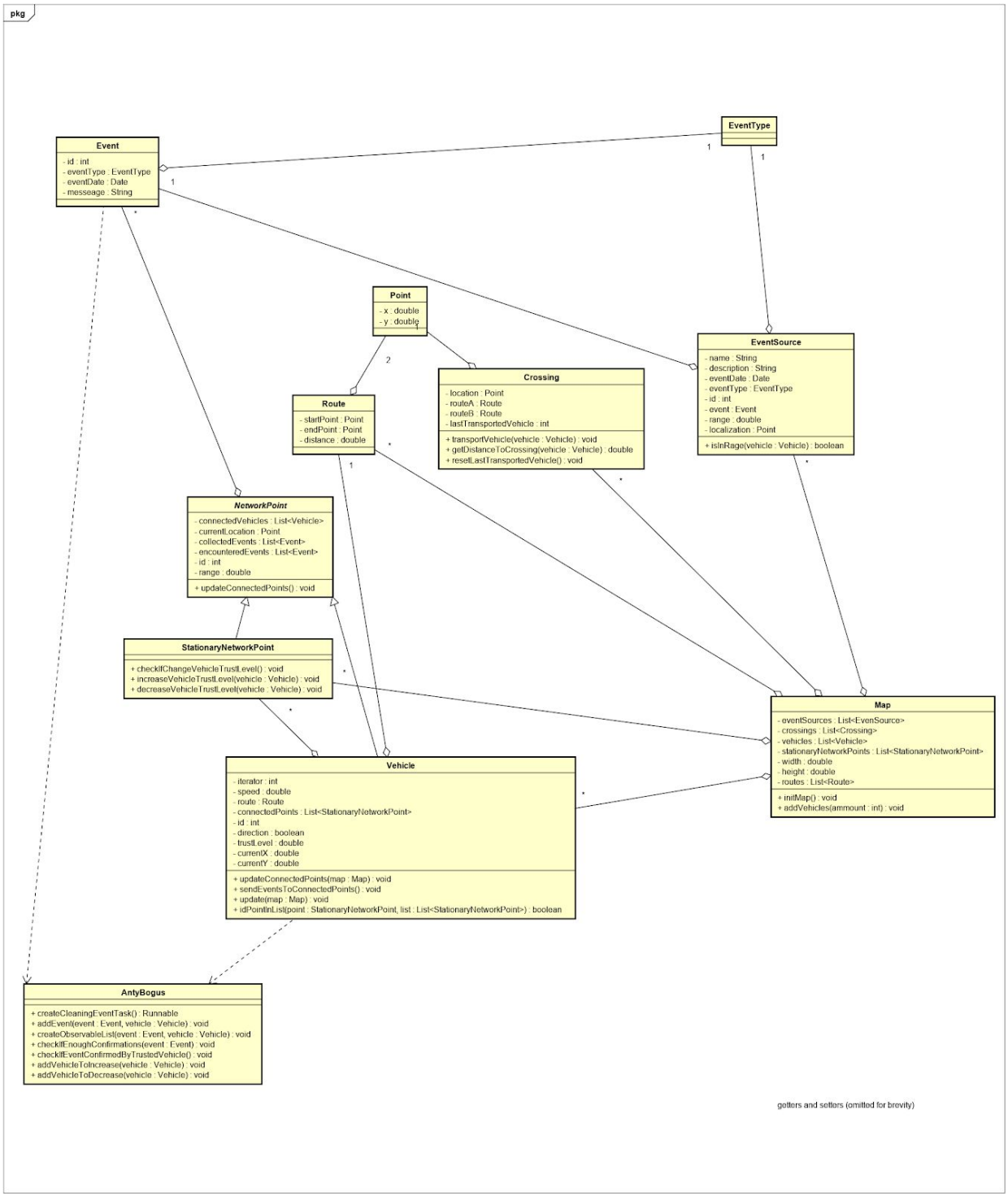
3.1. Functional requirements

1. Simulation and visualisation of cars movement on the street grid in a random way
2. Adding any number of cars to the street grid
3. Stopping and restarting simulation anytime
4. Displaying actual informations about selected car and modifying them anytime
5. Defining static network access points
6. Interacting individual cars between them and between static access point through creating small information exchange networks
7. Defining event points and propagating them through vanet network
8. Signal range visualisation of every car

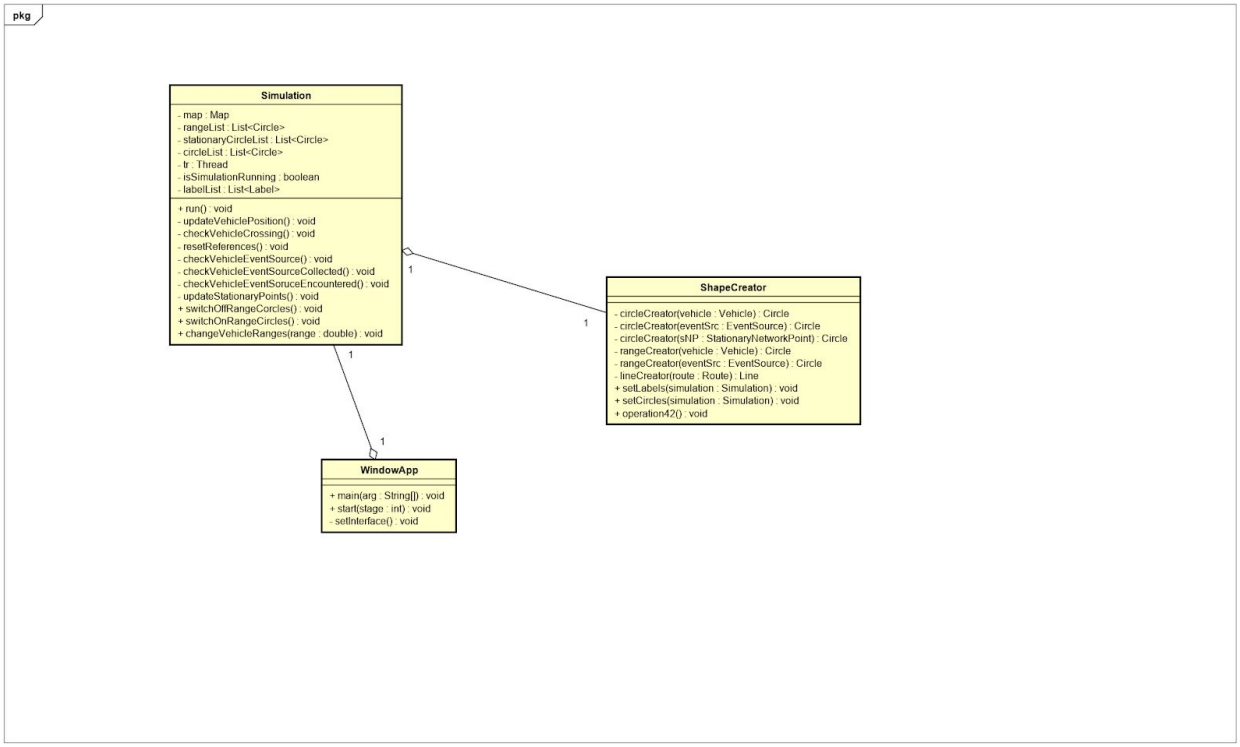
3.2. Nonfunctional requirements

1. System designed for one user without internet connection
2. Software is using Java SE Runtime Environment 8 to run
3. Software is using JDK 11 for building
3. Simulation and GUI is running on separate threads, but every move of simulation is adapted to not cause thread synchronization errors
4. JavaFX interface, version 11
5. Lombok version 1.18.6 for easy data class managing

3.3. Class diagram



Class diagram for logic layer. Represents basic functionality and repository for maintaining simulation.

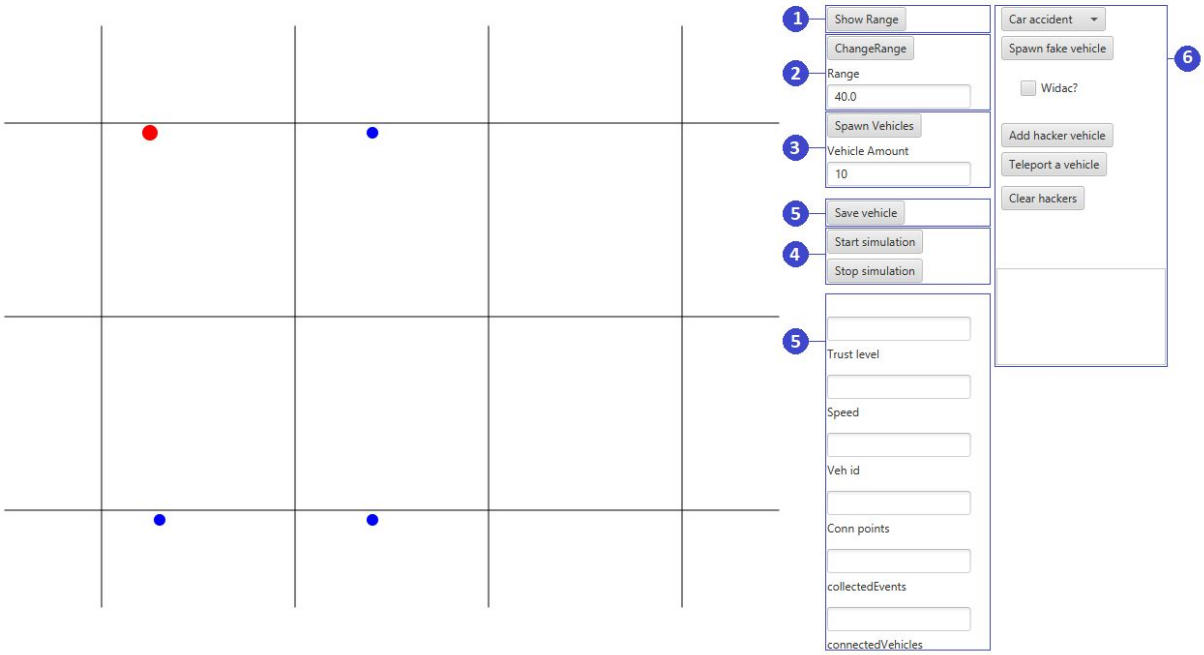


Class diagram for visualisation. It contains classes which allows to build interface, visualisation of map and manage simulation's flow.

3.4. Package diagram

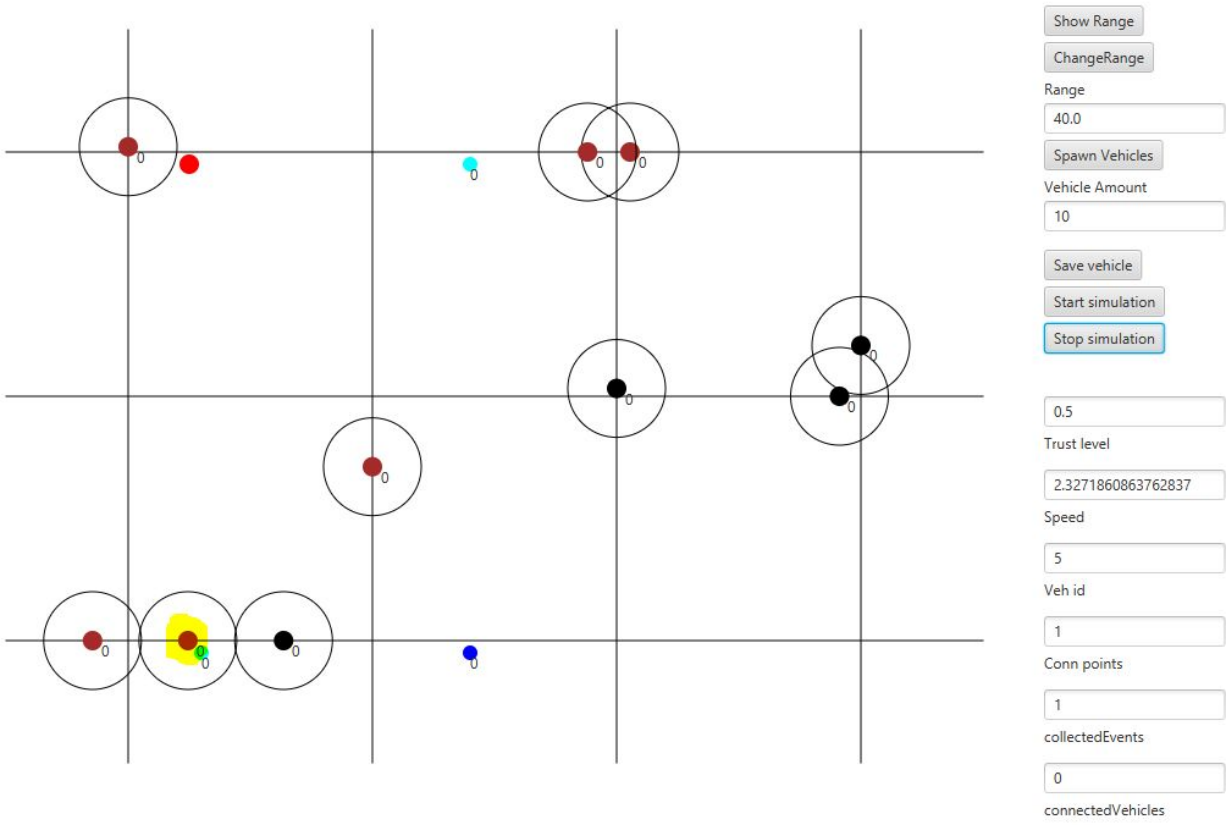
4. User documentation

After starting the application, we get the following view:



The basic operation is carried out using the following buttons:
[Show Range] - turns drawing the range of vehicles on and off (1)
[ChangeRange] - confirmation of the vehicle range change introduced in the text field “Range”; default value - 40.0 (2)
[Spawn Vehicles] - creates the number of vehicles specified in the "Vehicle Amount" field; default value - 10 (3)
[Start simulation] & [Stop simulation] - respectively plays or stops the simulation (4)

In the right section (6) there are buttons to handle attacks:
[Spawn fake vehicle] - spawns vehicle that broadcasts information choosen from dropdown beyond (car accident, speed camera, police control).
[Add hacker vehicle] - spawns hacker vehicle, marked red; hackers can be made invisible with [Widac?] checkbox and cleared with [Clear hackers] button
[Teleport a vehicle] - teleports random vehicle



After clicking on the vehicle, the following information appears (5):
- Trust level
- Speed
- Veh id
- Conn points
- collectedEvents
- connectedVehicles

The first two of them can be edited and then the changes can be confirmed by clicking [Save vehicle] (5).

5. Atak Bogus

5.1. Theory of Bogus Attack

Bogus information attack is a type of attack on the VANET network consisting in sending false information to gain personal benefits. In this attack, the attacker can be outsider/intruder or insider/legitimate user. The attacker broadcasts false

information in the vehicular network to affect the decisions of other vehicles by spreading the false information in the network. For example a vehicle can imitate a heavy traffic on one road to prevent another vehicle from choosing that road. This attack is an example of Application attack.

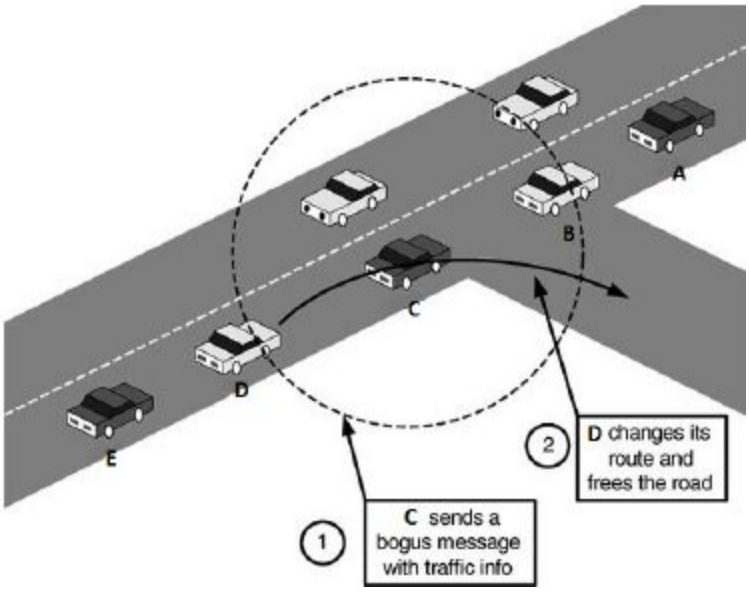


Fig. 1. Bogus information attack

Picture above demonstrates an example of bogus information attack, colluding attackers (**A** and **C**) disseminate false information to affect the decisions of other vehicles (**D**) and thus clear the way of attacker **E**.

In the attack scenario, the attacker chooses a node as its victim, and then prepares a RREQ or beacon packet for AODV and GPSR respectively as generated from the victim. The packets are generated for a randomly selected destination node, and the attacker node broadcasts these packets on behalf of the victim node every five seconds. The attacker attracts traffic by being the freshest node or the closest node to the destination in AODV and GPSR respectively. Again, any packets transmitting through the attacker will be dropped. This attack could also be used to isolate a node from the network; however, it will have little effect on the network due to the fast changing topology of VANETs.
Packets not transmitted through the attacker will remain unaffected.

Bush telegraph is a developed form of the bogus information attack. The difference in this case is that the attacker possesses multiple entities spread over several wireless hops. It is worth mentioning that after receiving a packet, a hop checks the error. If the error is small enough to be considered within tolerance margins, this error could be

tolerated and ignored. Abusing this vulnerability, a bush telegraph attacker appends incremental errors to the data at each hop. At each hop, the error is probably small enough to be tolerated and hence accepted by the neighbor. After passing several hops, the overall accumulation of these errors eventually yields to bogus information.

The situation in which there are more attackers in the VANET network than normal cars is an interesting case. In this scenario, we can not say which group of cars is a group of false information attackers. There is no reason to state that information shared by most of cars is fake.
This situation shows that in VANET network we can not state that shared information is in one hundred percent true. We always have to trust someone.

5.2 Ways of Defense

There are several solutions that provide protection against Bogus attacks. Below we will describe a few of them.

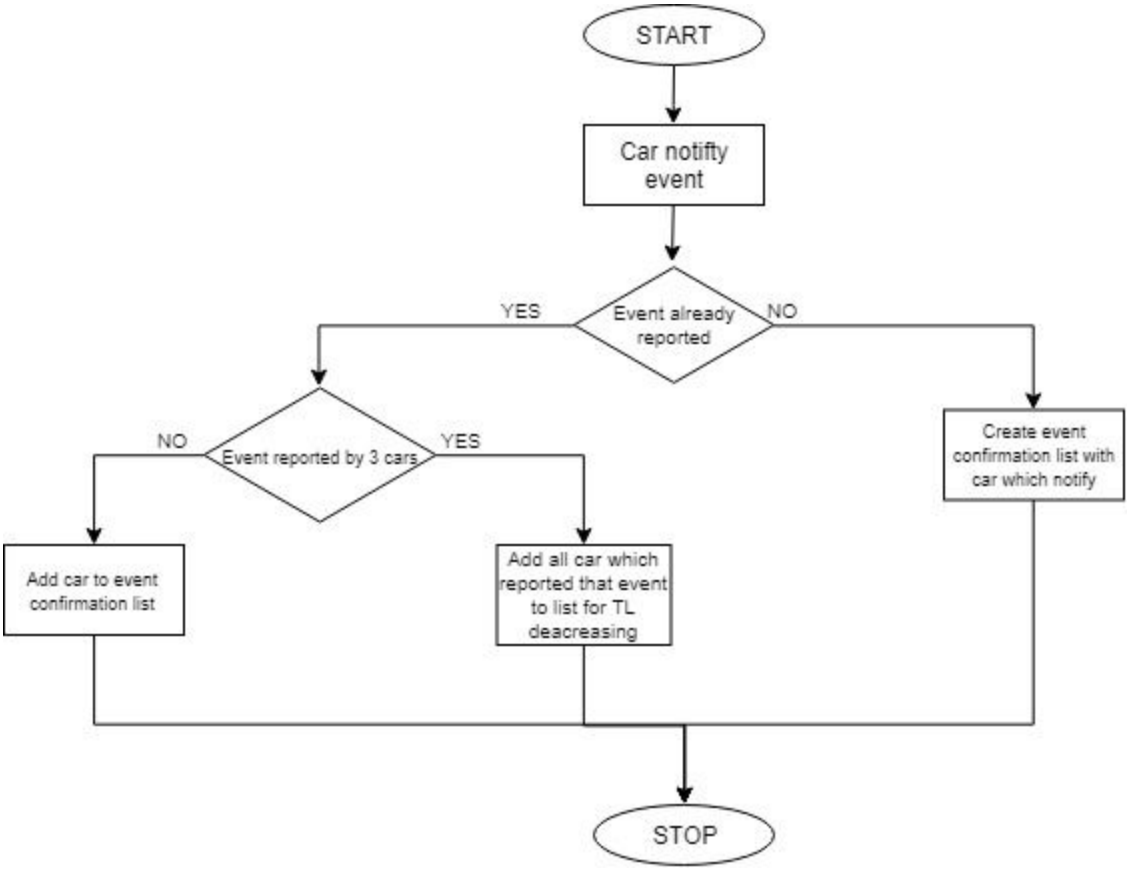
One of the solutions is the ECDSA algorithm (Elliptic Curve Digital Signature Algorithm). It uses a hashing technique to ensure a high level of authentication security for cars receiving information. To implement this solution each vehicle must have a public key as well as a private key. The car that wants to send the message firstly must use the hash function on messages and then encrypt them using the private key. The car, receiving information decrypts it using a public key that is visible to all network users. This algorithm is one of the types of encryption used DSA.

Another way is to ensure trust based on TRIP (Trust and Reputation infrastructure based proposal) - algorithm for traffic analysis. TRIP identifies malicious nodes that spread false or crafted information on the network VANET. Warning messages and messages are sent to another node which checks the reputation and credibility of the sender's node. If the node turns out to be malicious received messages are dropped. Fuzzy logic classifies a node based on the value obtained from three pieces of information: the previous value of reputation, surrounding vehicles and recommendations of the central body. There are three types of trust values: untrusted - reject all packets, trusted - accept all packets but not them send and reliable - accept all packages and send them on.

Another way to handle false information is RABTM, a system based on BTM (Beacon-based trust management) and RSU (Road-site unit). It involves the use of two different methods of trust - indirect and direct. The indirect method uses beacon signals to determine the position, speed and the direction in which the signal transmission object moves. Coefficient trust is calculated, which is then compared with the read data and determines the credibility. The direct method, on the other hand, determines the credibility of messages based on the relationship receiver / sender using RSU.

5.3. Algorithm description

Our implementation of anti-bogus algorithm depends on car trust level (TL). Each vehicle has its own degree. Maximum trust level value is 1.0 and minimal is 0.0. It can be increased by 0.1 when car notifies about an event and three cars confirm it. Level will be decreased by 0.4 if not enough cars confirm that event within 30 seconds. Responsibility for changing TL belongs to stationary network point. When vehicle is in its range it checks if trust level needs to be decreased or increased. Car is considered as untrusted when its TL drops below 0.5.



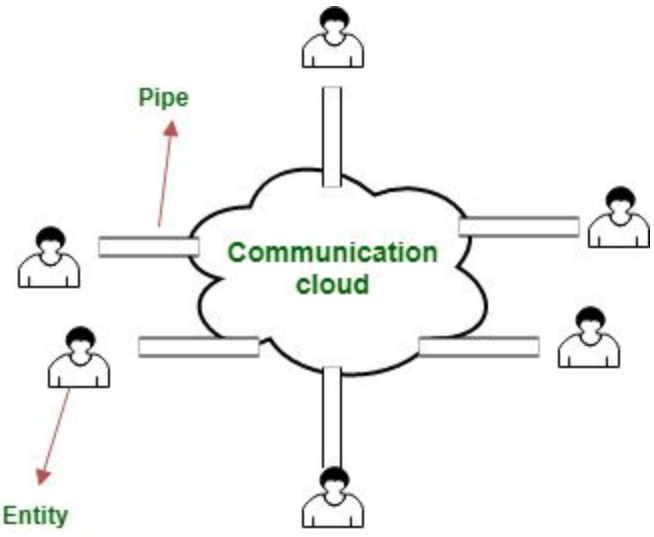
5.3.1. Algorithm flowchart

6. Sybil Attack

6.1. Sybil Attack in Theory

A Sybil attack is a widely known attack in the peer-to-peer network, wherein a reputation system is subverted by forging false identities. The name "Sybil" comes from a subject of a case study, a woman called Sybil Dorsett with a dissociative identity disorder and was first used in 2002 by Brian Zill at Microsoft Research.

In a civil attack, the attacker creates a large number of false identities and uses them to gain a disproportionately large influence. The network's vulnerability to this type of attack depends on how easy it is to determine which of the present nodes are false and on how easy it is to create the false ones. As of today we can easily say that an attack of Sybil type can be very easily carried out in a very cheap and efficient way.
An entity on a peer-to-peer network is a piece of software which has access to local resources. All of the entities present in the network, advertise themselves by presenting their identities and it's through them that the system has to see whether the node is real or just the illusion created by the attacker. To further describe the Sybil attack, we can use the diagram below.



The shown model used in a civil attack is a simple one. It consists of:

- The entities - which is a sum of both proper and improper nodes present in a system, they follow the protocols and rules present in the network.

- A communication cloud - a very general cloud, through which the nodes can communicate with one another and through which the messages can be sent.
- The pipes - which are connections between an entity and a communication cloud.

On VANET type network, the attack relies on creating additional illusions of vehicles to gain advantage in the network. To succeed in this process, the invader might use the authentication data of another vehicle to make it look like it's in multiple locations, to generate false information about the road or to steal the access data of other users. The illusions that are created like this by the attacker, are commonly called "Sybil nodes" and will be called accordingly in future.

An attack that is conducted this way, gives the invader a multitude of options to choose from:
- They can wreak havoc on the network to make it less reliable (ex. by simulating the presence of a vehicle in many locations),
- They can provide false data and make it believable by confirming it with other Sybil nodes,
- They can stress the system by generating large numbers of Sybil nodes and make the network run slower or even crash it.

6.2 Ways of Defense

There are already multiple solutions to the sybil problem. In the points 2.1, 2.2 and 2.3, they will be shortly described.

1. Signed timestamp

One of the possible solutions to the problem is assigning the vehicle in the VANET network a signed timestamp. Every RSU unit refreshes that timestamp in a way so it can be verified by the next RSU unit that the vehicle is driving by. Every other RSU unit calculates the time difference between the present time and the timestamp, and it checks whether it was physically possible for the vehicle to travel that distance in the calculated time. This method is one of the easiest ways to detect sybil nodes that are simulated and are jumping between locations to create chaos in the network.

2. Sybil node recognition by collecting data

Another solution to the problem is a data collection method. Even though it is used in cases of a highways and freeways, it is a valid and commonly known way of protecting the network. It consists of three phases: probing, confirmation and quarantine.
The probing phase the nodes collect information about other nodes that are geographically in front of and behind them. With this information, system is able to single out contradictions in the data and take appropriate action. The nodes that provided the false data and the nodes that were included in it are now being watched more carefully – this begins the confirmation phase. This phase's goal is to determine which of the suspicious nodes is or are the sybil ones. Once it is confirmed, the sybil node is put in quarantine in which the data provided by it is ignored.

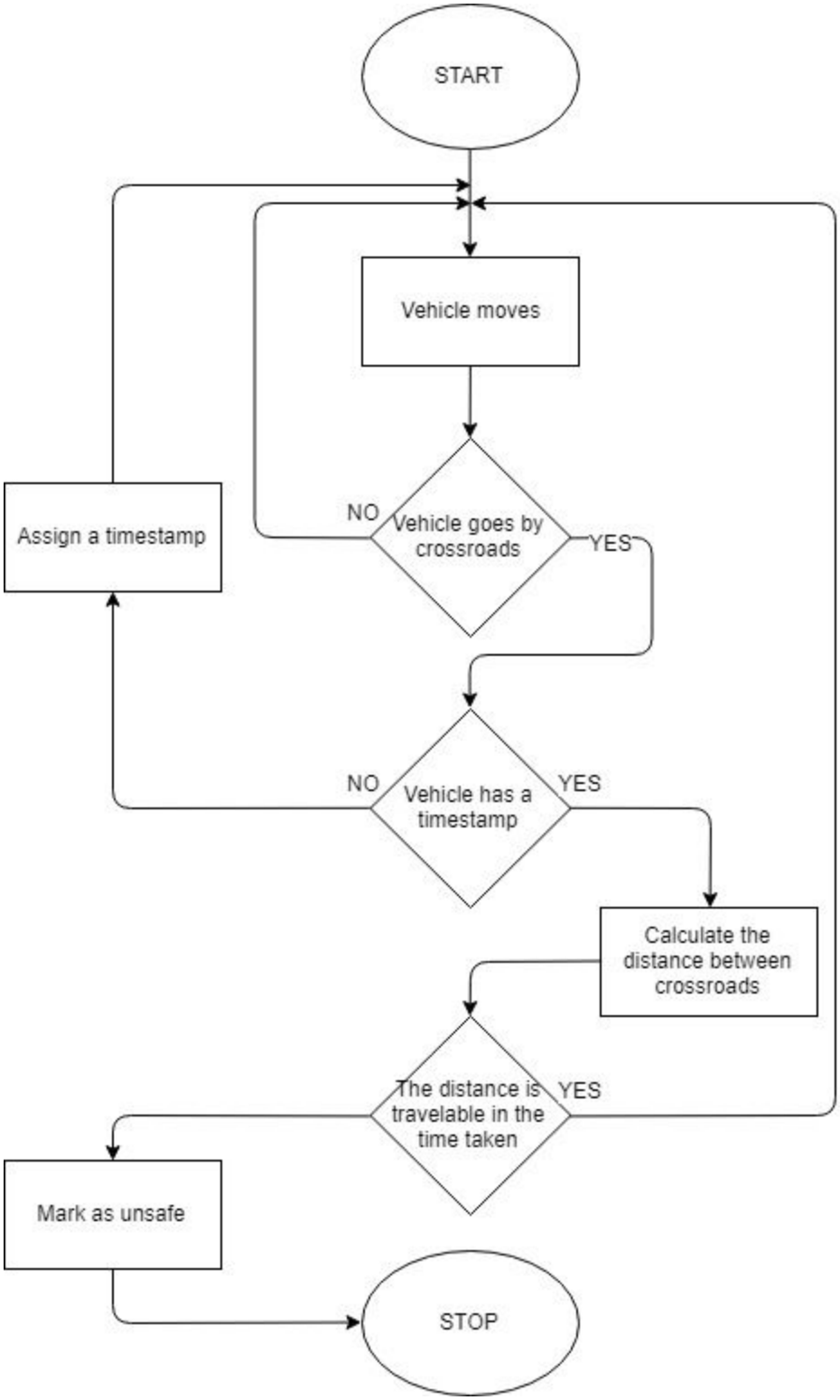
3. Resource test

The third method to distinct the sybil node is a resource node. The system randomly sends complicated math equations to the nodes (in the VANET network case, to the dock computers in the vehicles), which are expected to be resolved in a given time. Attacker that simulates multiple nodes will not probably be able to resolve all of sent equations, making his simulated vehicles compromised.

6.3. Description of the implemented solution

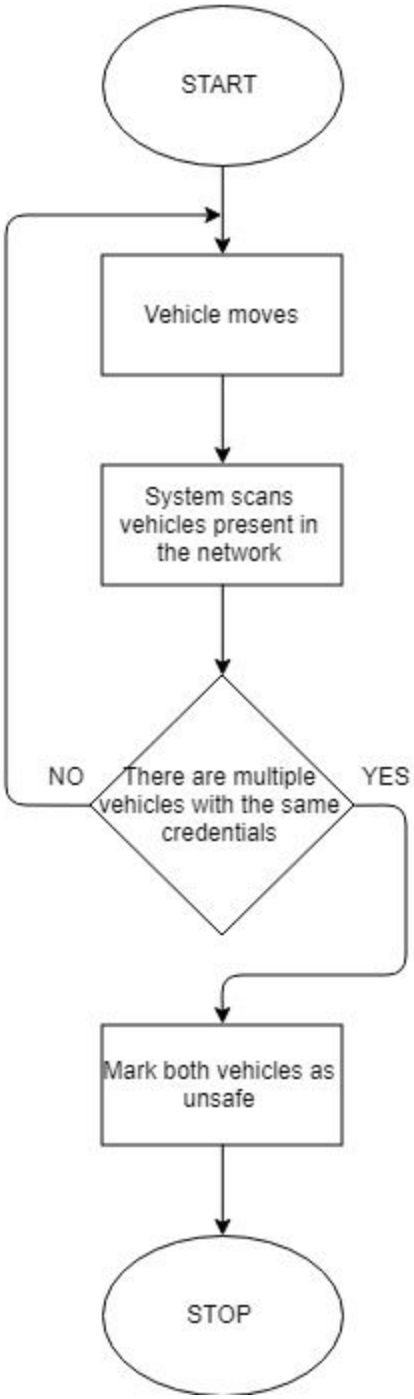
The solution that was implemented in the app is the one described in the point 6.2.1.

Each time, a node (vehicle) drives by a crossroads it is given a timestamp consisting of a present day and hour. When the node travels by another crossroads, after checking whether it was possible for the vehicle to travel the distance with its given speed, its timestamp is renewed. This kind of defense mechanism helps preventing spreading false information by the sybil node that changes its location. To simplify the algorithm it is was depicted in the picture below.



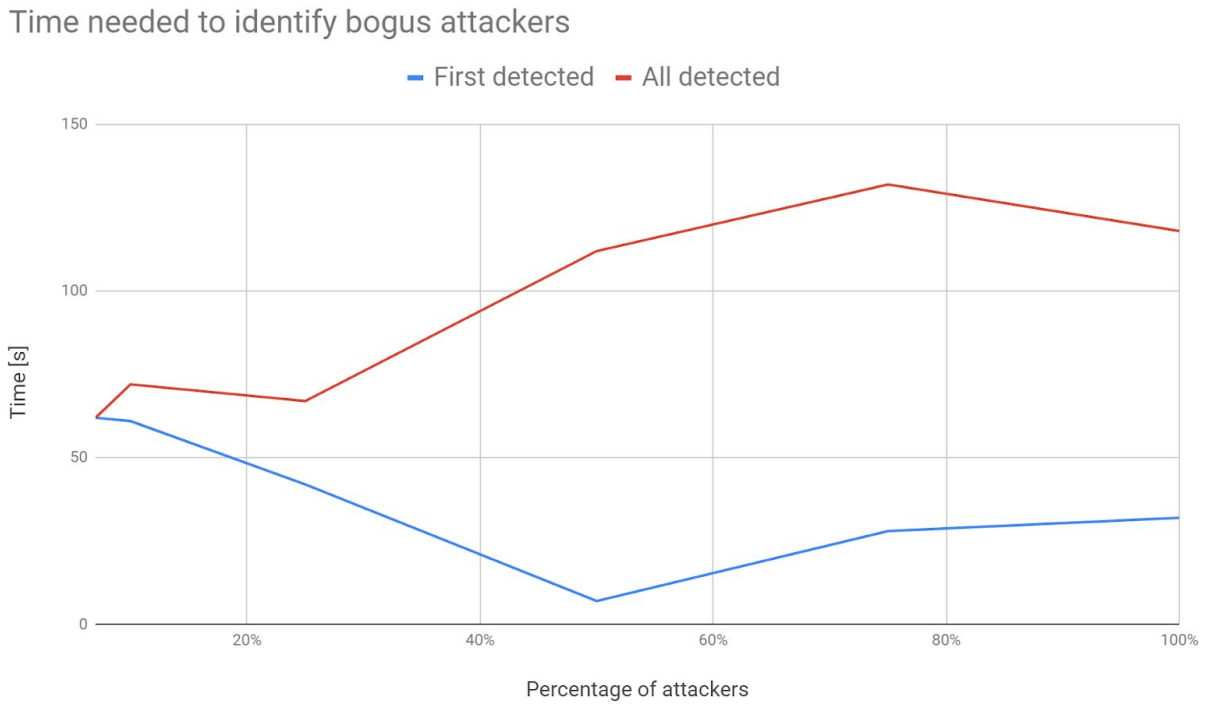
Also a mechanism that checks whether there are no multiple nodes in the system. The network is scanned once a second to find if there are any sybil nodes that use the same

credentials. If any of those algorithms finds a sybil node, it is put in a quarantine list that is shown in the bottom right. To simplify the second algorithm it is depicted in the picture below.



7. Experiments

Bogus attack				
Number of users	Number of attackers	Ratio of attackers to all nodes	Time needed to identify first attacker **[s]**	Time needed to identify all attackers **[s]**
12	1	7%	62	62
8	2	20%	61	72
12	4	25%	42	67
12	12	50%	7	112
4	12	75%	28	132
0	16	100%	32	118



8. Conclusion

The network lets its users share their knowledge about onroad events and traffic between themselves. Although this process of sharing information is incredibly useful, it also can be a subject of attacks. There are various ways to misinform others and various reasons to do it. Every user should be aware of the risks of any technology as well as every engineer should be prepared for his system to be attacked. We know many ways to spread misinformation through a VaNET network or just peer-to-peer networks and we have prepared safety measures against two types of attack: sybil and bogus.

Bogus attack is an attack used to share misinformation through replicating messages with false information. We tested a solution with applying to each node a trust level and here are our conclusions:

- Identifying vehicles that broadcast false information into the network (bogus attack) seems inevitable independently of their numbers, although the more of them, the more time it takes, which is something that contradicts our expectations. When the number of attackers overcomes 50% of all nodes, the network should not be able to distinguish the attackers from normal nodes. There could have been a difference if the nodes were more dense in the network we could observe a difference.
- When there are very few bogus attackers (less than 10% of all nodes in the network) it takes approximately thrice as much time to identify any of them, which might be an abusable weak spot of our network. Although it is harder to detect the first attacker, making more attacking nodes is more efficient.

Sybil attack relies on stealing other node's identity and share information as if the information originated from that node. Our safety measures consist of applying timestamps at each crossroads as well as checking each node's identity looking for duplicates. The conclusions are as follows:

- Sybil nodes that try to steal somebody's identity are almost immediately detected and both original vehicle and sybil node are marked as untrusted. This case might be a base for another attack, which we didn't prepare for. Since both vehicles are untrusted, attackers may duplicate every node in the network once and as a result paralyse all information traffic in that network.
- Network marks all vehicles that move between two RSUs too quickly immediately as untrusted, as they might be sybil nodes that fake their positions. This is especially useful against nodes that can falsificate their GPS signal or just inject their false position directly into the network.