**Detection and Prevention of Security attacks in VANET.**

Abstract

*Abstract*— In the current generation, road accidents and security problems increase dramatically worldwide in our day to day life. In order to overcome this, Vehicular Ad-hoc Network (VANETs) is considered as a key element of future Intelligent Transportation Systems (ITS). With the advancement in vehicular communications, the attacks have also increased, and such architecture is still exposed to many weaknesses which led to numerous security threats that must be addressed before VANET technology is practically and safely adopted. Distributed Denial of Service (DDoS) attack, replay attacks and Sybil attacks are the significant security threats that affect the communication and privacy in VANET. As simulators are being used in our work, we have discussed OMNET++ which is a new modernized latest mobility and network simulators as well as data network simulator. This is also integrated with the road traffic simulator SUMO with Veins, an open-source framework for VANET simulation. The objective of our work is to design an algorithm to detect and prevent various kinds of attacks using Java Security and Cryptography libraries. An analysis has also been done by applying four protocols on an existing scenario of real traffic simulator using OpenStreetMap and the best suitable protocol has been selected for further application.

Keywords—VANETs; Security Attacks; Certificate Authority; Java cryptography; Sybil attack; DDOS attack; Reply attack

**Introduction**

Vehicular Ad-hoc Network (VANET) provides a smart transportation system owing to Vehicle to Infrastructure (V2I) and Vehicle to Vehicle (V2V) message dissemination with an objective to provide safety on roads. VANET in comparison to Mobile Ad-hoc Network (MANET) points to an exceptional kind of networking with high mobility nodes which are vehicles. Major applications of VANET include electronic brake light, parking management and point crash notification. The topology in VANET varies according to vehicle movement scenario such as traffic light, highway and urban road scenario. The contribution to the area of VANET by the research community on various layers is on the rise. The simulator close to the real-time set up for VANET is the preferred choice of researchers as it involves less cost in comparison to really set up. The recent updates about security attacks in VANET. Trust and Privacy are explained in Ref. [1].

Nearly 1.25 million people die each year and on an average 3,287 deaths, a day is observed according to the world health organization [2]. Many engineers from a different area of studies such as vehicle designers and road engineers help in the reduction of the number of road accidents. VANETs are applicable in both rural and urban. Constraints of VANET is that the velocity of vehicles are purely depending on the speed limit of the roads they travel. Traffic signs and signals are bound to be followed by all vehicles which make VANET less difficult than MANET.

This paper proposes the methods for detecting and preventing security attacks of both V2V and V2I communication in VANETs. More specifically we are going to address security in VANET “**beaconing messages”**, i.e., messages sent from a vehicle to its neighbours with information of location, speed, braking and other sensorial data that aid safe decision making.

The main contribution of the paper has been given in the following points:

* Detecting the few security attacks which are highly probable and that exploit the both V2V and V2I communication by sending the false information to other vehicles and RSU.
  + Sybil Attack
  + Reply Attack
  + DDOS Attack
* Design of Algorithm for detecting the Sybil nodes by considering the timestamp and velocity parameter of the beacon messages sent from the vehicle nodes which estimates the distance.
* Provided various measures to prevent the attackers in VANET communication.
* Implemented a system which detect the best routing protocol for the VANET communication by considering the realistic scenario using SUMO and NS3.

The organization of the rest of the paper is as follows. In Section 2, discussed the background of the related work on detecting and preventing the Security attacks in VANET Communication. Further, discussion of the proposed network architecture are done in Section 3. Section 4 contains the proposed methods for detecting and preventing the Security attacks in VANETs. Then, Simulation Results and Analysis of the proposed mechanism has been done in Section 5, finally, last section concludes and discusses the future enhancement of the proposed Algorithm.

**Background**

Vehicular ad-hoc network (VANET) is a mobile, wireless, self-organized network characterized by high mobility of nodes, dynamic topology and low latency demands [6]. Vehicles are almost constantly in motion at relatively high speed, whilst often travelling in the opposite direction, resulting in frequent network topology changes. New connections are regularly established with vehicles joining the network, while on the other hand nodes leave the network and connections are lost.

The technology behind the inter-vehicular communication is being formalized into standards – ETSI ITS-G5 [7] in the EU and IEEE 1609 (WAVE) [8] in the US. Both specify common physical layer based on standard IEEE 802.11p and both operate in 5.9 GHz radio frequency band. Their differences are becoming a little more obvious in the higher layers of the protocol stack, even though they are still somewhat similar in principle. Regarding the safety-critical applications, there are two types of messages defined in the application layer - periodical awareness messages and event-based messages. The former, referred to as BSM (Basic Safety Message) in the US standard and CAM (Cooperative Awareness Message) in the EU standard, respectively, carries basic information about vehicle state. Each vehicle broadcasts this message in short regular intervals. In contrast, the latter, known as BSM part II in the US and DENM (Distributed Environmental Notification Message) in the EU, is being sent solely in response to certain events. In addition to information about the event, it also includes a number of further vehicle parameters and data on past and predicted future position.

The problem of secure vehicle-to-vehicle communication is the subject of several research projects, therefore we chose only a fraction of the most notable ones. Current state-of-the-art works are funded pre-dominantly by the European Commission in the EU and by the National Highway Traffic Safety Administration (NHTSA), federal agency of the U.S. DoT. SeVeCom (Secure Vehicle Communication), which ran from 2006 to 2009, was among the very first research initiatives in this field. The project focused on identifying security threats and specifying security architecture of the inter-vehicular communication. SeVeCom addressed key and identity management, secure communication protocols including secure routing, device tamper-proofing and privacy issue [9]. Aforementioned NHTSA, in collaboration with the consortium of automakers associated in CAMP (Crash Avoidance Metrics Partnership) set similar goals for their VSC (Vehicle Safety Communications) programme. Under the proposed security architecture each message would include substantial overhead and the message signatures would take time to process once they are received. Authors also stipulate that the management of a public key infrastructure (PKI) for roadside units would be necessary [10]. Building on theoretical foundations laid by a number of existing projects, authors of PRESERVE (Preparing Secure Vehicle-to-X Communication Systems) program successfully managed to implement and field test security subsystem of the communication system [11]. Results from the research projects have also contributed to the development of security architectures present in ETSI ITS-G5 and IEEE WAVE standards. The security is based on digital signatures and PKI which can be used to provide security services needed to mitigate common attacks (e.g. message replay, falsification, or eavesdropping) [12].

**Network Architecture**

In this section we describe about the attacker module used in our implementation system and then define Known type attack scenarios exploiting V2V and V2I communication in VANET System.

*A. Attacker Model*

We consider the pursuit of personal benefit as the most likely driving force behind the attack, i.e. an attacker freeing the road for him/herself. The adversary is an external active attacker with the ability to modify the software of the on-board unit within his/her possession. We consider a mobile attacker whose position and speed can change in time. To execute an attack successfully, the adversary must be in the victims’ communication range.

*B. Replay Attack*

Replay attack is a type of network attack in which the attacker listens to the communication channel and intercepts messages so that they can be rebroadcasted later. Utilizing this approach would allow attacker to impersonate an emergency vehicle (e.g. an ambulance) to free the road for him/herself. The scenario is as follows: attacker is following the emergency vehicle. Using BSM part II message, the emergency vehicle is notifying all vehicles in its communication range about its presence. Message is received by all vehicles in a close proximity including attacker’s vehicle. While legitimate users being approached from behind by the emergency vehicle respond by decelerating and pulling over to the side of the road, attacker captures and resends the message originating from the emergency vehicle. Legitimate users respond as if it was another emergency vehicle. Attacker takes the advantage of free road lane. Furthermore, messages could be stored or shared with other attackers and consequently exploited repeatedly in the future.

*C. Message Falsification*

Message falsification is an attack in which the attacker modifies existing valid message and retransmits this falsified message. The attack scenario is similar to the scenario of replay attack, the only difference is that in this attack the attacker does not need the presence of the emergency vehicle since he/she can modify any intercepted message or fabricate a completely new false message.

*D. Sybil Attack*

Sybil attack is an identity spoofing attack that is based on creating a forged identity in a network and misusing it for his needs. Sybil attack in VANET could be exploited to free the road by imitating a traffic jam. In this attack the attacker simulates several different counterfeit identities through which he/she disseminates status messages that give an impression of multiple standing or slow-moving vehicles. Upon receipt of those messages from counterfeit nodes fabricated by the attacker, legitimate user perceives inevitable traffic jam, hence responds by decelerating and remains in his/her current lane. The danger of this attack lies in the attacker being able to simulate complex traffic situations involving several vehicles which are, in fact, virtual, but perceived as real by legitimate users. Thus, victims could be navigated right where the attacker wants them to be. Sybil attack can be static or dynamic. Static Sybil attack generates counterfeit nodes (referred to as Sybil nodes) with fixed positions while the dynamic attack prepares Sybil nodes for each vehicle. For our scenario we have chosen the dynamic version of Sybil attack which is more efficient in achieving the attacker's goal. The scenario consists of several steps:

1) Monitoring of surroundings and collecting the periodic awareness messages in order to identify the victims.

2) Maintaining two lists that store the information about last known position of all victim vehicles and parameters of Sybil nodes for each victim.

3) Controlling the Sybil nodes. After receiving an awareness message, the attacker firstly checks if the corresponding sender is located before him/her. If the condition holds true (there is no sense to attack vehicles behind), the attacker evaluates if he has already a list of Sybil nodes for this vehicle. If yes, he can use them to further slow the vehicle down. In case of new vehicle the attacker proceeds with fabricating the Sybil nodes for it (i.e. broadcasting awareness messages simulating standing vehicles in front of the victim vehicle).

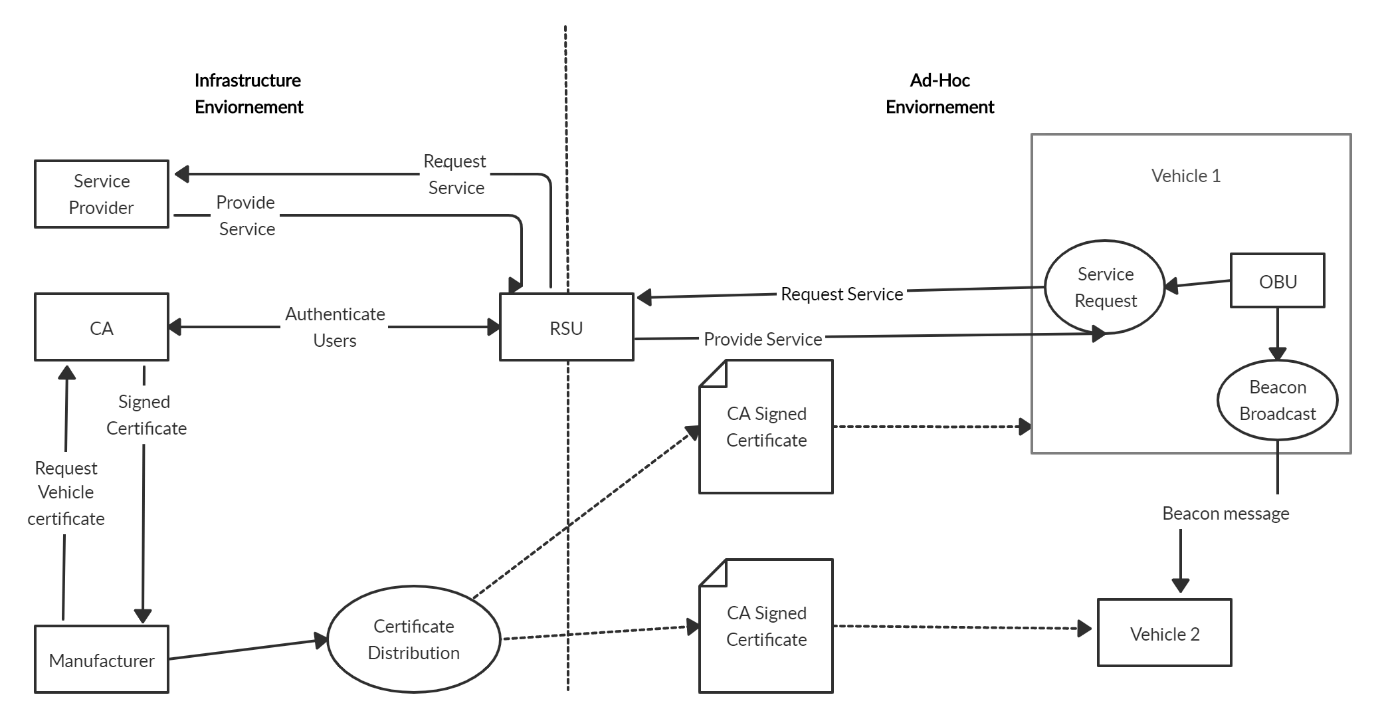


Figure . System Architecture of proposed mechanism

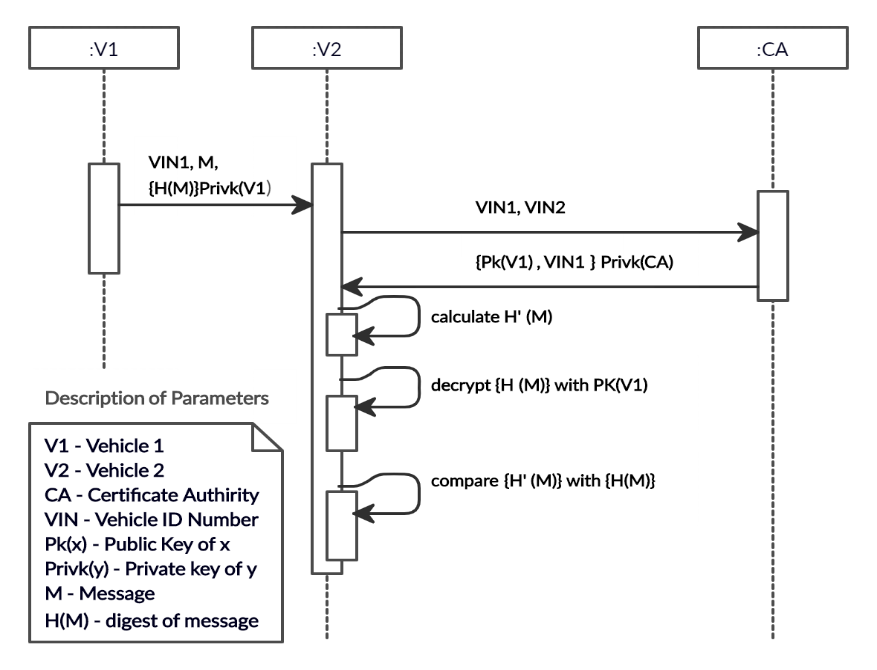


Figure . Message Exchange Mechanism

**Table 1**

**Simulation Parameters and Its Values**

|  |  |
| --- | --- |
| Parameter | Value |
| Max Beacon Range | 500 m |
| Too Dangerous Range | 100 m |
| Max RSU Range | 5000 m |
| Beacon Interval | 2 s |
| Danger Reset Interval | 1 s |
| Check RSU range Interval | 2 s |
| Max Interval Vicinity In Cache | 60 s |
| Network Position Update Interval | 2 s |
| Freshness Max Time | 2 s |
| Max Revoke Score | 2 |
| Held Beacons for Replay Attacks | 20 |
| Attacker Position Range | 5000 |
| CA Digest | SHA-256 |
| The Height of Network | 1800 m |
| The Width of Network | 1800 m |
| Encryption Technique | Vehicle RSA |
| mac/phy | IEEE 802.11p |

**Table 2**

**Notations and Definition**

|  |  |
| --- | --- |
| Notation | Definition |
| CA | Certificate authority |
| RSU | Roadside unit |
| *Vi* | *i*-th vehicle |
| VIN | Vehicle Id number |
| *PX* | The public key of x |
| *PrvtY* | The private key of y |
| *M* | Beaconing message |
| *H(M)* | Digest of message M |
| SUMO | Simulation of urban mobility |
| VANET | Vehicular ad hoc network |
| DDOS | Denial of Service |
| VANET | Vehicular ad hoc network |
| PDR | Packet delivery ratio |
| PLR | Packet loss ratio |
|  |  |

**Proposed Mechanism: Detection and Prevention of security attacks**

In this proposal we are going to define such requirements and ways in which we intend to solve the **goal of achieving secure VANETs**. More specifically we are going to address security in VANET **beaconing messages**, i.e., messages sent from a vehicle to it’s neighbours with information of location, speed, braking and other sensorial data that aid safe decision making.

Firstly, we are going to address the **identification** requirement. To uniquely identify a vehicle, we are going to use the **Vehicle Identification Number (VIN)** since it is emitted once the vehicle is produced and can never be changed unlike licence plates for example. Moreover, licence plates are designed so that they can be used to visually identify a vehicle making it a less appealing on the privacy standpoint.

We now need to ensure that the VIN is really coming from the vehicle it claims. For that purpose, we decided to use digital signatures, which not only solves the problem of **authentication** but also **message integrity** and **non-repudiation** of the sender.

1. **Replay Attack:**

* **Detection:** As explained in the previous section, to avoid the re-broadcasting of the same messages we decided to attach a timestamp to every message sent from vehicle nodes. RSU stores the first 20 messages from each vehicle for certain period, within that time if same vehicle tries to re-send the same messages, we then calculate the time stamp of the messages, if timestamp is above the given threshold, RSU will drop the messages sent by vehicle and detect it as Replay attack nodes. In Algorithm 1, SignedBeconDTO is the message sent by each vehicle node with signature and certificated signed by the CA to solve the problem of authentication.

Initially RSU stores the last 20 beacons sent from the vehicle in the Table, then which checks the freshness of the messages sent from the vehicle by calculating the timestamp of the vehicle with given threshold. If the calculated the timestamp is lesser than the current time, then RSU detect it as Relay attack node.

* **Prevention:** To prevent the Replay attack, RSU detects the Replay messages as explained in the Algorithm 1, and avoids sending the messages to the receiver vehicles nodes by dropping those type of the messages.

**A. Algorithm 1: Detection of Reply Attack**

**Input:** SignedCertificateDTO dto, RSU

**Output : Detect Reply node**

1. **Begin**
2. List<SignedBeaconDTO> *savedBeacons*: saving beacons for Reply Attack
3. double *N*: number of held beacons for Reply Attack
4. SignedCertificateDTO *dto*: signed data from the vehicle node
5. int *milliseconds* : Freshness Max Time
6. boolean *freshness*:

Save last 20 beacons for replay attacks

1. **if**  *(this.savedBeacons.size() > N )* **then**
2. this.savedBeacons.remove(0)
3. this.savedBeacons.add(dto);
4. **end if**

Checking for Reply of the beacons

1. **for** (SignedBeaconDTO dto : this.savedBeacons)
2. *freshness = dto.getTimeStamp.getTime() + milliseconds;*
3. **if** (*freshness >= System.currentTimeMillis()*) **then**
4. sender communication is fresh, No Reply Attack
5. **else**
6. Sender communication is not Fresh, Detected Reply Attack
7. **end for**
8. **End**
9. **Sybil Attack:**
   1. **Detection:** to achieve the detection of the Sybille nodes, we proposed the following solution. RSU verifies the every message’s sent by the vehicle node if its signed and certified by the CA as described in the Algorithm 4.

To better understand the proposed mechanism we present here its most features depicted in the 4 following algorithms.

* + Algorithm 2 details the procedure of validating the certificate and signature of the vehicle nodes checking with RSU and CA.
  + Algorithm 3 shows the procedure to estimate the speed of the vehicle by considering the velocity and time parameters and predict the position of the node.
  + Algorithm 4 details the procedure to check the estimated position is in the range of the network vicinity.
  + Algorithm 5 describes the whole algorithm that uses the three previous algorithms.

When vehicle sends the messages to another vehicle, RSU checks if the sender is authenticated by contacting the CA as described in the Algorithm 2. In the first step RSU checks if the message is replay. Then checks if the certificate is signed by the CA, verify the signature of each message sent from vehicle node.

After finding the sender is authenticated, RSU starts estimating the Position of the vehicle as explained in the Algorithm 3. Then RSU checks if the estimated position Is not in the dangerous range comparing with the threshold of the dangerous rang which is showed in the Algorithm 4. Finally, Algorithm 5. Describes the procedure of detecting the Sybil node. If the response from Algorithm 2, 3, 4 returns false RSU detect it as Sybil node.

1. **Prevention:** To prevent the Sybille attacker who sends the falsifies messages, RSU after detecting the sybil nodes, contact the CA to revoke the certificate of the attacker certificate, hence attacker will not be able to send the message’s again which explained in the Algorithm 6.

In Algorithm 6, RSU sends request to CA to revoke the certificate of the attacker after detecting the Sybil node described in the Algorithm 5. Then, CA verity’s the sender was authenticated and checks the request from RSU with threshold given. CA revokes the certificate and inform the RSU, then RSU stores the revoked certificate in its cache memory to check again if the same attacker tries to send the message. After that RSU informs the vehicle about the attacker as described in the Algorithm 6.

**B. Algorithm 2: RSU Checks if Sender is Authenticated for isRevoked Request**

**Input:** SignedCertificateDTO dto, Freshness Max Time (*Ft* ), RSU rsu

**Identify : sender is authenticated**

1. **Begin**

Verify Timestamp freshness

1. **If** (NOT dto.verifyFreshness(*Ft)*)
2. Senders communication is not fresh
3. return *false*;
4. **end if**

Verify if certificate was signed by CA

1. **If** (Not dto.verifyCertificate(this.rsu.getCACertificate()))
2. Invalid CA Signature on isRevoked request
3. return *false*;
4. **end if**

Verify if certificate has expired

1. **If** (Not dto.getSenderCertificate().checkValidity())
2. Sender's Certificate has expired
3. return *false*;
4. **end if**

verify if certificate is revoked

1. **If** (Not rsu.isCertInCache(dto.getSenderCertificate()))
2. Sender's Certificate is revoked
3. return *false*;
4. **end if**

Contact CA to verify if senders certificate is revoked

1. **If** (ca.isRevoked (new SignedCertificateDTO (dto.getSenderCertificate(), rsu.getCertificate()) )
2. Sender's Certificate is not revoked
3. return *false*;
4. **end if**

verify signature sent

1. **If** (NOT dto.verifySignature())
2. Invalid digital signature on isRevoked request
3. return *false*;
4. **end if**
5. return *true*;

Sender is authenticated

1. **End**

**C. Algorithm 3: Estimate the position**

1. **procedure** predictedNext(Vector2D velocity, double delta)
2. Vector2D *predicted*: estimated position
3. double *x* : x coordinate of the node
4. double *y* : y coordinate of the node
5. double *delta*: predicted value of timestamp
6. Vector2D *velocity*: velocity of the old beacon

Predict the next position of the old beacon

1. *predicted* = new Vector2D(x + (velocity.x \* delta), y + (velocity.y \* delta));
2. *return* *predicted;*
3. **end** **procedure**

**D. Algorithm 4: Check the Estimated position inRange**

1. **procedure** inRange( Vector2D newPosition, double range)
2. double *distance* : estimated distance
3. double *x* : x coordinate of the node
4. double *y* : y coordinate of the node
5. double *range*: threshold value of dangerous range
6. distance = Math.sqrt(Math.pow( x - newPosition.x, 2) + Math.pow( y - newPosition.y, 2));

check if the calculated distance is in range with

1. **if** (*distance* <= *range*) **then**
2. return *true*;
3. **else**
4. return *false*;
5. **end if**
6. **end** **procedure**

**E. Algorithm 5: Detection and prevention of Sybil Attack**

**Input:** SignedCertificateDTO dto, RSU

**Output : Detect and Prevent the Sybil node**

1. **Begin**
2. Vector2D *predictedDelta*: estimated TimeStamp
3. double *predictedPosition*: Estimated Position
4. double *range*: threshold value of dangerous range
5. double inRange: check if the new position is inRange
6. BeaconDTO oldBeacon: previously sent node
7. BeaconDTO newBeacon: newly sent node
8. SignedCertificateDTO certToRevoke: get the certificate of the data

Estimate the Timestamp and Position

1. *predictedDelta = (newBeacon.getTimestamp().getTime()*

*- oldBeacon.getTimestamp().getTime()) / 1000.0d;*

1. *predictedPosition = oldBeacon.getPosition()*

*.predictedNext(oldBeacon.getVelocity(), predictedDelta);*

Check if the predicted position is not in Dangerous Range

1. inRange = predictedPosition.inRange*(newBeacon.getPosition*(), *range* )
2. **If** (inRange) **then**
3. Estimated Position is in Range
4. response = RSU.tryRevoke(certToRevoke);
5. **if**(! response.getValue())
6. Reporting vehicle beaconing unplausible data! Sybil node detected
7. **else**
8. *return false*
9. **end if**
10. **end if**
11. **End**

**F. Algorithm 6: RSU tries to revoke the sender (attacker) certificate**

**Input:** SignedCertificateDTO dto, Freshness Max Time (*Ft* ), RSU rsu

**Output : Revokes the attacker certificate**

1. **Begin**

Verify that sender is trustworthy

1. **If** (NOT authenticateSender(dto))
2. Senders is not authenticated
3. return *false*;
4. **end if**

verify if certificate has expired

1. **If** (dto.getCertificate().checkValidity())
2. rsu.addCertificateToCache(dto.getCertificate())
3. Sender certificate has expired
4. return *false*;
5. **end if**

verify if certificate is in cache

1. **If** (rsu.isCertInCache(dto.getCertificate()))
2. Sender's Certificate is not in catch
3. return *false*;
4. **end if**

RSU contacts CA to revoke the attacker certificate

1. **If** (ca.tryRevoke(dto))

Adding attacker certificate to catch

1. rsu.addCertificateToCache(dto.getCertificate());
2. rsu.shareRevoked(dto);
3. return *false*;
4. **end if**

Inform other vehicle about revocation of attacker certificate

1. **procedure** informVehiclesOfRevocation(SignedCertificateDTO dto)
2. vehicle\_network.informVehiclesOfRevocation(dto, rsu.getPosition());
3. **end procedure**
4. **End**

**Simulation results and Analysis**

1. **Best routing protocol**

To continue further work on detecting and preventing the DDOS attack and Sybil attack using the Vehicular cloud computing (VCC), the best reliable routing protocol is required for the transmission in the real-world scenario. To achieve that, we have implemented system which generates the real-world scenario by selecting the specified area from the OpenStreetMap Framework. where number of vehicles, Trucks, Traffic Signals and pedestrians were selected. Then, we analysed the generated scenario in Urban simulator called SUMO.

After that, by using SUMO a Mobilty.tcl file was generated to analyse the communication between the nodes by using the four different routing protocol such as (Ad hoc On-Demand Distance Vector) AODV, Dynamic Source Routing (DSR), Destination-Sequenced Distance-Vector DSDV, Optimized Link State Routing Protocol (OLSR) with the Network Simulator NS3. The four-routing protocol was used for the transmission between the nodes and recorded the obtained results for each protocol which has the parameters such as Throughput, source node address, destination node address etc. Finally, we have done the graphical analysis on the obtained result over the simulation duration. For the given simulation duration, the DSR protocol was giving the high throughput and less network over head in VANET communication system as shown in the Fig.3. We are considering the DSR protocol for the further work on the Detecting and preventing the Security attacks in VANET by using VCC.

The performance parameters used for the evaluation of this work are: -

* + - ***Packet Loss: -***It is the ratio between the packets loss and the total packets sent by any node to the destination. Its value depends upon the congestion in the network due to which packets fail to reach their destination successfully [12][14].
    - ***Lifetime of network: -***Lifetime of a network is defined as time during which the vehicles of the network are able to route information successfully. If any number of nodes are out of energy or lose some functionality due to any reason, then the lifetime of the network ends.
    - ***Network Throughput: -***The percentage of data sent from the originator node to final node in specified time gives the value of network throughput. The high the throughput value, the maximum information is transmitted between source and destination.
    - ***Packet Delivery Ratio: -***The accurate delivery of the packets from the originator to the destination gives the value of packet delivery ratio. It is the ratio between the number of packets reached to the total number of packets [14].
    - ***Number of dead and alive nodes: -***It is the number of nodes which stop working are considered as dead nodes and the number of nodes which are disseminated the information in the network are under the alive nodes,

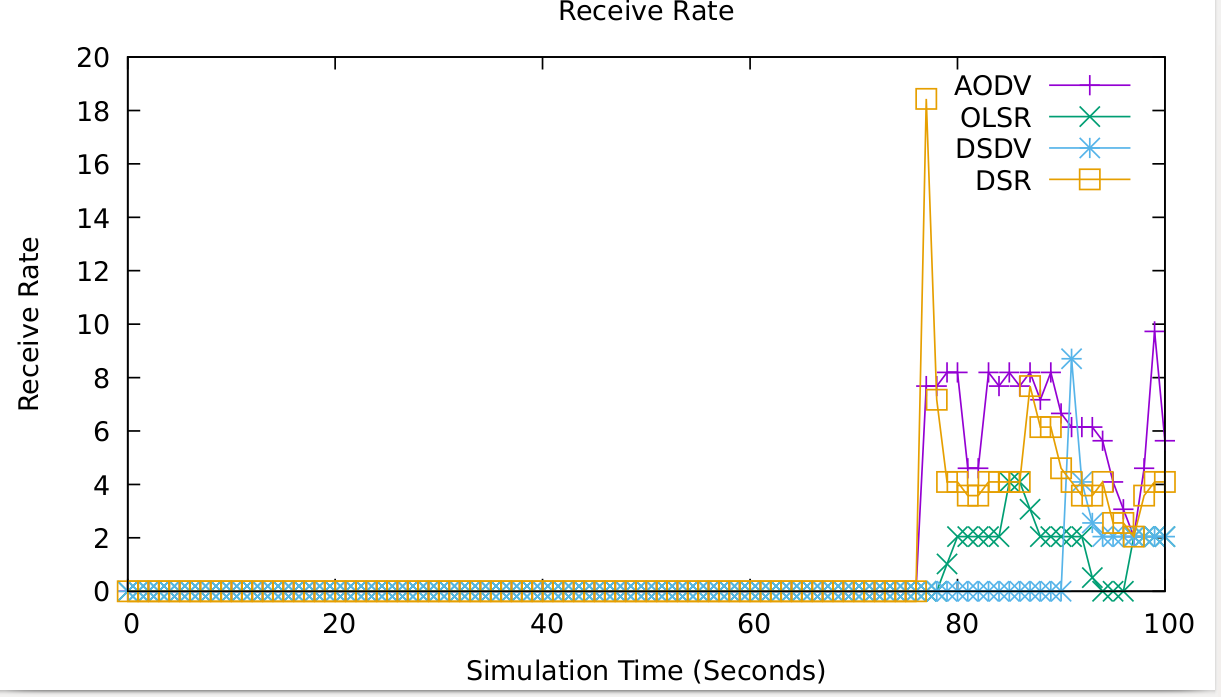


Figure 3. Found which Routing protocol is best for communication

**Table 3**

**Signing and Verification Delay of RSA and ECDSA**

|  |  |  |
| --- | --- | --- |
| Algorithm | Signing Delay (ms) | Verification Delay (ms) |
| RSA 2048-bit | 22.737 | 0.599 |
| ECDSA 224-bit | 0.800 | 3.000 |

**TABLE 4:**

**PERFORMANCE PARAMETERS TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of vehicles | Throughput of network | PDR | PLR | Lifetime of the Network |
| 5 | 250 | 58 | 300 | 41 |
| 8 | 300 | 59 | 190 | 39 |
| 10 | 350 | 62 | 152 | 38 |
| 12 | 360 | 68 | 130 | 37.5 |

**Conclusion and Future Enhancements**

This report highlights how the VANET environment can be improved by improving the driving experience, navigation services, road safety and other roadside services. Due to the characteristics of the VANET system and its architecture, VANET is vulnerable to many security attacks. There is a need to develop security solutions in the VANET environment. Many previous security efforts have applied the same old traditional security solutions without considering any special aspects of VANETs. We have studied the security challenges faced by VANETs and their architecture for our literature, which have helped us to come up with this solution to detect and prevent DDoS.

All recent studies in this field focus more on DDoS and Sybil attacks, such as how to detect and prevent them, from the security point of view of VANET. In addition, the report provides various security solutions for various attacks in VANETs using machine learning capable of detecting patterns in attacks. As part of our future work, we create an environment by using simulation tools that help us produce datasets that are greater than general and public datasets by using VCC (Vehicular Cloud Computing). By using this dataset, we can create a real-time context that lets us detect threats. We put forward to provide a solution for detecting and prevent DDoS attacks in VANET along with its prevention rate, detection rate and its own design.

**Table 1**

**Simulation Parameters and Its Values**

|  |  |
| --- | --- |
| Parameter | Value |
| Max Beacon Range | 500 m |
| Too Dangerous Range | 100 m |
| Max RSU Range | 5000 m |
| Beacon Interval | 2 s |
| Danger Reset Interval | 1 s |
| Check RSU range Interval | 2 s |
| Max Interval Vicinity In Cache | 60 s |
| Network Position Update Interval | 2 s |
| Freshness Max Time | 2 s |
| Max Revoke Score | 2 |
| Held Beacons for Replay Attacks | 20 |
| Attacker Position Range | 5000 |
| CA Digest | SHA-256 |
| The Height of Network | 1800 m |
| The Width of Network | 1800 m |
| Encryption Technique | Vehicle RSA |
| mac/phy | IEEE 802.11p |

**Table 2**

**Notations and Definition**

|  |  |
| --- | --- |
| Notation | Definition |
| CA | Certificate authority |
| RSU | Roadside unit |
| *Vi* | *i*-th vehicle |
| VIN | Vehicle Id number |
| *PX* | The public key of x |
| *PrvtY* | The private key of y |
| *M* | Beaconing message |
| *H(M)* | Digest of message M |
| SUMO | Simulation of urban mobility |
| VANET | Vehicular ad hoc network |
| DDOS | Denial of Service |
| VANET | Vehicular ad hoc network |
| PDR | Packet delivery ratio |
| PLR | Packet loss ratio |
|  |  |

**A. Algorithm 1: RSU Checks if Sender is Authenticated for isRevoked Request**

**Input:** SignedCertificateDTO dto, Freshness Max Time (*Ft* ), RSU rsu

**Identify : sender is authenticated**

1. **Begin**

Verify Timestamp freshness

1. **If** (NOT dto.verifyFreshness(*Ft)*)
2. Senders communication is not fresh
3. return *false*;
4. **end if**

Verify if certificate was signed by CA

1. **If** (Not dto.verifyCertificate(this.rsu.getCACertificate()))
2. Invalid CA Signature on isRevoked request
3. return *false*;
4. **end if**

Verify if certificate has expired

1. **If** (Not dto.getSenderCertificate().checkValidity())
2. Sender's Certificate has expired
3. return *false*;
4. **end if**

verify if certificate is revoked

1. **If** (Not rsu.isCertInCache(dto.getSenderCertificate()))
2. Sender's Certificate is revoked
3. return *false*;
4. **end if**

Contact CA to verify if senders certificate is revoked

1. **If** (ca.isRevoked (new SignedCertificateDTO (dto.getSenderCertificate(), rsu.getCertificate()) )
2. Sender's Certificate is not revoked
3. return *false*;
4. **end if**

verify signature sent

1. **If** (NOT dto.verifySignature())
2. Invalid digital signature on isRevoked request
3. return *false*;
4. **end if**
5. return *true*;

Sender is authenticated

1. **End**

**B. Algorithm 2: RSU tries to revoke the sender (attacker) certificate**

**Input:** SignedCertificateDTO dto, Freshness Max Time (*Ft* ), RSU rsu

**Output : Revokes the attacker certificate**

1. **Begin**

Verify that sender is trustworthy

1. **If** (NOT authenticateSender(dto))
2. Senders is not authenticated
3. return *false*;
4. **end if**

verify if certificate has expired

1. **If** (dto.getCertificate().checkValidity())
2. rsu.addCertificateToCache(dto.getCertificate())
3. Sender certificate has expired
4. return *false*;
5. **end if**

verify if certificate is in cache

1. **If** (rsu.isCertInCache(dto.getCertificate()))
2. Sender's Certificate is not in catch
3. return *false*;
4. **end if**

RSU contacts CA to revoke the attacker certificate

1. **If** (ca.tryRevoke(dto))

Adding attacker certificate to catch

1. rsu.addCertificateToCache(dto.getCertificate());
2. rsu.shareRevoked(dto);
3. return *false*;
4. **end if**

Inform other vehicle about revocation of attacker certificate

1. **procedure** informVehiclesOfRevocation(SignedCertificateDTO dto)
2. vehicle\_network.informVehiclesOfRevocation(dto, rsu.getPosition());
3. **end procedure**
4. **End**

**C. Algorithm 3: Check the Estimated position inRange**

1. **procedure** inRange( Vector2D newPosition, double range)
2. double *distance* : estimated distance
3. double *x* : x coordinate of the node
4. double *y* : y coordinate of the node
5. double *range*: threshold value of dangerous range
6. distance = Math.sqrt(Math.pow( x - newPosition.x, 2) + Math.pow( y - newPosition.y, 2));

check if the calculated distance is in range with

1. **if** (*distance* <= *range*) **then**
2. return *true*;
3. **else**
4. return *false*;
5. **end if**
6. **end** **procedure**

**D. Algorithm 4: Estimate the position**

1. **procedure** predictedNext(Vector2D velocity, double delta)
2. Vector2D *predicted*: estimated position
3. double *x* : x coordinate of the node
4. double *y* : y coordinate of the node
5. double *delta*: predicted value of timestamp
6. Vector2D *velocity*: velocity of the old beacon

Predict the next position of the old beacon

1. *predicted* = new Vector2D(x + (velocity.x \* delta), y + (velocity.y \* delta));
2. *return* *predicted;*
3. **end** **procedure**

**E. Algorithm 5: Detection and prevention of Sybil Attack**

**Input:** SignedCertificateDTO dto, RSU

**Output : Detect and Prevent the Sybil node**

1. **Begin**
2. Vector2D *predictedDelta*: estimated TimeStamp
3. double *predictedPosition*: Estimated Position
4. double *range*: threshold value of dangerous range
5. double inRange: check if the new position is inRange
6. BeaconDTO oldBeacon: previously sent node
7. BeaconDTO newBeacon: newly sent node
8. SignedCertificateDTO certToRevoke: get the certificate of the data

Estimate the Timestamp and Position

1. *predictedDelta = (newBeacon.getTimestamp().getTime()*

*- oldBeacon.getTimestamp().getTime()) / 1000.0d;*

1. *predictedPosition = oldBeacon.getPosition()*

*.predictedNext(oldBeacon.getVelocity(), predictedDelta);*

Check if the predicted position is not in Dangerous Range

1. inRange = predictedPosition.inRange*(newBeacon.getPosition*(), *range* )
2. **If** (inRange) **then**
3. Estimated Position is in Range
4. response = RSU.tryRevoke(certToRevoke);
5. **if**(! response.getValue())
6. Reporting vehicle beaconing unplausible data! Sybil node detected
7. **else**
8. *return false*
9. **end if**
10. **end if**
11. **End**

**F. Algorithm 6: Detection of Reply Attack**

**Input:** SignedCertificateDTO dto, RSU

**Output : Detect Reply node**

1. **Begin**
2. List<SignedBeaconDTO> *savedBeacons*: saving beacons for Reply Attack
3. double *N*: number of held beacons for Reply Attack
4. SignedCertificateDTO *dto*: signed data from the vehicle node
5. int *milliseconds* : Freshness Max Time
6. boolean *freshness*:

Save last 20 beacons for replay attacks

1. **if**  *(this.savedBeacons.size() > N )* **then**
2. this.savedBeacons.remove(0)
3. this.savedBeacons.add(dto);
4. **end if**

Checking for Reply of the beacons

1. **for** (SignedBeaconDTO dto : this.savedBeacons)
2. *freshness = dto.getTimeStamp.getTime() + milliseconds;*
3. **if** (*freshness >= System.currentTimeMillis()*) **then**
4. sender communication is fresh, No Reply Attack
5. **else**
6. Sender communication is not Fresh, Detected Reply Attack
7. **end for**
8. **End**

References

[1] “Transportation Forecast: Light Duty Vehicles,” [Online]. Available: http://www.navigantresearch.com/research/transportation-forecast-light-duty-vehicles. [Accessed 6-Jan-2019].

[2] WHO, “Road traffic injuries,” [Online]. Available: http://www.who.int/

mediacentre/factsheets/fs358/en/. [Accessed: 6-Jan-2019]

[3] L. Liu, S. Nie, and Y. Du, ”Free-Fall: Hacking Tesla from Wireless to CAN Bus,” In Proceedings of the Black Hat USA 2017, Las Vegas, NV, USA, 2017.

[4] C. Miller, and C. Valasek, “Adventures in Automotive Networks and Control Units,” [Online]. Available: http://illmatics.com/car\_hacking.pdf. [Accessed: 6-Jan-2019].

[5] R. Currie, “Hacking the CAN Bus: Basic Manipulation of a Modern Automobile Through CAN Bus Reverse Engineering,” STI Graduate Student Research, Maryland: SANS Technology Institute, 2017.

[6] F. Li, and Y. Wang, “Routing in vehicular ad hoc networks: A survey,” In IEEE Vehicular Technology Magazine, vol. 2, 2007, pp. 12–22.

[7] European Telecommunications Standards Institute (ETSI), “Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band,” EN 302 663 V1.2.1, Sophia-Antipolis: ETSI, 2013.

[8] Vehicular Technology Society (VTS), Intelligent Transportation Systems Committee (ITSC), Institute of Electrical and Electronics Engineers (IEEE), “1609.0-2013 IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture,” New York: IEEE, 2014.

[9] Transport Research and Innovation Monitoring and Information System (TRIMIS), “SEcure VEhicle COMmunication,” [Online]. Available: https://trimis.ec.europa.eu/project/secure-vehicle-communication. [Accessed: 6-Jan-2019].

[10] National Highway Traffic Safety Administration (NHTSA), “Vehicle Safety Communications Project, Final Report,” DOT HS 810 591, Washington: NHTSA, 2006.

[11] “Preparing Secure Vehicle-to-X Communication Systems,” [Online]. Available: https://www.preserve-project.eu/. [Accessed: 30-Mar-2019].

[12] B. Fernandes, J. Rufino, M. Alam, and J. Fereira, “Implementation and Analysis of IEEE and ETSI Security Standards for Vehicular Communications,” in Mobile Networks and Applications, vol. 23 Berlin, Heidelberg: Springer-Verlag, 2018, pp. 469–478.

[13] W. R. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. A. F. Loureiro, “Malicious node detection in wireless sensor networks,” In Proc. 18th International Parallel and Distributed Processing Symposium 2004, Santa Fe, NM, USA, 2004.

[14] B. Xiao, B. Yu, and C. Gao, “Detection and localization of sybil nodes in VANETs,” In Proc. 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks (DIWANS '06), New York, NY, USA, 2006, pp. 1–8.

[15] S. M. Bouassida, G. Guette, and M. Shawky, “Sybil Nodes Detection Based on Received Signal Strength Variations within VANET,” in I. J. Network Security, vol. 9, 2009, pp. 22–33.