Securing Hybrid SDN-based Geographic Routing Protocol using a Distributed Trust Model

Lylia Alouache, Mohamed Maachaoui, Rachid Chelouah

ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 5, No. 2, 567-577 (2020)
www.astesj.com
Special Issue on Multidisciplinary Sciences and Engineering

ASTES Journal
ISSN: 2415-6698

Securing Hybrid SDN-based Geographic Routing Protocol using a Distributed Trust Model

Lylia Alouache[*,1], Mohamed Maachaoui[2], Rachid Chelouah[1]

[1] *ETIS CNRS ENSEA UMR 8051. Computer science, CY Cergy Paris University, 95000, France*

[2] *Quartz Laboratory. Computer science, CY Cergy Paris University, 95000, France*

ARTICLE INFO

Article history:
Received: 15 January, 2020
Accepted: 10 March, 2020
Online: 12 April, 2020

Keywords:
IoV
Routing protocols
SDN
QoS
Security
Availability
Reliability
Integrity
Trust

ABSTRACT

In this paper, the vulnerabilities and the security attacks against vehicular networks, SDN architecture and some security solutions for SDVN are studied. Secondly, a complete and improved version of the secure HSDN-GRA routing protocol based on a distributed trust model is proposed where public keys of the vehicles are managed in a distributed way. Besides, in this approach, a weight is assigned to every vehicle, which is calculated from its freeload and its trust values per those of the neighbors. The trust value is deduced from the historical interactions stored in a log of communication errors. To measure the trust value, three unreliable behaviors occurred in a time interval are considered. The vehicle with the role of cluster head hosts a list of each vehicles misbehaviors forming a log of communication errors. A vehicle will be chosen as the next relay according to its weight. As a positive result of this complete proposed approach, the following security requirements are achieved: the vehicle's authentication and the data integrity are guaranteed by a signature mechanism, whereas an encrypted function is used to ensure the confidentiality of the exchanged data. The goal is to protect the routing process against malicious and unstable nodes. Finally, the implementation details and simulation analysis are given, and a comparative study between the secure and the insecure HSDN-GRA is presented in the presence of a percentage of malicious nodes in the network. The perspective of managing non-collaborative vehicles is briefly introduced as future work.

1 Introduction

The Vehicular Adhoc Networks (VANET) suffers from many lacks such as delays of communication because of the multi hop scenarios, frequent links failures because of the nodes mobility, and also a low security due to the several attacks and intrusions against the wireless and the heterogeneous network.
In our previous works [1][2], the SDN paradigm has been identified as a suitable approach for dealing with: the dynamic and large scale environment, the robustness of communication, the heterogeneity of the network concerning applications and the communication technologies, the routing strategies and also the security issues [3].

However, find the best way to implement the controller to suit to vehicular networks constraints is unresolved, especially because of the SDVN vulnerabilities. The security and the routing features are controlled by the SDN Controller. Whereas, the data plane applies according to the controller rules, since it is devoid of intelligence. As a result, the location choice of the controller is critical, as robust and secure communications depends on the robustness, the reliability and the availability of this controller.

This paper represents an extended version of our previous work [1] untitled "Securing Southbound Interface of HSDN-GRA Vehicular Routing Protocol using a Distributed Trust" that is published in 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC). We described an IoV communication use case based on the SDN architecture, to root IoV's data in a secure and robust manner from a source to a destination. The proposition was a preliminary approach without simulation analysis.

In this extended paper, we firstly discuss the security vulnerabilities of vehicular networks, SDN paradigm and SDVN networks. Secondly, we propose to secure the HSDN-GRA routing protocol by using a an encrypted function and trust model [1]. In this trust model, each new vehicle broadcast its public key to its neighbors,

[*]CY Cergy Paris University, Avenue du Parc 95000 Cergy Grance, +33769265726 & lae@eisti.eu

all the keys are managed in a distributed way. Besides, the described approach aims to assign a weight ω to each vehicle. ω is deduced from the free load value as well as the trust value of each vehicle per those of their neighbors.

To deduce the trust value of a vehicle, this approach uses the error log of HSDN-GRA protocol [4], as the historical misbehaviors of a vehicle gives its trust rate. three types of unreliable behaviors that a vehicle had in a period of time are considered. The list of each vehicle's misbehavior is stored in an error log and embedded on a specific cluster head vehicle. A vehicle can be the next relay according to its weight walue.

the advantage of the proposed solution is the improvement of security in the communication process, indeed, vehicle authentication and data integrity are obtained using a signature mechanism, while the confidentiality of the data exchanged is provided by an encryption function. the main purpose of secure HSDN-GRA is to protect the routing process from malicious and unstable vehicles.

The present paper is organized as follows: Section 2 studies the security attacks against vehicular networks and SDN architecture and some security solutions for SDVN. Section 3 describes our Secure HSDN-GRA approach. The implementation details and simulation analysis is presented in Section 4. Finally, Section 5 concludes the main contributions and summarizes the perspective of an incentive scheme in order to manage the non-collaborative vehicles.

2 Background and Related Work

In the survey [2] realized by Alouache et al., the SDN architecture is described as an alternative to secure the network and to ensure the data integrity. In fact, a global overview of the system is achievable thank to the SDN controller. It collects information about the entire network traffic instead of exchanging a large amounts of information. As a result, it provides a better and reliable security mechanism than those applied to the traditional network [5].

2.1 Security Attacks in Vehicular Networks

Vehicular networks suffer from several attacks which compromise the main security requirements. [6].

In Table 1 non exhaustive list of these attacks are given.

2.2 Security Attacks in SDN Architecture

The SDN paradigm also has its own vulnerabilities, they are principally related to the characteristics, the location and the crucial role of the controller, as well as the control data flow exchanged between the control plane and the data plane [7].

Table 2 exposes a non exhaustive list of attacks related to the SDN paradigm.

2.3 Security Solutions for SDVN

Despite the vulnerabilities exposed in Table 1 and those mentioned in Table 2, and since SDN exceeds the limits of traditional mechanisms, it can be beneficial for IoV security deficiencies.

Table 1: Vehicular Network Attacks vs Security Requirements

Attacks in Vehicular Networks	Authentication	Availability	Confidentiality	Integrity	Non-repudiation
Denial of Service (DoS)		x			
Distributed DoS		x			
Jamming		x		x	
Malware attack		x	x	x	
GPS Spoofing	x	x			
Hijacking of session	x				
Position faking	x				
Illusion attack	x	x			
Bogus information attack	x	x			
GPS Spoofing	x	x			
Snooping			x		
Identity reveling			x		
Location tracking			x		
Brute force	x		x		
Eavesdropping			x		
Sink Hole attack		x	x	x	
Black Hole attack		x		x	
Masquerade attack	x			x	
Message tempering				x	x
Message suppression				x	x
Message reply	x				x
Repudiation		x			x

In fact, the SDN controller labels and isolates suspect flows and their sources, consequently, the data plane will not process the

packets coming from these flows.
Besides, the IoV network is heterogeneous and uses a set of communication technologies such: WAVE, DSRC, LTE, etc. having their own security policies, therefore, due to the SDN controller and its given global overview, these security policies can be deployed without conflict, which reinforces the security of the whole IoV system.

In addition, the centralization and the abstraction provided by the controller give the administrators the possibility to update any security policy based on observed unreliable behaviors. Table 3 describes how SDN can be a solution for security issues.

Table 2: SDN Attacks vs Security Requirements

SDN attacks in SDN Architecture	Authentication	Availability	Confidentiality	Integrity	Non-repudiation
DoS on the Data plane		x			
Distributed DoS on the Control plane		x			
Controller identity spoofing	x	x	x	x	x
Flow based forwarding attack				x	x
Flow table alteration			x	x	

Table 3: Security in IoV with SDN Architecture

Action	details
Intrusion detection	The SDN controller manages the nodes authentication and detects malicious intrusions.
Attacks identification	The SDN controller monitors the exchanged flows emanating from all the nodes to detect any setting alteration.
Self recovery mechanism	The SDN controller hosts rules to automatic recovery against any attack.

Recently, some research papers deal with both routing and security issues in vehicular networks by adopting the SDN paradigm.

The Improvised Trust based Ad-hoc On-demand Distance Vector routing (I-TAODV) is proposed by Vasudes et al. in [8], it aims to secure routing in a multi hop scenario.
Based on a metric called trust value, the authors propose an algorithm to identify the trusted vehicles, and another one to identify the malicious ones.
This protocol is built on the SDN paradigm, where the control plane monitors forwarding, reversing, trust of forwarding Vehicle, trust of reverse vehicle, path trust and network performances.

Zhang et al. elaborate The Software-Defined Trust based Ad hoc On-demand Distance Vector routing (SD-TAODV) in [9]. The process of route discovery, route maintenance, the reverse and forwarding paths are chosen by the control plane.
They also use a trust management mechanism which is represented by a bi-objective function, it tries to optimize two objectives: the node trust and the path trust calculated respectively by Trust Node Calculation Process and Path Trust Calculation Process They are used to enhance the Route Discovery Process of TAODV.

3 Securing the HSDN-GRA

The HSDN-GRA protocol presented in the research [4] exploits a log of communication errors where three types of vehicle's misbehavior are stored: link failures, random reception of Beacon messages, and non-acknowledgement of previous messages.
To estimate the trust value, the vehicle communication history is observed and tracked from the log of communication error. In this extended paper, we use the trust value of each vehicle with comparison to the trust values of all its neighbors.

The trust is described as the expectation and the belief that a vehicle has about other vehicles concerning future behaviors.
Its estimation is based on 1) experiences and evidences collected in the past either directly or indirectly, and 2) the knowledge about the vehicles nature, and/or on recommendations from trusted entities [10]-[12].

This trust value in addition to the free load value of each vehicle are aggregated to assign a weight ω to each vehicle. This weight will be used to elect the best relay at each step of the routing process.

The routing rules are commonly governed by two types of controllers in HSDN-GRA: the semi-centralized controller represented by the cluster head, and the distributed controllers represented by the cluster members.

Figure 1 shows an IoV communication scenario using the HSDN-GRA routing protocol based on a semi-centralized SDN architecture.
A vehicle tries to access to an Internet service is requested by a vehicle, but the request fails because the closest infrastructure is down. So it switches to the vehicles on the road for routing its request.

The network is divided on clusters and Cluster heads are elected. Each Cluster Head contains a part of the control plane (the error log). The second part is distributed on the rest of the cluster members.

until reaching the destination, the request transits through vehicles with a hop by hop approach. Each vehicle selects its next relay with respect to the controller policy i.e,: The one who has the best relative trust value and the best relative free load value.

Hypothesis:

- In this extended paper, we opt for the highway scenario, this choice is guided by the stability of vehicle's speed.
- At first, the free load and the trust values of vehicles are equal and optimal.
- Malicious-Vehicles $\ll Reliable - Vehicles$.

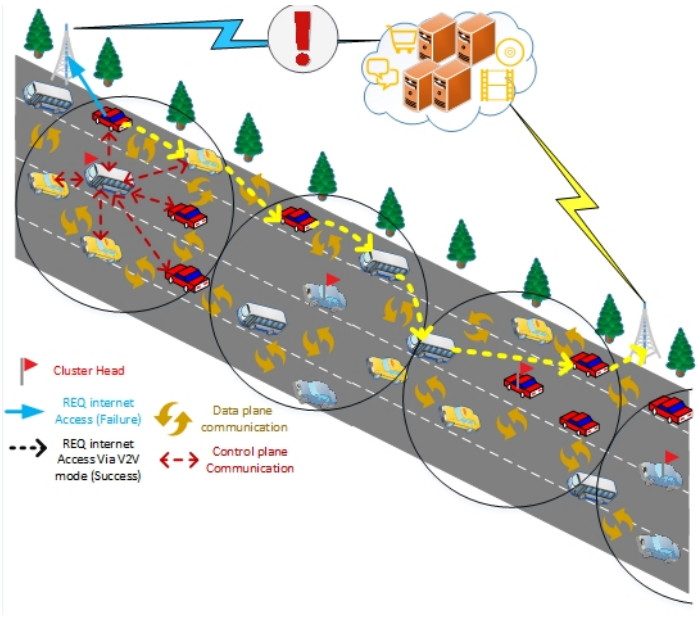

Figure 1: A Use Case of IoV Communication with a Hybrid SDN Architecture

The following subsections describe the securing HSDN-GRA steps.

3.1 Public key Distribution

- all the vehicles exchange a define Beacon to form a a one hop neighboring.
- Each vehicle sends its Public Key *Pkey* to the one hop neighbors, and receives back their *Pkey*.
- At this initialization step, because of the absence of cluster head, only the distributed controllers embedded on each vehicle are operational.
- Each vehicle has a weight ω, it exploits the load capacity l_i deduced from the received Beacons, and the trust value $Trust_i$ calculated from the log of communication errors according to Formula 1. The weight ω of each vehicle is calculated from Formula 2 by all the neighbors, and the true value will be confirmed by the majority.
- The election of the Cluster Head is realized by a distributed consensus after the initialization period of time T has passed, the one with the largest weight ω will be the Cluster Head.

$$Trust = (\frac{1}{Popularity}) \quad (1)$$

$$\omega(v)_i = (\frac{l_i}{\sum_{j=1}^{m} l_j} + \frac{Trust_i}{\sum_{j=1}^{m} Trust_j}) \quad (2)$$

where m is the number of neighbors.

-To monitor the communication errors of each neighbor, in the beginning, before electing cluster heads, logs are distributed within all the vehicles. The trust rate is estimated by exploiting all the distributed logs. So, initially each vehicle:
-Monitors its neighbors and has it own log
-Calculates the weight ω of each neighbor
-Then, each vehicle broadcasts in a map the $Pkey_i$ of each neighbor and their associated weight ω values:

$$NeighborPkeyMap = \{Pkey_i, \omega_i)\}, \text{With } i \in \{1, m\}$$

The algorithm 1 describes how the public key distribution as well as the weight of each vehicle is managed.

Algorithm 1: Public Key Distribution (PKD)

Let $Pkey_i$ be a the public key of vehicle V_i;
Let $NeighborsPkeyMap < Pkey, \omega >$ be a the map in which a vehicle V store the public key and the weight of its neighbors;
input : Vehicle V
output : $NeighborsPkeyMap$
Initialization: $NeighborsPkeyMap = \emptyset$; ω_{Vi}=*null*; $Pkey_i$=*null*;
for *each vehicle* V_i **do**
  $Broadcast(Beacon_{Vi})$;
  $Broadcast(Pkey_i)$;
end
for *each vehicle* V **do**
  $Receive(Beacon_{Vi})$;
  $Receive(Pkey_{Vi})$;
  $Trust_i = (\frac{1}{Popularity_i})$;
  $\omega(V)_i = (\frac{l_i}{\sum_{j=1}^{m} l_j} + \frac{Trust_i}{\sum_{j=1}^{m} Trust_j})$;
  $Insert(< Pkey_{Vi}, \omega(Vi) >, NeighborsPkeyMap)$;
end
$Broadcast(NeighborsPkeyMap)$;

3.2 Cluster Head Election

-Using all the received $NeighborsPkeyMap_i$, a vehicle V confirms the reliable weight value ω_i of each neighbor V_i and the association $(V_i/Pkey_i)$. i.e, a coherent contain of the $NeighborsPkeyMap$ is deduced by the majority who give the same values.
-On each vehicle a consensus occur to chose as Cluster Head the vehicle with the largest weight ω.
-The variable "IsCh" will be $True$ in the Beacon for the Cluster Head, while the rest of vehicles will keep the value $False$.
-The elected Cluster Head, representing the Semi-Centralized SDN controller, will now hosts the log of communication errors related

to it cluster.
The algorithm 2 details the Cluster Head election.

Algorithm 2: Secure Cluster Head Election (SCEA)

Let $NeighborsPkeyMap_i < Pkey, \omega >$ be a the maps received by V from each neighbor;
input : Vehicle V
output : $ClusterHead$
$Initialization$: $Max\omega = \omega$;$PkeyMax=Pkey$; $ClusterHead = null$;
for *each vehicle V* **do**
  $Receive(NeighborsPkeyMap_i)$;
  $FixTrueValueof(NeighborsPkeyMap)$ By the Majority;
  if $(NeighborsPkeyMap.\omega_i > Max\omega)$ **then**
    $Max\omega = NeighborsPkeyMap.\omega_i$;
    $PkeyMax = NeighborsPkeyMap.Pkey_i$;
  else
    if $(NeighborsPkeyMap.\omega_i == Max\omega)$ *and* $(NeighborsPkeyMap.Pkey_i < PkeyMax)$ **then**
      $Max\omega = NeighborsPkeyMap.\omega_i$;
      $PkeyMax = NeighborsPkeyMap.Pkey_i$;
    end
    $ClusterHead = V_{Max\omega}$;
    if $(ClusterHead = V)$ **then**
      $Beacon_V.isCH = True$;
      $Send(NewBeacon)$;
      $Receive(ErrorLog)$ from previous Cluster Head or from the neighbors at initialization;
      $Filter(ErrorLog)$;
    else
      $Reset(NeighborsPkeyMap_i)$;
    end
  end
end

3.3 Secure Control Plane of HSDN-GRA

In this section, the Secure HSDN-GRA is detailed.
Once the first election of the Cluster Head occurred, the vehicles signs all their Beacons. The objective is to guarantee the authentication.Figure 2 expose the Beacon message structure.

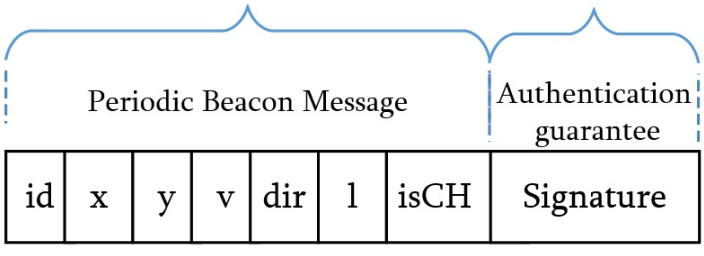

Figure 2: Beacon Message structure in Secure HSDN-GRA

The structure of the messages exchanged via the southbound interface are represented in Figure 3.

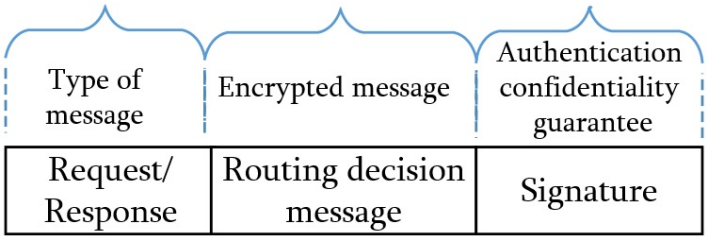

Figure 3: Structure of Southbound Messages in Secure HSDN-GRA

As the confidentiality of the control decisions, especially about the next hops, need to be preserve, we also propose to secure the southbound interface used in HSDN-GRA routing protocol.

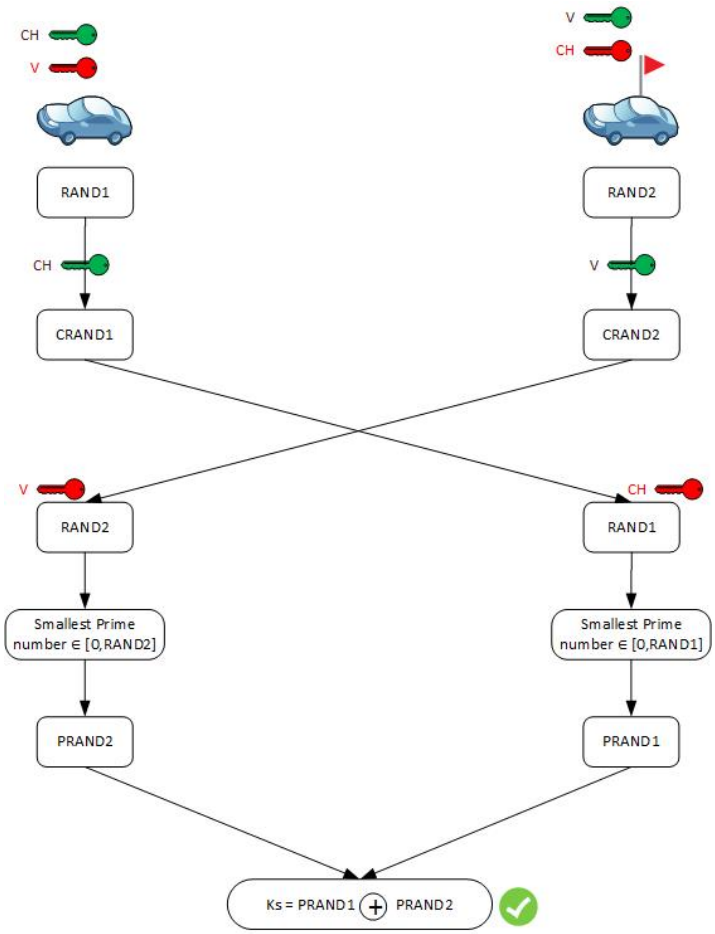

Figure 4: Secret Key creation between the Cluster Head and a vehicle V

To encrypt the data flow of the southbound interface, we are inspired from the symmetric encryption algorithm AES [13].A secret key K_s is exchanged between each vehicle and its Cluster Head. The public keys $Pkey$ already shared are exploited to deduce K_s.

A vehicle V and its Cluster Head exchange two random numbers $RAND1$ and $RAND2$ encrypted respectively with $Pkey_{CH}$ and $Pkey_V$.

Then, the Cluster Head decrypt with its private key $RAND1$

and choose the smallest prime number $PRAND1$ in the interval [0,RAND1]. The vehicle V decrypt also with its private key $RAND2$ and choose the smallest prime number $PRAND2$ in the interval [0,RAND2].

Finally, the secret key K_s is given using the formula 3. The symbol $\bigoplus$ representing the Exclusive Or logical operation.

$$Ks = (PRAND1 \bigoplus PRAND2) \quad (3)$$

Figure 4 illustrate how the secret key K_s is formed by the Cluster Head and a vehicle V in order to encrypt the southbound interface communications.

Figure 5 illustrate communications inside a cluster of Secure HSDN-GRA.

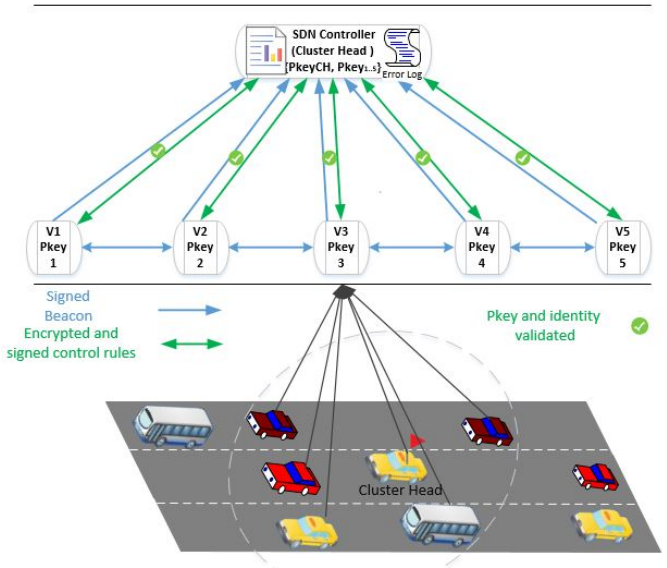

Figure 5: Communication inside a Cluster of Secure HSDN-GRA

3.4 Secure Incoming Vehicles in a Cluster

Three steps are required to integrate a new vehicle in a cluster:

- The new vehicle V_n broadcast its Beacon and its $Pkey$ to announce its self.
- All the neighbors inside the Cluster reply to V_n with the identifier and the $Pkey$ of the Cluster Head, while sending the triplet (V_n,$Pkey_n$, ω_n) to the Cluster Head.
- The Cluster Head analyzes all the received triplets (V_n,$Pkey_n$,ω_n) in order to confirm the identity of V_n. It deduces the percentage of veracity of this triplet, from the number of vehicles who affirm it. After that, a Challenge-Response mechanism [14] is exploited associate definitively V_n to $Pkey_n$.

These steps are respectively illustrated by Figure 6, Figure 7 and Figure 8.

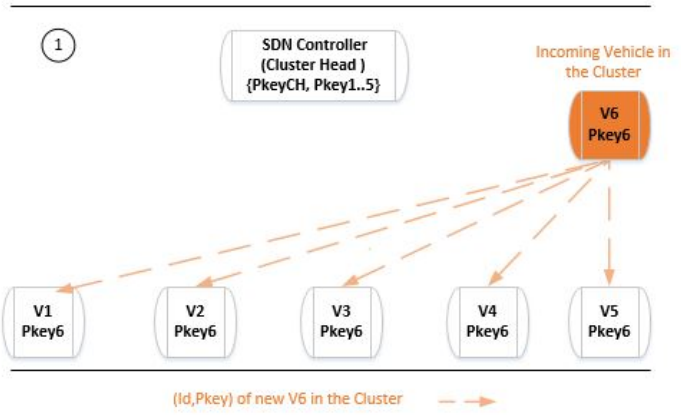

Figure 6: Incoming Vehicle in the Cluster: Phase 1

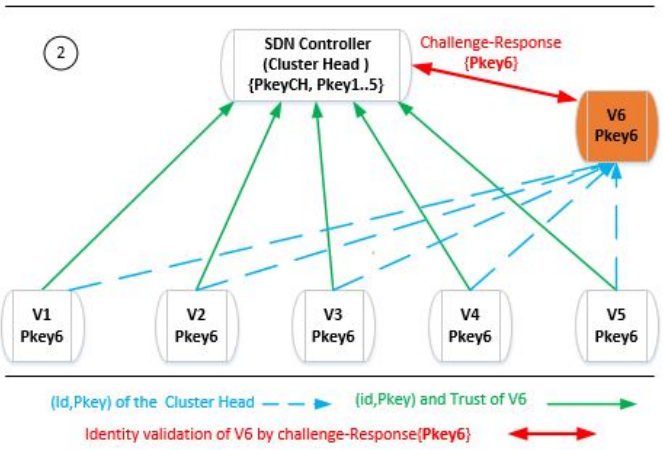

Figure 7: Incoming Vehicle in the Cluster: Phase 2

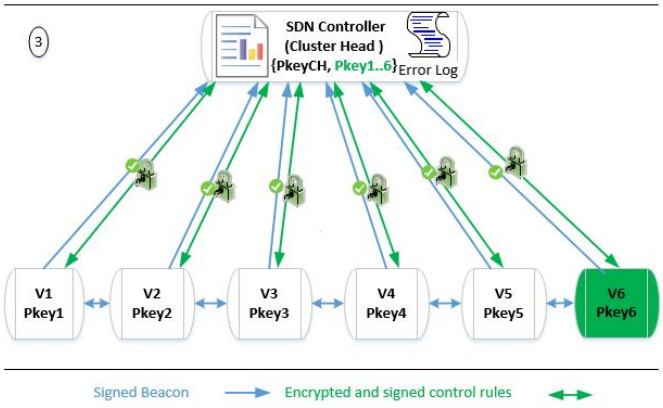

Figure 8: Incoming Vehicle in the Cluster: Phase 3

3.5 Secure Outgoing vehicle from a Cluster

This section shows the behaviour of secure HSDN-GRA in case of outgoing vehicle. Two cases are identified:

The first case deals with a cluster member which formulates a Request-To-Leave message the steps are:

- The Cluster member sends a Request-to-Leave message to the Cluster Head
- A Challenge-Response mechanism [14] is launched to confirm that the request is formulated by the pretending cluster member.
- After the authentication, an ACK message is sent by the Cluster Head to the leaving vehicle as an acknowledgement of its request. Meanwhile, the Cluster Head also broadcasts a $Revoke(V, Pkey)$ message inside the cluster in order to delete the keys of the outgoing vehicle.
- ACK messages confirming the revocation of this vehicle are sent to the Cluster Head by all the cluster members.

Algorithm 3: Secure Outgoing Vehicle From a Cluster (SOC)

Let τ be the Beacon interval and ToF be the maximum delay of receiving Beacons;
input : Vehicle V
output : $Cluster$
$Initialization$: $ToF = 5\tau$;
for *each Cluster* **do**
  if *(V wants to leave a Cluster)* **then**
    $Send(Request - To - Leave)$ to the Cluster Head);
    if *($Challenge - Response(ClusterHead, V)$ is verified by the Cluster Head)* **then**
      $Send(ACK)$ by the Cluster Head to V;
      $Broadcast(Revoke(V, Pkey))$ to the cluster members;
    end
  else
    if $(Receive(Beacon_V) \notin [\tau,5\tau])$ **then**
      $Broadcast(Revoke(V, Pkey))$ to the cluster members;
    else
      $Reset(ToF)$;
    end
  end
end

The second case concerns the revocation of the vehicles that don't respect the periodicity of Beacons, as they are considered as unreliable.
In this paper, at every time-out τ, vehicles are supposed to send and receive back Beacons. Besides, a Timer-of-Refresh ToF, initialized to 5τ, is assigned to each Cluster member sending a Beacon. The Cluster Head launches and decrements ToF when it receives a new Beacon message from a Cluster member.

If the Cluster Head receives t Beacon in the time interval $[\tau,5\tau]$, the ToF value is refreshed and reset to its initial value. Otherwise, the Cluster member is considered as unreliable and need to be excluded from the cluster according to the following steps:

- If the ToF has passed, the Cluster Head broadcasts a $Revoke(V, Pkey)$ message within its cluster.
- ACK messages confirming the revocation of this vehicle are sent by all the Cluster members.

The two cases are combined in the algorithm 3.

Figure 9 shows a Cluster member outgoing scenario.

It encompasses the two cases. The black circle details the Request-to-Leave scenario, while the rest explains the revocation procedure.

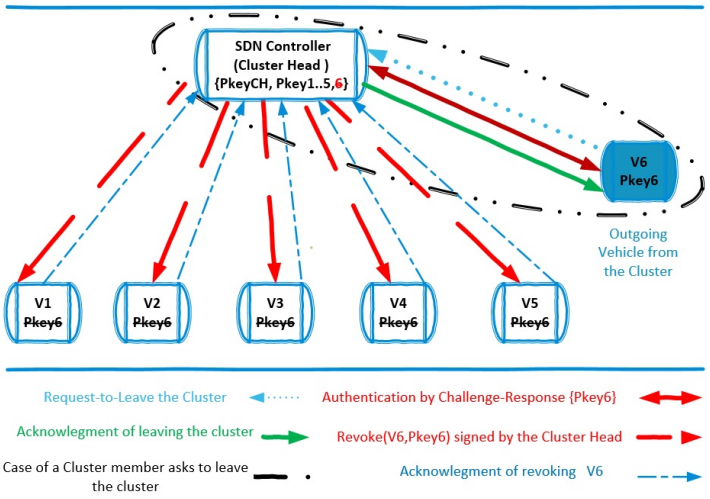

Figure 9: Outgoing Cluster Member Management

4 Simulation Analysis

In this simulation, we simulate a scenario of the Secure HSDN-GRA routing packets in IoV.
We develop a simulation with traffic condition similar to the situation in Figure 1.

Table 4: NS2 Simulation Parameters

Parameters	Specification
Simulation time	300s
Simulation area	1000m X 1000m
Number of nodes	[20-300]
Speed	[50-140] km/h
Propagation model	Two Ray Ground
Medium capacity	6 Mbps
Transmission range	310 m
Transport layer	UDP

4.1 NS2 Implementation

The simulation is done under a Ubuntu 10.04 Linux machine where we have installed the new version of network simulator NS2.34[15] and VanetMobiSim[16].
The simulation parameters are summarized in Table 4.

Some simulation functions in C++ are summarized in listing 1 (cf. Appendix 5).

4.2 Experimental Results

The goal of this evaluation consist of the study of attack resilience and routing performances of the secure HSDN-GRA protocol against the insecure HSDN-GRA protocol.

Percentage rate of data delivery and average end to end delay are chosen as indicators of routing performances.

The comparative study is done for different combinations of metrics like: the presence of malicious nodes, the clusters density and the vehicles speed.

the routing decisions of HSDN-GRA protocol are guided by the one-hop neighbors, besides, the secure HSDN-GRA protocol aims to detect and avoid malicious nodes. So, the routing performances will be positively affected by the improvement of routing behavior in presence of malicious nodes

4.2.1 Effect of Malicious Nodes

In this section, we observe in Figure 10 that the Secure HSDN-GRA presents better packet delivery ratio than the baseline HSDN-GRA once an increased number of malicious nodes are introduced in the simulation.

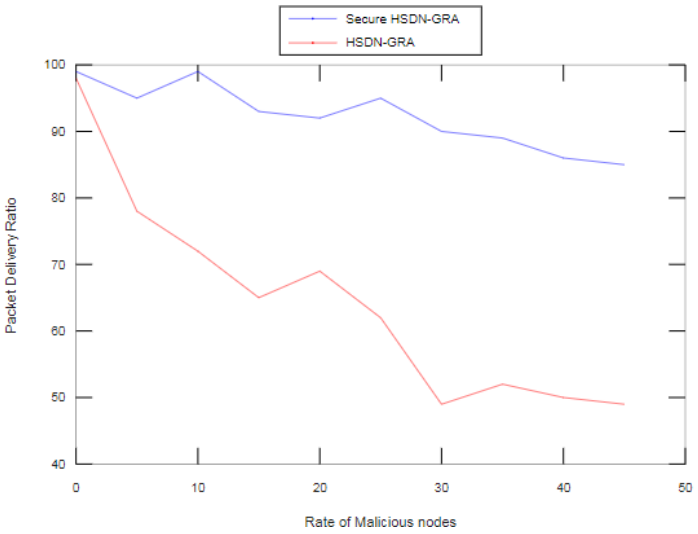

Figure 10: Packet Delivery Ratio vs. Number of Malicious Nodes

Indeed, the Secure HSDN-GRA elects as a relay the neighbor with the best trust value as well as the maximum free load with comparison to the whole neighbors, furthermore, the exchanged packets are signed and encrypted with a secret key Ks which makes the alteration and the destruction of packets more difficult.

Besides, the Secure HSDN-GRA also outperforms the baseline HSDN-GRA of delay in the presence of an increased number of malicious nodes in the simulation as shown in Figure 11.

In fact, the Secure HSDN-GRA takes into account the trust value of each relay with comparison to the whole neighbors as well as the corresponding average free load before choose it. It makes the HSDN-GRA more resilient.

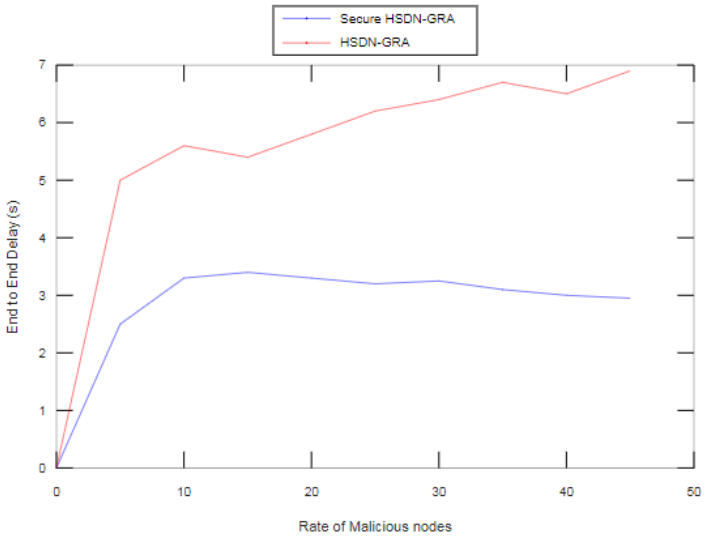

Figure 11: Average End to End Delay vs. Number of Malicious Nodes

4.2.2 Effect of Cluster nodes Density

Figure 12 and Figure 13 show respectively the impact of clusters density on the behaviour of both Secure HSDN-GRA and baseline HSDN-GRA in term of packet delivery ratio and delay especially in the presence of 20% of malicious nodes.

Firstly, for both Secure HSDN-GRA and baseline HSDN-GRA the increase density inside clusters offer better packet delivery ratio because more relays can be exploited. However, the packet delivery ratio is more satisfying when the relays are chosen according to the trust analysis estimated by the network.

Then, we observe in Figure 12 that Secure HSDN-GRA deliver an average of 30.28% of packets more that baseline HSDN-GRA in the presence of 20% of malicious nodes.

Besides, we can deduce that the authentication and the encrypted communications decrease the number of packets alteration, packet lost and malicious intrusion.

Secondly, we can observe that Secure HSDN-GRA reduce the delay comparing to baseline HSDN-GRA because the relays are chosen according to their trust values in order to guarantee the local acknowledgement of each packet sent and reduce the error recovery, consequently the delay is reduced.

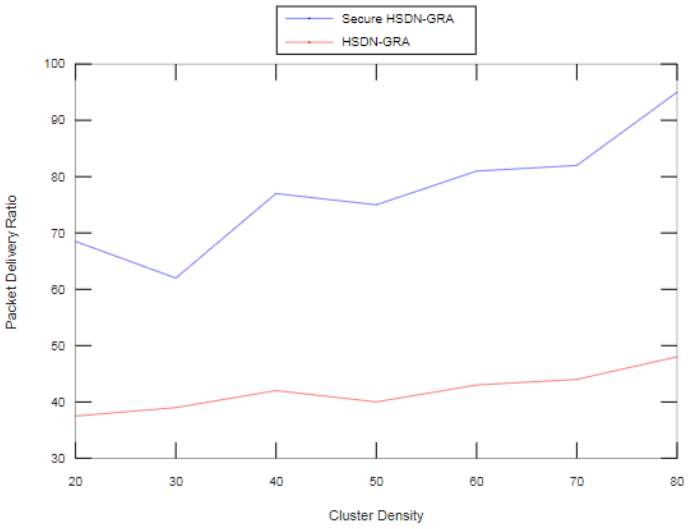

Figure 12: Packet Delivery Ratio vs. Clusters Density with 20% of malicious nodes

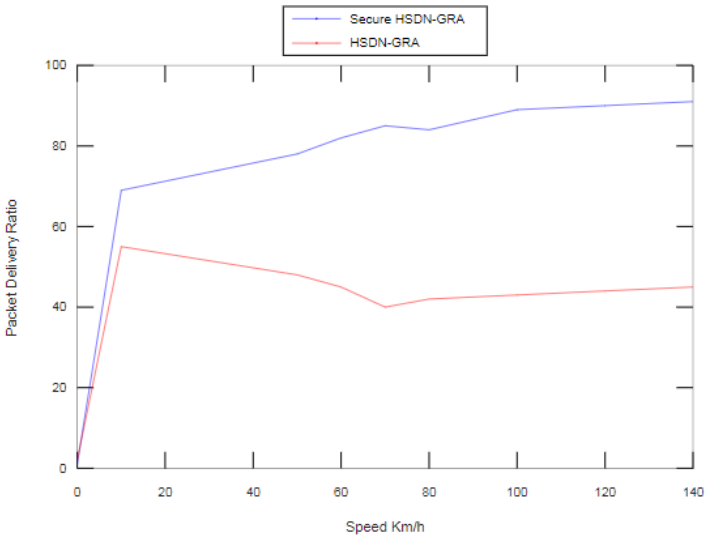

Figure 14: Packet Delivery Ratio vs. Speed with 20% of malicious nodes

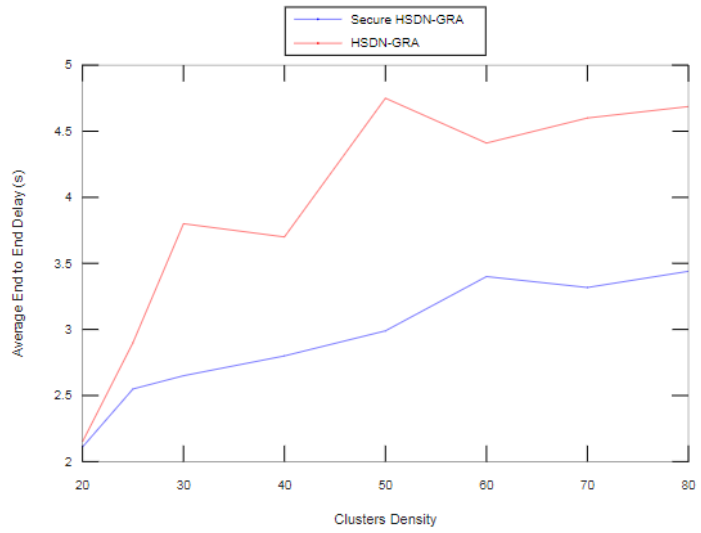

Figure 13: Average End to End Delay vs. Clusters Density with 20% of malicious nodes

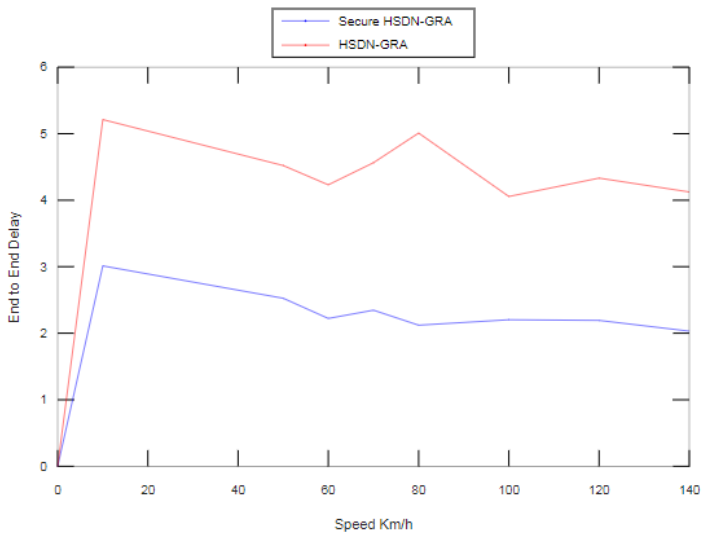

Figure 15: Average End to End Delay vs. Speed with 20% of malicious nodes

4.2.3 Effect of Vehicles Speed

Figure 14 and Figure 15 show the impact of vehicles speed against the Secure HSDN-GRA and baseline HSDN-GRA in the presence of 20% of malicious nodes.

The results in Figure 14 show that the secure HSDN-GRA outperform the baseline HSDN-GRA in term of packet delivery ratio, besides, we can observe that the increase speed stabilizes the packet delivery ratio for the both protocols.

For the delay performances, we can observe in Figure 15 that Secure HSDN-GRA reduces the delay when the speed, furthermore, it surpasses the baseline HSDN-GRA, it is due to the stability of the clusters on the highways and the packets destination are reached faster, since the relays are chosen according to the direction of the destination.

5 Conclusion

The result of applying the SDN architecture on vehicular network consist of making the routing strategies optimal, especially for with load balancing, security policies and also for the network heterogeneity management.

Indeed, it aims to select the most suitable channels and frequencies for data transmission at a specific time according to the context and the requirements.

Besides, it implements various security policies and exploits them adaptively according to the requirements. Finally,the nodes are balanced thanks to the global view given by the control plane.

In this paper, a secure HSDN-GRA for more robustness is exposed, a trust model is built between nodes, and the communications

between the nodes as well as those of the southbound interface are secured with an encryption function.
Even if we have a performing and securing routing protocol, in a totally distributed system it is difficult to guarantee the cooperation of all the entities without any rewards in turn, the selfish nodes constitute an obstacle of communication in a totally distributed network, so as a future perspective, a solution based on blockchain paradigm will be integrated in secure HSDN-GRA in order to tackle selfish nodes and stimulate their cooperation.

Acknowledgment This project was supported by Initiative d'Excellence Paris//Seine.

References

[1] Lylia Alouache et al. "Securing Southbound Interface of HSDN-GRA Vehicular Routing Protocol using a Distributed Trust". In: *Fourth International Conference on Fog and Mobile Edge Computing, FMEC 2019, Rome, Italy, June 10-13, 2019*. 2019, pp. 90–97.

[2] Lylia Alouache et al. "Survey on IoV routing protocols: Security and network architecture". In: *International Journal of Communication Systems* 32.2 (2019), pp. 38–49.

[3] Antonio Di Maio et al. "Enabling SDN in VANETs: What is the Impact on Security?" In: *Sensors* 16.12 (2016).

[4] L Alouache et al. "Toward a hybrid SDN architecture for V2V communication in IoV environment". In: 2018 Fifth International Conference on Software Defined Systems (SDS). Barcelona, Spain, 2018, pp. 93–99.

[5] P. Baskett et al. "SDNAN: Software-Defined Networking in Ad-Hoc Networks of Smartphones". In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA, 2013, pp. 861–862.

[6] M. Azees, P. Vijayakumar, and L. Jegatha Deborah. "Comprehensive survey on security services in vehicular ad-hoc networks". In: *IET Intelligent Transport Systems* 10.6 (2016), pp. 379–388.

[7] M. Liyanage et al. "Opportunities and Challenges of Software-Defined Mobile Networks in Network Security". In: *IEEE Security Privacy* 14.4 (2016), pp. 34–44.

[8] H. Vasudev and D. Das. "A trust based secure communication for software defined VANETs". In: *2018 International Conference on Information Networking (ICOIN)*. 2018, pp. 316–321.

[9] D Zhang et al. "Software-defined Vehicular Ad Hoc Networks with Trust Management". In: *the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. DIVANet '16. Malta, Malta: ACM, 2016, pp. 41–49. ISBN: 978-1-4503-4506-4.

[10] Seyed Soleymani et al. "Trust Management in Vehicular Ad-Hoc Network: a Systematic Review". In: 2015 (2015).

[11] N. Bimeyer et al. "Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters". In: *2012 IEEE Vehicular Networking Conference (VNC)*. 2012, pp. 78–85.

[12] Junhai Luo, Xue Liu, and Mingyu Fan. "A Trust model based on fuzzy recommendation for Mobile Ad-hoc Networks". In: *Computer Networks* 53.14 (2009), pp. 2396 –2407. ISSN: 1389-1286.

[13] Jim Schaad. *Use of the Advanced Encryption Standard (AES) Encryption Algorithm in Cryptographic Message Syntax (CMS)*. RFC 3565. 2003. URL: `https://rfc-editor.org/rfc/rfc3565.txt`.

[14] Margaret Rouse. *Challenge - Response Authentication Definition: What does Challenge - Response Authentication mean?* `https://searchsecurity.techtarget.com/definition/challenge-response-system`. Accessed: 2018-05-30.

[15] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. 1st ed. Springer Publishing Company, Incorporated, 2008. ISBN: 0387717595, 9780387717593.

[16] Jrme Hrri et al. "Vehicular mobility simulation with VanetMobiSim". In: *SIMULATION* 87.4 (2011), pp. 275–300.

Appendix

Table 5 presents the abbreviations used throughout the paper.

Table 5: Abbreviations used in the paper

ACK	Acknowledgment
AES	Advanced Encryption Standard
DoS	Denial-of-Service
DSCR	Dedicated Short-Range Communication
FMEC	International Conference on Fog and Mobile Edge Computing
GPS	Global Positioning System
HSDN-GRA	Hybrid SDN-based Geographic Routing Protocol with Multi-agent Approach
IoV	Internet of Vehicle
I-TAODV	Improvised Trust based Ad-hoc On-demand Distance Vector routing
LTE	Long Term Evolution
SDN	Software-Defined Networking
SD-TAODV	Software-Defined Trust based Ad hoc On-demand Distance Vector routing
SDVN	Software-Defined Vehicular Networking
TAODV	Trust based secure routing in AODV routing protocol
ToF	Timer-of-Refresh
UDP	User Datagram Protocol
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad Hoc Network
WAVE	Wireless Access in Vehicular Environments

Listing 1: Some NS2 Simulation Functions

```
static inline double jitter (double max, int be-random-)
static class ANGLEclusterHeaderClass : public PacketHeaderClass
static class ANGLEclusterClass
ANGLEclusterClass() :
TclClass("Agent/ANGLEcluster"):
public TclClass ("PacketHeader/ANGLEcluster",sizeof(hdr-all-
    anglecluster))
void ANGLEclusterHelloTimer::expire(Event *e)
UpdPosTimer::expire(Event *e)
StateTimer::expire(Event *e)
PERTimer::expire(Event *e)
EndElecTimer::expire(Event *e)
ANGLEclusterAgent::ANGLEclusterAgent() : Agent(PT-ANGLECLUSTER)
int ANGLEclusterAgent::command(int argc, const char*const* argv
    )
void ANGLEclusterAgent::SendHello()
void ANGLEclusterAgent::SendBeacon(u int8 t type)
void ANGLEclusterAgent::SendMSG()
void ANGLEclusterAgent::SendRoutingMSG(nsaddrt s,nsaddrt d, int
     i)
void ANGLEclusterAgent::sendACK(nsaddrt node, float seqnumber)
void ANGLEclusterAgent::sendPkey()
void ANGLEclusterAgent::sendRequestToLeave(nsaddrt node)
void ANGLEclusterAgent::sendRevoke(nsaddrt node)
void ANGLEclusterAgent::receive(Packet* p, Handler*)
void ANGLEclusterAgent::RecvHello(Packet *p)
void ANGLEclusterAgent::RecvBeacon(Packet *p)
void ANGLEclusterAgent::RecvCHBeacon(Packet *p)
void ANGLEclusterAgent::RecvRoutingMsg(Packet *p)
void ANGLEclusterAgent::ReceiveACK(Packet *p)
void ANGLEclusterAgent::ReceiveRevoke(Packet *p)
void ANGLEclusterAgent::timeout()
tosend ANGLEclusterAgent::caluclWeight()
int ANGLEclusterAgent::search(nsaddrt inode)
void ANGLEclusterAgent::getSpeed()
void ANGLEclusterAgent::getmovspeed(double * sp)
void ANGLEclusterAgent::getLoc()
void ANGLEclusterAgent::GetLocation()
void ANGLEclusterAgent::updcoord()
void ANGLEclusterAgent::signalPre()
void ANGLEclusterAgent::signalRouting()
void ANGLEclusterAgent::showState()
void ANGLEclusterAgent::PurgeTable(u int8t num)
void ANGLEclusterAgent::secretKey()
void ANGLEclusterAgent::signature(Packet* p, Handler*)
void ANGLEclusterAgent::crypt(Packet* p, Handler*)
int ANGLEclusterAgent::numofNeighbors()
int ANGLEclusterAgent::existmember()
void ANGLEclusterAgent::Popularity(nsaddrt node,int cas)
double ANGLEclusterAgent::Frequency(nsaddrt node,int cas)
int ANGLEclusterAgent::getPopularity(nsaddrt node)
double ANGLEclusterAgent::getFrequency(nsaddrt node)
nsaddrt ANGLEclusterAgent::unReliableNode()
float ANGLEclusterAgent::getTrust(nsaddrt node)
void ANGLEclusterAgent::updatePositions()
void ANGLEclusterAgent::endelection()
void ANGLEclusterAgent::VehicleState()
void ANGLEclusterAgent::RecvPresMSG(Packet *p)
void ANGLEclusterAgent::BestNode(nsaddrt node, double pd)
nsaddrt ANGLEclusterAgent::NextRelay()
void ANGLEclusterAgent::Revoke(nsaddrt node)
```