# A TRUST MANAGEMENT FRAMEWORK FOR VEHICULAR AD HOC NETWORKS

Rezvi Shahariar and Chris Phillips

# School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, UK

# ABSTRACT

Vehicular Ad Hoc Networks (VANETs) enable road users and public infrastructure to share information that improves the operation of roads and driver experience. However, these are vulnerable to poorly behaved authorized users. Trust management is used to address attacks from authorized users in accordance with their trust score. By removing the dissemination of trust metrics in the validation process, communication overhead and response time are lowered. In this paper, we propose a new Tamper-Proof Device (TPD) based trust management framework for controlling trust at the sender side vehicle that regulates driver behaviour. Moreover, the dissemination of feedback is only required when there is conflicting information in the VANET. If a conflict arises, the Road-Side Unit (RSU) decides, using the weighted voting system, whether the originator is to be believed, or not. The framework is evaluated against a centralized reputation approach and the results demonstrate that it outperforms the latter.

#### **KEYWORDS**

VANET, Trust Management, Security, Tamper Proof Device, Malicious Behaviour

# **1. INTRODUCTION**

Vehicular Ad Hoc Networks (VANETs) can provide traffic advice, safety announcements, and infotainment services to road users. Typically, a vehicle may report an emergency event with other road users or may request the location of a petrol pump or a nearby parking area. VANETs are also deployed to mitigate the aftereffect of road incidents and to warn vehicles in advance. However, as this application involves wireless communication, it is at risk of security attacks. Additionally, drivers can fraudulently broadcast false messages. To be successful, messages must be accurate and trustworthy, otherwise, with a malicious untrue message, a vehicle can mislead many others causing congestion or other undesirable phenomena.



Figure 1. Typical Mechanism for Reporting an Accident

For example, as shown in Figure 1, suppose a vehicle "V" broadcasts a message reporting a crash. This message could be truthful or false. If other vehicles receive a false message, their subsequent detour will impact their travel time. However, for a truthful announcement, the detour permits them to avoid potential congestion.

In a VANET, outsider attacks can be thwarted using a cryptographic scheme but not insider attacks. Trust-based approaches are used to thwart insider attacks from malicious authorized users [1, 2, 3]. It is noted in [4, 5] that trust schemes can improve security by identifying dishonest vehicles and revoking messages from them. Even so, trust approaches cannot protect VANETs completely [1]. Basically, the trust that vehicle W attributes to vehicle V is the confidence W places in a set of actions from V. Typically, the reliability of relayed information is periodically evaluated using predefined metrics and computational methods [6]. Vehicles that consistently maintain a good trust score can be considered trustworthy by others as their current trust scores rely on their previous trustworthy announcements. However, it is not guaranteed that a trusted vehicle will always broadcast trustworthy messages.

In existing approaches [2, 3, 6-8] both trusted and untrusted vehicles can broadcast messages. Untrusted vehicles are expected to broadcast more malicious messages than trusted vehicles, which produces an additional demand on the network both in terms of message volume and the verification process. This places a considerable burden on the receivers. Methodologies based on direct and indirect trust require regular monitoring of activities across both single and multi-hop transmission ranges. Some approaches [2, 4, 9] can result in considerable trust metric exchanges to verify the original announcement. These messages, along with the event announcement, complicate the situation as it is necessary to evaluate their validity in a short time frame due to fast vehicle movement [10]. The authors in [6] claim receivers should decide the trust of messages in a short timeframe. However, when receivers independently compute trust from their neighbours' trust metrics, they suffer from high response times [2, 6, 9]. Alternatively, approaches that allow trust computation at a centralized server need to communicate to obtain updated trust information concerning the sender vehicle. This introduces an additional delay in the decision-making process concerning emergency events. Consequently, some vehicles may drive into the event zone despite being previously warned, as suggested in [11]. Also, there is an open debate [4, 8] regarding how often a centralized server should communicate revised trust data. Therefore, this paper proposes a novel Tamper-Proof Device (TPD)-based sender-side trust management framework for VANETs with the following features:

- To the best of our knowledge, for the first time, this framework employs a sender-side trust model to regulate access control using information accuracy, delay, and positional differences collected from the sender vehicle itself. Unlike other approaches [2, 4, 7, 8, 10, 12, 13], there is no flow of trust metrics unless a reporter vehicle refutes an announcement.
- Various classes of messages along with their associated trust threshold are defined for regulating access control which confirms that only trusted vehicles can announce messages.
- The scheme employs a collaborative untrue message discovery algorithm for detecting various forms of attack.
- Receivers can instantly act on a sender's message knowing that the sender must have sufficient trust, or they can report an untrue attack if they do not see the event on the roads.
- The framework is simulated in the Veins to verify its satisfactory operation. The communication overhead and response time are compared against a typical reputation approach [8] with varying vehicular densities and speeds. Moreover, the accuracy of the framework is presented with differing percentages of malicious and benevolent feedback.

Our paper is organized as follows. Section 2 reviews existing trust models. Section 3 presents the core parts of the proposed trust model. Section 4 illustrates the system model and verification of the framework. Section 5 then analyses the framework and provides a performance comparison against a baseline approach. Finally, in Section 6, we conclude this work.

# 2. RELATED WORK

Till now, many trust models for VANETs have been proposed for evaluating message trustworthiness [5]. These models are categorized into three groups. The first group is called the Entity-Oriented Trust Model (EOTM) which verifies only an entity's trust. The second group is called the Data-Oriented Trust Models (DOTM) which focus on evaluating the trustworthiness of data only. The third group is called the Hybrid Trust Model (HTM) which evaluates both entity's trust and the reliability of data. Below trust models from these categories are highlighted briefly.

Entity-oriented trust models are typified by [7], where the researchers use a Tamper Proof Module (TPM) on every vehicle to find the cost for the transmission and then adjust trust from the receivers' feedback. However, the trust score updating leads to excessive communication. Conversely, the authors in [14] apply fuzzy logic to calculate trust using experience, plausibility, and location accuracy. It can detect bogus messages and alteration attacks. However, additional communication is required as vehicles consult with fog nodes to check the location accuracy. Ref [9] uses both fuzzy logic and Q-learning for trust calculation. This approach is evaluated in terms of precision and recall with varying numbers of malicious vehicles. However, the model requires repeated sensing of messages from neighbours. In contrast, Ref [2] uses the Bayesian rule and Dempster-Shafer Theorem (DST) for trust calculation. It combines independent beliefs to determine the trust of a vehicle. However, an erroneous recommendation can bias the trust calculation. Reference [4] uses reputation and receivers' feedback on received messages to calculate trust. The scheme is evaluated in presence of false messages in both urban and highway scenarios. Nevertheless, it may suffer from excessive trust metric dissemination. The researchers in [15] propose a past interactions-based reputation management scheme for VANETs. A vehicle collects a signed reputation from the server to attach to its messages. A receiver verifies the message to accept or reject it and updates the reputation of vehicles on the server. However, this approach requires periodic reputation exchanges with the server. Another trust model is proposed in [16]. They consider familiarity, packet delivery ratio, timeliness, and interaction frequency as parameters to manipulate final trust. This model is analyzed considering the recent history of interactions but not considering any attacker model. In [17], the authors introduce a blockchainbased trust management protocol for VANETs to update of trust of vehicles as low, medium, and high at the Trust Authority (TA) using tamper-proof logs periodically. However, this is not validated using any known attack.

Data-oriented trust models are typified by [18], where packets are forwarded along the most trusted path using a trusted routing protocol. An intrusion detection module thwarts only denial of service (DoS) attacks. Ref [19] presents an infrastructure-less, data-oriented trust model which verifies content similarity and conflict as well as route similarity. Conversely, the reference [20] proposes a distance and geolocation-based probabilistic approach to estimate the trust from received data. This model does not forward a message beyond a certain distance. Ref [21] proposes a Bayesian Inference-based voting mechanism and vehicles run Dijkstra's algorithm as a route update requires upon every message arrival. Messages carry a time parameter as road ID while being forwarded. However, this requires frequent maintenance of routes. The researchers in [22] propose a trust model called FACT for achieving reliable information dissemination in VANETs using two modules. The first module evaluates the trust of the message, whereas the

second one finds a highly trusted path for forwarding messages. However, this is an applicationoriented scheme that does not use any infrastructure to monitor activities.

The researchers in [5] evaluated one data-oriented, one entity-oriented, and one hybrid trust model under various adversary scenarios. In addition, a risk assessment model is also presented for the identification of critical vulnerabilities. Ref [6] checks the reliability of messages using direct interactions and stores previous interactions and the trust of neighbours in a local database. However, there is no false trust message detection scheme. Conversely, the authors in [12] calculate data trust from multiple vehicle responses and vehicle trust from functional and recommendation trust. Though this model considers simple, bad mouth and zigzag attacks, but not compared with any model. In [3], RSUs use hash message authentication code (HMAC) and digital signature to evaluate the trust of vehicles based on neighbour trust values and only reward. They measure communication overhead and suggest integrating the ID-based and batch signatures in the future. The researchers in [8] update vehicle reputations at the RSUs using the noticed events from vehicles. The RSU next announces updated reputations to vehicles. Receivers store all the messages about an event until a timer expires to decide whether an event is true or false. However, this model suffers from high response times and communication overhead. Alternatively, in [23], a self-organizing trust model is considered for both urban and rural settings which can detect fake locations, and fake times and revoke messages as necessary. This model validates the trust of messages and then accepts the message with the highest trust for an event. However, its efficacy is not analyzed. In [24], the authors present a trust model, where the entity-centric trust model of this scheme thwart the black-hole attack and selective forwarding attack. The data-centric trust model is used to discover relations among data and performs trust evaluation based on utility theory. The data trust model can be further improved by selecting the appropriate utility parameter. Alternatively, in [25], a risk-based hybrid trust model is proposed and compared with a multi-facet-based trust model. The result suggests that it always selects a low-risk action which is different from what the trust-based approach suggests. However, this work is only suitable for a clustered architecture which is unrealistic for VANET. Ref [26] uses a Bayesian inference-based direct trust and recommendation trust to calculate the final trust. This model finds the confidence of direct trust to avoid the costly recommendation trust calculation. Though this approach is compared with two models, they consider only packet drop and interception as malicious behaviours.

Many blockchain-based trust models are also present in the literature. Here, we only highlight two approaches though our proposed model is not blockchain-based. Ref. [27] presents a three-layer blockchain-based trust model for VANETs using Dirichlet distribution, regression, and revocation. They consider simple, slander, and strategic attacks along with both normal, and malicious servers. However, the work does not reward benevolent activities from vehicles. Also, in [28], the authors present a decentralized blockchain-based trust model which selects a message evaluator through RSU collaboration. The approach finds the rating for messages, the sender, and the evaluator. Next, they calculate the global trust of a node based on the rating, and message quality. They preserve trust data in the blockchain and use a consensus process to insert blocks. They claim that their approach can prevent Sybil, message spoofing, bad-mouthing, and ballot-stuffing attacks. However, this model is not compared against other trust-based models.

# **3. PROPOSED SYSTEM**

Ideally, a trust model promoting security should incur little or no extra burden in terms of computational and communication cost. In a VANET, vehicles typically meet each other randomly and fleetingly. Thus, there is little time for decision-making based on trust. With a receiver-side trust model, vehicles with a poor trust score can still send messages although these

will typically be ignored by receivers once their trust level is verified. However, this takes time, so vehicles may unnecessarily experience events such as traffic jams. To this end, our research proposes a novel sender-side trust management framework that reduces the amount of trust information passed over a VANET and blocks untrusted transmission attempts. It uses a sender-side TPD on vehicles to prevent unauthorized access and regulate transmissions based on the level of trust. Different classes of messages are created that associate different trust thresholds to permit their broadcast. Drivers improve their trust score from valid announcements, forwarding, beaconing, and clarifying events to an RSU. Also, a driver builds trust from RSU rewards whenever "wins a dispute" over another vehicle. The beaconing reward is only given when a driver is not blocked, and trust is less than 0.5. The TPD also punishes drivers when announcements are delayed, or the vehicle is travelled more than a threshold distance besides the arrangement of RSU punishments whenever a driver "loses a dispute" to another vehicle. Since announcements are regulated by the sender's TPD, receivers can believe messages and the sender's trust instantly.

# **3.1. System Assumptions**

The framework assumes that the security of the TPD is beyond the scope of the proposed framework as it relates to physical layer protection. Also, we do not consider other security aspects with this trust framework as we believe that existing security techniques can address authentication, privacy, and integrity. A security approach that supports these functionalities could be incorporated with the framework to confirm the authenticity of the driver and/or messages with other entities, secure the privacy of the driver, and any Hash Message Authentication Code (HMAC) for achieving integrity. For example, when a driver registers with the TA, the driver can obtain private and public keys to encrypt and decrypt messages and can obtain a pseudo-identity associated with his driver ID for securing privacy with other drivers, and HMAC can be used to maintain integrity [1]. RSUs, official vehicles, and the Trust Authority (TA) are also considered fully trustworthy. Both the TA and TPDs are governed and owned by the Road Transport Authority (RTA). The resilience of the TA infrastructure is beyond the focus of this work. We assume a driver has a built-in dashboard with designated touch buttons to display the classes of messages available given his/her current trust score and to generate specific emergency events for other vehicles. Furthermore, a TPD can access GPS to determine the location of the vehicle.

# 3.2. Registration, Blocklisting, and Redemption

Drivers may register themselves with the TA directly using an online form with the vehicle plate number as vehicle ID and driver's license number as driver ID. Since this is an external means, this is outside the scope of the framework. Alternatively, if the system chooses to send a registration message from the driver when they start initially; then the RSU forwards the registration and confirmation of registration to and from the TA.

RSUs send the decisions of disputed events to the TA to store in a driver profile database which keeps driver and vehicle information, the event information, and the reward/punishment. When the TA receives a decision on a disputed event, then it searches the driver profile database. If three malicious events have been found in a limited timeframe, the TA sends a blocking confirmation message to the RSUs in the driver's vicinity. When the vehicle receives a blocking confirmation message, the TPD blocks further access of the driver and acknowledges the blocking confirmation message with the TA via an RSU. The blocked driver can only send/receive beacons into the VANET. Additionally, a blocking message can be generated from the vehicle's TPD when the driver's trust score crosses the lowest acceptable trust limit. This message is forwarded by an RSU to the TA and the same mechanism is followed to block the

access of traffic events for the driver in the VANET. By default, based on experimentation, regular drivers obtain access to traffic events in the VANET with a trust score between 0.06 to 0.9. Whenever the trust score becomes lower than 0.06, an external mechanism requires the driver to communicate with the TA to obtain redemption from blocking. We assume this is within the jurisdiction of the road transport authority (RTA) and may involve issuing a monetary penalty or other sanctions.

#### **3.3. Framework Components**

Both regular vehicles and official vehicles are present within the framework. The framework is extensible so that more vehicle classes could be added. The approach supports the same vehicle accommodating multiple drivers via individual driver trust management. The actions and responsibilities of each vehicle type are limited to their role. Every vehicle is pre-equipped with a built-in On-Board Unit (OBU), comprising a Global Positioning System (GPS) for location access, a transceiver to communicate with other entities, and a TPD that manages the trust and regulates transmissions. We define the following actors based on their roles:

- Senders: are drivers that can originate both true and untrue announcements relating to an incident, such as an accident, subject to their trust score. For a true announcement, the trust manager within the TPD rewards the driver if the claim is not disputed within a given time.
- Reporters: are drivers that refute an announcement of a sender and receive a reward or punishment if the challenge is confirmed or dismissed, respectively. If they do decide to make a report, they may do so either truthfully or falsely. Failure to make a report carries no penalty.
- Receivers: are drivers that receive messages from any entity and relay them automatically provided the hop limit is not reached and their trust is sufficient.
- Clarifiers: If a dispute is detected at an RSU, the RSU transmits a query message seeking clarification concerning a disputed incident. Vehicles that receive this message can choose to answer the query, i.e., to respond to the RSU, confirming or denying that the incident has taken place, or ignore it. If they respond, they are considered clarifiers.
- Road-Side Units: are automated units that receive information from senders, reporters, and clarifiers, either directly or via intermediate vehicles that rebroadcast the received messages. If information from multiple senders, reporters, or a combination of these conflicts, then the RSU will rule on the dispute. RSUs act as an intermediary between the vehicles and the TA.
- Trust Authority: is the ultimate authority in this framework which validates registration and blocklisting of drivers. The TA blocklists a driver whenever it receives a blocklist message initiated from a TPD or if it finds the three malicious events (3ME) for the same driver within a configurable timeframe. The TA then replies with a blocking confirmation to the RSUs in the vicinity of the last disputed event to reach the vehicle's TPD. Incidents reported by RSUs are saved by the TA in an incident database including the location, timestamp, and incident information. The TA also maintains a driver profile database containing the reward/punishment history of drivers.
- Official Vehicles: This framework considers police, ambulance, and fire service vehicles as official vehicles. Their primary task is to respond to emergency issues on roads by cooperating with RSUs. They are always trusted.

# **3.4. Trust Evaluation Mechanism**

This framework sets an initial trust score of 0.45 for regular vehicles to avoid the cold start problem and with the expectation that they will achieve a trust score of 0.5 relatively quickly so that they can then announce all events that the framework supports. Regular vehicles are

considered trusted when their trust score is 0.5 or higher. A regular vehicle's trust score cannot go above 0.9 or below 0.05. The trust T of a regular vehicle i is expressed by Eqn (1).

$$T_i = \{ t \mid t \in \mathbb{R} \mid 0.05 < t \le 0.9 \}$$
(1)

The following rules govern the actions of regular vehicles considering T as trust. If  $T \le 0.05$  (blocked state) of a driver and/or vehicle, then TPD sends a blocking message automatically to the TA, and the vehicle can only generate periodic beacons. Then the TPD waits and blocks network access for all traffic events upon blocking confirmation from the TA. If  $0.05 < T \le 0.25$  (not trusted state), then the vehicle can announce periodic beacons as well but cannot forward events from others. If 0.25 < T < 0.5 (lowly trusted state), the vehicle can send beacons, make limited announcements, and can forward events from others. If  $0.5 \le T \le 0.9$  (trusted state and highly trusted when (T = 0.9)), the vehicle can forward and announce all classes of events. If regular vehicles spread untrue messages multiple times, then they receive incremental punishments from RSU. If this count becomes three for severe situations like false accident announcements, then the network access for the driver is blocked. Since the framework considers official vehicles, they are assigned a higher trust score over regular vehicles (i.e. T = 1.0) as regular vehicles should not be trusted more than an official vehicles.

We envision a driver's dashboard as consisting of a set of buttons for supported actions in the framework. Appropriate buttons can be pressed relevant to a specific type of road incident. There are three main classes of messages in the hierarchy. The lowest class consists of beacons and "wave" service announcements, though a blocked driver cannot use the "wave" service facility. The next class of messages consists of announcements of poor road conditions, debris, road defects, and so forth. These can only be broadcasted by drivers with a trust score greater than 0.25. The highest class of messages consists of announcements for accidents, traffic jams, road closures, etc., as well as untrue attack reporting messages. To announce a message from this class, a driver needs to have a trust score of at least 0.5. Algorithm 1 shows the announcement, retransmission, relaying, feedback, and reporting activities of regular vehicles. In Algorithm 2, the TPD trust update and blocking management are shown for regular vehicles. Notations and symbols for Algorithms 1 and 2 are provided in Table 1.

#### Table 1. List of Notations

Notation	Meaning	Notation	Meaning
RSUr	r <sup>th</sup> RSU	HL and $RT_L$	hop limit and retransmission
			limit
$T_s(D_s(V_s))$	trust <sub>s</sub> of driver s of vehicles	Reward <sub>f</sub> ,	reward for forwarding,
		Reward <sub>clar</sub> , and	clarification, and reporting
		Reward <sub>unt-atck</sub>	
Vrep, Vrec, & Vtrust-	reporter, receiver, and trusted	LowTrust <sub>msg</sub>	forwarding is not possible
cla	clarifier vehicle		with low trust
timer <sub>reward-withhold</sub>	when to process	timer <sub>bilst</sub>	to check blocking condition
	reward/punishment		
evt <sub>e</sub> and	traffic event and reporting the	driver_List	registered driver list
untrue(evt <sub>e</sub> )	<i>evt</i> <sub>e</sub>		
RSU <sub>clarif_query</sub>	clarification query from RSU	Trust <sub>s</sub> , Trust <sub>d</sub>	saved and initial trust
$T_{dis}$ , and $T_{int}$	time threshold to send	Complaint_List	list of reported
	feedback and to report $evt_e$		announcements
TTL, and M <sub>cls</sub>	Time-To-Live, class of	longDelayed	driver delayed than the upper
	messages		limit
ATT(M <sub>cls</sub> )	associated trust threshold of	Msg <sub>block &amp;</sub>	blocking and blocking
	M <sub>cls</sub>	Msg <sub>block-conf</sub>	confirmation message
Rew <sub>r</sub> /Pun <sub>r</sub> and	reward/punishment from a	PosDiff	distance between the event
Rew <sub>tpd</sub> /Pun <sub>tpd</sub>	RSU <sub>r</sub> , and TPD		and announcement location

Algorithm 1. regular vehicle traffic event management

**Input**: Driver ID, Vehicle ID, events, trust of drivers, hop and retransmit limit **Output**: controlled broadcasting, relaying, reporting, and sending feedback

- 1. case eventType of
- 2. *witnessed-event://* to warn others.
- 3. **if**  $(T_s(D_s(V_s)) \ge ATTL(evt_e))$
- 4.  $D_s(V_s)$  prepares and broadcasts the *evt*<sub>e</sub>
- 5. Send metrics to TPD to find Rew<sub>tpd</sub>/Pun<sub>tpd</sub>
- 6. **end if**
- 7. *reported-event:*// to report the received event.
- 8. **if** (V<sub>s</sub> decides  $evt_e$ =false) and (V<sub>s</sub> visits event place within T<sub>int</sub>) and (T<sub>s</sub> (D<sub>s</sub> (V<sub>s</sub>))  $\geq 0.5$ )
- 9. Send *untrue(evt<sub>e</sub>)* towards RSU
- 10. Notify TPD to add  $T_s = T_s + Reward_{unt-atck}$
- 11. end if
- 12. *relayed-event://* to relay event up to hop limit.
- 13. **if** (V<sub>s</sub> gets an *evt*<sub>e</sub> or an *untrue*(*evt*<sub>e</sub>) from a V<sub>rep</sub> first time)
- 14. **if** ( $V_s$  sends *evt*<sub>e</sub> or *untrue* (*evt*<sub>e</sub>))
- 15. Return
- 16. **end if**
- 17. **if** ( $T_s \in (T_s > 0.05 \text{ and } T_s <=0.25$ )
- 18. Send a LowTrust<sub>msg</sub>
- 19. else

- 20. **if** TTL( $evt_e$  or  $untrue(evt_e) \ge HL$ )
- 21. Stop resending *evt*<sub>e</sub> or *untrue*(*evt*<sub>e</sub>)
- 22. else
- 23. Resend  $evt_e$  or  $untrue(evt_e)$  up to HL
- 24. Notify TPD to add  $T_s = T_s + Reward_f$
- 25. **end if**
- 26. end if
- 27. end if
- 28. *retransmit-event:* // to repeat the broadcasting
- 29. if (no\_of\_time  $\leq RT_L$ )
- 30. Resend  $evt_e$
- 31. end if
- 32. *feedback-event:* // to send feedback.
- 33. if (V<sub>s</sub> receives a RSU query about *evt*<sub>e</sub>)
- 34. **if** ( $V_s$  is a  $V_{rep}$  or is the sender of  $evt_e$ )
- 35. Return
- 36. **end if**
- 37. **if** TTL(RSU<sub>clarif\_query</sub>) < HL
- 38. Resend RSU<sub>clarif\_query</sub> message
- 39. **end if**
- 40. **if** ( $V_s$  visits the event location within  $T_{dis}$ )
- 41. Send feedback

42.	Notify	TPD	to	add	$T_s =$	T <sub>s</sub> +	44. end if
reward <sub>clar</sub>							45. end case
43. <b>end if</b>							

Alg	gorithm 2: trust and blocking management at	the TPD
Inp	out: Announced evte, reporting status of evte,	Msg <sub>block-conf</sub> , PosDiff, delay, Rew <sub>r</sub> /Pun <sub>r</sub> ,
Ou	tput: Trust update and access blocking.	
1.	case eventType of	35. <b>else</b>
2.	periodic-blocking-checker://	36. start timer <sub>reward-withhold</sub> to process
3.	if $D_s(V_s)$ is unblocked) and (timer <sub>blist</sub>	reward
	expires) and $(T_s(D_s(V_s)) \le 0.05))$	37. end if
4.	TPD <sub>s</sub> issue a Msg <sub>block</sub> to reach TA.	38. end if
5.	end if	39. TPD reward/punishment://add with
6.	<b>if</b> Msg <sub>block-conf</sub> comes from TA for $D_s(V_s)$	trust
7.	Disable the network access for D <sub>s</sub>	40. if (broadcasted msg_id $\in$
8.	end if	Complaint_List)
9	<b>RSU reward/punishment:</b> // add with	41. Update $T_s(D_s(V_s) = T_s(D_s(V_s)$
	trust.	42. Return
10.	<b>if</b> ( $\text{Rew}_r/\text{Pun}_r$ from an $\text{RSU}_r$ for the	43. <b>else</b>
	$D(s(V_s)))$	44. Update $T_s(D_s(V_s) = T_s(D_s$
11.	$(T_s(V_s(D_s)) = (T_s(V_s(D_s)) + \text{Rew}_r/\text{Pun}_r)$	$(V_s)$ +reward <sub>tpd</sub>
12.	end if	45. <b>if</b> $(T_s(D_s(V_s))>0.9$
13.	<i>metrics</i> : // Applies reward withholding.	46. Update $T_s(D_s(V_s)=0.9)$
14.	if ( $evt_e$ =false by receiving a complaint)	47. end if
15.	reward <sub>evt-e</sub> =0	48. <b>if</b> $(T_s(D_s(V_s)) \le 0.05)$
16.	else	49. Update $T_s(D_s(V_s)=0.05)$
17.	case [PosDiff   D]	50. Start timer <sub>blist</sub>
18.	$0 < PosDiff < 300m \mid 0 < D < 15s:$	51. end if
19.	reward <sub>tpd</sub> = $0.08$ .	52. end if
20.	$301 < PosDiff < 500m   16 < D \le 30s$ :	53. complaint-on-broadcasted-event:// save
21.	reward <sub>tpd</sub> = $0.06$ .	report.
22.	$501 < PosDiff < 800m   31 < D \le 60s$ :	54. If the broadcasted msg_1d has a
23.	reward <sub>tpd</sub> = $0.05$ .	complaint
24.	$801 < PosDiff < 1200m   61 < D \le 120s$ :	55. Save the complaint into the
25.	reward <sub>tpd</sub> = $0.01$ .	Complaint_List
26.	1201 <posdiff<1500m 121<d<="" td=""  =""><td>56. end if <math>57.4</math></td></posdiff<1500m>	56. end if $57.4$
27	$\leq 1508$ :	5/. <i>ariver-change-event:</i> // to change driver
27.	reward <sub>tpd</sub> =-0.01.	58. Extract the driver_name $D_s$
28.	IongDelayed=true	59. If $(D_s \text{ exists in the driver_List})60$ . Use
29.	POSDIII > 1500 m   D > 150S:	$D_s$ as the current driver and Trust <sub>s</sub>
30. 21	reward <sub>tpd</sub> =-0.05.	
31.	longDelayed=true	62. Add $D_s$ in driver_List and $T_s=Trust_d$
32. 22	end case if long Delayed true	03. end II
33. 24	ii longDelayed=true	04. end cas
34.	can reward/punishment process	
	inimediately	

A potential reward is initially assessed at a TPD and then withheld for a period before adding it to the current trust. During this time, a TPD checks whether any complaint has been raised by any reporter. Rewards are calculated based on message accuracy (no complaint), location difference,

and delay/responsiveness. Thus, the framework promotes emergency event announcements at the earliest possible opportunity. The distance a vehicle moves between the event location and the vehicle's current position is passed to the TPD to determine the location difference. Delay is calculated as the difference between the announcement and the observation time on the road. The TPD uses this information to assess the reward/punishment for the announcement. Also, the framework suggests a vehicle should not travel more than 500 meters or the next traffic signal to earn a higher reward from the announcement. The trust  $T_i$  is updated inside the TPD using Eqn (2).

$$T_i = T_{i-1} + R_i "or" P_i$$
 (2)

Here, Ti is the revised trust score of a vehicle after adding a reward/punishment to its current trust Ti-1. Ri "or" Pi is the estimated reward or punishment for the i<sup>th</sup> message announcement. The set of rules used by the TPD for deciding the appropriate reward/punishment magnitude for a given announcement is shown in the "metrics" event handling in Algorithm 2. In the reward/punishment assessment, the accuracy of the announcement, delay, and location difference is considered and then the current trust is updated using the assessed reward/punishment. The assessed reward is then withheld for a period and the reporting status of the announced message is checked before the trusted update. If the message is accurate but the driver exceeds the thresholds, then no reward or a nominal punishment is issued. But the assessed punishment is deducted immediately from the current trust. However, if the message is inaccurate (i.e., a complaint received during the reward withholds period), then the driver receives no reward for the announcement from the TPD and defers the reward/punishment decision to the RSU. As a VANET is a time-critical system, vehicles should disseminate information promptly. Thus, the delay and distance traveled are given prime importance in the reward calculation besides the message accuracy.

#### **3.5. Functional Diagram of the Proposed Framework**

In Figure 2, assume a driver sees an incident and wishes to announce it. The framework first checks the trust score from the TPD and determines if the action is eligible with the driver's current trust. If this test is passed, then the driver announces the incident. Other "receivers" forward it up to the configurable hop limit. As this is an original announcement, the driver is classified as a "sender". If this announcement reaches subsequent drivers who visit the same location later, they can notice whether the said event occurs or not. However, if any driver believes the announcement to be untrue, that driver can send a complaint to the RSU. When the RSU receives this complaint, the RSU requests "trusted clarifiers" to respond, confirming or denying the claim.

It collects feedback from these trusted clarifiers who have recently visited the event location. After this, the RSU rules on the validity of the event and penalizes or rewards the respective vehicles. An RSU always informs the TA of the outcome of a dispute, which could be a driver being malicious. It is then up to the TA to check prior behaviour for three malicious activities from a specific driver over a configurable time and block this driver. The TA sends a blocking confirmation message to RSUs in the vicinity from where the TA receives the last dispute decision. These RSUs forward it to the concerned vehicle which receives and acknowledges the instruction. Alternatively, if a driver's trust reduces to 0.05, the TPD generates a blocking request message which a nearby RSU forwards to the TA. Then the TA blocks this driver and informs the respective TPD via the RSUs in the vicinity of the vehicle.



Figure 2. Functional Diagram of Proposed Framework

# 3.6. RSU Traffic Event Management/Functionality

RSUs always listen to traffic events and share them as necessary based on their severity. RSUs receive regular beacons from vehicles. In response, RSUs send beacons periodically to notify of their roadside existence so that other entities can request services. When RSUs receive an emergency traffic event message from a sender vehicle, they rebroadcast the same towards neighboring vehicles so that oncoming vehicles whose route includes the problematic road may avoid it. RSUs also share certain events with nearby RSUs so that vehicles in a greater region may avoid the problematic road, if appropriate. For some events, RSU will continue to periodically announce it until they receive notification from an official vehicle to confirm the event is resolved. When a traffic event occurs and if the RSU receives an AttendingBY-Voff message from an official vehicle, the RSU can confirm the event has occurred. The RSU continues to announce the traffic event periodically until the reception of a sorted traffic event from the official vehicle. When the event sorted message arrives, the RSU stops rebroadcasting the original traffic event towards oncoming vehicles. Rather it starts broadcasting only the sorted traffic event up to a retransmission limit as well as forwarding this message to nearby RSUs based on the severity of the original traffic event. RSUs rebroadcast and forward traffic incidents to the TA from sender vehicles besides storing traffic incident information until it is resolved. Each local service point, for example, petrol pumps, and parking is registered in advance with the nearby RSU. Whenever any vehicle sends any query seeking information regarding any service, the local RSU sends a reply to the service query containing the information of queried service or it says it has no information if it does not know.

An RSU assigns a fixed amount of reward and sets the punishment for disputed announcements using an Incremental Punishment Policy (IPP). An RSU forwards the decision of a disputed event to the TA. Then TA checks the malicious event count for relevant drivers. If the 3ME condition holds then the TA sends a blocking confirmation message to the RSUs in the vicinity of the last disputed event. After that, these RSUs broadcast this message to the vehicle. Besides these functionalities, an RSU also resolves disputes when untrue attack messages arise which is illustrated in Section 3.7.

#### 3.7. RSU Untrue Message Detection

If an RSU receives conflicting information from a sender and a reporter, it initiates a "collaboration" process to determine the validity of the disputed event. To this end, first, an RSU broadcasts a send-a-reply message to all trusted clarifiers in the vicinity including possible

official vehicles and waits for a timer to expire when the feedback collection is finished, as depicted in Algorithm 3. Notations and symbols for this algorithm are listed in Table 2.

Notation	Meaning
evt <sub>e</sub> -sorted	sorted event <i>evt</i> <sub>e</sub>
untrueHandledList	list of disputed cases which RSU <sub>r</sub> has decision
send-a-reply( <i>evt</i> <sub>e</sub> )	ask $V_{clas}$ to send feedback on $evt_e$
Loc <sub>req</sub> (service <sub>i</sub> ) and Loc <sub>rep</sub> (service <sub>i</sub> )	location of i <sup>th</sup> service query and i <sup>th</sup> service reply
RSU <sub>r{known service}</sub>	known services at RSU <sub>r</sub>
timer <sub>sc</sub>	timer to collect feedbacks by an RSU
reply <sub>off</sub>	reply from V <sub>off</sub>
untrue_id	untrue_attack message id
AttendingBY- $V_{off}(evt_e)$	attending $evt_e$ by a V <sub>off</sub>
rewV, punV	rewarded and punished vehicle

Table 2. Li	st of Notations.
-------------	------------------

Algorithm 3: RSU untrue attack handler

Inpu	Input: untrue attack, feedback message, trust, lists to save events				
Out	Output: initiate feedback collection for a timer, find rewarded/punished vehicle				
1.	while running	29.	Update rewardList and		
2.	case eventType of		punishmentList		
3.	untrue attack: // deals with untrue	30.	forwardMsgtoRSU <sub>s</sub> (decision_untrue)		
	attacks.	31.	if $count(3ME(V_s \text{ or } V_{rep})) \ge 3$		
4.	<b>if</b> unique $untrue(evt_e)$ from $RSU_s/V_s$	32.	Send a Msg <sub>block</sub> (count(3ME(V <sub>s</sub>		
5.	Insert into untrueAddedList		or $V_{rep} \ge 3)$ ) to TA		
6.	<b>if</b> $untrue(evt_e) \in untrueHandledList$	33.	end if		
7.	Return	34.	end if		
8.	else	35.	else		
9.	Insert into untrueHandledList	36.	The feedback is for different RSU <sub>s</sub>		
10.	<b>if</b> $untrue(evt_e)$ from a V <sub>s</sub>	37.	end if		
11.	Broadcast a <i>send-a-reply(evt<sub>e</sub>)</i>	38.	end while		
12.	Start a timer <sub>sc</sub> to collect	39.	<i>decision-of-untrue</i> : // to resolve		
	feedbacks		dispute		
13.	end if	40.	<b>if</b> timer <sub>sc</sub> expires		
14.	end if	41.	<b>if</b> the $untrue(evt_e)$ has a decision		
15.	else	42.	Return		
16.	$RSU_r$ receives an <i>untrue</i> ( <i>evt</i> <sub>e</sub> ) from	43.	else		
	a V <sub>off</sub>	44.	Sum=0		
17.	Call rew-pun-generator(V <sub>off</sub> , V <sub>s</sub> )	45.	case feedbackType of		
18.	end if	46.	Positive: $F_i = 1$		
19.	<i>feedback:</i> // collect all the feedbacks.	47.	Negative: $F_i = -1$		
20.	while (timer <sub>sc</sub> is not expired)	48.	Unsure: $F_i = 0$		
21.	<b>if</b> unique feedback $f_u$ from $V_{cla}$ is for	49.	end case		
	RSU <sub>r</sub>	50.	<b>for</b> each F <sub>i</sub> from feedback vector		
22.	Insert in vector $\langle f_0, f_1, \dots, f_n \rangle$		$<\!\!F_n, T_n\!\!>$		
23.	<b>if</b> $f_u$ is from a $V_{off}$	51.	$Sum += T_i * F_i$		
24.	<b>if</b> $f_u$ is the same as the $V_s$ 's event	52.	end for		
25.	Call rew-pun-gen(V <sub>s</sub> ,V <sub>rep</sub> )	53.	if Sum>0		
26.	else	54.	$V_s$ send true event, $V_{rep}$ send		
27.	Call rew-pun-gen (V <sub>s</sub> ,V <sub>rep</sub> )		false report		
28.	end if	55.	Call rew-pun-generator(V <sub>s</sub> , V <sub>rep</sub> )		

56.	else if (sum<0)	64. <b>if</b> (sum>0 or sum<0	))
57.	$V_s$ send false event, $V_{rep}$ send	65. forwardMsgtoRS	U <sub>s</sub> (untrue_
	true report	dec)	
58.	Call rew-pun-gen ( $V_{rep}$ , $V_s$ )	66. <b>end if</b>	
59.	else	67. Clear the vector <fe< td=""><td>edback&gt; on</td></fe<>	edback> on
60.	Undecided conflict	untrue_id	
61.	Insert attack into	68. end if	
	unresolvedUntrueList	69. end if	
62.	Send an unresolvedUntrue( <i>evt</i> <sub>e</sub> )	70. end case	
	to a $V_{\rm off}$	71. end while	
63.	end if		

Sender(s) and reporter(s) involved in the dispute are not permitted to participate in this clarification process. It is reasonable to consider that there are some trusted vehicles around the event. Also, there may be several malicious vehicles, as considered in [2, 13]. The effect of malicious feedback will be nullified when the true feedback outweighs the malicious feedback while taking a decision. In this framework, feedback can only be generated by trusted clarifiers with trust scores greater than 0.5 and official vehicles. The possible feedback messages are 'YES' or 'NO'. Eligible vehicles that respond, known as clarifiers, reply 'YES' if they had visited the event location recently and confirmed the event or 'NO' if they had visited the event location and did not see the event. In some cases, drivers neither notice the event nor visit the event location in the recent past. These drivers will simply ignore the RSU query. Also, official vehicle feedback is treated as the decider for a dispute which bypasses the collaboration process since collected feedbacks from the trusted clarifiers are not used in forming a decision. When an RSU receives official vehicle feedback in Algorithm 3, it instantly invokes the reward-punishment generator as shown in Algorithm 4.

Algo	rithm 4: rew-pun-gen (rewV, punV)			
Inpu	t: rewarded vehicle, punished vehicle			
Output: send reward/punishment message and blocking message to TA, if required				
1. w	hile running	4.	if (rewV!=V <sub>off</sub> )	
2.	<i>reward/punishment:</i> //estimate	5.	Send the reward_msg(rewV)	
	reward or punishment for disputed	6.	end if	
	event	7.	Send the punishment_msg (punV)	
3.	Store reward(rewV) and	8.	Call forwardMsgtoTA(untrue_dec)	
	punishment(punV) in rewardList and	9.	end while	
	punishmentList			

The RSU dispute resolution mechanism in Algorithm 4 uses these feedbacks to decide the truthfulness of a dispute. Here, the RSU performs a sum of product calculation of the feedback and trust of the clarifiers to decide on the disputed event. For example, suppose a vector of feedback is ('YES', 'YES, 'NO', 'NO', 'YES') which are represented programmatically as (1, 1, -1, -1, 1) and the clarifier's corresponding trust scores are: (0.5, 0.7, 0.65, 0.68, 0.9), then the RSU decides by using Eqn (3). It should be noted that only trusted clarifiers can join the collaboration process. Generally, Eqn (3) can be expressed as in Eqn (4) for n feedbacks collected from n trusted clarifiers, where  $F_i$  is the *i*<sup>th</sup> feedback and  $T_i$  is the *i*<sup>th</sup> clarifier's trust score.

$$Decision = [1*0.5] + [1*0.7] + [-1*0.65] + [-1*0.68] + [1*0.9]$$
(3)

 $Decision = \sum_{i=1}^{n} F_i * T_i \tag{4}$ 

If the outcome is positive, then the RSU decides the sender has disseminated a true event and thus receives an RSU reward; the conflicting reporter(s) receive an RSU punishment. If the outcome is negative, the converse actions are followed. When the decision is reached, the RSU calls the reward-punishment generator, shown in Algorithm 4. During the punishment assessment, an incremental punishment policy (IPP) is applied to influence the future good behaviour of drivers. However, for an unresolved issue when RSU has no feedback data or Decision=0 in Eqn. (4), the RSU stores them in an unresolved dispute list and later may ask an official vehicle to inspect the event location physically and report its findings so that the RSU can take an action on the dispute. It should be noted that if during the collaboration process, any official vehicle receives an RSU message, but they have not visited the disputed event location recently, then they reply with a far-from-event message. However, if the RSU receives a decisive message from an official vehicle, then it always decides on the event using this message and bypasses the collaboration mechanism.

# 4. IMPLEMENTATION

The framework is implemented in Veins 5.0 [29] which is a tightly coupled framework for simulating VANETs comprising the SUMO traffic simulator [30] and OMNeT++ discrete event network simulator [31]. We extend Veins in several respects. First, we create four types of vehicles, namely: "official" vehicles (police, ambulance, and fire service) and regular vehicles. A TA module has been created that registers and blocks drivers. Additionally, the TA unit keeps a driver's most recent reward/punishment history in a driver profile database to facilitate the blocking of malicious drivers and to record incident information (location, timestamp, incident) in an incident database. In addition to this, an RSU internetwork is developed which also connects to the TA unit via wired communication. Inside each RSU, besides event management, a dispute resolution process is implemented to detect untrue/inconsistent attacks.

# 4.1. System Model and Environment

As there are four types of vehicles, we have created four distinct modules in OMNeT++ and the C++ implementation of them according to the functionality specified in Section 3. A TPD module is added to regular vehicles which primarily implements the trust update and access blocking of each driver. The module can exchange messages with the vehicle application layer using an internal connection. When drivers broadcast messages, they send the delay, and the location of the event from the current location of the vehicle identified on the map. Also, the TPD can check the current location of the vehicle and then determine the amount of reward/punishment from an announcement using Algorithm 2. The TPD withholds rewards for a given period to allow disputes to arise via reporters. Also, the TPD disables the transmission of a blocked driver to stop the generation of event announcements; however, whilst in the blocking state a driver still broadcasts beacons. The vehicle application first reads the trust from the TPD. This trust needs to satisfy the associated trust threshold for the message class to proceed with the announcement. The reward varies for activities like beaconing, forwarding, and broadcasting announcements. Vehicles can only obtain a beaconing reward if they are classified as not trusted or *lowly trusted* as defined in Section 3.4. The TPD can also support multiple driver profiles in case different people share a vehicle.



Figure 3. Road Networks Used in the Simulation

The application layer of official vehicles is different than the regular vehicles. Official vehicles respond to RSU queries differently than regular vehicles. Also, when an emergency arises, an RSU gives precedence to their messages over those from regular vehicles. There is a built-in mobility module from Veins that advances all vehicles at regular time steps. The system starts with no vehicles in the terrain model and once a vehicle is added, it remains in the system until the simulation terminates. Once a predetermined number of vehicles enter the system, no more are permitted. The simulation commences by assigning periodic events to specific vehicles and then the resultant data are collected regarding the specific experiment. The framework is simulated using road networks from the Veins default Erlangen city map [29] as in Figure 3a, the Manhattan grid map in Figure 3b, and one alternate route scenario in Figure 3c.

# 4.2. Verification – Thwarting Untrue and Inconsistent Attacks

This framework is verified in the presence of malicious and benign behaviours of trusted vehicles. To this end, this experiment is conducted at least thirty times for 5000 simulation seconds with 10-100 vehicles. One result from these experiments is depicted in Figure 4 to illustrate trust management in the presence of attacks. The horizontal axis represents the simulation time in seconds, and the vertical axis represents the trust score. All vehicles start with a trust score of 0.9. The reward is fixed for a single announcement or RSU interaction which is set to 0.08. and RSU punishments for three untrue announcements are set to 0.1, 0.3, and 0.5 (applying IPP) consecutively. Receivers report an event with a probability of 40% and the event supporting probability P from clarifiers is set to 20%. Vehicle V0 constantly sends messages at 200s intervals (simulation seconds) starting from 100s. Figure 4 records the inconsistent behaviour of V0 with the consistent behaviours of V1 and V2.



Figure 4. Trust Increment/Decrement with Untrue and Inconsistent Attacks

When V0 sends untrue messages, the RSU punishes it by 0.1 and 0.3 consecutively at 220s and 640s. Conversely, when V0 announces trustworthy messages consecutively at 500s, 1100s 1300s, 1500s, 1900s, the TPD adds a reserve reward at 620s, 1220s, 1420s, 1620s, and 2020s; these rewards are withheld for 120s. V0 receives complaints for message announcements at the 700s, 900s, and 1700s. Thus, the TPD does not add any reward for these announcements. After this, the RSU punishes V0 by 0.5 which is shown by a large reduction in the trust score in the 2020s followed by a blocking message which sets its current trust score to 0.05 irrespective of whatever it previously had. In this way, V0 is blocked from network access by the framework. From this simulation result, it can be concluded that the Incremental Punishment Policy (IPP) will demotivate vehicles from attacking repeatedly. The IPP provides a flexible means of punishing and blocking vehicles for their inconsistent behaviours although they may sometimes announce trustworthy messages in between their malicious activities. As vehicles receive higher punishment in each subsequent untrue announcement, vehicles with inconsistent behaviour will be isolated as well. Additionally, we allow only three malicious actions within the simulation timeframe (for example 5000s) from a trusted driver to become blocked to limit further harmful actions. So, in summary, this trace represents an example confirming the system can successfully detect the inconsistent behaviour of a malicious vehicle and punish it accordingly.

# 5. PERFORMANCE EVALUATION

In this section, the evaluation of the proposed approach is investigated. The simulation model described in Section 4 is used in the presence of varying traffic densities. Clarifiers generate varying percentages of malicious and benevolent feedback to classify events as true negative, true positive, false positive, and false negative observational data. Analysis shows the proposed framework can classify events as expected. This means that when there is more benevolent feedback the RSU can classify an event correctly and vice versa.

This set of experiments considers the generation of varying ratios of positive and negative feedback when classifying a disputed event. Moreover, we show the minor impact of vehicle density on the results as RSUs ignore repeated complaints regarding the same event. As expected, when density increases, more vehicles complain about an untrue event. The RSU forwards the first complaint to nearby RSUs which avoids invoking costly concurrent collaboration procedures at other RSUs. Furthermore, the proposed approach is compared against a reputation approach [8] in terms of response time and communication overhead. The response time is the decision time of

receiver vehicles when they receive an event message in the network. A trust management model for a VANET can be considered the most efficient one when receivers can decide about an event in the fastest possible time relative to trust systems where additional computation and communication are required after the arrival of messages. Hence, response time is a good indicator of performance. Another useful metric is the communication overhead since a trust model with lower communication overhead reduces the burden of message transmission and processing. For this reason, a trust model with a lower response time and communication overhead can be regarded as superior to one where these values are higher. Furthermore, these two metrics affect the performance of the network communication, such as channel availability, hence the proposed approach is compared with a reputation approach which suffers from these two factors. Results from the analysis show the proposed framework outperforms the existing one as a receiver vehicle in the proposed framework can decide on the appropriate action without further communication within the VANET. However, the proposed framework requires broadcasting feedback other than the traffic event if a reporter invokes an untrue attack event.

# 5.1. Scenario 1 - Accuracy of the Proposed Framework

#### **5.1.1. Simulation Setup**

This series of simulations have been conducted using one predefined route for both regular and official vehicles on the Erlangen city map from Veins [29]. Selected parameters used for conducting the series of simulations are listed in Table 3.

We repeat each experiment five times to collect trial data for every vehicle density and probability of supporting an event. This sample data is then averaged for analysis. We use the probability P to control the support or denial of events from clarifiers through YES/NO responses. For example, with a probability of P=0, clarifiers always send NO. For P=1, clarifiers always send YES. For probabilities of P=0.2...0.8, clarifiers send YES/NO responses accordingly. In this way, the analysis considers varying ratios of benevolent and malicious feedback. We consider, the senders always announce true events in one set of experiments, whereas, in another set, they always announce untrue events. One or more reporter vehicles may send untrue attacks upon reception of these events which are also randomized with a probability of 0.4.

Selected Parar	neter	Value
Simulation	Simulation area	2.5km X 2.5Km
Parameter	Number of vehicles	[10, 30, 50, 70, 90,
		100]
	Number of RSUs	12
	Speed of vehicles	Max 80 m/s
	Simulation time (seconds)	4000s
	Transmission range	300m
	Warm-up period	700s
	Number of event source	3
	Periodic announcement	At 100s
	Event supporting probability	[0, 0.2, 0.4, 0.6, 0.8, 1]
	Reward withhold timer	120s
	Collaboration timer	120s
	Initial trust	0.8
Attacker	Untrue and Inconsistent attacks	Applies to regular
Model	initiated	vehicles

Table 3. Parameters for Simulation Setup.

The subsequent analysis considers P, the probability of being truthful or not, and D, the vehicle density. The possible RSU judgements are given in Table 4.

	Predicted true	Predicted false
A true event	True Negative (TN)	False Positive (FP)
A false event	False Negative (FN)	True Positive (TP)

Table 4. Results Classification Matrix

#### 5.1.2. Results and Analysis

In Figure 5, the x-axis represents the vehicle density, the y-axis represents the probability of being truthful or not and the z-axis represents the normalized likelihood of classified cases. We define the normalized likelihood of TN/FP classified cases as the ratio of the average number of classified TN/FP cases to the average number of reported events which the RSU classifies as TN/FP using the dispute resolution process. When the vehicle density increases, the number of reporters also increases. However, throughout the simulations, increasing vehicle density is shown to have only a marginal impact on the results as the RSUs ignore repeated complaints concerning the same event from multiple reporters. This is possible as the RSU which received the first complaint concerning an event forwards notification of it immediately to other RSUs in the vicinity to prevent invoking further costly and redundant collaboration procedures.

Figure 5a shows the TN results for a series of simulations where sender vehicles announce only true events. Overall, as P increases, the possibility of classifying TN cases also increases. This means, the framework correctly classifies disputed events if most clarifiers send truthful feedback. As expected, at P=0, there are no TN cases because all clarifiers deny the original event. Alternatively, at P=1, all reported events are TN as all clarifiers only send YES. The TN cases increase rapidly from P=0.4 to P=0.6 as the proportion of received YES responses is sufficient to support the sender announcement at RSUs. Also, the number of TN cases increases with the rise of vehicle density as the increased traffic supports the sender's announcement. Figure 5b shows the FN results of classifying FN cases where sender vehicles announce only untrue events which show a similar trend like TN cases. Conversely, Figure 5c shows the TP chart for a series of simulations when sender vehicles announce only untrue messages. Overall, as P increases, the possibility of classifying TP cases decreases. That means the approach correctly classifies untrue events if most clarifiers rebuff the original, malicious sender message(s). The rapid fall in TP cases noticed between P=0.4 to 0.6 arises when most clarifiers confirm the untruthfulness of the sender's announcement to the RSU query. Here also the number of TP cases decreases more with increasing vehicle density as the increased traffic leads the RSU to receive more YES responses (maliciously supporting the sender announcement). Finally, Figure 5d shows the false FP when the sender announces only trustworthy messages which also shows a similar trend like TP chart.



International Journal of Security, Privacy and Trust Management (IJSPTM) Vol 12, No 1, February 2023

Figure 5. Normalized Likelihood of Classified Cases

Thus, the normalized likelihood of classified TN/FN cases is lower at lower P values and is greater at higher P values than the expected trend. To classify an event as TN/FN, an RSU needs more YES/NO feedback than NO/YES, respectively. The proportion of YES and NO feedback received at an RSU is reflected in the decision which causes the curve to vary nonlinearly with P. Also, the TP/FP curves show a nonlinear relationship for a similar reason.

#### 5.2. Scenario 2 – Comparison with Baseline Approach

#### 5.2.1. Simulation Setup

We implement a baseline approach [8] alongside our scheme in order to compare message flows. We see that the receiver-side trust evaluation approach suffers badly from communication overhead due to trust metric dissemination as receivers are busy with trust verification after the arrival of messages. In [8] the trustworthiness of a sender is decided using one of the following schemes: majority voting, weighted voting by reputation, and highest reputation level. The feedback is collated at the RSU, and the trust score is subsequently interrogated. This set of experiments runs for 800 simulation seconds and is repeated 10 times to obtain the average number of messages exchanged in the presence of 10 to 70 vehicles. An event is introduced deliberately at 400 seconds in both approaches. In approach [8], all vehicles upon observing the event, announce it. Conversely, in the proposed framework the announcement of an event from one vehicle is adequate with receivers relaying it up to 4 hops.

#### 5.2.2. Results and Analysis

In Figure 6, the x-axis represents the number of vehicles present in the simulation, and the y-axis represents the communication overhead for a single event. This framework is compared against the approach in [8] with 30 and 45-second interval timers. It is clear from Figure 6 that the overhead is higher in [8] with both timer durations than in the proposed framework. With a 30s

timer in approach [8], the communication overhead is two, three, and four times higher than the proposed framework when the number of vehicles is 50, 60, and 70, respectively. In most situations, the overhead in the proposed framework is significantly lower than the approach [8], which suffers from a higher overhead due to the need of generating feedback towards RSU for regular reputation updates.



Figure 6. Communication Overhead Comparison

In addition, as expected, the proposed framework is better when we compare its response time against a receiver end-based trust approach. Approach [8] starts a timer for a predefined period i.e., the 30s, to collect additional messages about the same event. When it expires, receivers decide on an event. That is why it suffers from a higher response time. During this time, vehicles may enter a "problematic" road area as they are typically moving fast. On the other hand, the proposed framework quickly decides an event without further communication unless it is disputed. Thus, the proposed framework exhibits a faster response time in comparison with schemes such as [2, 8, 9, 12, 14, 20, 23].

# 6. CONCLUSION

In summary, this research proposes a TPD-based sender-side trust management framework for VANETs. The framework reduces trust metric communications at the expense of equipping every regular vehicle with a TPD. TPD installation is a one-time cost whereas, the circulation of trust metrics is continuous assuming the communication is ongoing. Senders are trusted by default and so response times are reduced at the receivers as trust confirmation is avoided. Additionally, results suggest that the RSUs can successfully resolve any true/false complaints and can detect untrue and inconsistent attacks if the majority of clarifiers send truthful feedback. Furthermore, the framework can be enhanced with additional functionality. For example, a driver profile-based reward/punishment database could be added where historical information concerning driver misdemeanours can tune subsequent rewards and punishments.

#### ACKNOWLEDGEMENTS

This work is supported by the University of Dhaka under the Bangabandhu Overseas Scholarship scheme funded by the People's republic of Bangladesh government for improving quality of education and research.

#### REFERENCES

- [1] Shrikant Tangade, Sunilkumar S. Manvi, "Trust management scheme in VANET: Neighbour communication-based approach," *in Proc. IEEE Int. Conf. on Smart Technol. for Smart Nat.* (*SmartTechCon*)., Bengaluru, India, 2017, pp. 741-744.
- [2] Zhexiong Wei, Fei Richard Yu, Azzedine Boukerche, "Trust based security enhancements for vehicular ad hoc networks," in Proc. of the 4th ACM Int. Symp. on Dev. and Anal. of Intell. Veh. Netw. and Appl. (DIVANet)., Montreal, Canada, 2014, pp. 103-109.
- [3] Shrikant Tangade, Sunilkumar S. Manvi, "CBTM: Cryptography based trust management scheme for secure vehicular communications," in Proc. IEEE 15th Int. Conf. on Control., Autom., Robot. and Vis. (ICARCV)., Singapore, 2018, pp. 325-330.
- [4] Dahiya, Rohan, Frank Jiang, Robin Ram Doss, "A Feedback-Driven Lightweight Reputation Scheme for IoV," in Proc. IEEE 19th Int. Conf. on Trust. Secur. and Priv. in Comput. and Commun. (TrustCom)., Guangzhou, China, 2020, pp. 1060-1068.
- [5] Farhan Ahmad, Virginia NL Franqueira, Asma Adnane, "TEAM: A trust evaluation and management framework in context-enabled vehicular ad-hoc networks," IEEE Access, vol. 6, pp. 28643-28660, May. 2018.
- [6] Tahani Gazdar, Abdelfettah Belghith, Hassan Abutair, "An enhanced distributed trust computing protocol for VANETs," IEEE Access, vol. 6, pp. 380-392, Oct. 2017.
- [7] Nadia Haddadou, Abderrezak Rachedi, Yacine Ghamri-Doudane, "A job market signaling scheme for incentive and trust management in vehicular ad hoc networks," IEEE trans. on Veh. Technol., vol. 64, no. 8, pp.3657-3674. Sept. 2014.
- [8] Ricardo Mühlbauer, João Henrique Kleinschmidt, "Bring your own reputation: A feasible trust system for vehicular ad hoc networks," J. of Sens. and Actuator Netw., vol. 7, no. 3, p.37, Sept. 2018.
- [9] Siri Guleng, Celimuge Wu, Xianfu Chen, Xiaoyan Wang, Tsotumu Yoshinaga, Yusheng Ji, "Decentralized Trust Evaluation in Vehicular Internet of Things," IEEE Access, vol. 7, pp. 15980-15988, Jan. 2019.
- [10] Zhaojun Lu, Gang Qu, Zhenglin Liu, "A survey on recent advances in vehicular network security, trust, and privacy," IEEE Trans. on Intell. Transp. Syst., vol. 20, no. 2, pp.760-776, Apr. 2018.
- [11] Yu-Chih Wei, Yi-Ming Chen, "Adaptive decision making for improving trust establishment in vanet," in Proc IEEE 16th Asia-Pacific Netw. Oper. and Manag. Symp. (APNOMS)., Hsinchu, Taiwan, 2014, pp. 1-4.
- [12] Wenjia Li, Houbing Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," IEEE Trans. on Intell. Transp. Syst., vol. 17, no. 4, pp. 960-969, Nov. 2015.
- [13] Zhe Yang, Kan Yang, Lei Lei, Kan Zheng, Victor C.M. Leung, "Blockchain-based decentralized trust management in vehicular networks," IEEE Internet of Things J., vol. 6, no. 2, pp.1495-1505, May. 2018.
- [14] Seyed Ahmad Soleymani, Abdul Hanan Abdullah, Mahdi Zareei, Mohammad Hossein Anisi, Cesar Vargas-Rosales, Muhammad Khurram Khan, Shidrokh Goudarzi, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," IEEE Access, vol. 5, pp. 15619-15629, Jul. 2017.
- [15] Qin Li, Amizah Malip, Keith M. Martin, Siaw-Lynn Ng, Jie Zhang, "A reputation-based announcement scheme for VANETs," IEEE Trans. on Veh. Technol., vol. 61 no. 9, pp.4095-4108, Nov. 2012.
- [16] Sarah Ali Siddiqui, Adnan Mahmood, Quan Z. Sheng, Hajime Suzuki, Wei Ni, "A time-aware trust management heuristic for the Internet of Vehicles," In 2021 IEEE 20th Int. Conf. on Trust., Secur. and Priv. in Comput. and Commun. (TrustCom)., Shenyang, China, Oct. 2021. pp. 1-8,
- [17] Muhammad Usman Aftab, Mehdi Hussain, Anders Lindgren, Abdul Ghafoor, "Towards a distributed ledger based verifiable trusted protocol for VANET", in 2021 Int. Conf. on Digit. Future and Transp. Technol. (ICoDT2),. Islamabad, Pakistan, May. 2021, pp. 1-6.
- [18] Chaker Abdelaziz Kerrache, Nasreddine Lagraa, Carlos T. Calafate, Abderrahmane Lakas, "TFDD: A trust-based framework for reliable data delivery and DoS defense in VANETs," Veh. Commun., vol. 9, pp. 254-267, Dec. 2017.
- [19] Sashi Gurung, Dan Lin, Anna Squicciarini, Elisa Bertino, "Information-oriented trustworthiness evaluation in vehicular ad-hoc networks," in Proc. Netw. and Syst. Secur. (NSS)., Berlin Heidelberg, Germany, Springer, 2013, pp. 94-108.

- [20] Danda B. Rawat, Gongjun Yan, Bhed B. Bista, Michele C. Weigle, "Trust on the Security of Wireless Vehicular Ad-hoc Networking," Ad Hoc Sens. Wirel. Netw., vol. 24, no. 3/4, pp. 283-305, Jan. 2015.
- [21] Guanghao Wang, Yue Wu, "BIBRM: A bayesian inference-based road message trust model in vehicular ad hoc networks," in Proc 13th Int. Conf. on Trust. Secur. and Priv. in Comput. and Commun. (TrustCom)., Beijing, China, 2014, pp. 481-486.
- [22] Karim Rostamzadeh, Hasen Nicanfar, Narjes Torabi, Sathish Gopalakrishnan, Victor C. M. Leung, "A context-aware trust-based information dissemination framework for vehicular networks," IEEE Internet of Things J., vol. 2, no. 2, pp.121-132, Apr. 2015.
- [23] Ibrahim Abdo Rai, Riaz Ahmed Shaikh, Syed Raheel Hassan, "A hybrid dual-mode trust management scheme for vehicular networks," Int. J. of Distrib. Sens. Netw., vol. 16 no. 7, pp.1550147720939372, Jul. 2020.
- [24] Yao Xuanxia, Zhang Xinlei, Ning Huansheng, Li Pengjian, "Using trust model to ensure reliable data acquisition in VANETs," Ad Hoc Netw., vol. 55, pp.107-118, Feb. 2017.
- [25] Rasha Jamal Atwa, Paola Flocchini, Amiya Nayak, "Risk-based trust evaluation model for VANETs," In 2020 Int. Symp. on Netw., Comput. And Commun. (ISNCC)., Montreal, QC, Canada, Oct. 2020, pp. 1-6.
- [26] Honghao Gao, Can Liu, Yuyu Yin, Yueshen Xu, Yu Li, "A hybrid approach to trust node assessment and management for vanets cooperative data communication: Historical interaction perspective," IEEE Trans. on Intell. Transp. Sys. pp. vol. 23, no. 9, pp. 1-10, Nov. 2021.
- [27] Zhigang Yang, Ruyan Wang, Dapeng Wu, Boran Yang, Puning Zhang, "Blockchain-enabled Trust Management Model for the Internet of Vehicles," IEEE Internet of Things J., Oct. 2021.
- [28] Xiao Chen, Jie Ding, Zhenyu Lu, "A decentralized trust management system for intelligent transportation environments," IEEE Trans. on Int. Transp. Sys., vol. 23, no. 1, pp.558-571, Jan. 2022.
- [29] Car2x, Vehicles in Network Simulation (Veins) v5.0 (Version 5.0), car2x, September 19, 2022. https://veins.car2x.org/download/.
- [30] OpenSim Limited, Objective Modular Network Testbed in C++ (OMNeT++) v5.5.1 (Version 5.5.1), omnetpp.org, September 19, 2022. https://omnetpp.org/software/2019/05/31/omnet-5-5-released.
- [31] The German Aerospace Center, Simulation of Urban Mobility (SUMO) v1.2.0. (Version 1.2.0), sourceforge, September 19, 2022. https://sourceforge.net/projects/sumo/files/sumo/version%201.2.0/.

#### AUTHORS

**Rezvi Shahariar** received his B.Sc. degree in Computer Science from the University of Dhaka, Bangladesh in 2006; and an M.S. degree in Computer Science in 2007 from the same institution. After some time as a lecturer at the University of Asia Pacific, Dhaka, Bangladesh, he is now an Assistant Professor at the Institute of Information Technology, University of Dhaka, whilst pursuing a PhD at Queen Mary University of London. His research interests include wireless network analysis with an emphasis on trust, security in VANETs, and the application of machine learning to security.

A A

**Chris Phillips** (MIEEE) received a BEng. Degree in Telecoms Engineering from Queen Mary, University of London (QMUL) in 1987 followed by a PhD on concurrent discrete event-driven simulation, also from QMUL. He then worked in industry as a hardware and systems engineer with Bell Northern Research, Siemens Roke Manor Research and Nortel Networks, focusing on broadband network protocols, resource management and resilience. In 2000 he returned to QMUL as a Reader. His research focuses on management mechanisms to enable limited resources to be used effectively in uncertain environments.

