2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering

Trust and Reputation in Vehicular Networks: A Smart contract-based approach

Nisha Malik, Priyadarsi Nanda, Xiangjian He, Ren Ping Liu
Faculty of Engineering and IT, University of Technology Sydney, Australia
Email: Nisha.Malik@uts.edu.au, Priyadarsi.Nanda@uts.edu.au,
Xiangjian.He@uts.edu.au, RenPing.Liu@uts.edu.au

***Abstract*—Appending digital signatures and certificates to messages guarantee data integrity and ensure non-repudiation, but do not identify greedy authenticated nodes. Trust evolves if some reputable and trusted node verifies the node, data and evaluates the trustworthiness of the node using an accurate metric. But, even if the verifying party is a trusted centralized party, there is opacity and obscurity in computed reputation rating. The trusted party maps it with the node's identity, but how is it evaluated and what inputs derive the reputation rating remains hidden, thus concealment of transparency leads to privacy. Besides, the malevolent nodes might collude together for defamatory actions against reliable nodes, and eventually bad mouth these nodes or praise malicious nodes collaboratively. Thus, we cannot always assume the fairness of the nodes as the rating they give to any node might not be a fair one. In this paper, we propose a smart contract-based approach to update and query the reputation of nodes, stored and maintained by IPFS distributed storage. The use case particularly deals with an emergency scenario, dealing against colluding attacks. Our scheme is implemented using MATLAB simulation. The results show how smart contracts are capable of accurately identifying trustworthy nodes and record the reputation of a node transparently and immutably.**

Keywords—trust, reputation, vanets, smart contracts, blockchain, IPFS

I. INTRODUCTION

Vehicular ad-hoc networks (VANETs) are wireless, Dedicated Short Range Communications (DSRC) [1] enabled networks where each vehicle works both as a network node and a router, thereby facilitating communication between nearby vehicles and standard roadside units (RSU) [2]. For decades, these networks have encountered multitudinous security issues, such as authentication, privacy, trust and confidentiality [3]. These security issues are attributed to the unique architecture of the network coupled with its dynamic features. For instance, there is intermittent connection and disconnection in the network as nodes travel both in and out of range with one another.

Several works in the past decade have been proposed to achieve authentication of nodes, conditional privacy preservation [4], and securing communication, but very less contribution has been made towards evaluating trustworthiness and data credibility of these participating nodes in vehicular networks. Trust in nodes specially needs to be evaluated when nodes are authenticated.

The self-made decentralized network demands the potentiality to grant each node capability to derive the reputation of every other node. It becomes important that unlike some centralized proposals [5] they are self-sufficient, due to the absence of any reputed (centralized or decentralized entity) to continuously monitor the functional and behavioral capabilities of the nodes.

Over the years, multiple works [6-13] have been done for accurate evaluation of a vehicle's trustworthiness. These rely on various criteria for assessment. A few models draw reputations from data-based trust model, while, others are more focused on entity-based trust model and role-based trust model [6]. But a definitive trust model should drive upon a hybrid approach in order to build a vehicle's trust and it should not solely be dependent on a single trusted party.

Among all these works, some are centralized, while a recent shift has been observed towards decentralization. In centralized trust inference, the centralized trusted source manages information by gathering multiple inputs and producing an output without any transparency with other vehicles. As per our understanding and based on the current drawbacks of the trust model, we believe each vehicle should not just have independent rights to query the reputation of a node, but also, contribute to its evaluation process, with a clear view of what happens behind the scenes. While many decentralized works have been proposed in literatures [10-11], they still rely on a centralized party for disseminating and concealing the reputation information. We aim for complete decentralization and cost effectiveness, so rather than storing the data in the blockchains as proposed in [14,15], we store most of the data in Interplanetary file system (IPFS) network which is cheap and thus allows us to store more data. The former creates an immutable record of transactions that happened amongst the peers, while the later strives to execute a set of instructions upon triggering of an event.

We aim to record the messages transmitted (in the form of transactions), as it can be a proven history to ensure non-repudiation and establish reputation of vehicles based on the validity of messages transmitted and the reputation score maintained. The distributed IPFS [16] network is the storage and retrieval repository used for sharing and storing the reputation of nodes. We analyzed some of the critical requirements of a trust and reputation evaluation model and worked towards addressing those requirements in our proposed model. The proposed trust and reputation model stand by to substantiate impregnable requirements in VANETs with proven resolution to decentralization, transparency, reduced latency, dependency on a centralized trust system and more accurate detection of false messages. The remainder of our paper is organized as follows.

Section II gives a brief discussion on requirement of Trust model. Section III focuses on some of the past works with their pros and cons. Section IV drills down the background concepts used in the proposed model, as elaborated in section V, along with the network entities, assumptions and the working model for our proposed scheme. Section VI, provides theoretical and mathematical analysis of our scheme, followed by conclusion and future directions in section VII.

II. TRUST AND REPUTATION MODEL IN VANETS

The trust model in VANETs must intractably encompass following characteristics in order to have an unimpeachable, robust and reliable outcome.

2324-9013/19/$31.00 ©2019 IEEE
DOI 10.1109/TrustCom/BigDataSE.2019.00015

A. Light, scalable and fast

Highly dynamic topology: The frequently changing topology requires a distributed approach as most traffic conditions require minimum processing time with minimum computation overheads.

Latency issue: There ought to be minimum dependency of information from surrounding mobile nodes or static units (such as the RSUs). The amount of time needed in gathering information to assess the trustworthiness of node is directly proportional to the latency in tackling such situations.

B. Accuracy of Reputation evaluation

An accurate algorithm should consider previous history of the node (identity and its behaviors) and evaluate accordingly.

C. Protection against Collusion attacks/ Fair evaluation

No-bad mouthing: To avoid bad mouthing, a decentralized blockchain network with a quick consensus algorithm should be deployed for reputation evaluation. Before a node adds the reputation, score corresponding to a node, it should be verified and then added.

Collusion attack: Colluded bad-mouthing or commendation should be completely avoided.

D. Independence of node's movements

As the nodes move along different roads, highways and pathways, the deployed trust model should be accurate irrespective of the paths taken by a node. It should be independent of the route taken and should never presume for a specific path for evaluation.

E. Privacy Preservation

In the process of reputation score evaluation, accepting and forwarding messages, the real identity of the nodes should not be revealed.

F. Reputation Evaluation

How trustworthy is the data and the node?

Upon reception of a message, following factors should be considered in evaluating the extent to which a node can be trusted:

1. The reputation of the node sending the information
2. How many nodes are sending the similar information over a period and their corresponding locations?
3. How many other nodes are recommending this node sending the information? How can these nodes be queried for reputation scores?

III. Related works

VANETs are the most vulnerable categories of adhoc networks, considering not just the dynamic nature of VANETs but exposure to various forms of attacks [7]. It is particularly important to identify the insider nodes, which are behaving maliciously as they are most threatening. The study of multiple misbehaviors in [8], classifies them according to different intentions and consequent actions on road by the node. There are selfish node attacks and malicious attacks. Selfish nodes do not cause any active damage to the network, but they intend not to participate in order to save their resources and power. Malicious nodes on the other hand can be involved actively in multiple attacks to fulfil a selfish motive, such as sending fake messages of an emergency to get it cleared for itself. Other malicious attacks causing trust issues include message tampering attack, replay attack.

A trust model for the VANETs should evaluate trustworthiness based upon previously discussed factors. This is true particularly for the latter, as multiple nodes might collude together to send false information, bad-mouth about a reputable node or state good reputation of a malicious node [9].

In [10], authors rely on multiple parameters for trust evaluation, particularly the recommendation from the certificate authority or RSU, to reward both the reporting and recommending nodes. The reputation score is also the result of assessment and analysis by the centralized party. The authors in [11] evaluate the trust of both the data and node based on the message received and data sensed from multiple vehicles. It is primarily used to assess how trustworthy the received information is. Node trust is to assess the trustworthiness of node.

Evaluation of different solutions [6-11] highlights a limitation, that, they are all centralized in nature. A key technology that overcomes this centralization limit and assures users of security and trustworthiness is by blockchain based schemes [12]. On the other hand, in [13] a decentralized data credibility system is proposed. This system selects a vehicle from the group of vehicles which is going to validate the received messages and broadcast the rating block. Other vehicles will then validate the received block using their local knowledge and decide if it should be added to blockchain or not. The ratings received by the vehicles on observation of the traffic are stored in a block and are chained together using their *HASH VALUE*. Then, a temporary center amongst the chain of blocks is selected and broadcasts its rating to others. But, limited by the message's timeliness and vehicle's sensing capacity, the posteriori ratings may have some mistakes.

The proposed system runs on a smart contract embedded in the proposed *RepuChain*. First an off-chain reference of the occurred event is created, and then the reputation contract facilitates to provide feedback regarding the same. The feedback is positive for the occurrence of the event, with rewards top up the node's wallet, and updates the reputation score. But, if fake event is detected, the nodes' reputation degrades along with reduction in wallet amount. The contract just like any other smart contract has their own storage, for data reference. The distributed and decentralized storage for maintaining node's reputation is the Inter- Planetary File System (IPFS) [16]. This is also responsible to maintain copies of node's identity certificates. With multiple nodes rendering requisite data in IPFS [16], there is a reduced latency, dependency, and bandwidth consumption for accessing and storing data.

IV. Background Concepts

Blockchain

The blockchain has attained popularity as a decentralized, peer-to-peer, distributed ledger enabling the storage and distribution of data, finance or other digital assets. Each participant maintains a consistent copy of the transactions that ever occurred in the network, which are added to the chain of blocks after a mutual agreement among these nodes as decided by the chosen consensus [17]. Most blockchains serve rewards to the nodes performing the most crucial operation of block mining, as it requires extensive computations to prove the validity of the block. Fig.1 below

represents the functions associated with a blockchain.

The blockchain based environments promote the trustworthiness of the network and its data, as all peers in the network are involved in verifying the distributed shared data. Authors in [18] further add that blockchains are tamper proof as they are often infeasible to modify due to their distributed nature. Such top-notch security features assure users of confidentiality and privacy of their data.

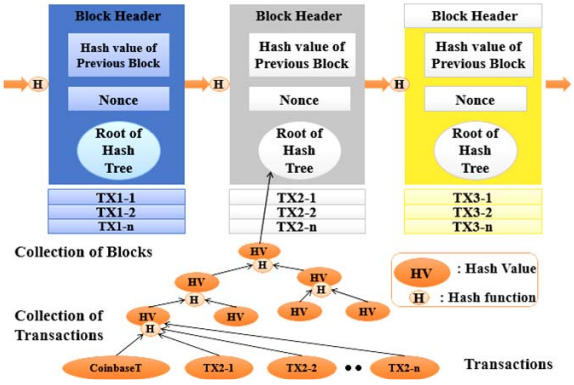

Fig. 1. A typical blockchain [19]

Smart Contracts

They are the piece of code running on top of blockchain, containing a set of rules and conditions for executing the code within upon invoking the contract. The contract beforehand contains the parties that can invoke them and conditions required to execute the code upon meeting the set conditions.

Interplanetary File System (IPFS) storage

The interplanetary file system storms the storage, sharing and database technology with its new distributed, decentralized and 'bit torrent' approach [16]. It plans to overtake the http for retrieval, storage and update of any resource over the internet in future, owing to 'each node a storage node' mechanism. While http continues to rule retrieval, storage and update of resources for nearly about decades now, its disadvantages have really started to matter now more than ever with the overflow of data and an increasing number of users.

Advantages

Avoids the breaking links (no duplicates, but availability): The most popular '404' error occurs due to broken links and unreachability of exact server hosting the data. On another note, if the host decides to remove a certain data which no one else owns becomes an issue for dependent users.

Offline accessibility: Offline has become the new online with multiple distributed systems. In IPFS multiple nodes keep the copies of the files/data, and the nearest nodes can be easily queried instead of reaching out the server every time.

Reduced bandwidth consumption with decentralization: IPFS functioning depends on data content unlike http which is an IP-addressing protocol. For any file that the owner plans to upload for sharing and storage purpose, is stored with the node within the node's directory, while the hash of the document stores the hash of the content along with the node's ID in the Distributed Hash Table (DHT). The DHT stores this key value pair, where key is the node ID and hash of content it provides as shown in Fig 2. Anyone in the network looking for that file can query for the hash of the document, which is provided to the IPFS network is resolved and content is served. This is much simpler as any node which is nearest to the data demanding node can provide the data, unlike Http where there is latency, bandwidth consumption and denial of service sometimes, considering centralization.

Disadvantages

Data deletion issue: If a data owner wishes to delete data permanently, there is no way to ensure this, as multiple nodes might be keeping a copy of the file. If deletion happens before any other node downloads the file, it can be ensured as permanent deletion, but otherwise no.

Access control limitation: The IPFS network is a publicly available network, hence in order to make sure any file is not publicly available, files can be encrypted, which provides access control, but if it needs to be shared with multiple parties again becomes an issue. Solutions such as proxy re-encryption have been proposed to resolve this issue in one of the works, Nucypher [20].

We propose to use private permissioned set of IPFS nodes for registration, where records of user details, time of registration, authority issuing the certificate and if modifications happened, then latest values are recorded via. Smart contracts. The smart contract ensures that, only authorized party modifies the content and shares and update on the blockchain in the form of IPFS hash ID.

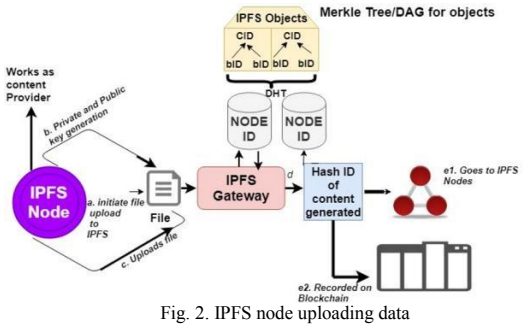

Fig. 2. IPFS node uploading data

V. PROPOSED MODEL

The previous section highlighted various advantages of IPFS, smart contracts and blockchain in VANETs. Based on these advantages, the current study proposes an algorithm, based on the two technologies, that secures the VANET data and infrastructure.

The proposed scheme consists of three phases: registration of vehicles, reputation evaluation (trustworthy message transmission and verification), reputation update and query. The registration of the nodes follows up in a constrained permissioned environment, where a *proof of authority consensus* algorithm is used to add vehicles onto the private IPFS network and blockchain. On road the reputed nodes verify vehicles and store a local copy of recently verified vehicles. This becomes essential as other nodes when wish to query the state of these vehicles do not require to query the IPFS again. The proof of reputation model is used in selecting these reputed nodes from nearby vehicles. There can be more than one of these around other vehicles. The RSUs are also considered reputed here.

The analogy is like owning a higher number of coins in Bitcoin blockchains where the wealthiest nodes validate new blocks and add them to the chain. However, in this case, proof of reputation model is used to select the most 'reputable' to validate new entries to the blockchain.

The process of reputation evaluation and computation is through the deployed smart contract. The flow of interaction between multiple layers is shown in Fig 3.

A. Network Composition

The network is a blockchain enabled network w.r.t registration, verification and reputation management. The entities and data structures building up the network include the following:

Entities

1. Motor Vehicles Division (MVD): Vehicle owners register with details to obtain the Electronic License Plate no. (ELP).

2. Law Enforcement Authority (LEA): LEA provides the blockchain platform for registration and reputation management, but doesn't necessarily control it, as we would discuss. The IPFS nodes in the scheme are privately permissioned set of nodes, which hold the reputation values of other vehicles. The deployment of smart contracts takes place here for registration and reputation. This ensures traceability of malicious nodes.

3.Light IPFS vehicles (LIV): These are regular vehicles, which have registered to the IPFS network, obtained their network parameters, keys and certificate information such as hash ID, asset creation block pointer on the ledger. They are light nodes as they do not have write accessibility to the ledger and the IPFS objects. Due to computation power limitations, they cannot synchronize with the complete blockchain or indulge in mining but can query other IPFS nodes. They are, therefore, the light vehicles.

4. Reputable IPFS vehicles (RIV): The nodes who have established a reputation in the network via proven functionality and honest message delivery as verified by other reputable nodes come under this category. These have more rights as compared to the LIVs and their recommendations are important in the assessment of reputation of other nodes.

5. The edge nodes (RSUs): The RSUs form the computational layer of the network, where mining of transactions takes place to validate the transactions and add them to the block. Their position in the network layer and functionality comes from their increased computation power and resources as required by the miners.

Data Structures

1. Registration smart contract: This is included in the registration and authentication ledger, which ensures only valid and authorized users make a new entry, update an old entry with any changes and generate new addresses with these modifications. The transactions are verified by the private permissioned nodes part of MVD and LEA.

2. Reputation smart contract: This can be initiated by the LIVs to put their reputation scores in their respective IPFS objects w.r.t surrounding recommendations and truth scores. The reputation of each node is an IPFS object, where any modification is subject to positive reviews from neighboring nodes and the multi-signature transaction generated by the node and a minimum of RIVs.

3. The Ledger-Repuchain: This ledger is solely for the purpose of vehicle management. The nature of the ledger is private and only for vehicular environment. The consensus running on the ledger includes both proof-of-authority and proof-of-reputation.

4. Off-Chain event information storage: The emergency event information and details of informing vehicles is stored here.

5. Reputation review storage: This stores the different reviews of the vehicles w.r.t to any event reference information. It is used to evaluate the final rating of vehicles after analyzing the reviews.

6. RepuWallet: This is user's wallet to manage the public private keys, reputation score and reward points.

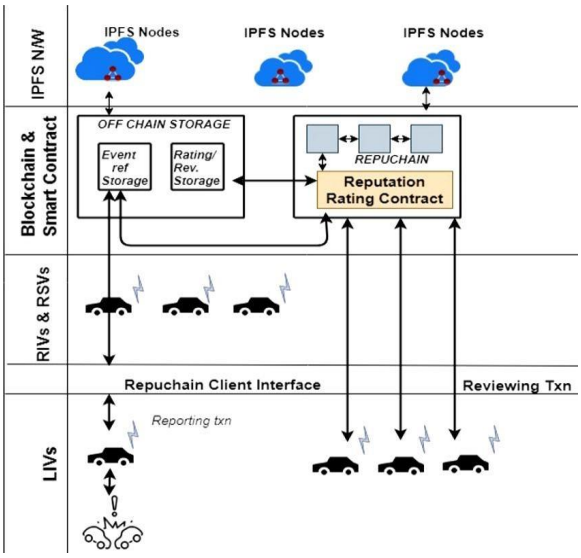

Fig. 3. The IPFS, Blockchain and Smart Contract in VANET

B. Assumptions

1. *LEA* and *MVD* are assumed to be trusted parties as they are maintaining the set of private IPFS nodes and ensuring the traceability in case of disputes.

2. A second assumption made which is, RSUs are the demarcation point for different networks, that is, and only vehicles that are in the proximity of a given RSU can form a network. This allows easier control of blockchain activity and privacy preservation. Various notations for the model are given in Table 1 below:

Notation	Meaning
E_{ID}	Event ID- depends for type of event *01-Accident occurrence* *02-Collision warning* *03-Road blocked*
PID_a	Pseudo ID of a^{th} vehicle
Node ID	Hash of actual identity of a vehicle
RSU_i	i^{th} RSU with which a vehicle is associated
l_m	Location coordinates as detected by m^{th} entity
t_n	Timestamp as recorded by n^{th} entity.
RS_a	Reputation score of a^{th} entity
DS_a	Digital Signature of a^{th} entity
ERN	Event Reference as generated by the vehicle, using location and associated RSU
RS_a	*Reputation of node with PID_a, computed as H(NodeID||accumulated_reputationscore)*
TRS	Threshold Reputation Score
H ()	Sha-256 hash function
Txn_i	i^{th} transaction generated and broadcasted by a vehicle.

Table 1: Notations

VI. REPUCHAIN

Use Case- Emergency Scenario

RIVs ensure decentralized trust and transparency among the untrusted vehicles. They are the vehicles on road, but with a high reputation score than neighboring nodes. They are selected based upon the number of reputation points gained over a period:

Message Type 1 (MT_1): These are regular messages which a vehicle would broadcast (in case it loses control due to internal flaws, or bad driving sense) to the neighboring vehicles. These include collision warnings, sudden brakes applied.

Message Type 2 (MT_2): This is when vehicles spots and emergency and proceeds to broadcasts to the neighboring and probable far away vehicles for caution. This can be an accident, road damage, fallen tree or construction on road, etc.

We are only considering the message broadcasts and not investigating the path traversal of a node to identify if a malicious node delivers or drops a packet on its path, as investigated in [21]. Authors in [21], proposed *watchdog* and *path rater*, in order to identify and remove malicious nodes, who are part of the network but intentionally silent and not forwarding packets. Our reputation check only works for MT_1 and MT_2 message broadcasts for evaluating trust, but not detecting or isolating nodes which aren't forwarding packets.

Vehicles broadcast the above kinds of messages as per the situation. But, it is the neighbors who verify their authenticity and upload a positive or negative score corresponding to their PIDs. Based upon how many truly verified messages have been received for a vehicle, reputation points are evaluated for the RIVs or LIVs.

Each vehicle has a reputation object in the IPFS, which contains relevant information. The link to the object is stored in the *RepuChain*. While content in the object contains the node ID and its reputation score; it is stored as it is in the DHT, the link is the hash representation of this object stored in the blockchain and used to retrieve the actual object. When vehicles receive a message, they query the IPFS nodes. As the LIVs also have rights to download a copy of any vehicle's reputation object, they can easily provide the copy to neighboring vehicles. The vehicles can match the hash to verify if data has been tampered. In this way the vehicles can fairly decide on the reputation of a node without falling for bad-mouthing or colluding attack. Fig. 4 demonstrates the vehicular node as a repuchain client, and various interacting modules.

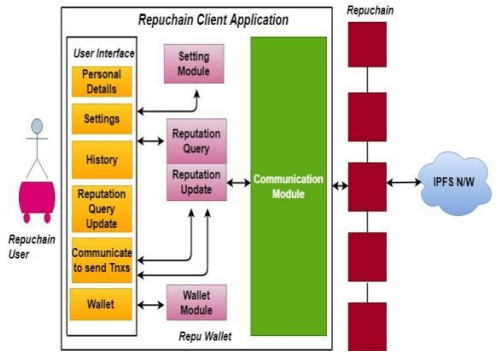

Fig. 4. Vehicular node – The *Repuchain* client

A. Emergency Detected, reported and recorded

A vehicle V_a detects an emergency event E_{ID}, $MT_{1/2}$ and broadcasts an emergency transaction to the emergency contract with the transaction, $\mathbf{Txn_1}$= *{PID_a, E_{ID}, MT_1, RSU_i, l_m, t_n, RS_a, DS_a}*. The nearest RIVs and RSUs work towards validating the transaction based upon two criteria –

-vehicle's location details 'l_a' in the past time 't_a' and
-querying neighboring nodes to verify reputation score and if any recent messages received regarding the same event.
-the location as mentioned is associated with the RSU_i near the event. V_a's association is stored with the other RSUs for some stipulated time.

Algorithm I Emergency event generation and record

Inputs: PID_a, location l_i, current RSU association, RSU_a of the witnessing vehicle. There can be several witnessing and reporting vehicles.
Outcome: E_{ID}, and *ERN* generation
Update: Event reference storage updated

1. **for** each vehicle reporting occurrence of an alarming situation **do**
2. PID_a broadcasts Txn_1 with all the information
3. Txn_1 received by nearest *RSUs* and *RIVs*
4. **if** (location of reported event == vehicle's last
5. location && last RSU associated can verify location, at time '*t*')
6. **then** validate Txn_1 and generate block with new *ERN*
7. **else** reject *Txn1* with warning to reporting vehicle with a probable
8. fake event.
9. **Store** *ERN* in Event reference storage
10. (**key** = *blockhashID*, **value** = *ERN*)
11. **Broadcast** block with the *ERN* to the nearest vehicles
12. **end if**
13. **end for**
14. **EXIT;**

The location match was the only possibility of verification of the vehicle's message with the reputation score topping trustworthiness of the source. The RSUs and RIVs query the IPFS network for the reputation score of the vehicle with its PID_a as the search value, and the nearest available node with the block reverting this key, value pair. If $RS_a > TRS$, the message is accepted, else to avoid taking any chances of true message, it is not rejected but added to the cache and more messages regarding the event are waited for a time interval. After validation, block is broadcasted with the event reference details.

The event reference is important as nodes use this to submit a rating later to the *reputation contract.*
The transaction $\mathbf{Txn_2}$ = *{Txn_{1ID}, ERN, RIVi /RSUi, DSi}*, which generates the event reference number, *ERN* is generated with the Txn_1 as input and output includes event reference and details of RIV or RSU whoever verifies and generates the reference number for taking reviews and feedback regarding the same. The nodes accept and take actions according to the event.

The event reference should remain the same, even if multiple vehicles report the event. For this, the nearest RSU association of the event is used to computer event reference number. When multiple vehicles report the same event, the event reference still resolves to same hash as **ERN** = *(E_{ID}||RSU_i)*. The *ERN* should keep track of how many vehicles have reported the same event. For Multiple RSUs coming in the region of the event occurred, **ERN** = *(E_{ID}||RSU_i||RSU_j)*. The block formation process is as depicted in Fig 5**.** the event reference storage is an off-chain storage, which only costs when writes are performed, but not during reads. The process is explained in algorithm I.

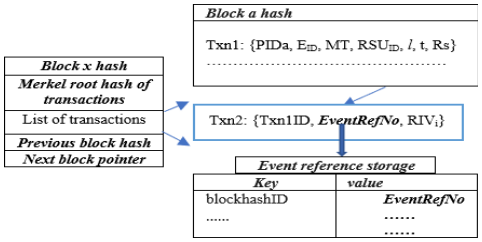

Fig. 5. Transaction updating Event reference storage

B. Reputation Update- Disseminating feedback

Now, vehicles are required to give feedback regarding this event, and if the feedback falls in the majority category and matches the location of the node, reward points are given in the *RepuWallet.* The steps are as given in algorithm II.

The feedback is waited for reputation evaluation. The feedbacks invoke the reputation contract, which can only be invoked by vehicle with a good reputation score. The feedback transaction, **Txn3**: *{PID_b, E_{ID}, ERN, l_x, t_y, RSU_j, 'feedback:1, -1', RS_b, DS_b}* goes to the reputation smart contract, which validates the inputs, such as RS_b, location l_x, and once verified by the RSUs and RIVs, the smart contract updates the reputation score for V_a.

Here, l_x = *current vehicle's location,* RSU_j= *the same RSU* which falls in region of event.

Now, for a positive feedback and event recognition, V_b is rewarded as well with an update of reputation score. The feedbacks are accumulated in the reputation contract storage. The contract utilizes these to update the reputation object of the node in IPFS.

Algorithm II Reputation Update Algorithm

Inputs: PID_a of the reporting vehicle, N user reviews and ratings {R_a, R_b, R_c…. R_n}, E_{ID}, ERN, location l_i, l_j, l_k, l_t of N users, last RSU association of these vehicles {RSU_1, RSU_2, RSU_3……RSU_N}and reputation scores {RS_a, RS_b, RS_c, RS_d…… RS_n}

Outcome: Ratings accepted or rejected based upon location match, reputation score verification and transaction validation.

Updates: -Updated reputation Object of PID_a
-Updates reputation objects of reviewing vehicles.

```
1.  for each vehicle disseminating the review Txns to the Reputation
    Contract on RepuChain corresponding to an EID and ERN, do
2.      If (EID && ERN == valid && location li of reviewing vehicle in
3.  range)
4.          then Txn -> validated by RSUs and RIVs && Contract function
5.  1 executed;
6.
7.      else if (only PID is truly identified, but other parameters not found)
            then Contract function 2 is invoked;
            // Vehicles punished with bad mouthing and colluding attack
        warning.
        else
            Txn rejected
8.  Contract function1 {  // invoked by a valid vehicle with valid inputs
9.          RS of reviewing vehicle = previous score + 1; {
10.         if reviewing vehicles rated reported vehicles truthful
11.            then RS of reporting vehicles vehicle = previous score +1;
12.         else
13.            RS of reporting vehicles = previous score -1;}
14.         }
15.         endif
16. Contract function2 { //invoked by identified, but with malicious
17. intention
18.         RS of invoking vehicle = previous score -1;
19.   }
20.   end if
21. reputation object is modified in IPFS and new hash value is generated
22. new block broadcasted which contains updated reputation scores of the
23. reporting and reviewing vehicles
24. end for
    EXIT;
```

C. Querying Reputation under ordinary situation

If the surrounding vehicles are willing to find out the reputation of a node from a message received (whether alarming or not), then IPFS provides minimum amount of response time owing to its distributed nature. Unlike in a centralized reputation evaluation and management environment, the nodes' request is routed to the nearest nodes first, and if they have downloaded the copy of the node in question, its reputation score can be forwarded.

IPFS works with an incentivizing and motivating environment for the nodes. The nodes providing the reputation score of one node, can supply some other verification information, or any other node's reputation score in return.Vehicles supplying blocks of data to other nodes are rewarded by the network, which motivates the network to share as much as possible.

Algorithm III Query Reputation Score under normal circumstances

Description: If vehicles V_a receives message from V_b, which might be suspicious, and V_a wants to query reputation score of V_b

Inputs: PID_a, PID_b, RS_a, RS_b
Output: Reputation score of vehicles V_b, RS_b

```
1.  if vehicles Va receives a probable suspicious message from Vb
2.  then Va broadcasts a query messages to nearby nodes,
3.       QM = {PIDa, RSa, RSb,' query for RSb', PIDb}
4.       if nearby vehicles possess block with RSb, then respond with RM
5.            RM = {PIDi, RSi, RSb, blockhash}, blockhash is the block with
6.                 updated reputation score in repuchain.
7.       else query the actual IPFS network for updated reputation object hash.
8.       end if
9.  end if
10.    if RSb > TRS
11.       accept message
12.    else
13.       reject message.
14.    end if
15. end if
16. EXIT;
```

VII. Model Analysis

This section discusses the viability of the proposed model in the dynamic VANET environment. In this section, we present our model analysis and carry out security analysis through implementation of our proposed scheme in order to determine trustworthy nodes and also, how well the scheme performs in a malicious environment.

Theorem: A centralized party cannot tamper the rating of a node.

Proof: The rating of a node is the hash of the file content holding the reputation of the node, *H (NodeID|| accumulated_reputationscore)*. If a centralized party tampers the hash, no querying node can reach the actual value, and would keep discarding messages from such a node. But, since the node itself serves as the content provider and all the modification/deletion transactions are recorded on repuchain, no centralized node would want to lose its reputation by data poisoning.

Theorem: The reputation score is available to individual nodes, when queried. There is no centralized dependency.

Proof: Without any latency and bandwidth consumption, the nearest nodes are contacted first followed by the permanent nodes. The delay hardly varies in the round-trip time for fetching results.

Theorem: The process of updating a node's reputation is fair and transparent to avoid questioning, and completely prevent colluding or bad-mouthing.

Proof: As discussed in the scheme, each update, and modification is a transaction on the chain. Though, older reputation objects might not be on-chain to preserve space

and promote scalability, but then they can always be traced for an audit trail.

Theorem: The system works with minimum computation and overhead.
Proof: In a blockchain based decentralized environment, validation by the network does not rely on a heavy consensus such as proof-of-work, instead we used proof-of- authority, and proof-of-reputation or PBFT as discussed in [18].

A. Security analysis:

The model is secured against message integrity, data poisoning and DDOS attack.

Data Integrity: Both blockchain and IPFS strive to achieve data integrity and immutability because of the hash values and pointers safeguarding the content value in IPFS and the blockchain respectively. Uploading the reputation score involves the input of multiple reviews, and smart contract disables manual inputs. The hash value of a node's reputation object is what represents the content, and the file is maintained with private permissioned IPFS nodes. The multihash ID of the content (serving as a pointer to the actual content), is stored on blockchain.

Data Poisoning: Multiple vehicles generate review transactions to provide feedback regarding an event occurred. All these are accumulated after verification in the smart contract storage, to calculate the reputation score of the reporting vehicle. Given the pre-coded contract running on If-Then-That logic, it is hard to tamper the reviews or modify the reputation of a node. Also, each transaction as inputs to the smart contracts for event recording or reputation calculation are included in validated block, for transparency.

Data Availability:
The decentralized IPFS DHT storage network as well as blockchain render a transparent, peer-to-peer, immutable and tamper-proof storage. The IPFS storage for files and corresponding pointers to the files in blockchain, facilitate peer-to-peer rendering of data. There are two kinds of data providers in IPFS, one that are temporary, and have downloaded a copy of the file, which is served upon query if the cache isn't cleared out. The other kind of providers are the permanent ones, which always host that data as they are the ones owning and providing the data. Due to the multiple copies of data circulating in the network, DDOS becomes impossible.
When the update reputation transaction is created, transaction validation, the smart contract execution, and updating of the reputation object in IPFS takes around 0.050 seconds, whereas querying of the score takes around 0.040 seconds.

Traceability: Each transaction is recorded in the blockchain and the smart contract storage evidently sourcing the provenance of the transaction or any update caused by the execution of the smart contract. The transactions are recorded with timestamp, while the smart contracts for the reputation evaluation keep track of the latest update, time of update, sourcing transactions causing the update. All of these ensure traceability in case of any denial, thus providing non-repudiation.

B. Latency

The total delay in the network is a sum of multiple message validations, verifications, processing and finally cryptographic operations.
Reputation score update:
Txn$_1$= *{PID_a, E_{ID}, MT, RSU_i, l_m, t_n, RS_a, DS_a}*
The complete process of validation of this transaction after reception by the RSU_j and RIV_k, depends on following three steps:

a. Verification of RS_a by querying the other LIVs or RIVs. The query contains key value as hash of the current reputation object file in IPFS and expects the value in return.
b. Then computation of the ERN is done by the RSU or RIV, as $(E_{ID}||RSU_i)$ The OR operation requires minimal computation. The block after validation is generated with the ERN.
c. Now, the smart contracts take time to verify transactions received for feedback and uses the reviews to update the reputation object of the node. $Total\_latency=Txn_1\_validation+Txn_2\_validation+Txn_3\_validation$

Reputation score query
The query time totally depends on the number of nodes hopped upon to acquire the required value corresponding to the object hash asked for.
1. Best Case: Node is just in the next hop with a copy in cache.
2. Worst case: The copy is with none of the nearby temporary IPFS nodes, but instead needs to be downloaded from the IPFS network. This takes time depending on the file size.

C. Implementation and Results

To evaluate the successful working of the reputation evaluation and update algorithm, the functionality of 4 vehicles is recorded and evaluated using MATLAB, analyzed over a period of 90 hours which also included broadcast of fake messages created manually by giving wrong input value for the current location or the emergency location or both. The graph in Fig 5 show how the network prevent colluding attacks, as to how the contract ensures that the rating is evaluated based on true scores as sent by the neighboring vehicles.
Vehicle A participates benevolently in the successful achievement of the network motive. As it disseminates true messages over the period, its reputation score rises in an exponential manner. The increase in reputation score is directly proportional to the rewards in the *repuwallet*. For vehicles B, the graph shows a decrement in reputation score after 40-50 hours as fake messages were broadcasted by the node. Vehicles C has not been actively participating to disseminate any true or fake messages and has been a quite spectator, hence a constant nature of his graph. But, for Vehicle D, there has been a rise initially in the score, with a continuous fall after proof of involvement in fake message broadcast, which later improves when the vehicle takes warnings seriously.
The graph in Fig 6 shows how the increasing number of malicious vehicles can affect the functionality of network as with more fake messages coming from a location and associated RSU can verify, it becomes hard to discard such messages. But, when numbers of malicious vehicles are below 40%, it is still easier to distinguish between malicious

and trustworthy nodes. Here, a very important role is played by the TRS. It is important to set an accurate value for TRS, so that fake and true messages can easily be distinguished.

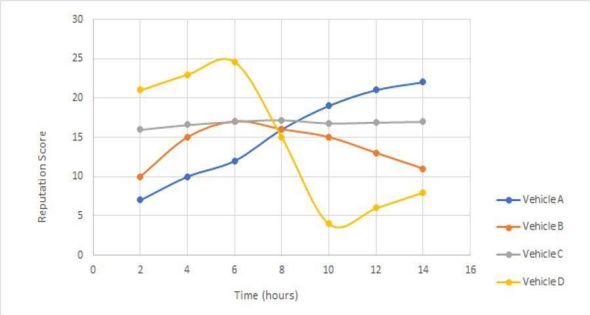

Fig. 5. Reputation score update over a period

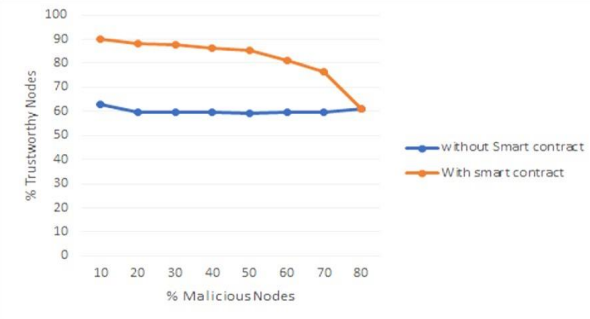

Fig. 6. Trustworthy nodes detected with increasing malicious nodes

VIII. Conclusion and Future Works

In this paper we evaluate the schemes providing trust in the nodes and the system claiming to provide credible reputation score. Limitations in the existent centralized PKI framework, as well as some of the decentralized works have been identified and as a result, a blockchain and smart contract-based solution is envisaged as the best solution. The framework provides decentralization, transparency, immutability and peer-to-peer availability and consistency of every user's reputation value. We evaluated the proposed scheme both from ordinary and alarming situation perspective. The nodes have their pseudo IDs obtained after registration, which provide privacy to some extent, but they do not prevent location tracking. In future, we plan to work where, the nodes supplying the rating should be anonymous and should be able to generate dynamic identities for themselves. Also, we would like to explore the use of multiple platforms as many have evolved over the past two years with fast and reliable consensus mechanisms with less transaction costs. The project moves forward to explore EOS, Stellar, Hyperledger, IBM Watson IoT platform, among many others.

References

[1] Kenney, J.B., 2011. Dedicated short-range communications (DSRC) standards in the United States. Proceedings of the IEEE, 99(7), pp.1162-1182.

[2] Malik, N., Puthal, D. and Nanda, P., 2017, December. An Overview of Security Challenges in Vehicular Ad-Hoc Networks. In *2017 International Conference on Information Technology (ICIT)* (pp. 208-213). IEEE.

[3] Lu, Z., Qu, G. and Liu, Z., 2018. A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems*, (99), pp.1-17.

[4] Malik, N., Nanda, P., Arora, A., He, X. and Puthal, D., 2018, August. Blockchain Based Secured Identity Authentication and Expeditious Revocation Framework for Vehicular Networks. In 2018 17th IEEE International Conference on Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 674-679). IEEE.

[5] (https://www.its.dot.gov/factsheets/pdf/CV_SCMS.pdf)

[6] Zhang, J., 2011, March. A survey on trust management for vanets. In 2011 IEEE International Conference on Advanced Information Networking and Applications (pp. 105-112). IEEE.

[7] Raya, M. and Hubaux, J.P., 2005, November. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks* (pp. 11-21). ACM.

[8] P.-W. Yau and C. J. Mitchell, "Security vulnerabilities in ad hoc networks," in Proc. 7th Int. Symp. Commun. Theory Appl., 2003, pp. 99–104

[9] S. Buchegger and J.-Y. L. Boudec, "The effect of rumor spreading in reputation systems for mobile ad-hoc networks," in Proc. WiOpt, Model. Mobile, Ad Hoc Netw., 2003, pp. 131–140.

[10] Mármol, F.G. and Pérez, G.M., 2012. TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks. *Journal of network and computer applications*, *35*(3), pp.934-941.

[11] Li, W. and Song, H., 2016. ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, *17*(4), pp.960- 969.

[12] Singh, M. and Kim, S. (2018). Branch based blockchain technology in intelligent vehicle. Computer Networks, 145, pp.219-231.

[13] Yang, Z., Zheng, K., Yang, K. and Leung, V.C., 2017, October. A blockchain-based reputation system for data credibility assessment in vehicular networks. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (pp. 1-5). IEEE.

[14] Lu, Z., Wang, Q., Qu, G. and Liu, Z., 2018, August. Bars: a blockchain-based anonymous reputation system for trust management in vanets. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 98-103). IEEE.

[15] Lu, Z., Liu, W., Wang, Q., Qu, G. and Liu, Z., 2018. A privacy-preserving trust model based on blockchain for vanets. *IEEE Access*, *6*, pp.45655-45664.

[16] Benet, J., 2014. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561.

[17] Puthal, D., Malik, N., Mohanty, S.P., Kougianos, E. and Yang, C., 2018. The blockchain as a decentralized security framework [future directions]. IEEE Consumer Electronics Magazine, 7(2), pp.18-21.

[18] Minoli, D. and Occhiogrosso, B., 2018. Blockchain mechanisms for IoT security. Internet of Things, 1, pp.1-13.

[19] Puthal, D., Malik, N., Mohanty, S.P., Kougianos, E. and Das, G., 2018. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consumer Electronics Magazine*, *7*(4), pp.6-14.

[20] https://www.nucypher.com/

[21] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proc. ACM 6th Annu. Int. Conf. MobiCom Netw., Boston, MA, USA, 2000, pp. 255–265.