Journal of Parallel and Distributed Computing 152 (2021) 144–156

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

A scalable blockchain based trust management in VANET routing protocol☆

Sowmya Kudva [a], Shahriar Badsha [a,*], Shamik Sengupta [a], Hung La [a], Ibrahim Khalil [b], Mohammed Atiquzzaman [c]

[a] University of Nevada, Reno, USA
[b] RMIT University, Melbourne, Australia
[c] University of Oklahoma, Norman, USA

ARTICLE INFO

Article history:
Received 29 June 2020
Received in revised form 1 January 2021
Accepted 26 February 2021
Available online 12 March 2021

Keywords:
Blockchain
VANET
Distributed system
Routing protocol

ABSTRACT

Critical event information dissemination has been proliferating on VANET allowing road safety via connected vehicular communications. Despite the prospect of promising applications in vehicular networks, it faces unresolved challenges that hold the capability to slow down network performance upon deployment, especially in terms of security. Particularly, insider attacks such as Blackhole attacks that are carried out against VANET systems can disrupt the networks' average performance and prevent communication between vehicles entirely. Many state-of-the-art solutions have been proposed to detect and eliminate such nodes based on reputation systems and broadcast routing. However, if the network consists of multiple malicious nodes, the message dissemination could fail due to broadcast message tampering attack or packet dropping. In this study, we explore to answer the question of *"can we improve the insider attacks mitigation in VANET by enhancing the trust in the network system so that the possibility of successful attacks can be reduced?"*. To answer this question, in this paper, we present the blockchain-based decentralized trust score framework for the participating nodes to detect and blacklist insider attackers in VANET proactively. We propose a two-level detection system, in which at the first level, neighboring nodes calculate the trust individually. In the second level, a consortium blockchain-based system with authorized Road Side Units (RSUs) as validators, aggregate trust scores for vehicular nodes. Then, based on trust scores reported by the neighboring nodes, the blacklist node tables are dynamically modified. The experimental analysis shows that the proposed system is efficient and scalable in terms of the network's practical size. Finally, we also present evidence that the proposed system improves the VANET performance by mitigating and blacklisting insider attack launching nodes.

1. Introduction

Vehicular Ad-hoc Networks (VANETs) technology is gaining significant attention from the research for the reason that it is a promising technology to improve the Intelligent Transportation System (ITSs) [10,47] via Vehicle-to-Everything (V2X) communications [8]. V2X enables information exchange between vehicles and its surroundings to support safety message dissemination [43], traffic management applications such as traffic congestion information [23], dynamic route planning [26], commercial applications such as content distribution [25], gaming, and entertainment [40] and ride sharing [19]. These applications function mainly based on a vehicle sensor's ability to intelligently perceive the conditions around it and disseminate messages via inter-vehicular communication to bring significant improvements in city road traffic.

Although VANET applications make driving a better experience, it lags in actual deployment and its extensive usage [11] due to numerous unresolved security threats. VANETs are exposed to increased risk of being a target for various types of attacks, as security by design has not been considered. Hence the system interfaces and routing protocols have vulnerabilities that hinder the application availability. Several researchers have put their efforts into their work that broadly covers various types of attacks. Various schemes have been designed and developed so far to provide security solutions to attacks in VANETs [21]. Most of the existing security solutions for VANETs use improved hardware components [6,42] and enhanced software components such as certificates [31], public key infrastructure (PKI) schemes to authenticate vehicles, which contain the digital signature [30].

☆ Submitted to Information-Sciences Special Issue.

* Corresponding author.
E-mail addresses: skudva@nevada.unr.edu (S. Kudva), sbadsha@unr.edu (S. Badsha), ssengupta@unr.edu (S. Sengupta), hla@unr.edu (H. La), ibrahim.khalil@rmit.edu.au (I. Khalil), atiq@ou.edu (M. Atiquzzaman).

https://doi.org/10.1016/j.jpdc.2021.02.024
0743-7315/

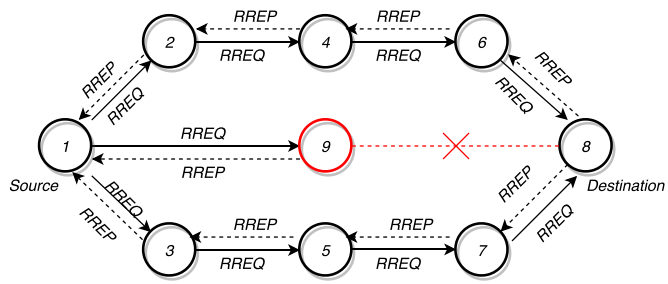


Fig. 1. Balckhole attack in AODV protocol.

1.1. Motivation

In V2X communication-based applications, vehicles highly rely on routing protocols that propagate the information. A routing protocol determines the path for packet transmission from source to destination. Regardless of the application considered, the underlying routing mechanism at the network layer is crucial and significantly affects the applications' overall performance. Out of the many routing algorithms, Ad-hoc On-Demand Vector (AODV) [28] is an extensively adopted reactive routing protocol in a dynamically changing network such as VANET [13,39]. Additionally, less memory consumption for processing makes AODV the best fit for resource-limited vehicular nodes.

In AODV, whenever a node wants to communicate with other nodes, a route discovery process is initiated. AODV protocol uses three control messages that are Route Request (RREQ), Route Reply (RREP), and Route Error (RERR), as shown in Fig. 1. RREQ packets are broadcast to the nodes in the network by the source to find a path. All the other nodes that receive the RREQ packet keep transmitting them until they find a fresh enough route to the destination. On receiving RREQ, if the node is the destination or if the intermediate node has a new route to the destination, it sends the RREP packet back to the source. The Hop count of every node increases by one on receipt of RREQ message, and route entry is updated with new data by intermediate nodes on receipt of RREP messages. However, if the link is broken between two nodes, then the RRER message is sent back to the source node via the reverse path. Each node on receiving RRER invalidates the route in their table to an unreachable destination. A node increases its destination sequence number each time a new RREQ, RREP messages are sent. Destination Sequence Number (DSN) is a 32-bit integer associated with every route and is used to decide the freshness of a particular route. The larger the sequence number, the fresher is the route.

Although AODV has been around for quite some time, few security issues make it vulnerable to various attacks. An internal malicious node makes use of the AODV routing protocol's vulnerability to advertise itself for having the shortest path to the destination node irrespective of its routing table entry. It intercepts the passing through packets and drops all the packets transmitted from the source node, causing a disconnect in the network. This attack is known as a blackhole attack [32]. In Fig. 1, node 9 represents a member launching a blackhole attack. This attack can be perceived as a denial-of-service attack by a node or a router that either refuses to participate in the network or drops the data packets instead of transmitting them. This is an extremely dangerous attack as the data packets containing important information is lost permanently during its transmission to the destination. This kind of attack relates to the attacker that is authenticated in the network and knows best about the network, making it easy to disrupt the network's normal behavior. The situation worsens when multiple blackhole nodes exist in the network. Consequently, the network may become unavailable and may lead to crashes and congestion in road traffic.

To achieve a more efficient packet forwarding process and to mitigate packet drop attacks in VANETs, trust management models are usually adopted against inside attackers. Trust-based solutions help to detect selfish nodes that act as blackholes in the network. Authors of [17] propose computing a distrust level for every neighbor acting as a blackhole through a watchdog technique. This distrust level will be sent to the cluster head and, in turn, delivered to a trusted third party, which revokes the attacker's certificate. Nevertheless, untrustworthy intermediate vehicles can modify the message containing trust values. They can even generate and insert a new trust value in the VANET causing broadcast tampering attacks [33]. Furthermore, a certificate revoking third party may also get compromised.

In this context, various initiatives have been recently launched to investigate the suitability of trustless and decentralized ledger technology (DLT), also known as blockchain, in securing vehicles' trust scores. Blockchain (BC) is an emerging decentralized and distributed computing paradigm that underpins the Bitcoin cryptocurrency [24], which provides immutability and security in Peer-to-Peer (P2P) networks. Blockchain makes it possible to transparently store and transmit information securely and without a third party control point in vehicular networks. As the core of any blockchain-based system, a consensus algorithm directly affects the performance of the blockchain system [34] in terms of transaction confirmation delays. Public blockchain incorporating the classic Proof of Work (PoW) [27] might not be the best way to contribute to the vehicular network due to its focus on high computation power. Hence in most VANET applications, consortium blockchain [9,15], which is a specific variant of blockchain, is incorporated. Practical Byzantine Fault Tolerance (PBFT) [7] is utilized as the underlying consensus protocol in this system. PBFT works efficiently, even when a portion of the network is faulty. However, PBFT only scales to few tens of nodes since it needs to exchange messages to reach consensus on a single operation among n servers resulting in $O(n^2)$ complexity. Therefore, consortium blockchain operating with a fixed number of pre-authorized nodes as validators solves massive consensus overhead [18]. Thus we incorporate consortium blockchain for our trust management system as it is an appropriate fit to compute and disburse trust scores faster in VANET based on blockchain.

1.2. Contribution

In this work, a decentralized trust score system that considers the trustworthiness of vehicular nodes based on specific quality metrics is designed. For this purpose, we incorporate the Trusted AODV protocol in our framework as the routing protocol. Most of the works, such as [4] and [36] alter the mechanism of AODV to incorporate trust models. Hence, in our work, the existing AODV routing protocol has been modified to calculate the trust scores of neighbor nodes. Trust in this model is calculated based on various metrics concerning routing protocol such as packet delivery ratio (PDR), the time difference in response to new route query RREQ and difference in destination sequence number (DSN). In the second-level, authorized validators perform the trust score aggregation logged as transactions by multiple nodes. Also, they update the blacklist node table based on the pre-configured trust-score threshold. The blacklisted node table is designed as a global referencing table immutable by malicious nodes, transparent, and quickly distributed to all the nodes via blockchain.

The main contribution of this paper is summarized as the following;

- First, we exploit blockchain features to implement a blockchain-based two-level trust score system as a

solution to detect and blacklist multiple blackhole nodes from the network much different from conventional methods of elimination.
- Second, we design the processing logic for decentralized transaction pool processing and trust score aggregation in a VANET system.
- Finally, we present the results in terms of the impact of the proposed blockchain-based trust model on network metrics such as throughput rate and packet drop ratio through simulation.

The remainder of this paper is organized as follows. Section 2 reviews some related work. Section 3 describes the proposed system model for the detection and elimination of blackhole nodes from the network. Section 4 consists of a brief discussion of security analysis. Section 5 describes the experimental results and some discussions. Finally, Section 6 concludes this paper.

2. Related work

In this section, we briefly review the literature on the various solutions for a specific insider attacks in VANET named blackhole. A blackhole attack is one of the attack vectors defined earlier that presents a threat across a broad spectrum of networks and is not unique to the vehicular networking domain.

In [3], the authors proposed a combination of greedy geographic routing protocol and trust tables to include a reliability parameter for vehicular nodes. However, their solution is suitable for only a low-density network where the car moves in straight lines. In [2] authors built an intelligent intrusion detection system (IDS) based blackhole detection system for VANETs. It can detect novel attacks (variations of blackhole attack) as well. However, it is time-consuming and requires more computational resources.

Several solutions were proposed based on trust value such as in [29], every node maintains a trust table in addition to the routing table in which every individual node stores the trust value of its one-hop neighboring nodes by monitoring in promiscuous mode to observe the forwarding and dropping of packets by the neighboring nodes. They also implement a trust manager that sends and receives trust tables to/from RSUs. However, there is a high potential for a single point of failure with this design.

In [16], authors have proposed a technique incorporating a trust table for holding the honest nodes. The RREP is overloaded with an extra field that indicates the reliability of the replying node. The source node sends its' data if and only if a reliable node propagates the RREP. Otherwise, it shall wait for another RREP. However, if other malicious nodes exist in the network, causing false accusations, this approach will fail in rightly identifying the blackhole node.

As we have seen from the above discussion, most of the proposed methods brought some novelty to the attack detection scheme. Besides, they suffered from drawbacks such as scalability [3], computational overhead issues on intermediate, and source node [2]. Besides, a trust-based system like [29], fails in the case of a large network. When multiple trust scores of various neighbor nodes are sent to a central trust manager, this can cause network congestion and communication delay. This becomes critical, especially in safety-related scenarios. Additionally, the trust score update process might become a victim of a broadcast tampering attack where the intermediate nodes or the trust manager alters the original message or might drop packets without forwarding. The need for a better trust management system has attracted researchers towards evaluating blockchain, a new decentralized distributed technology, which guarantees trust in untrustworthy environments.

The development trend of blockchain technology is across various domains, as addressed in several works in various areas, such as smart transportation, supply chain, manufacturing [1], crowdfunding [41], healthcare [44], mobile crowdsourcing [12] and trust evaluation. Based on the decentralization nature of blockchain, trust management issues can effectively be conducted using RSU's on the blockchain network in VANETs. Authors of [20] explore a blockchain-based anonymous reputation system to establish distributed trust management in VANETs. Their trust model is to improve the trustworthiness of messages exchanged, and it relies on the reputation of the sender. All the broadcasted messages are recorded on a blockchain as persistent evidence to evaluate each vehicle's reputation. Similarly, the authors of [45] proposed a blockchain-based decentralized trust management system in vehicular networks using the Bayesian inference model for the received messages to assess its credibilities. Vehicles can query the trust values of neighbors. Trust values are aggregated in the RSU based on ratings generated by message receivers using blockchain.

In [39], authors developed a countermeasure for blackhole attacks in VANETs consisting of detection, identification, and prevention of blackhole nodes using a backtracking algorithm. If a consensus is reached by both source and destination nodes that the accused node is acting maliciously, they can ban that node for a temporary period of time. The banned node entries are kept in a Blockchain. However, the solution mainly focuses on the case of a single malicious node.

Table 1 summarizes the features proposed by various other works that we have discussed. As we notice from the table, our work is consistent with other blockchain-related works. However, we aim to resolve the blackhole attack and propose a technique considering entity (vehicular node's) trust rather than reputation based on historic interaction. Due to the high mobility that is restricted to road topology and variable speed in VANETs, the nodes of the network change frequently resulting in no long-term interaction among vehicles. In most cases, two vehicles may exchange a handful of messages just for once which makes it impractical to accumulate interaction information to calculate the trust of the other entities. Thus it is crucial in such environments to precisely infer the level of trust quickly based on the factors that are relevant for that instant such as network throughput and packet delivery ratio. As a result, we design an efficient decentralized trust score management system based on entity trust and a blacklist node table management system based on blockchain. The main objective of this work is to eliminate packet droppers from the network, minimize the routing overhead of trust scores. Moreover, establishing a decentralized system in place decreases the latency time for a source node in finding a legitimate route to the destination.

3. Proposed system model

In this section, we first give an overview of the proposed trust score management system based on blockchain. Then we introduce the main components of the system architecture as illustrated in Fig. 2 as well as the threat model for the system. Finally, we present the system methodology in detail.

3.1. Components of system model

Before elaborating on the methodology of the proposed architecture, we first introduce the main components of the system design. As illustrated in Fig. 2, the considered system model has the following entities.

Consortium Blockchain Network: The consortium blockchain network [46] is the core of our proposed scheme. In a consortium blockchain, the nodes that participate in the consensus are pre-authorized, and they determine the generation of each block. In

Table 1
Comparative analysis of various approaches of the existing trust models For VANET.

Approaches	[20]	[3]	[2]	[29]	[16]	[45]	[39]	OurWork
Decentralized Trust Management	✓			✓	✓	✓	✓	✓
Centralized Trust Tables		✓						
Detect Blackhole Greyhole attack			✓	✓			✓	✓
Detect Multiple Blackhole nodes								✓
Based on Entity Trust	✓							✓
Reputation on Messages Exchanged					✓	✓		
Leverage Blockchain	✓					✓	✓	✓

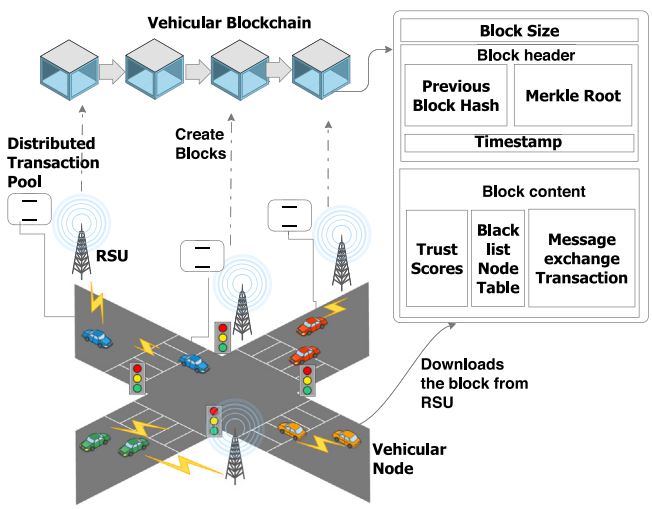


Fig. 2. Consortium Blockchain-based VANET system architecture.

this design, RSU is a pre-authorized node. RSU is granted the right to write data into the blockchain and participate in the consensus. These are considered as full nodes which authoritatively verifies all transactions in the network [34]. On the other hand, a vehicle is a lightweight node that can access the data replicated on the RSU storage, but it does not participate in the consensus.

Local storage in the RSU is responsible for collecting data uploaded by the vehicular nodes as well as obtaining data shared by other RSU. The consensus mechanism solves the problem of mutual trust between the nodes in the system. Furthermore, the trust score information in one country need not be shared with other countries if the border-crossing traffic is not allowed between those countries. Hence for simplicity, we consider regional blockchain [35] specific to a geographic area maintained by the roadside units (RSU). Incorporating regional blockchain into the design for VANETs ensures that the blockchain is shared among nodes in a geographically bounded area.

Blocks: A block consists of a block header and a block body. The block header consists of the previous block's hash, timestamp, and Merkle root of transactions. The block body consists of a list of trust score messages that behave as transactions uploaded by vehicular nodes. Apart from that, the body also holds the aggregated trust scores and a blacklisted node table.

Transactions T_x: In a blockchain-based VANET, records of message exchanged, services utilized, etc. can be a part of each transaction. In our model, transactions are specifically referred to as uploading trust scores of neighbor vehicular nodes to the nearest RSU, about 1000 m in the communication range.

Road Side Units (RSU): We consider RSU as the edge nodes upgraded to have computational capabilities and storage space. In our model, RSU maintains local storage that collects transactions. After verifying the transaction signature, transactions are broadcast to other RSU. RSU also acts as an aggregator for calculating cumulative trust values for a vehicular node and performs block creation to append new blocks to the blockchain.

Vehicular Nodes: These are vehicles equipped with sensors and OBU (which is in charge of all communication and computation tasks) that can communicate with each other and RSU through radio. In our blockchain-based VANET environment, each vehicular node is represented as V_i where $i \in \{1, 2, 3, 4, \ldots, N\}$ and N is the total number of vehicular nodes in the network. These nodes are assumed to be lightweight and are not part of the block creation process. Information exchange between any two nodes in VANET takes place through dedicated short-range communication (DSRC) [33] radio protocol via which vehicles exchange messages with nearby vehicles in V2V or V2I connectivity mode. Vehicular nodes are responsible for uploading transactions into the shared ledger maintained by RSUs. Vehicular nodes have the lowest security level.

3.2. Threat model

Both RSUs and vehicular nodes are vulnerable to attacks, which can cause network performance deterioration. RSUs are considered semi-trusted with a medium level of security. Some of the vehicular nodes' operations may as well be taken control of by the adversaries. These malicious nodes may act individually or in collaboration to drop the packets passing through them. Although all the communication between the RSUs is assumed to take place via a secured channel, we consider the following types of attacks that can be launched to jeopardize the running system.

- Defaming or Bad-mouthing attack: It is possible to sense a bad-mouthing attack in this model, which means that the vehicle can generate a false trust score for an honest vehicle and upload the transaction to the distributed ledger.
- Identity Spoofing attack: Vehicular nodes may try to spoof the identity of the other nodes in the network and try to upload the trust scores to the blockchain.
- Tampering Blacklist Node table: Malicious internal nodes might try to perform add/delete or modify the blacklisted node table to hamper the integrity of the system.

- Byzantines RSU's: It is possible that some of the RSUs may act maliciously or might be under the control of external attackers during the validation process to damage the network.

While we consider the attackers not to control all the nodes within the network causing eclipse attack [14], we build our model with an assumption of having about 25% of the malicious nodes in the network and design countermeasures against the blackhole attack.

3.3. Design goals

Under the threat model defined, our goal is to design a tamper-proof trust scores and blacklist node table in a vehicular network, which should be effective and an efficient trust score management system with the following key requirements.

- The proposed system should be (1) scalable to support a very high range of vehicular nodes that join the network (2) transparent so that all the authorized members of the system should have access to the same immutable records (3) tamper-resistant so that it ensures the integrity of stored trust scores and blacklist node tables. (4) enabled to audit to produce tamper-proof evidence.
- The proposed system should be free from a single point of failure (SPOC). Thus it necessitates the need for incorporating decentralization in our design so that no single entity is holding control of the entire system.
- The processing and execution speed of the proposed system should be of the order of a few milliseconds so that each transaction is processed and an updated trust score is available to the entire system without having to wait too much. This in turn should minimize the routing overhead in VANETs.
- The cost of data storage associated with the proposed system should be of an acceptable range which is a very crucial design requirement.

3.4. Overview

In a VANET, every vehicular node maintains a routing table for known destinations. For unknown destinations, the route is updated by using RREQ and RREP messages over the Trusted AODV routing protocol. In this protocol, the node can promiscuously monitor its neighbor node for generating trust. A promiscuous mode is where an honest node taps the packets being forwarded by its neighboring node so that a node can determine whether a neighboring node forwards a packet or drops [38]. A caching mechanism is implemented in the TAODV protocol to verify that a neighboring node forwards packets. To determine if it is the same packet, the node verifies the tapped packet with the cached packets. If cached packets cannot be tapped from their neighbor, then those packets are considered to be dropped. We make use of this protocol in our design (see Fig. 3).

In this method, the node transmits dummy packets to its neighbor node over the UDP transport layer protocol. Using promiscuous mode, the node can judge its neighbor node for a pre-determined short duration of time and assign a trust score. In our experiments, we have set the time limit to 120 s.

Instead of broadcasting trust score by each node to the network and causing computation overhead of trust value aggregation, in our proposed model, the trust score is managed in a decentralized manner. It is uploaded as a transaction to the nearest RSU. RSU verifies the signature to validate that the message is from an authenticated node and further broadcasts the transactions to other RSUs, thus maintaining a distributed ledger.

Since the distributed ledger system's state has to be agreed by the peers, the consensus has to be achieved. RSUs in our model, which are pre-selected validator nodes, follow the Practical Byzantine Fault Tolerant (PBFT) consensus mechanism to achieve consensus. PBFT is an improved version of Byzantine Fault Tolerance that ensures a consensus regardless of malicious behaviors on the part of some participating nodes [7]. After every interval of window time t_i, a leader node is chosen that aggregates all the transactions together from the processed transaction pool to publish a new block of data. This block contains the previous block header, current block hash, Merkle root [22] of transaction records, aggregated trust scores of nodes, and a blacklisted node table.

As we are dealing with essential event messages, reliable and quick message dissemination is of high priority. Vehicular nodes, which are light in the network, periodically download the latest trust scores and blacklisted node table from the blockchain to refer before performing any message dissemination through the network nodes. Thus insider attack launching nodes are detected and eliminated based on trust score. A safe, reliable, and tamper-free route to communicate messages in the network is ensured via blockchain technology.

3.5. System methodology

In this section, we outline the overall system methodology of our proposed framework step by step.

1. *System Initialization*: The vehicular nodes joining the blockchain network for the first time submit their identification details such as name, address, Electronic License Plate (ELP) number of vehicles, and other required identification details to the RSU. It, in turn, assigns a pseudo-identity id_{vi} which is a unique number for each vehicular nodes V_i along with generating a public–private key pair by using Elliptic-Curve Diffie–Hellman(ECDH) key agreement protocol. Each license plate is also mapped to a renewal count RC_{vi} that counts the number of times a node has re-registered in the network. The RSU generates a mapping list $\{id_{vi}, PK_{vi}, SK_{vi}, RC_{vi}\}$ for each V_i. This identification vector of the V_i is generated every time when a vehicular node rejoins the network incrementing RC_{vi}. It will be digitally signed by the RSU and stored as a single transaction in the identification ledger.
2. *Trust Score Assignment*: As an additional initialization step, at the time of joining the network, each vehicular node V_i gets a default trust score. For simplicity, we have considered the default trust score as 0.5 on a scale of 1.0. The value of the default trust score is solely a design decision at the time of system initialization. Assigning default trust scores to each node is an important factor that draws a boundary between honest and dishonest peers as well as new members of the network. Trust scores vary dynamically with parameters such as time and packet delivery ratio. As the packet delivery ratio diminishes below a stipulated threshold, the trust score is dropped to 0.0. Consequently, the corresponding vehicular node is blacklisted. Thus trust scores of 0.5 signify a newly joined member of the network. Trust scores beyond 0.5 signify greater levels of trustworthiness of the member node.
3. *Genesis Block Creation*: The blockchain begins with a genesis block on top of which the successive blocks are stacked. Genesis block contains a previous block hash as "0", blacklisted node table, and transaction lists as null along with the current timestamp placeholder, the placeholder for the highest destination sequence number. When V_i joins the

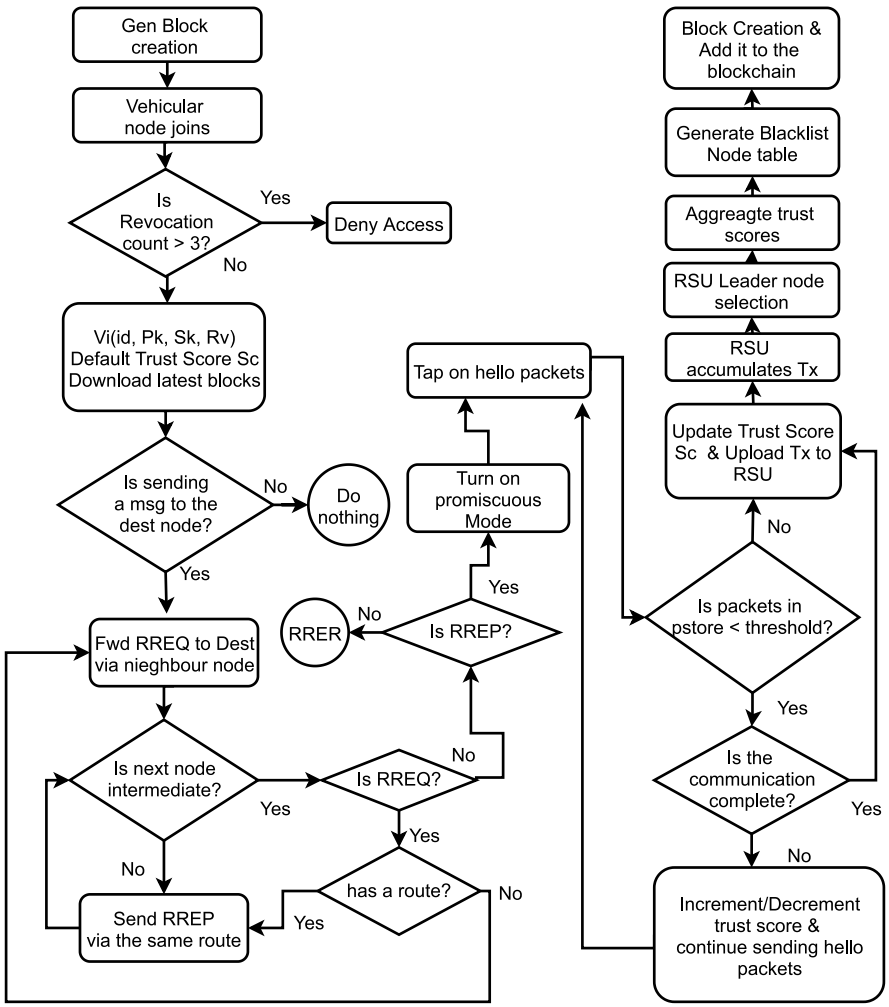


Fig. 3. Flowchart of the proposed blockchain based VANET trust score system.

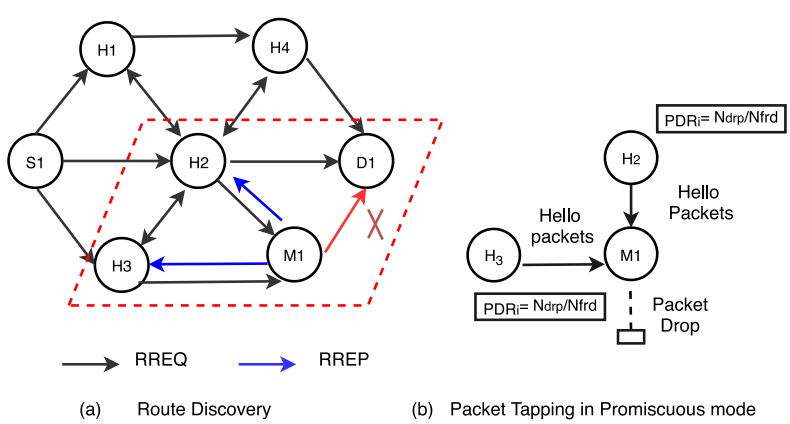


Fig. 4. Flow of route discovery and packet tapping in promiscuous mode.

network, then it only knows the genesis block. V_i will have to download all the blocks from the nearest RSU starting from the genesis block to synchronize with the blockchain network [34]. In the blacklisted node table, aggregated trust scores get updated as per the latest communication.

4. *Sending Route Discovery*: Let us assume the node S_1 to be the source node desiring to communicate with node D_1 as in Fig. 4(a). Thus, as per AODV protocol, node S_1 floods an RREQ packet in the network and waits for the RREP packet to obtain a fresh route to the destination node D_1. All nodes forward the request further in the network until a fresh route notification is returned. When the RREQ reached M_1, it returns RREP with the highest destination sequence number (DSN) to its neighbor node H3 and H2, as shown in Fig. 4(a). When the RREP packet is sent from an intermediate node M_1, nodes preceding the node which

sent the RREP packet, i.e., H_3 and H_2 in Fig. 4(b) gets alerted to certify the RREP sending node. Node H_3 performs the preliminary checks from the RREP message to check two conditions. (1) DSN received vs. current maximum DSN recorded in the blockchain network. (2) Timestamp difference between RREP and RREQ.

5. *Entering Promiscuous Mode*: Once the preconditions are met, node H_3 switches on its promiscuous mode to tap onto M_1. Promiscuous mode [38] is the mode in which a node can overhear the packets transmitted by its neighbor node by tapping it as represented by pseudo code in algorithm 3.1 & 3.2. Node H_3 sends a series of hello packets to the destination node D_1 via node M_1. Variable *pstore* holds the list of all hello packets sent out from node H_3. Information can be fetched by calling *packetLookUp* method on each packet. Algorithm 3.1 loops through each packet in the *pstore* and calls *MethodTap* on it. Struct *hdr* holds the header information of each packet sent out from node M_1. *MethodTap* verifies if the packet sent out from M_1 matches the packet stored in source node's *pstore* to certify that the node is not malicious. If it matches, it deletes the packet from *pstore* invoking *deletePacket* on that particular packet. We make use of these algorithms for VANETs, which allows honest nodes H_3 and H_2 to intercept and read neighbor node M_1 within its range, sending network packets.

Algorithm 3.1 Caller Method

Input: *tstore* → an object of *trustStore* of node *i* contains methods *trustUpdate* and *trustLookUp*
pstore → an object of *packetStore* that contains a list of packets stored in cache, has access to *packetLookUp* method and has access to *deletePacket* method
Output: *tstore* → trust score obj of neighboring node i

```
Initialization : Packet *p → uid, src, dest, fwrdId
                 Default trust of node i → 0.5
Loop through every packet in pstore and invoke MethodTap
for Packet *p do
  Invoke MethodTap
end for
```

Algorithm 3.2 MethodTap: Packet Tapping

Input: *Pointer to packet *p*
Output: Invokes *UpdateTrustScore*(*hdr.orgby*)

```
Initialization : Declare Struct hdr → fetch header info of *p
hrd → { uid = p.uid,
        orgBy = p.fwrdId,
        src =p.src,
        dest = p.dest}
var pb → pstore.packetLookUp(hdr.uid)
if pb != null then
  if (pb.packetId == hdr.uid) & (pb.source == hdr.src) &
     (pb.dest == hdr.dest) &(pb.fwdrId == hdr.orgby) then
      call pstore.deletePacket(hrd.uid) → to delete the p*
    from cache
      invoke UpdateTrustScore(hdr.orgby)
  end if
end if
```

6. *Trust Value Calculation Sc*: Every node which receives RREP from its neighbor in a network determines the trust value that represents the trustworthiness of the neighboring node. This is the first level of trust score calculation that happens in our proposed model. The trust value *Sc* of node $M1$ is calculated dynamically based on the number of packet forwarded N^{fwd} by node M_1.

Algorithm 3.3 UpdateTrustScore: Updating Trust Score

Input: header struct *hdr* for each *packet *p*
Output: Trust score for a node is updated UpdateTrustScore-hdr.orgBy

```
trust → tstore.trustLookUp(hdr.orgBy)
if trust ≤ 1 then
    tstore.trustUpdate(hdr.orgBy, trust+0.001)
else
  tstore.trustUpdate(hdr.orgby, 1)
end if
trust
```

Algorithm 3.4 checkMalicious: Checking Malicious Node

Input: header struct *hdr* for each *packet *p*
Output: TrustScore for a node is updated checkMaliciousorgBy

```
if (pstore.packetCount ≥ 50) then
    trust_orgBy → tstore.trustLookUp(orgBy)
  if trust_orgBy then
      tstore.trustUpdate(orgBy,0)
  else
    tstore.trustInsert(orgBy,0)
  end if
end if
trust_orgBy
```

As shown in pseudo-code 3.3 for every matched packet that goes out of node M_1, the trust score is incremented by 0.001 by invoking *UpdateTrustScore* method. However, if the accumulated packet in the *pstore* is greater than the threshold (λ) at any point in time during the transmission, the trust score *Sc* is decremented to 0.0 as shown in algorithm 3.4 by *checkMalicious* method. The threshold (λ) is a dynamically set value based on the criticality of the message that is being transmitted via V2V. For instance, the threshold value for life-critical message dissemination such as Emergency Electronic Brake Light (EEBL) or Collision Avoidance can be set low so that no packets can be stacked in the queue for a long time. For our experiments, we have considered a threshold value of 50 as the maximum number of packets that can be accumulated in the packet store. This is the total number of packets sent out in 2.5s at the rate of 1 per 0.05s during the simulation of the transmission. Once the set threshold is reached, the trust value is decremented to 0.0. This approach helps us to rightly identify the fake node that tries to deceive the system. Consider a case in which a node V_i is a busy intermediate node for many communications. This node might probably try to deceive the system by transmitting packets only until it achieves higher trust scores. Thereafter it either drops the packets intentionally or is being controlled by an adversary. In this case, decrementing the trust score by some percentage would still consider the node as honest since it is not blacklisted yet. Consequently, this results in a malicious node being considered trustworthy. Hence to overcome this situation, in our design we immediately drop the trust score to 0.0 once the node is identified as dropping packets. However, the threshold value can be set high for noncritical events and it can be set by using learning algorithms [37]. If no packets are dropped and the number of packets in *pstore* is below the threshold, the

trust value Sc is incremented until it reaches 1.0. Once the trust score is calculated, the traditional process is executed where RREP is forwarded until it reaches the source node. At this point, node $H3$ also uploads a transaction containing trust scores to the nearest RSU.

Algorithm 3.5 Creating Vehicular Nodes and Ratings List

Input: $T_x \rightarrow$ Stream of transactions in a pool
Output: V_x and r_x
1: *Initialization* : *Total Number of transaction T_x for 120s*
$V_x \longleftarrow$ arrayList of all vehicular ids
$r_x \longleftarrow$ arrayList of all vehicular ratings
2: *Loop through every transaction in T_x*
3: **for** Transaction T_i **do**
4: Push $V_i \longleftarrow$ to V_x
5: Push $r_i \longleftarrow$ to r_x
6: **end for**
7: **return** V_x and r_x

Algorithm 3.6 Creating Multimaps of Flattened Trust Scores

Input: $V_x \rightarrow$ arrayList of vehicular ids
$r_x \rightarrow$ arrayList of vehicular ratings
Output: *nodes* $\rightarrow$ Map of nodes : corresponding multiple trust scores
1: Create a multimap with vehicle id as key and multiple ratings as value.
2: **for** $i \leq V_i$ size **do**
3: put V_i into *nodes* multimap and map it with multiple r_i
4: **end for**
5: **return** *nodes*

Algorithm 3.7 Aggregating Trust Scores

Input: *nodes* $\rightarrow$ multimap of nodes : corresponding trust scores
Output: $avg_{ri} \rightarrow$ Average rating for V_i after aggregation
1: **for** $node_i \in nodes$ set **do**
2: Create r_i list
3: **for** every rating $r \in r_i$ list **do**
4: Aggregate and Calculate Average avg_{ri}
5: **end for**
6: **end for**
7: **return** *nodes*, avg_{ri}

Algorithm 3.8 Blacklisting Nodes

Input: *nodes* $\rightarrow$ List of all nodes
$avg_{ri} \rightarrow$ Average rating for V_i after aggregation
Output: *nodes* $\rightarrow$ mutated list of active nodes
1: **for** $node \in nodes$ set **do**
2: **if** $avg_{ri} \leq threshold$ **then**
3: *nodes*.remove(node)
4: **end if**
5: **end for**
6: **return** *nodes*

7. *Transaction Logging*: In our model, vehicular node V_i invokes an object of transaction class and posts an updated trust score of monitored neighboring vehicle V_j as a transaction in the distributed storage maintained by RSUs. Every transaction must be signed by a digital signature $Sig(SK(V_i))$ of the initiating entity. A typical transaction is represented in Table 2. Each transaction is uniquely identified by a transaction *id* represented by Tx_{id}.

Table 2
Typical transaction by a member node.

Tx_{id}	$ACRAF23DB3C4$
TimeStamp	$YYYY - MM - DDTHH : MM : SS$
SourceNodeId	PubKey(V_i)
NeighbourNodeId	PubKey(V_j)
TrustScore	$0.0 - 1.0$
Hopcount	n
digital signature	$Sig(SecKey_{vi})$

8. *Transaction Pool Processing*: Over time, unconfirmed transactions created by the vehicular nodes throughout the network get accumulated in the distributed ledger transaction pool. This processing of the transaction pool starts with algorithm 3.5 where the system reads each transaction T_x from the pool and creates a list of vehicular nodes V_x and its corresponding list of ratings r_x. It is further processed as shown in pseudo-code 3.6 where each unique V_x is mapped to multiple ratings associated with it. Once the pool of transactions is processed, it outputs a list of vehicular nodes and their corresponding trust scores. This enables the authorized validators, which are RSUs in our model, to get an accurate list of recent transactions that have to be added to the block.
9. *Leader RSU Selection*: We use Practical Byzantine Fault Tolerance (PBFT) as the underlying consensus protocol. Developed by Castro and Liskov [7] in 1999, PBFT has gained wide recognition for practicality. In our system, authorized RSUs are the validator nodes that follow PBFT protocols to generate and broadcast blocks. One of the RSUs is randomly chosen as the primary, or the leader node, and others are secondary.
The leader node collects all received records of transactions and generates a Merkle hash value of the records linked to the previous block in the vehicular blockchain and successfully generate a block. Once a leader creates a block, it is validated by the secondary nodes, and all honest nodes help reach a consensus regarding the state of the system using the majority rule. A PBFT enabled distributed system provides a practical byzantine state machine replication that can work even when malicious RSUs are operating in the system, assuming that honest RSUs are more than $2f+1$ where f is the number of faulty RSUs.
10. *Trust Score Aggregation by RSUs*: There is a possibility that each node could be verified by several neighbor nodes. In our e.g. $H3$ and $H2$ both nodes validated $M1$. Hence ratings have to be aggregated before creating blocks. Leader RSU node picks all the logged transactions from the processed transaction pool in every defined time interval. It aggregates every vehicular node's ratings based on Algorithm 3.7. Algorithm 3.8 provides the pseudo-code for blacklist node table generation. Further to this, it also invokes a Merkle tree module to create a Merkle root hash of all the transactions.
11. *Block Generation and Addition to Blockchain* : Once the trust scores are aggregated, and the blacklisted node table is generated, it is added to the block along with the previous block hash and Merkle root. The created block is pushed for verification from secondary nodes. They validate the correctness of the block and send approval messages to the leader node. After receiving 2/3rd approval, the leader node adds the block to the blockchain and notifies all the nodes as per the PBFT protocol.
12. *Download Blacklist Node Table*: All RSUs update their local chain with the latest one to reflect the latest transactions. All vehicular nodes update the local database with

the updated blacklisted node table and the highest DSN recorded. Source nodes waiting to deliver messages confirms whether the intermediate node responding with the RREP message is blacklisted or not. If yes, it sends out fresh RREQ. Else, communication packets are sent via the found root.

13. *Public Key Revocation and Reactivation*: Once a vehicular node V_i is blacklisted, its associated identity vector $\{id_{vi}, PK_{vi}, SK_{vi}, RC_{vi}\}$ will be revoked before its intended expiration date. A node can retry at most 3 times to re-register in the network. RSU checks the license plate number and fetches the RC_{vi} count for the node V_i requesting re-registration. If RC_{vi} is below 3, the request for a new unique identity vector containing a unique public key PK_{vi} is issued. This gives vehicular nodes a fair amount of chance to rejoin and correct the misbehavior.

4. Security analysis

In this section, we discuss the security properties of our proposed model framework. Specifically, this analysis is focused on the resilience against the attacks discussed in Section 3.2. Table 3 below summarizes a comparative analysis of security features of the various other approaches discussed in Section 2.

4.1. Defense against defaming attack

A malicious vehicular node (different from a blackhole node) may evaluate the honest neighbor node to calculate the trust score. It may upload a fake trust score to eliminate the honest behaving node from the network. However, the proposed scheme is secure against the defaming attack. In our model, this attack is defended in two ways. Firstly, each node can submit trust score once for a specific neighbor node and upload the transaction tuple $T_j = Src_{vid}|Node_{vjd}|TrustScore_{vjd}|Hop|DigSign_{vid}$. Duplicate tuples with the same Src_{vid} and $Node_{vjd}$ are eliminated by RSU's. Secondly, trust score for specific node T_j is aggregated by RSUs by averaging over multiple m records of trust score i.e. $\frac{1}{m}\sum_{x=1}^{m} T_j$ as reported from different neighbor nodes. Lastly, due to the limited number of malicious nodes, these unfair trust scores can hardly disrupt the system.

4.2. Defense against identity spoofing and tampering blacklist node table

During the transaction logging phase, each transaction T_j needs to be signed by the current vehicle node $DigSign_{vid}$ before they are sent. The authorized RSU then verifies the signature. Consortium blockchain, combined with the digital signature technique, ensures that any external attacker cannot disrupt the network as the attacker's digital signature cannot be verified. Additionally, an external attacker cannot launch an identity spoofing attack as no entity can falsify the digital signature of another entity without the private key of the actual member of the network. This in turn, ensures that only legitimate and authenticated vehicles can upload the trust scores for the network entities.

Tamper-proof is another essential feature of this framework. Since the blacklisted node table is distributed in a decentralized manner via blockchain, it is free from any internal or external entity performing add, delete, or modification to the blacklisted node list. This is because of the inherent properties of blockchain; any changes made to the stored node table will inevitably change the hash value of the block resulting in the mismatch and invalidation of the block by the majority of the honest nodes.

4.3. Defense against Byzantine RSUs

The proposed system discussed that a small portion of RSUs might get controlled by an attacker. Data might get altered or deleted by these malicious RSUs. However, the PBFT consensus mechanism employed in the system ensures the network's normal operation even when 33% of the nodes are damaged. In PBFT consensus mechanisms, the block proposer is bound to get at least $\frac{2}{3}$rd of votes from the secondary RSUs that are honest. If we suppose that there are f malicious RSU nodes in the whole network and the total number of RSUs satisfies $n \geq 3f+1$, the system can defend against malicious tampering data attacks initiated by faulty RSU. This makes our system byzantine fault-tolerant reducing the impacts of compromised RSUs.

5. Experimental evaluation

5.1. VANET simulation setup

To study the impact of insider attack scenarios and test the network performance with the proposed blockchain-based trust score management solution in VANET, we utilize the NS2 simulation tool installed on Virtual Linux OS with Ubuntu 16.04 distribution having 8.00GB RAM.

In our study for the simulation of real-time roads, junctions, and traffic light, we used OpenStreetMaps (OSM),[1] which provides free editable maps of the world. OSM helps generate realistic street structures considering two-way, four-way streets, traffic lights, and the buildings. For vehicular mobility, we use the Simulation of Urban Mobility tool (SUMO) [5] version 1.23. The generated traffic models are then imported into NS2 to simulate various attack scenarios. We restrict the simulation area to 800×800 m and repeat evaluation for ten iterations with 20, 40, 60, 80, and 100 number of vehicular nodes each time. Since the area under simulation is quite smaller; we test with a maximum of 100 nodes for our experiments. The simulations generate trace files analyzed using AWK scripts to calculate average packet delivery ratio (PDR), average throughput, and average delay when the messages are communicated in a network.

We used the TwoRayGround propagation model with a maximum speed of 15 m/s. WirelessPhy was used as the network interface type in the configuration file. The UDP traffic was used to send data from source to destination nodes faster as it does not require a 3-way handshake to establish a connection. Packets among the nodes were transmitted with a constant bit rate (CBR) of one packet per 0.05 s. We used a constant size of 512 bytes for each packet for all our simulations.

We also vary the number of sources and destination nodes as two for 20, three for 40 and 60, four for 80, and 100 number of total nodes. These nodes sent and received data packets throughout the simulation. The simulation was done for 120 s (2 mins).

5.2. Blockchain simulation setup

We have implemented trust score management in the Java environment using a laptop with 2.3 GHz Intel Core i5 and 8 GB 2133 MHz LPDDR3. Our simulated blockchain framework receives trust scores data from vehicular nodes through V2I communication for the trust score aggregation. We consider that a network containing N nodes, requires at least 15% of maximum network strength as validator nodes. Hence we perform all the tests with 15 validator nodes, i.e., RSUs. We implement various methods to calculate each node's aggregated trust scores, which are disseminated in the blockchain network along with the list of blacklisted nodes.

[1] http://www.openstreetmap.org

Table 3
Comparative security analysis of the existing trust models For VANET.

Secured Against	[20]	[3]	[2]	[29]	[16]	[45]	[39]	OurWork
UnAuthorized Identity	✓	NA	NA	NA				✓
ID Spoofing						✓		✓
Defaming						✓		✓
Byzantine RSU						✓		✓
Data Tampering						✓	✓	✓

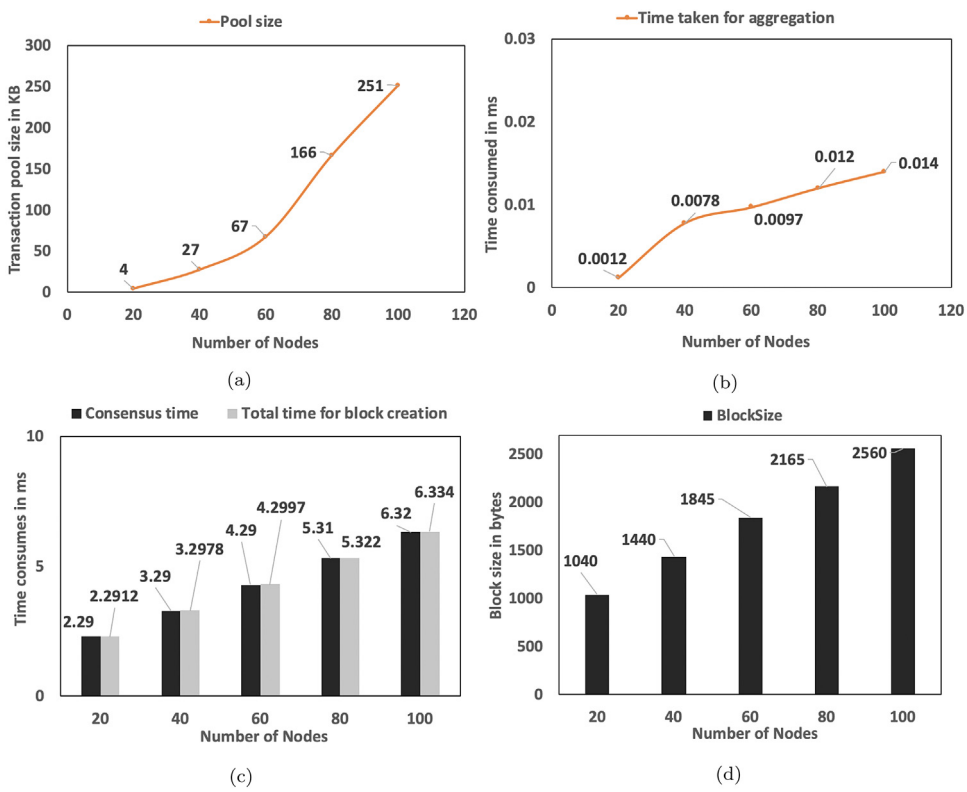


Fig. 5. Analyzing the data metrics in blockchain (a) Size of the transaction pool created with different number of nodes (b) Time consumption of transaction pool processing and block creation vs the number of network nodes.

5.3. Analysis

We mainly consider the following different scenarios.

1. Computation cost of transaction pool and trust score aggregation logic, and derive insights on the scalability of the proposed trust score system.
2. Analyze the total time taken to create a block with 15 entities as validators using PBFT consensus.
3. Effects in network performance of VANET by removing multiple insider attack launching nodes, which consists of approximately 25% of the given network, using our framework.

5.3.1. Computation cost of transaction pool processing and trust score aggregation

In this experiment, we have used the data generated from the simulation of Trusted AODV protocol (NS2 tool) as an input to the trust management prototype built based on simulated blockchain. Trust scores data generated as the output from the simulation was used for logging as transactions in the transaction pool. This data was processed as per Algorithm 3.5 to 3.8, and the time consumption for processing various sizes of data pool was evaluated. Different configurations in NS2, i.e., 20, 40, 60, 80, and 100 nodes generated 4 KB, 27 KB, 67 KB, 166 KB, and 251 KB size of trust scores as transactions in the transaction pool respectively, as shown in Fig. 4(a). Time taken by the validator nodes, i.e., RSUs, to process these transactions and aggregate the trust scores is of the order of few milliseconds, as seen in Fig. 4(b). Although with the increase in the number of transactions, time consumed to aggregate changes linearly, we infer that with the highest of 100 vehicular nodes, the maximum transaction pool size of 251 KB was processed, and aggregated trust scores were calculated in as low as 0.014 ms.

5.3.2. Computation cost of block creation using PBFT consensus using a varied number of validators

In this experiment, we evaluate the proposed model in terms of time taken for reaching PBFT consensus among validators

Fig. 6. Variation of network performance of VANET nodes in the presence of and after removing malicious node.

on the state of transactions and then creating a block with a blacklisted node table. We perform this test with 15 number of validator nodes, i.e., RSUs, to process different block sizes. Fig. 4(c) represents the time taken to reach consensus and the total time consumed for each block creation. Block creation with a maximum of 15 validators for a network size of a maximum of 100 vehicular nodes is taking 6.334 s to create a new block. This shows that blacklisted node tables are disseminated to all the network participants within a short duration. Additionally, a block is being created every 120 s with an average size of 1.8 KB. Fig. 4(d) represents the blocksize with a different number of network nodes. Storage overhead is calculated to be around $1.8 * (60/120) * 24 * 365$ which is approximately 7884 KB per year for the entire blockchain. Hence, the techniques introduced also requires very low storage in the blockchain for the 800×800 area considered in the experiment.

5.3.3. Influence of trust model in network performance

To analyze if the network performance could be improved by incorporating the proposed model, we ran the simulation twice under the same configuration by injecting 25% malicious nodes. First, we ran it with traditional AODV protocol, and then we ran the proposed AODV with trust model on Blockchain. Note that these malicious nodes could be single blackhole nodes causing the packet drop or multiple nodes forming tunnels to consume the data packets. In the first case, without trust and having 25% of malicious nodes, it is clear that the network throughput in bps and packet delivery ratio extremely deteriorated as shown in Fig. 6(a). From Fig. 6(b) it can be inferred that a significant reduction in packet drop ratio and improved throughput in bits per second can be achieved by incorporating the proposed model into the VANET system. Furthermore, it is evident that when the number of nodes is 40 and 60 with the 25% malicious nodes, the packet drop ratio is of the same range as we are testing with two pairs of source and destination. Similar is the case with 80 and 100 nodes.

6. Conclusion

In this paper, a consortium blockchain-based approach for mitigating insider attack in the VANET system using Trusted AODV protocol is proposed. In this work, the vehicular nodes offload the mining process to the RSUs to speed up the block generation, suitable for the proposed VANET system. We use promiscuous mode to assign a trust value to neighbor vehicular nodes that responded dynamically and periodically with RREP messages. We also show how the trust score gets aggregated by authorized RSUs and we evaluated the block time consumption concerning PBFT consensus. The results show an improved packet delivery ratio and throughput of the entire network by incorporating blockchain-based VANET. It is proved to be more efficient for message dissemination in the VANET by efficiently eliminating the blackhole nodes.

CRediT authorship contribution statement

Sowmya Kudva: Conceptualization, Methodology, Experimentation, Validation, Writing - original draft. **Shahriar Badsha:** Supervision, Methodology, Conceptualization, Methodology, Investigation, Writing - review & editing. **Shamik Sengupta:** Supervision, Conceptualization, Writing - review & editing. **Hung La:** Supervision, Writing - review & editing. **Ibrahim Khalil:** Supervision, Writing - review & editing. **Mohammed Atiquzzaman:** Supervision, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Saveen A. Abeyratne, Radmehr P. Monfared, Blockchain ready manufacturing supply chain using distributed ledger, Int. J. Res. Eng. Technol. 5 (9) (2016) 1–10.

[2] Khattab M Ali Alheeti, Anna Gruebler, Klaus D McDonald-Maier, An intrusion detection system against black hole attacks on the communication network of self-driving cars, in: 2015 Sixth International Conference on Emerging Security Technologies (EST), IEEE, 2015, pp. 86–91.

[3] Hanin Almutairi, Samia Chelloug, Hanan Alqarni, Raghda Aljaber, Alyah Alshehri, Dima Alotaish, A new black hole detection scheme for VANETs, in: Proceedings of the 6th International Conference on Management of Emergent Digital EcoSystems, 2014, pp. 133–138.

[4] Neeraj Arya, Upendra Singh, Sushma Singh, Detecting and avoiding of worm hole attack and collaborative blackhole attack on MANET using trusted AODV routing algorithm, in: 2015 International Conference on Computer, Communication and Control (IC4), IEEE, 2015, pp. 1–5.

[5] Michael Behrisch, Laura Bieker, Jakob Erdmann, Daniel Krajzewicz, SUMO–Simulation of urban mobility: an overview, in: Proceedings of SIMUL 2011, the Third International Conference on Advances in System Simulation, ThinkMind, 2011.

[6] Subir Biswas, Jelena Mišić, Vojislav Mišić, Ddos attack on WAVE-enabled VANET through synchronization, in: 2012 IEEE Global Communications Conference (GLOBECOM), IEEE, 2012, pp. 1079–1084.

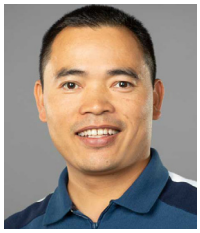
[7] Miguel Castro, Barbara Liskov, Practical byzantine fault tolerance and proactive recovery, ACM Trans. Comput. Syst. (TOCS) 20 (4) (2002) 398–461.

[8] Shanzhi Chen, Jinling Hu, Yan Shi, Ying Peng, Jiayi Fang, Rui Zhao, Li Zhao, Vehicle-to-everything (V2X) services supported by LTE-based systems and 5G, IEEE Commun. Stand. Mag. 1 (2) (2017) 70–76.
[9] Omar Dib, Kei-Leo Brousmiche, Antoine Durand, Eric Thea, Elyes Ben Hamida, Consortium blockchains: Overview, applications and challenges, Int. J. Adv. Telecommun. 11 (1&2) (2018).
[10] George Dimitrakopoulos, Panagiotis Demestichas, Intelligent transportation systems, IEEE Veh. Technol. Mag. 5 (1) (2010) 77–84.
[11] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, Alejandro Quintero, VANET Security surveys, Comput. Commun. 44 (2014) 1–13.
[12] Wei Feng, Zheng Yan, MCS-Chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain, Future Gener. Comput. Syst. 95 (2019) 649–666.
[13] P. Sai Gautham, R. Shanmughasundaram, Detection and isolation of black hole in VANET, in: 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT, IEEE, 2017, pp. 1534–1539.
[14] Daniel Germanus, Stefanie Roos, Thorsten Strufe, Neeraj Suri, Mitigating eclipse attacks in peer-to-peer networks, in: 2014 IEEE Conference on Communications and Network Security, IEEE, 2014, pp. 400–408.
[15] Shafkat Islam, Shahriar Badsha, Shamik Sengupta, A light-weight blockchain architecture for V2v knowledge sharing at vehicular edges, in: IEEE Smart Cities, 2020.
[16] Yaser Khamayseh, Abdulraheem Bader, Wail Mardini, Muneer Bani Yasein, A new protocol for detecting black hole nodes in ad hoc networks, Int. J. Commun. Netw. Inf. Secur. 3 (1) (2011) 36.
[17] Uzma Khan, Shikha Agrawal, Sanjay Silakari, Detection of malicious nodes (dmn) in vehicular ad-hoc networks, Procedia Comput. Sci. 46 (9) (2015) 965–972.
[18] Sowmya Kudva, Shahriar Badsha, Shamik Sengupta, Ibrahim Khalil, Albert Zomaya, Towards secure and practical consensus for blockchain based vanet, Inform. Sci. 545 (2020) 170–187.
[19] Sowmya Kudva, Renat Norderhaug, Shahriar Badsha, Shamik Sengupta, A.S.M Kayes, PEBERS: Practical ethereum blockchain based efficient ride hailing service, in: IEEE International Conference on Informatics, IoT and Enabling Technologies, 2020.
[20] Zhaojun Lu, Qian Wang, Gang Qu, Zhenglin Liu, 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), IEEE, 2018, pp. 98–103.
[21] Shirshak Maskey, Shahriar Badsha, Shamik Sengupta, Ibrahim Khalil, BITS: blockchain based intelligent transportation system with outlier detection for smart city, in: 2020 IEEE 18th Annual International Conference on Pervasive Computing and Communication Worshop, 2020.
[22] Ralph C. Merkle, A digital signature based on a conventional encryption function, in: Conference on the Theory and Application of Cryptographic Techniques, Springer, 1987, pp. 369–378.
[23] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, Liviu Iftode, Trafficview: traffic data dissemination using car-to-car communication, ACM SIGMOBILE Mob. Comput. Commun. Rev. 8 (3) (2004) 6–19.
[24] Satoshi Nakamoto, et al., Bitcoin: A peer-to-peer electronic cash system, Working Paper, 2008.
[25] Alok Nandan, Shirshanka Das, Giovanni Pau, Mario Gerla, MY Sanadidi, Cooperative downloading in vehicular ad-hoc wireless networks, in: Second Annual Conference on Wireless on-Demand Network Systems and Services, IEEE, 2005, pp. 32–41.
[26] Josiane Nzouonta, Neeraj Rajgure, Guiling Wang, Cristian Borcea, VANET Routing on city roads using real-time vehicular traffic information, IEEE Trans. Veh. Technol. 58 (7) (2009) 3609–3626.
[27] Karl J. O'Dwyer, David Malone, Bitcoin mining and its energy footprint, in: Proc. of the 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, 2014, pp. 280–285.
[28] Charles Perkins, Elizabeth Belding-Royer, Samir Das, RFC3561: Ad hoc on-demand distance vector (AODV) routing, 2203.
[29] Nazish Rafique, Muazzam A. Khan, Nazar A. Saqib, Faisal Bashir, Cory Beard, Zhu Li, Black hole prevention in vanets using trust management and fuzzy logic analyzer, Int. J. Comput. Sci. Inf. Secur. 14 (9) (2016) 1226.
[30] Maxim Raya, Jean-Pierre Hubaux, Securing vehicular ad hoc networks, J. Comput. Secur. 15 (1) (2007) 39–68.
[31] Maxim Raya, Panagiotis Papadimitratos, Imad Aad, Daniel Jungels, Jean-Pierre Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, IEEE J. Sel. Areas Commun. 25 (8) (2007) 1557–1568.
[32] Fatih Sakiz, Sevil Sen, A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV, Ad Hoc Netw. 61 (2017) 33–50.
[33] Muhammad Sameer Sheikh, Jun Liang, A comprehensive survey on VANET security services in traffic management system, Wirel. Commun. Mob. Comput. 2019 (2019).
[34] Rakesh Shrestha, Rojeena Bajracharya, Anish P Shrestha, Seung Yeob Nam, A new type of blockchain for secure message exchange in VANET, Digit. Commun. Netw. (2019).
[35] Rakesh Shrestha, Seung Yeob Nam, Regional blockchain for vehicular networks to prevent 51% attacks, IEEE Access 7 (2019) 95021–95033.
[36] Sridhar Subramanian, Baskaran Ramachandran, QOS Assertion in MANET routing based on trusted AODV (ST-AODV), Int. J. Ad Hoc Sens. Ubiquitous Comput. 3 (3) (2012) 135.
[37] Nasrin Taherkhani, Samuel Pierre, Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm, IEEE Trans. Intell. Transp. Syst. 17 (11) (2016) 3275–3285.
[38] Fidel Thachil, K.C. Shet, A trust based approach for AODV protocol to mitigate black hole attack in MANET, in: 2012 International Conference on Computing Sciences, IEEE, 2012, pp. 281–285.
[39] John Tobin, Christina Thorpe, Liam Murphy, An approach to mitigate black hole attacks on vehicular wireless networks, in: 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), IEEE, 2017, pp. 1–7.
[40] Ozan K. Tonguz, Mate Boban, Multiplayer games over vehicular ad hoc networks: A new application, Ad Hoc Netw. 8 (5) (2010) 531–543.
[41] Iman Vakilinia, Shahriar Badsha, Shamik Sengupta, Crowdfunding the insurance of a cyber-product using blockchain, in: Proc. of the 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York City, NY, USA, 2018, pp. 964–970.
[42] Karan Verma, Halabi Hasbullah, Ashok Kumar, Prevention of dos attacks in VANET, Wirel. Pers. Commun. 73 (1) (2013) 95–126.
[43] Lars Wischhof, Andr Ebner, Hermann Rohling, Self-organizing traffic information system based on car-to-car communication: Prototype implementation, in: International Workshop on Intelligent Transportation (WIT), 2004, pp. 49–53.
[44] Nicholas J. Witchey, Healthcare transaction validation via blockchain, systems and methods, US Patent 10,340,038, 2019.
[45] Zhe Yang, Kan Yang, Lei Lei, Kan Zheng, Victor CM Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE Internet Things J. 6 (2) (2018) 1495–1505.
[46] Xiaohong Zhang, Xiaofeng Chen, Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network, IEEE Access 7 (2019) 58241–58254.
[47] Junping Zhang, Fei-Yue Wang, Kunfeng Wang, Wei-Hua Lin, Xin Xu, Cheng Chen, Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1624–1639.

Sowmya Kudva is a graduate student in computer science and engineering from University of Nevada Reno. She has also worked with Tesla as intern. Her current research interests include improving consensus mechanism in resource limited blockchain based vehicular network.

Shahriar Badsha is an Assistant Professor in cybersecurity at the Department Computer Science and Engineering at University of Nevada, Reno, US since 2019. Before joining the University, Dr. Badsha worked as a postdoctoral researcher at the same Department from August 2018 to June 2019. His main research interests are cryptography and blockchain based applications, privacy preserving (homomorphic cryptography based) applications, access control, and IoT (Internet of Things) privacy. Dr. Shahriar Badsha obtained his Ph.D. degree in Computer Science and Software Engineering, RMIT University, Australia in 2019. He was also affiliated with the data61, CSIRO in Melbourne, Australia. He serves as reviewer in some of the top journals including IEEE Transactions in Information Forensic and Security, IEEE Transaction on Network and Service Management, IEEE Transaction on Services Computing, Computer Networks and Future Generation Computer System.

Dr. Shamik Sengupta is an Associate Professor in the Department of Computer Science and Engineering and Executive Director of the Cybersecurity Center at University of Nevada, Reno (UNR). His research interests include cognitive radio and DSA networks, game theory, cybersecurity, network economics and self-configuring wireless mesh networks. He has authored over 100 international conferences and journal publications. He is the recipient of an IEEE GLOBECOM 2008 best paper award, and an International Symposium on Performance Evaluation of Computer and Telecommunication Systems 2017 best paper award. He is the recipient of NSF CAREER award in 2012 and UNR CSE Best Researcher award in 20152016 and 2017–2018. Shamik serves on the organizing and technical program committee of several IEEE conferences. He is serving in the Editorial Boards of several journals. For more information, please visit: https://www.cse.unr.edu/shamik/.



Hung M. La (IEEE SM'2014, M'2009) received his B.S. and M.S. degrees in Electrical Engineering from Thai Nguyen University of Technology, Thai Nguyen, Vietnam, in 2001 and 2003, respectively, and his Ph.D. degree in Electrical and Computer Engineering from Oklahoma State University, Stillwater, OK, USA, in 2011. He is the Director of the Advanced Robotics and Automation (ARA) Lab, and Assistant Professor of the Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA. From 2011 to 2014, he was a Post Doctoral research fellow and then a Research Faculty Member at the Center for Advanced Infrastructure and Transportation, Rutgers University, Piscataway, NJ, USA. Dr. La is an Associate Editor of the IEEE Transactions on Human–Machine Systems, and Guest Editor of International Journal of Robust and Nonlinear Control.

Ibrahim Khalil is an Associate Professor with Computer Science and Software Engineering, RMIT University, Melbourne, Australia. Ibrahim obtained his Ph.D. in 2003 from the University of Berne in Switzerland. He has several years of experience in Silicon Valley based companies working on Large Network Provisioning and Management software. His research interests are in scalable efficient computing in distributed systems, network and data security and secure data analysis including big data security.

Mohammed Atiquzzaman (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering and electronics from the University of Manchester, Manchester, U.K., in 1984 and 1987, respectively. He currently holds the Edith Kinney Gaylord Presidential professorship with the School of Computer Science, University of Oklahoma, Norman, OK, USA. He is the Editor-in-Chief of Journal of Networks and Computer Applications, the Founding Editor-in-Chief of Vehicular Communications and has served/serving on the editorial boards of various IEEE journals and cochaired numerous IEEE international conferences including IEEE Globecom. His research has been funded by the National Science Foundation, National Aeronautics and Space Administration, U.S. Air Force, Cisco, and Honeywell. His research interests include communications switching, transport protocols, wireless and mobile networks, satellite networks, and optical communications.