Al-Azhar Engineering 16th International Conference

Vol. 19, No. 72, July 2024, 31- 52

SECURING VANETS USING A BLOCKCHAIN-BASED APPROACH WITH PROOF OF TRAJECTORY CONSENSUS

Manal S. Gamal[1*], Sayed A. Nouh[2], Abdurrahman A. Nasr[2]

[1]Department of Electrical Engineering, Faculty of Engineering, 6th of October University, Giza, Egypt

[2]Department of Computers and Systems Engineering, Faculty of Engineering - Al-Azhar University, Nasr City, Cairo, Egypt

***Correspondence:** manal.shehab@gmail.com

Citation:

M.S. Gamal, S.A. Nouh and A.A. Nasr, "Securing VANETs using a blockchain based approach with proof of trajectory consensus", Journal of Al-Azhar University Engineering Sector, vol. 19, pp. 31- 52, 2024.

Received: 28 November 2023

Revised: 09 January 2024

Accepted: 25 January 2024

DoI:10.21608/auej.2024.251815.1494

ABSTRACT

This research focuses on addressing the critical issue of Sybil attacks in Vehicular Ad-hoc Networks (VANETs), which pose serious threats to the security and reliability of communication among vehicles and Roadside Units (RSUs). The proposed solution introduces innovative protocols leveraging concepts from the Secure Electronic Transactions (SET) protocol and Blockchain technology. Notably, the research introduces a novel concept, "Proof of Trajectory (PoTr)," enhancing vehicle authentication and countering identity spoofing—the primary technique employed by Sybil attackers. The study includes implementation details, simulation experiments, and evaluation results demonstrating the effectiveness of the proposed protocols in detecting and preventing various types of attacks, including Identity Spoofing, Man-in-the-Middle, and Replay attacks. The outcomes present VSCI as a robust and secure communication framework for VANETs, contributing to the advancement of security solutions in the context of dynamic and mobile ad-hoc networks.

KEYWORDS: VANET, Vehicular Ad-hoc Networks, Blockchain, Proof of Trajectory, Secure Electronic Transactions protocol, SET protocol, Sybil Attack.

تأمين الشبكات اللاسلكية للسيارات باستخدام تقنية توافق سلسلة البلوكات

منال شهاب احمد[1*] ، سيد عبد الهادى نوح[2]، عبد الرحمن على نصر[2]

[1]قسم الهندسة الكهربيه - كلية الهندسة – جامعة 6 اكتوبر ، الجيزه، مصر

[2]قسم الهندسة النظم و الحاسبات - كلية الهندسة - جامعة الأزهر، مدينة نصر، القاهره

***البريد الاليكتروني للباحث الرئيسي:** manal.shehab@gmail.com

الملخص

يركز هذا البحث على معالجة هجمات Sybil في الشبكات المخصصة للمركبات (VANETs) ، والتي تشكل تهديدات خطيرة لأمن وموثوقية الاتصال بين المركبات والوحدات على جانب الطريق (RSUs). يقدم الحل المقترح بروتوكولات مبتكرة تستفيد من مفاهيم بروتوكول المعاملات الإلكترونية الآمنة (SET) وتقنية Blockchain. حيث يقدم البحث مفهومًا جديدًا، "إثبات المسار" (PoTr) ، لتعزيز مصادقة المركبات ومكافحة انتحال الهوية الذي يعتبر الأسلوب الأساسي الذي يستخدمه مهاجمو Sybil.تتضمن المقالة تفاصيل التنفيذ وتجارب المحاكاة ونتائج التقييم التي توضح فعالية البروتوكولات المقترحة في اكتشاف

ومنع أنواع مختلفة من الهجمات، بما في ذلك هجمات انتحال الهوية (Identity Spoofing) وهجمات الرجل في الوسط (Man-in-the-Middle) وهجمات إعادة التشغيل (Replay). تقدم النتائج VSCI كإطار اتصال قوي وآمن لشبكاتVANET ، مما يساهم في تطوير الحلول الأمنية في سياق الشبكات الديناميكية والمتنقلة.

الكلمات المفتاحية : شبكات المركبات المخصصة، سلسلة البلوكات، إثبات المسار، بروتوكول المعاملات الإلكترونية الآمنة، بروتوكول SET، هجوم Sybil.

1. INTRODUCTION

Vehicular ad hoc Network (VANET) is the most popular real-life paradigm of ad hoc networks in which the nodes are mostly moving vehicles. VANET is a recent promising technology in a ubiquitous environment. It is a specific type of Mobile ad hoc Networks (MANETs).

There is a wide range of applications that are especially designed for VANETs [1] to ensure road safety and to enhance the drivers' comfort. In fact, the range of VANET applications is endless.

These applications usually communicate real time data and information, such as road accident warnings, traffic jam alerts, or even asking for traffic information. Therefore, like all other networks, they are susceptible to various types of malicious attacks that would misbehave the traffic and cause road disasters [2]. Therefore, securing VANETs communication is mandatory.

There are two models of VANET networks, namely, public open VANETs and private closed VANETS. Public street systems follow the public open former model where vehicles are unrelated and mostly do not know each other, hence, having no communication coordination. Roadside Units (RSU) are publicly installed to play the role of coordinating services among vehicles having no leader, where all mobile communications go through the RSU, hence the name centralized V2R. On the other hand, private closed VANETS are mostly between closely related vehicles with a well-defined leader, such as troops, hence, their communication model is mainly decentralized V2V where the vehicles communicate directly with each other with no need for an RSU coordinator. This article focuses more on the centralized V2R. All VANET involved nodes are assumed properly equipped with the necessary computing, storage, and wireless communication devices.

Like all other networks, VANETs communications are susceptible to various types of malicious attacks that would misbehave the traffic and cause road disasters [3]. Therefore, securing VANETs communication is mandatory. Sybil attacks [4] are the most serious, harmful, and hardest to detect types of threats in VANETs as they work on falsifying two communication dimensions, namely, message content tampering and sender identity spoofing. Sybil attackers not only pretend the identity of other nodes of the network but also replicate them and create multiple faked identities to falsify the traffic scenarios. Therefore, if there are many faked nodes in the network sending malicious messages, the drawn traffic situation will, of course, be inaccurate if not incorrect and totally misleading and harmful. Hence, Sybil attackers target the content tampering via identity spoofing.

Therefore, this research focuses on securing VANET networks and applications against Sybil attacks by introducing a VANET secured infrastructure (VSCI) and a set of security protocols aiming at avoiding Sybil attacks. Moreover, and as a second defence line against uncaught attacks, an application development frame of reference is also proposed (VADF) as a preventative approach to vaccinate the application algorithms against Sybil attacks, and through which security holes are detected and hence are prevented. Noteworthy, VADF is outside the scope of this article.

VSCI follows a merge between the techniques of the SET (Secure Electronic Transactions) protocol [5] that is employed to secure against the tampering of message data, while the Blockchain technology [6] is employed to secure against identity spoofing and replication. It is wort noting that the nodes in the original Blockchain model are static (not moving) which is not the case for VANET systems where the nodes are continuously moving and changing their positions, which makes the vehicle position an important factor to trace, hence, it is considered the vehicle transaction rather than the monetary transactions of the original Blockchain system. Clearly, the

Proof of Work consensus does not work for VANET systems, hence, this research introduced the concept of “the proof of trajectory consensus (PoTr)”to enhance the vehicle authentication and avoid identity spoofing.

Few researches that focused on the security of VANET systems are represented in Section 1.1 discusses those researches using ad hoc techniques, while Section 1.2 discusses those researches focusing on Blockchain-based solutions.

1.1. Ad Hoc-Based VANET Security

B. K. Lee, E. H. Jeong, and I. Jung proposes a DTSA (Detection Technique against a Sybil Attack) protocol so that it can provide vehicles with a more secure information for the road situation and the traffic flow among vehicles. The DTSA uses SKC (Session Key-based Certificate) to verify the IDs among vehicles to detect the Sybil attack [7].

J. Grover, M. S. Gaur, N. Prajapati, and V. Laxmi presented a distributed solution based on the use of Received Signal Strength (RSS) for detecting Sybil nodes in VANET. This approach relies on similarity of RSS values of nodes instead of inferring the position of nodes using RSS. This technique is lightweight as it considers only a single parameter RSS value for detecting Sybil attacks [8].

N. Dutta and S. Chellappan proposed a fuzzy time-series clustering based approach that does not require any additional hardware or infrastructure support for Sybil attack detection in VANETs. The proposed technique leverages the dispersion of vehicle platoons over time in a network and detects Sybil nodes as those that are traveling closely in a cluster for an unreasonably long time [9].

B. Su and L. Tong proposed a geographic routing strategy based on trusted nodes to accommodate the change in VANETs to achieve the reliable and fast transmission of emergency messages. The strategy is used to transmit emergency messages in flow and dense traffic conditions [10].

1.2. Blockchain-Based VANET Security

K. Parmar, S. Patil, D. Patel, V. Patel, B. Parikh, and P. Padaria propose a privacy-preserving authentication scheme using blockchain technology. To preserve the privacy of vehicles as well as the authentication of vehicles in VANETs, we need a suitable mechanism. In order to authenticate vehicles, we use blockchain technology that make TA transparent and accountable while authenticating vehicles [11].

K. Bala, R. Upadhyay, S. R. Anwar, and G. Shrimal introduced a Blockchain-based trust management architecture for safe and secure automotive networks. It used a private blockchain and smart contracts to construct a trust assessment blockchain that all vehicles may use to submit feedback comments without fear of privacy breaches or the value of their distributed RSUs being impacted [12].

Tianhong Su, et.al proposed a Blockchain-based privacy protection system for VANETs. They designed a two-way authentication and key agreement algorithm through encryption and signature. They tried to also solve the central dependency problem of traditional VANETs. The system uses malicious behavior voting system to detect the message sent by the vehicle through the Blockchain node, which prevents the vehicle from maliciously publishing illegal location information [13].

Shrestha, et al. proposed a type of Blockchain to resolve critical message dissemination issues in the VANET. By creating a local Blockchain for real-world event message exchange among vehicles within the boundary of a country. They also presented a public Blockchain that stores the node trustworthiness and message trustworthiness in a distributed ledger that is appropriate for secure message dissemination [14].

Hassija, et.al. proposed a distributed Directed Acyclic Graph (DAG) To overcome the VANET security challenges. They viewed a VANET as comprising several requesting vehicles and RSUs. The proposed model is based on advanced Blockchain to provide a strong level of security and data immutability. A sample auction-based smart contract is also proposed to model the V2R cost bargaining for data offloading [15].

Shahid Khan, et.al. proposed a secure trust-based architecture that utilizes Blockchain technology to increase security and privacy of users in VANETs. The proposed Blockchain architecture was developed to mitigate networks attacks while maintaining the privacy of users. They used the timestamp and hash techniques to maintain the freshness of messages delivered. They also used the message rating and credibility approach via the Blockchain technology to prove the trust management among vehicles during information exchange [16].

Yao-Tsung Yang, et. al. proposed a proof-of-event (PoE) consensus applicable to vehicular networks instead of proof-of-work or proof-of-authority approaches. The traffic data are collected through the RSU, while the passing vehicles will verify the correctness when receiving the event notification. They also proposed a Traffic Event Validation (BTEV) framework that employs the PoE consensus mechanism to achieve the reliability of confirming the event occurrences. This framework verifies traffic incidents through vehicles near by the RSU, while accomplishing the role of event alert [17].

Section 2 discusses the proposed infrastructure VSCI and its components, while Section 3 discusses the proposed protocols of communication through VSCI. Section 4 discusses the simulation experiments used for evaluating the proposed solution, while Section 5 conclusion and future work.

2. The Proposed Blockchain-Based VANET Infrastructure (VSCI)

The proposed Blockchain secured communication infrastructure (VSCI) is chosen as a base framework for securing the communication of both open and closed VANET networks, of course after making the suitable adaptations. Because of the inherent security nature of the Blockchain technology and architecture, this research employed its technology to design the VSCI with a central goal to enhance the vehicle authentication and to avoid identity spoofing—the key technique used by the Sybil attackers.

Before defining and discussing new Blockchain model and defining the meanings and new definitions of its elements, let us first shed lights on the idea and base foundation behind the updates made on the traditional Blockchain model.

In VANETs, the monetary-type debit-credit transactions that are traced by the ledger is of no meaning; more importantly is to be able to trace the movement of the vehicle from location to another at which time, i.e., trajectory tracing. This trajectory tracing, when analyzed, can give a picture on the road behavior of the vehicle. Accordingly, the definitions of the Blockchain elements: Transaction, Ledger, and Consensus must be changed to support the new VANET requirement of tracing the trajectory. In the new proposed model, the vehicle's p-tuple (p,t) (the position-timestamp tuple) is the transaction that is maintained in the Ledger; in other words, each time (at certain sampling time period) the vehicle position changes, a p-tuple is added to the ledger, hence the Ledger is maintaining he vehicle's trajectory. Noteworthy, unlike the monetary transactions, the p-tuple transactions do not practice debit-credit directions, but only moving forward in time.

The new concept of trajectory tracing mandates the need for a new consensus (PoTr Consensus), in which the changes of p-tuple at Δt (trajectory transaction ΔT) can be used to validate the message sender identity to avoid Sybil's identity spoofing and replication.

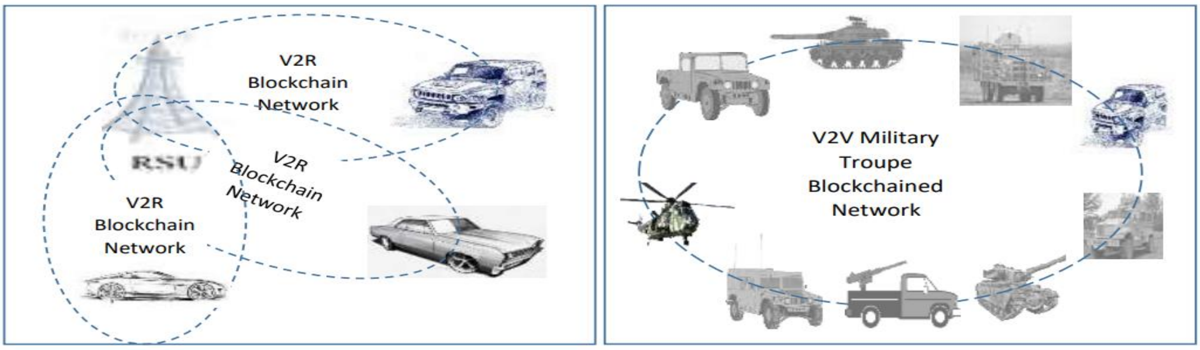
Now, let us discuss the mapping of each of the Blockchain components [18] to their corresponding VANET's VSCI. **Table 1** summarizes this mapping, while the discussion of how each component is adapted to suite the new requirements mandated by VANETs comes next.

Table 1: The Components of the Adapted Blockchain.

Blockchain Components	**Adapted Blockchain Components**
A network node	RSU or a Vehicle
Leader node	RSU in V2R and an elected V (RAFT) in V2V
Block	pt-transaction
Ledger	Trajectory of V
Distributed Blockchain (in V2R)	Private ledger distributed between RSU and V
Distributed Blockchain (in V2V)	Private Ledger for each V distributed among all Vs.
Consensus	Proof of Trajectory consensus
Smart Contracts	SSC and ASC smart contracts
Encryption	Symmetric and Asymmetric
Certificate Authority	RSU and TCA
Adding and revoking a participating node	"Request_for_RPC" smart contract (SSC)

Network Nodes

Each VANET-member vehicle is a Blockchain node. For a vehicle to be a member in the VANET network system, it has to register itself to the VANET system. Two types of registrations take place, one to the Global (countrywide) VANET system by the public traffic authority (TCA) that authorizes the vehicle and give it the license to navigate through the country streets. The second is a registration at a zone-dominating node (RSU) to be authorized to navigate/tour through its specific territory (zone).



a) V2R Blockchain networks.

b) V2V Blockchain distributed network

Fig. 1 : Blockchain-based VANET Network Topologies.

Network Topologies

In the proposed model, two topologies are considered, one for the open public VANETs and one for the closed private VANETs, as shown in Figure 1. As envisioned by the Blockchain technology, the former topology can be depicted as many Blockchain networks of only two nodes, the RSU and a single vehicle, let us call it dual-node Blockchain VANET. Accordingly, there would exist a number of dual-node Blockchain networks (n) as there are vehicles (n) in the zone of the RSU, as shown in **Fig. 1.a**. However, there would be a single Blockchain VANET network for each closed private VANET, as shown in **Fig. 1.b**.

Transactions

For open Public Street VANETs, the most important transaction to maintain in the proposed model is the trajectory of the vehicle within its trip that we call it Movement-Tracing Transactions (MVTT). Each MVTT is a pt-tuple, a tuple (p, t), where p stands for location (x-y for on surface vehicles, or x-y-z for flying vehicles), and t stands for its corresponding timestamp of when the vehicle was at that location. A vehicle's trajectory is traced by tracking the sequence of transactions in order.

Blocks

Transactions together with other useful data e.g., the vehicles speed at the transaction time, are collected and stored in a Block to draw a full picture of the vehicle's behavior. The hash of the previous transaction is also stored in its Block as a checksum for validation purposes.

Block chain

The transaction Blocks for each vehicle are chronologically ordered and chained together in sequence of occurrence. Each Block is given a sequence number (its order in the chain) for synchronization purposes.

Ledger

The Ledger is a chain of the Blocks of transactions over a certain defined period, e.g., a tour within a certain zone. Simply, a ledger is a set of such MVTT transactions. If the transactions are continuously collected at Δt sampling time, then the ledger represents a full historical tracing of the timestamp-annotated trajectory of the vehicle along its trip, which gives a full live picture on the behavior of the vehicle in motion.

One can think of the vehicle ledger as an improved odometer that can be built-in into vehicles (a recommendation to manufacturers) with the aid of a built-in GPS. Otherwise, **a smart contract software** for ledger management and maintenance should be installed in each vehicle to maintain and communicate such transactions with each corresponding zone-managing RSU. Like odometers, ledgers can be reset before each trip or can be maintained for the vehicle lifetime, or even can be reset at the entrance of a new zone, if memory is of concern. The ledger can be represented as follows:

LV_y = the private ledger for Vehicle $V_y = \{T_{xij}\} = \{(R_x, p_i, t_j)\}$, where: $T_{xij} = T_y$ is a single pt-tuple transaction (p_i,t_j) for vehicle y at zone x; R_x is the current zone-id (RSU-id, assuming that each zone is managed by one single RSU); p_i represents the coordinates of the current location of vehicle V_y; and t_j is the timestamp of when the vehicle was at location p_i.

LR_x = the public ledger for RSU $R_x = \{LV_{yij}\}= \{(V_y, p_i, t_j)\}$, i.e., the set of all ledgers of all vehicles y inside the associated zone as maintained in the private ledger of those vehicles. Each RSU maintains only the ledgers for those vehicles at its zone.

Proof of Trajectory Consensus (PoTr).

In the proposed solution, there are two types of consensuses, namely, the Current Position Consensus (CPC), and the full Zone Trajectory Consensus (ZTC). The former consensus is simpler as it considers the last n transactions (at least 2) to describe the trajectory, e.g., the last/current transaction. Noteworthy, the current pt-tuple can be inferred by other nearby vehicles; hence, CPC is not secured enough in terms of Sybil attacks.

The two types of consensuses can be described as follows:

CPC_{yj} = The CPC for vehicle V_y at the current time t_j
= $H(T_{yj})$, where T_{yj} is the current or last transaction in the private ledger of vehicle V_y,

ZTC_{xy} = The ZTC for a vehicle V_y at zone Z_x at the current time t_j
= $H(LV_{xy})$, where LV_{xy} is the full trajectory ledger for vehicle V_y regarding its stay at zone Z_x.

If the vehicle resets its ledger each time it enters a new zone, then the query is simpler as it takes the whole trajectory ledger.

H, in the above formulas is a hash function that generates a unique key of a predefined length. This unique key can be used as a symmetric key unique for each transaction/vehicle, a key that both the RSU and the vehicle can generate at its side without a need for exchange.

Smart Contracts (SC).

There are two types of Smart Contracts (SC) according to the proposed protocol. The first type of SCs is called System SCs (SSC) as they are at the core of the proposed security protocol, and hence, they are mandatory for all vehicles and must be active and running for each vehicle to get involved in the proposed Public Open VANET system. A vehicle must register itself at the time it joins the VANET in order to be able to communicate with the VANET, at which time the necessary SSCs are downloaded to the vehicle's computing system and then activated so that the vehicle becomes a VANET member to leverage all the VANET services.

Smart Contract Invocation and Execution Model.

In our proposed VSCI, each Smart Contract has predefined invocation conditions (triggers), besides its execution code, which must be defined before the smart contract can be plugged into the VANET system. The smart contract is automatically invoked and executes immediately when its invocation conditions are satisfied, which is the responsibility of the implementation of the proposed VANET infrastructure; hence guarantees robust, efficient, uninterrupted execution to assure its expected behavior.

Suggested System Smart Contracts (SSCs).

Table 2 lists some of the SSCs as used in this research:

Table 2: SSCs as used in this research.

SSC	Description	Invocation Conditions
Register-to-VANET( )	Register the vehicle to become a member of the VANET. It is the responsibility of the public traffic authority TCA.	a request issued by a vehicle *v* to join VANET
Request-Navigation-Passport (LIC)	When a vehicle enters a new zone z-id, this smart contract is automatically activated by the underlying SSC of the VANET management system installed at the vehicle by the "Register-to-VANET" SSC. If the vehicle is not registered to VANET, this SSC is not activated. This SSC is automatically sent to the appropriate RSU Rz that is managing zone z-id zone. Rz will in turn do the necessary validations and then issues a temporary roaming passport RpCv,z allowing the vehicle to communicate with the Rz during its stay in the zone.	A vehicle passed through the borders of a new zone, which can be detected by the RSU that, in turn, sends an alert to the vehicle to issue the request
Register-to-Application (App-id)	A vehicle must register for the use of VANET services. This SSC allows the vehicle to download the application software and its corresponding daemon functions, hence become ready to use the application.	The vehicle issues a request to register in the application
Register-to-VANET( )	Register the vehicle to become a member of the VANET. It is the responsibility of the public traffic authority TCA of the country, which reviews all physical credentials and issues the vehicle's license.	A request issued by a vehicle v to join VANET.
Upgrade & Download-Daemons (Application-id)	This function is used by the VANET system to push software upgrades to the vehicle.	When an upgrade is available in the cloud for a specific application.
Maintain-Ledger ( )	This SSC software works unattended. It continuously updates the vehicle's private ledger at predefined periods of Δt, and then communicates each new transaction to the corresponding RSU by sending a message Update-Ledger (T).	Every preset time interval Δt a scheduler invokes the SSC.
Get-Current-Transaction ( )	It issues a new transaction, add it to its local ledger, and then communicates it to the requesting RSU. It is required for those vehicles that do not have a manufacturer-provided updated odometer. This SSC works only under the control and upon a request from the RSU in contrary of the Maintain-Ledger SSC that works continuously and initiates the communication.	Automatically invoked at the time of sending a new message.
Generate-Embedded-Symmetric-Key (LV)	It issues a secret key by hashing the vehicle ledger LV.	At the time of preparation of a message to send or at the time oof receiving a message.
Verify-Vehicle-location (ID, Reported-location)	This daemon function verifies whether the vehicle can possibly be in a nearby location to the reported-location.	At the time, a message is received.

3. The Proposed VANET's VSCI and Protocols

One of the prominent attacks in VANETs is the Sybil attack, in which the attackers create multiple false identities to disturb the functionality of the VANET. Sybil attackers work along two dimensions: Replicating faked nodes and Interfering with messages sent by honest vehicles. Hence, an avoidance and prevention set of protocols against Sybil attacks are proposed, at the heart of which is the PoTr consensus that combines three common security techniques, namely, trusted certification, position verification, and time stamping to give more accuracy during VANET communications.

The proposed secured communication infrastructure works on three dimensions: namely, authenticating sender identity, verifying the message integrity, and Proof-of-Trajectory consensus PoTr.

The Proposed V2R Attack-Avoidance Protocol—an Overview.

Fig. 2. summarizes the communication protocol for V2R network model as envisioned by the proposed solution. The proposed solution avoids and detects against Sybil attacks through two stages: **Stage 1:** authorizes the vehicle to join a VANET system, which without it no communication takes place, **Stage 2:** works at the time of receiving a new message.

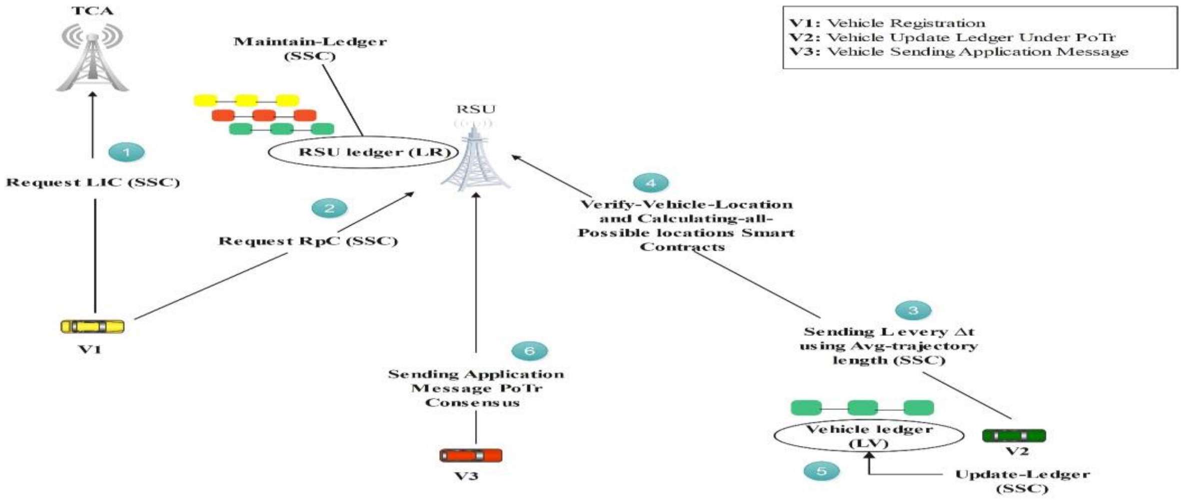

TCA
Maintain-Ledger (SSC)
V1: Vehicle Registration
V2: Vehicle Update Ledger Under PoTr
V3: Vehicle Sending Application Message
RSU
RSU ledger (LR)
1
Request LIC (SSC)
2
Request RpC (SSC)
4
Verify-Vehicle-Location and Calculating-all-Possible locations Smart Contracts
V1
3
Sending L every Δt using Avg-trajectory length (SSC)
6
Sending Application Message PoTr Consensus
Vehicle ledger (LV)
V2
V3
5
Update-Ledger (SSC)

Fig. 2: V2R Message Communication in Open Public VANETs.

It is worth noting that this two-stage protocol uses different types of secrete keys:

1. The public and private keys of the TCA, namely, K_{PUt}, K_{PRt}, respectively. These are given and their management are beyond the scope of this research.
2. The public and private keys of the RSU, namely, K_{PUr}, K_{PRr}, respectively. These are given and their management are beyond the scope of this research.
3. The public and private keys of the Vehicle *v* as obtained from the TCA-generated LIC, namely, K_{PUv}, K_{PRv}, respectively.
4. The public and private keys of the Vehicle *v* as obtained from the RSU-generated RpC, namely, K_{PUv}, K_{PRv}, respectively.
5. The symmetric key K_S = a symmetric key that is generated by the RSU in the algorithm of exchanging the RpC with the vehicle. This symmetric key has to be communicated between the two entities.
6. Embedded Symmetric key EK = HF (LV_x), where HF is a hash function that hashes the vehicle ledger. This symmetric key is computed by both RSU and the vehicle, and it does not require exchanging it between them. There can be different versions of the this computed symmetric key by taking as many parameters as the complexity of the key is required, e.g., HF (LV_x, LIC, RpC).

Stage1: Vehicles Identity Registration and Certification

Each RSU is considered the sole authority responsible for authenticating vehicles participating in the VANETs network at its territory (zone). The key to authentication is the unique temporary PKI-based certificate (RpC) that is specially issued for each legal vehicle at an entry point to the territory. The proposed protocol uses a two-level certification system—namely, (1) the TCA's LIC and (2) the RSU's RpC certificates, respectively—to strengthen the authentication security. **Fig. 3.** demonstrates the sequence of events of how the two-level registration protocol works for zone registration. Noteworthy, an RpC is never issued unless the vehicle is legally authenticated by using its LIC.

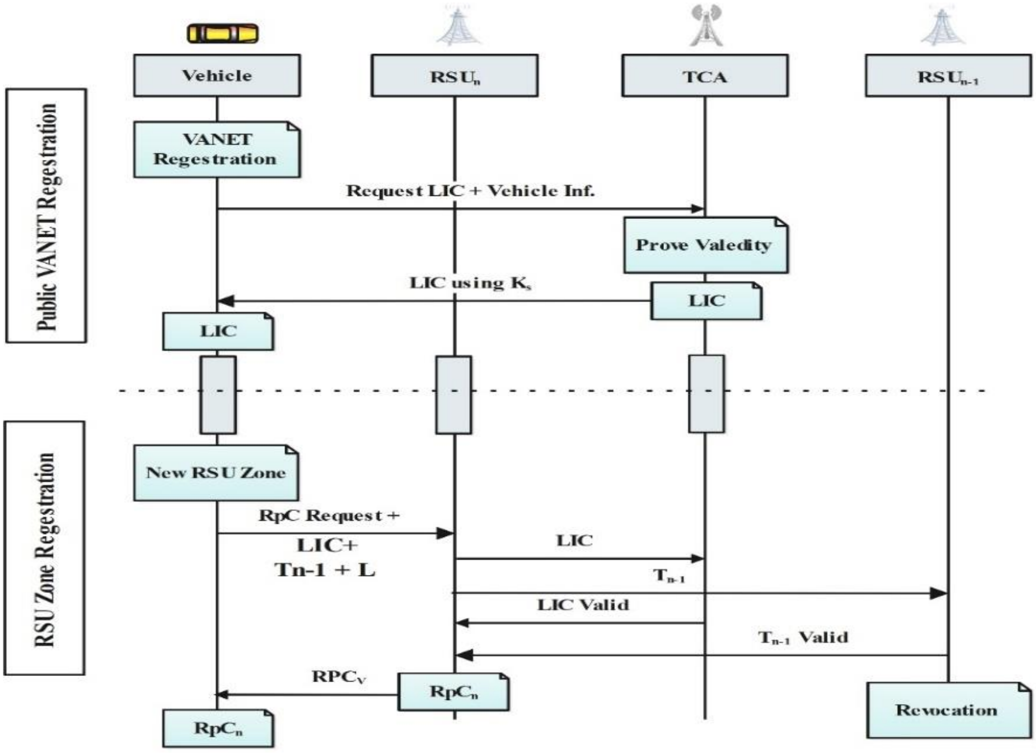


Fig. 3: The Two-Level Vehicle Registration Protocol.

The "Register-to-VANET" Smart Contract Algorithm.

Vehicles have to, first, register to the VANET system to become an active VANET member who is able to communicate with the countrywide VANET. The processor of this step is the public traffic authority TCA. If the vehicle satisfies all the legal requirements, it will be granted a valid LIC—the output of the process—however, the registration fails otherwise.

The *"Request-Navigation-Passport"* Smart Contract Algorithm.

In this smart contract, three entities are involved, namely, The Current RSU R_n and the previous RSU R_p and the TCA, as shown in **Fig. 4**. The request of issuing a Roaming pass is received by R_n that takes the initiative of leading the process. R_n, then, communicates with the other two entities, namely, TCA to validate the legal status of the vehicle, and R_p to "hand-in the stick" (as in relay racing) and to verify the last position p_l at the previous zone to get assured that the vehicle can realistically pass to the current position p_c at the given elapsed time. The "Request Roaming Pass" SSC .

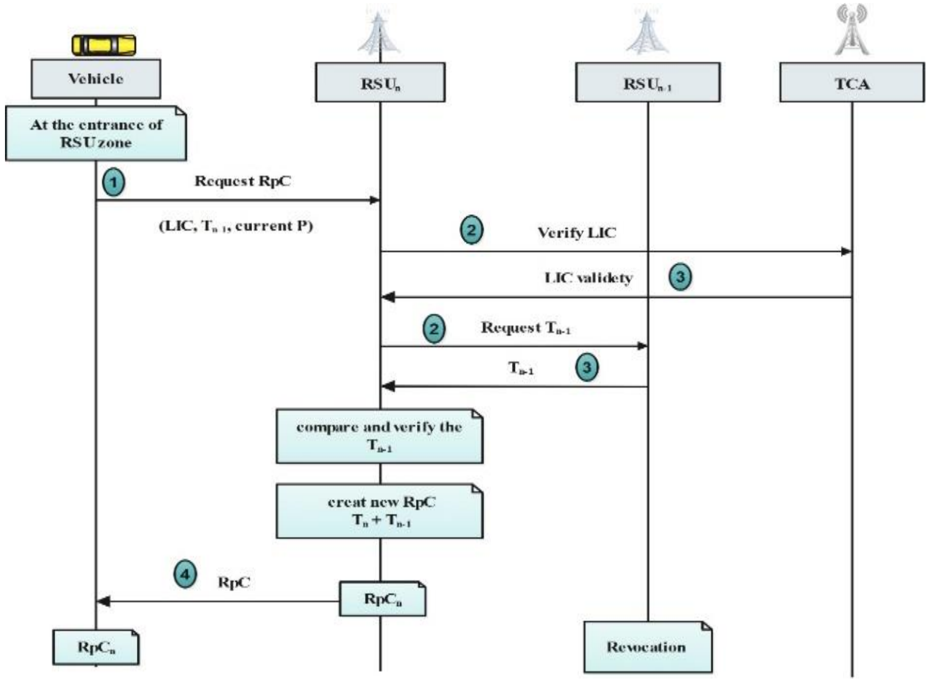


Fig. 4: The “Request-Navigation-Passport” Smart Contract Process.

The “Request Roaming Pass” process can be described as follows:

1. When a Vehicle *v* enters a new RSU zone R_n, it sends a communication request to the RSU concatenated with its LIC certificate, the last transaction T_{n-1} at the previous zone z_{n-1}, and its current location at the entry point to the new zone z_n.
2. RSU should prove the validity of the vehicle by:
   a. Sending the LIC certificate to TCA to confirm its validity, and
   b. Sending T_{n-1} to the previous RSU R_{n-1} to confirm its credibility.
3. Upon authenticating the vehicle identities by both TCA and R_{n-1}, and verifying the last transaction T_{n-1}:
   a. R_n authenticates the vehicle and issues the RpC_v.
   b. The RpC_v is then securely communicated to *v* using the R2V secured communication protocol as described later.
   c. A new Ledger is then created at both R_n and *v*.

Communicating the Roaming Passport.

The RpC generated by an RSU for a vehicle *v* must be securely communicated to the vehicle. This section proposes a secured communication protocol for sending the RpC to the vehicle, as shown in **Fig. 5**. This protocol is again inspired by the SET protocol. It employs the PKI, hashing, digital signature, and digital envelop techniques.

As shown in **Fig. 5**, the RpC certificate is digitally signed (K_r) by being encrypted using the RSU’s private key (E_{PRr}). K_r is then hashed into a digest $H(K_r)$ that is then concatenated with the digital signature K_r and the RSU’s public key PU_r to compose a composite message Y. Y is then put in an envelope Z using the RSU-generated symmetric key S. It is worth noting that S is used here because the ledger LV_t at the time of issuing the RpC certificate has only one transaction (entry point of the zone), which makes it not strong enough Embedded Symmetric key, especially because there might be a sybil attacker near to the vehicle at that time, hence it can predict the transaction. Therefore, to strengthen the Embedded Symmetric key, LV may include few more past transactions to start with.

$K_r = E_{PR_r}(RpC)$,
$Y = Kr \| H(Kr) \| PU_r$,
$Z = E_S [Y]$.

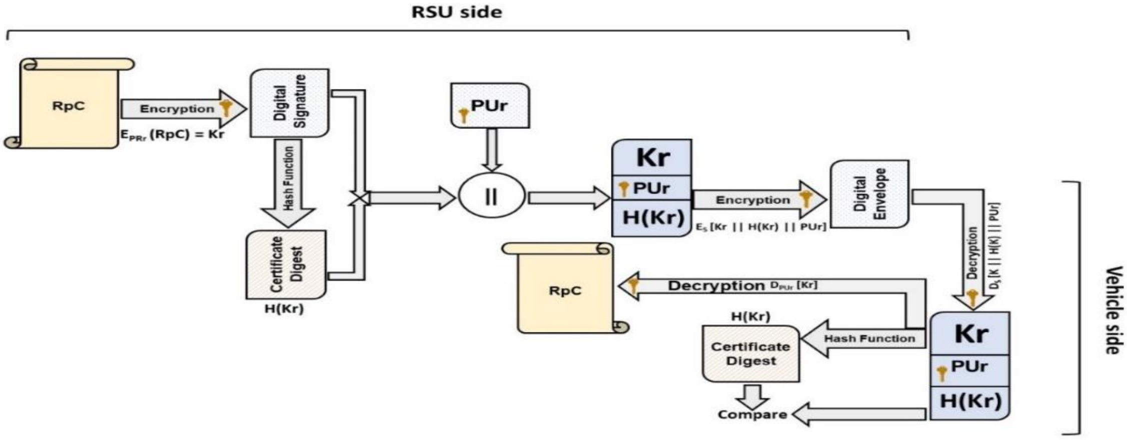


Fig. 5: The Proposed Protocol for Communicating the RpC Certificate.

On the other side (the Vehicle v receiver side), the vehicle *v* receives the digital envelop Z and does exactly the inverse. It decrypts Z using the computed symmetric key S and then deconcatenates Y into its three constituting elements: the RSC public key PU_r, the hashed digest $H(K_r)$ and the encrypted certificate (K_r). The encrypted certificate goes through two processes the first is decrypting using the RSU public key (D_{PU_r}) reaching the RPC and the second is hashing the received digital signature to be compared with the received hash digest if they match then the vehicle is sure that this is the RPC given by the RSU.

$R = D_s [K_r \| H(K_r) \| PU_r] = K_r, H(K_r)$ and PU_r
$D_{PU_r}(K_r) = RPC$

Noteworthy, this protocol contains two extra steps that are used for exchanging the LIC. First, the public key of the sender RSU is sent with the message because there are unlimited number of RSUs in the VANET unlike the case when the sender is the TCA. The second difference is the use of the symmetric key and envelop, since Sybil attackers are more interested in interfering with the messages to corrupt the behavior of the VANET at a certain zone, and they have no interest in the legal registrations and licensing.

Stage 2: The Message Sender's Authentication and Contents Verification.

All messages received by either the RSU or a vehicle are assumed vulnerable until both the sender and the message content are authenticated, otherwise the message is refused, and the sender is requested to resend the message.

Noteworthy, in PoTr consensus, each Vehicle maintains its unique ledger LV_R of all movement transactions within a certain zone. Since this ledger is unique for each vehicle, so will be its hash. Therefore, any of the following formulas can be used to generate a unique hash based on the required complexity:

$S = HF(LV_v)$,
$S = HF(LV_i, LV_{i-1}, LIC_v)$.

Notice that the vehicle's unique ledger LVv is synchronized and maintained at both sides of the RSU and the vehicle itself, and only at those two entities. Therefore, the generated hash S can be used as a strong embedded symmetric key ES that does not require to be exchanged between the RSU and the vehicle as it can be computed by each entity separately.

If a Sybil attacker can guess the last transaction of a certain vehicle, it cannot guess the whole ledger; hence, the hash of the ledger can be securely used as a strong unique symmetric key that does not require exchange between the RSU and the vehicle since it can be computed at each side. In addition, this symmetric key lifetime is very short, as it changes each time a new message is exchanged due to the continuous change of the ledger. This makes the protocol of message exchange between RSU and Vehicle (R2V) simple and AES [5] can be used gaining all of its advantages, as shown in **Fig. 6**.

Before a message is sent, it has to be:

a. Digitally signed by the sender's private key to assure the identity of the sender, then the signed message is:
b. Ciphered using the computed Embedded Secrete key ES to assure the integrity of the message contents, which is then
c. Digitally enveloped by being encrypted using the public key of the receiver to assure no intrusion by malicious Sybil attackers.

When the message is received, the algorithm is reversed with an extra two validation steps, one comes at the beginning and the other at the end, as below:

a. Decrypted using the private key of the receiver. If not properly decrypted, then the message is refused as it is assumed sent by a Sybil node who manipulated the original message, then
b. It is deciphered by guessing and computing the symmetric key. if it is not properly deciphered then the assumption is that the ES key is malicious and the message contents are tampered, hence the message is refused, then lastly,
c. It is decrypted using the public key of the sender. If the key did not work then the sender identity is assumed suspicious, and hence, the message is refused.
d. Verifying the application-specific information received in the body of the message using the especially designed ASCs.
e. Verifying the vehicle location SSC is applied to verify that the current location of the vehicle is reachable from the last received transaction within the time interval using the PoTr consensus.
f. If the message is accepted , validate the messages information content using the appropriate ASC, which is application dependent, though automatically invoked by VSCI.

Therefore, if the received message is refused for any reason, the sender is requested to resend the message.

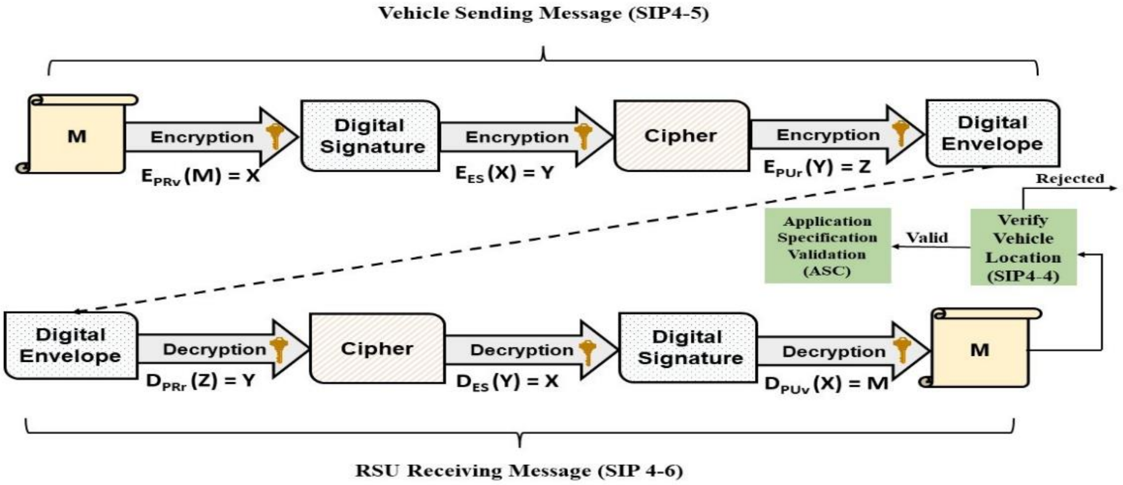


Fig. 6: The Proposed Message Authentication Protocol.

Ledger Update

Since the ledger is very crucial and is at the center of the proposed protocol, its accuracy and integrity must be guaranteed. This section discusses those special issues regarding the update of ledger in such a way to avoid updates with malicious transactions.

Ledger transactions must be continuously synchronized between the vehicle and the RSU in charge. The transactions are created by the vehicle at intervals of time Δt, and is then sent to the RSU as a special type of message (Update-Ledger) to update its version of the vehicle ledger. The Update-Ledger SSC works like all other messages, but with few extra validations of the contents (the transaction). In other words, the message body content is validated after the message is being accepted through passing the three validation steps of PoTr consensus validation, identity authentication, and content integrity verification.

After validating the message by the proposed VSMI receive message protocol and before updating the ledger of the sending vehicle at the receiving RSU, the following actions take place:

1. The transaction# sent with the transaction message must be verified. To explain, if the transaction# of the received message has the value of i+1, while that of the last transaction saved in the ledger is <= i-1, then there are missing transactions that are assumed either lost in their way of communication, or a Sybil attacker had wrongly guessed it. Therefore, the update message is rejected, and the sender is requested to resend the missing transactions first. This is the idea of the validation of the transaction#, however, the handling details are left to the VANET designer since there may exist many possible scenarios.
2. The checksum is validated next. It is the hash of the last saved transaction (with transaction# = i-1). Therefore, the validation algorithm does the hashing of the last stored Block, and then match it with the received checksum, if does not match, then the message is rejected.

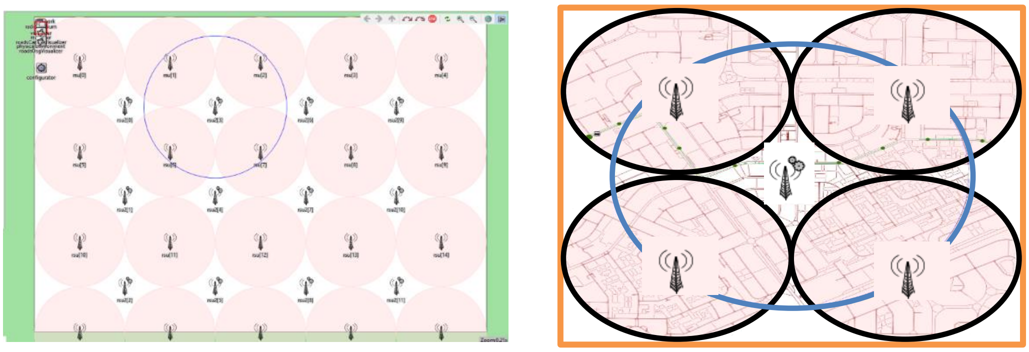
If the two extra validation steps pass, the RSU composes the new Block, and then adds it to the block chain of the sender vehicle.

4. Implementation, Simulation, and Evaluation

A simulation environment was set for evaluation purposes. The evaluation has two folds: assessment of the computational requirements as well as the solution defense ability. This environment was made of three main types of simulators, namely, VANET, Network, and Mobility simulators. This research used OMNet++ version 5.6.2, and SUMO version 1.12.0 for Windows 64, respectively. In addition, OMNet++, in turn, uses two libraries, namely, INET version 4.2.5 and Veins version 5.1.

4.1. The Architecture of the Simulated VANET

The geographical area of the 5th settlement of the New Cairo City is chosen for the experimental simulation because of its clear urban design with wide streets and moderate traffic density, **Fig. 7**. shows the 32 RSUs distributed to cover the whole selected area with 15 Master RSUs and 17 Sub-RSUs.



a) Master and Sub-Master RSUs Map.

b) Zones Intersection.

Fig. *7***:** Intersecting Zones of the Master and Sub-Master RSUs.

Moreover, vehicles are simulated, where each has an id and a predefined static route that specifies the vehicle's trip inside the simulated area. The movement of a vehicle is synchronized between SUMO and Veins via the mobility module of Veins and the road traffic simulation of SUMO.

4.2. The Simulation Process and Parameters

The messaging protocol of the proposed VSCI is the most intensively used protocol during the operation of the VANET if compared to those other protocols, e.g., vehicle registration protocols. It communicates all types of information messages including Ledger-Update messages and application messages; hence, it is the chosen protocol for simulation. **Fig. 8** depicts the simulated messaging protocol showing all of its components that were implemented as part of the simulation environment.

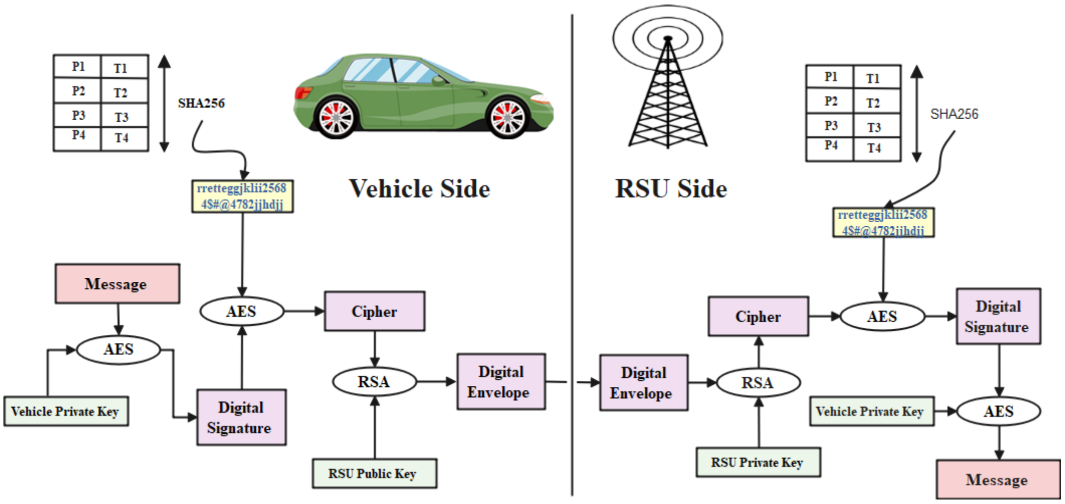
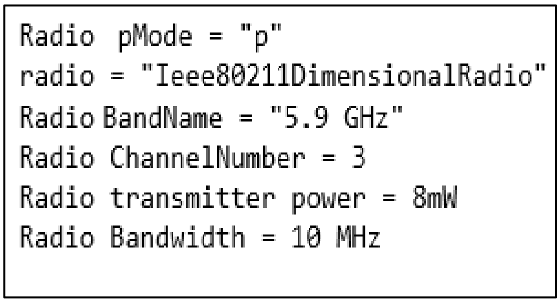


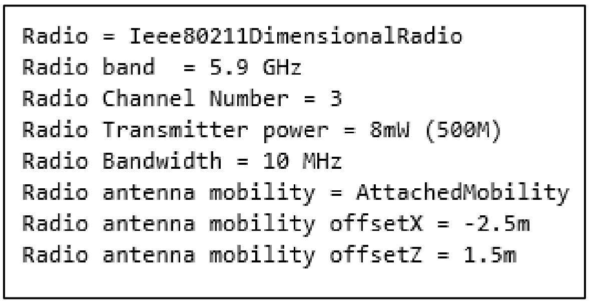
Fig. *8*: The Simulated Messaging Protocol.

The simulated RSU's communication parameters are shown in **Fig. 9**, while the vehicle's communication parameters are shown in **Fig. 10**. The vehicle acceleration is assumed = 2.6 m/sec2, while its deceleration = 4.5 m/sec2.



```
Radio  pMode = "p"
radio = "Ieee80211DimensionalRadio"
Radio BandName = "5.9 GHz"
Radio ChannelNumber = 3
Radio transmitter power = 8mW
Radio Bandwidth = 10 MHz
```

Fig. 9: Simulated Master RSU's Communication Parameters.



```
Radio  = Ieee80211DimensionalRadio
Radio band  = 5.9 GHz
Radio Channel Number = 3
Radio Transmitter power = 8mW (500M)
Radio Bandwidth = 10 MHz
Radio antenna mobility = AttachedMobility
Radio antenna mobility offsetX = -2.5m
Radio antenna mobility offsetZ = 1.5m
```

Fig. 10: Simulated Vehicle's Communication Parameters.

4.3. Attack Simulation and Penetration Test

Three types of attacks are simulated, namely, Identity Spoofing, Man-in-the-middle, and Replay attacks.

- **Identity Spoofing attack**. Authentication is the verification of the identity between vehicles and RSUs and the validation of integrity of the information exchange. Additionally, it ensures that all vehicles are the right vehicle to communicate with it. This type of attack destroys two important security conditions, which are authenticity of the sender and integrity of the message. The scenario of handling the simulated identity spoofing attack is shown in **Fig. 11**.
- **Man-in-the-middle attack**. MITM attack is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. The attacker must be able to intercept all relevant messages passing between the two victims and inject new ones. The scenario of handling the Man-in-the-Middle attack is shown in **Fig. 12**.

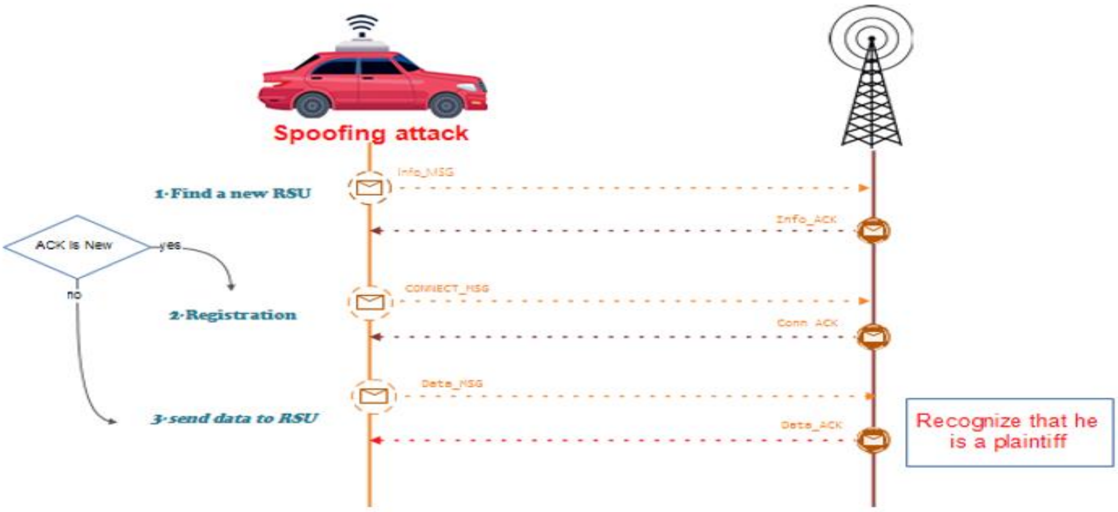


Fig. 11: The Scenario of handling the Identity Spoofing Attack.

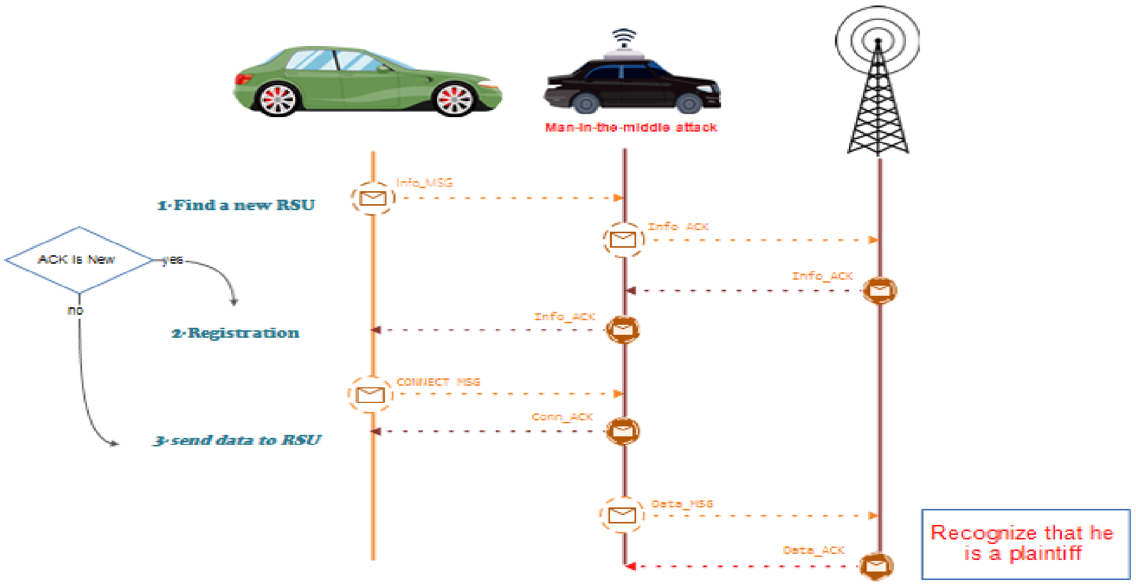


Fig.12: The Scenario of handling Man in the Middle Attack.

- **Replay attack**. A replay attack is a kind of man-in-the-middle attack in which an attacker sniffs messages being sent on a channel to intercept them and resend them under the cloak of authentic messages. What makes the replay attack particularly harmful is that the attacker does not even need to decrypt the message they resend but can still fool the receiver into thinking that the received message is legitimate. The scenario of handling the simulated Replay attack is shown in **Fig. 13**.

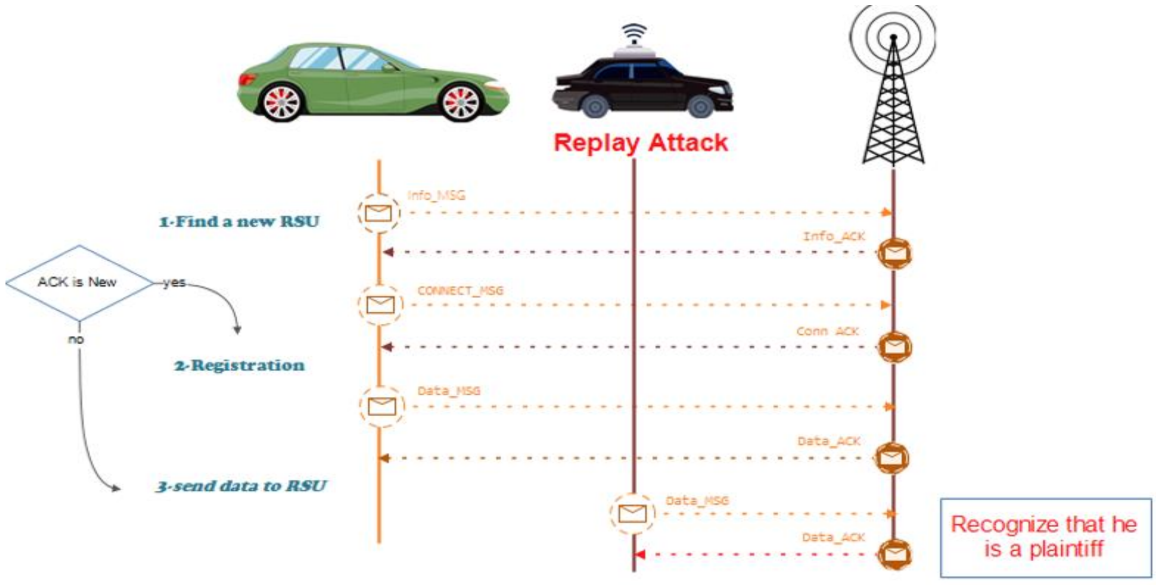


Fig. 13: The Scenario of handling the Replay Attack.

The results of this experimental simulation of the three types of attacks, are shown in **Fig. 14**, **Fig. 15**, and **Fig. 16**. In the simulation experiment, 5 attackers were simulated for each type of attacks.

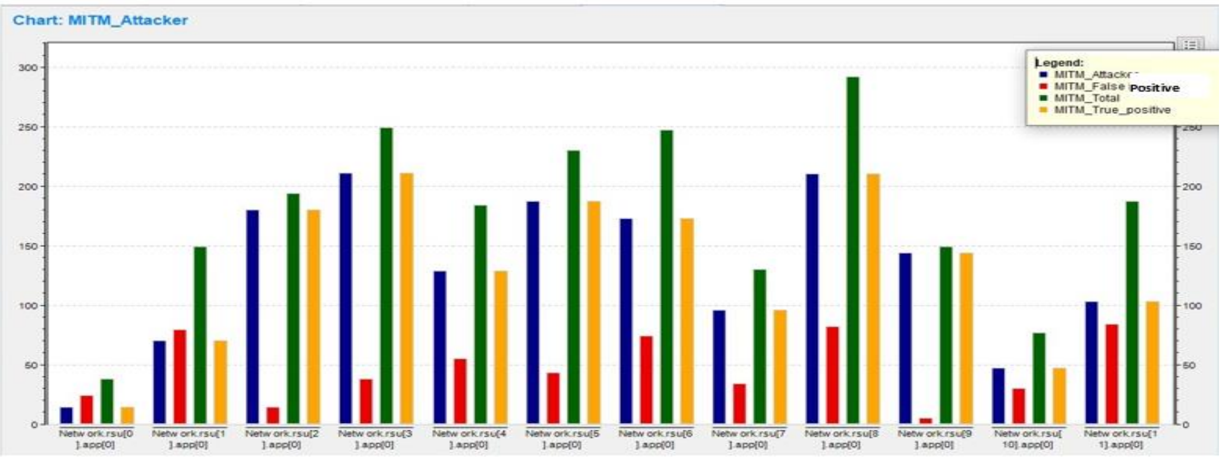


Fig. 14: The Simulated Man-in-the- Middle Attack on the level for each of the RSUs.

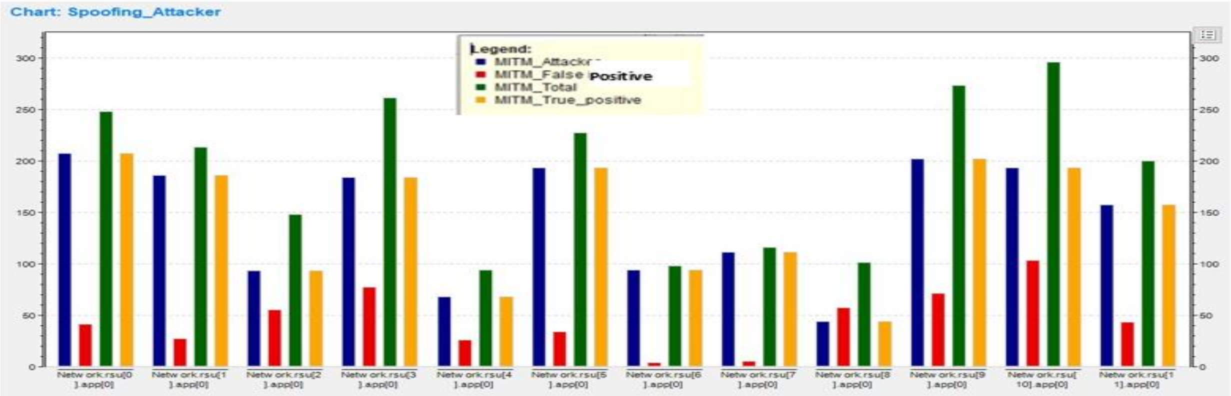


Fig. 15: The Simulated Identity Spoofing Attack on the level for each of the RSUs.

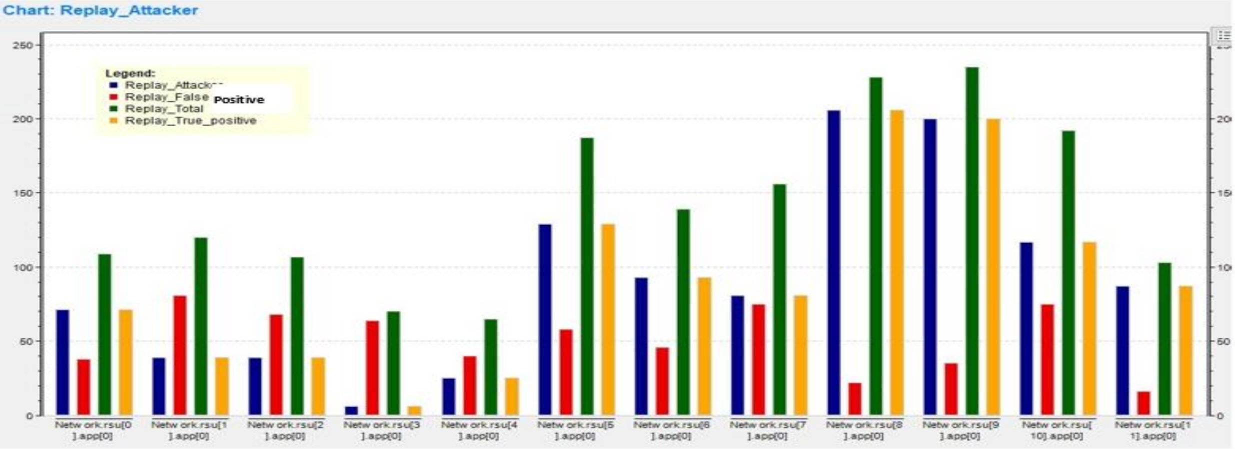


Fig. 16 The Simulated Replay Attack on the level for each of the RSUs.

Table 3 shows the confusion matrix for all attacks for the whole simulation experiment over the whole network. The simulation results show that the proposed "Send-Receive" protocol is robust enough where none of the attacks for the three simulated attack types had been able to breakthrough without being detected and rejected. Unfortunately, it had also rejected some other good messages as shown in the matrix.

Table 3: The Confusion Matrix.

TP	FN
1.0	**0**
0.072	**0.928**
FP	**TN**

4.4. Performance Analysis

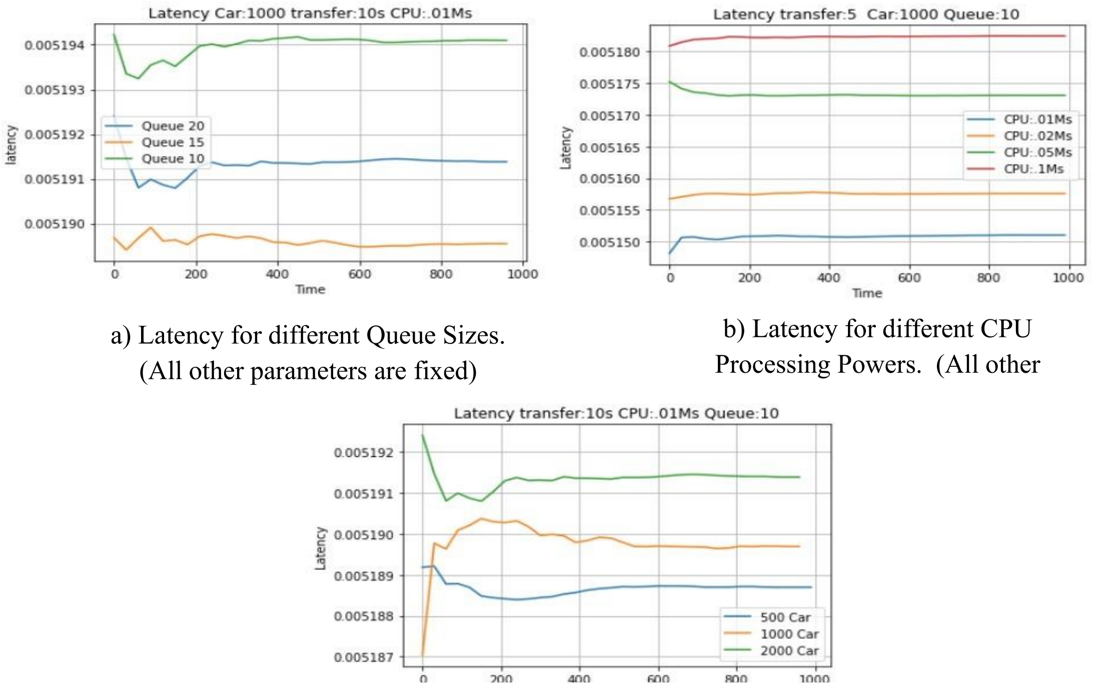
A large number of simulation experiments were conducted for performance evaluation of the simulated proposed solution. The most important factor to measure was latency that is defined as the time taken form the first timestamp of sending a message by a vehicle until the time of receiving the acknowledgment after the message is being processed by the RSU. **Table 4** shows the most important factors affecting the performance, where different values were tried in the simulation experiments for each parameter in order to assess the impact of the changes of that parameter on latency.

Table 4: The Values of the Important Factors as Varied in the Different Runs of the Simulation Experiments.

Elements	5th Settlement of new Cairo City (5km X 3.5Km)			
RSU Range	250 m	500 m	750 m	1000 m
No. of RSUs	70	20	12	6
Transmission Rate (No. of received messages by RSU per sec.)	5	10	15	2
No. of Vehicles	100	500	2000	5000
CPU Capacity (Processing time msec./message)	0.01	0.02	0.05	0.1
RAM Queue Size (Max. no. of messages)	10	15	20	25

Fig. 17 shows a sample of the results of some experiments, to give an idea how the parameter change would affect the system performance. In each experiment, all parameters were kept fixed while only one parameter changes.

It was evident that increasing the processing power and the message queue size would improve the latency. In addition, as the number of vehicles decrease as the performance is better, same as the rate of messaging.



a) Latency for different Queue Sizes. (All other parameters are fixed)

b) Latency for different CPU Processing Powers. (All other

c) Latency for different Number of Vehicles. (All other parameters are fixed)

Fig. 17: Latency for Different Simulation Parameters

5. Conclusion and Future Work

Security vulnerability in VANETs threatens people's lives. Therefore, this research has proposed a secured communication infrastructure (VSCI) for VANET nodes to communicate securely. This infrastructure builds on the techniques of the SET protocol, PKI infrastructure, and Blockchain technology. In this framework, each vehicle maintains its own private ledger that contains the pt-transactions representing the vehicle's trajectory, hence, introducing the concept Proof of trajectory consensus (PoTr). Another copy of the private ledger for each vehicle is also maintained by the central controlling unit (RSU) that maintains the ledger of all vehicles as long as they are inside its territory (controlled zone).

The proposed infrastructure assumes both V2R and V2V secured message exchange. It detects Sybil faked pretended identity via a complicated send-receive protocol based on the concepts of PKI and SET. This same protocol also verifies the message content before accepting it for assuring its integrity and being untampered.

In the case of a V2R communication, the proposed infrastructure assumes that the RSU is the central moderator that is responsible for message validation and verification, however, in the case of V2V, a distributed model is proposed where the decision on the message validity is shared among all network nodes with a voting mechanism.

Finally, the proposed protocols are implemented as a proof of concept, and simulation and evaluation experiments were conducted.

Future research directions the proposed model of VSCI secure against many types of attacks as proved by the experimental simulations, namely, man in the middle, replay, and identity spoofing, which motivates researchers for further investigation on how other types of attacks would impact and update the VSCI model and its proposed protocols.

Another future research question to investigate is "How the vehicles as well as the sub-RSUs take part of the role of the Master RSU" to reduce its load and hence reduce the cost of the establishment of the VANET by, may be, reducing the computing power of the RSUs.

References

[1] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, “VANet security challenges and solutions: A survey,” Veh. Commun., vol. 7, no. January. Elsevier Inc., pp. 7–20, 2017, doi: 10.1016/j.vehcom.2017.01.002.

[2] M. S. Sheikh and J. Liang, “A comprehensive survey on VANET security services in traffic management system,” Wirel. Commun. Mob. Comput., vol. 2019, 2019, doi: 10.1155/2019/2423915.

[3] A. Singh and S. Kad, “A Review on the Various Security Techniques for VANETs,” Procedia Comput. Sci., vol. 78, pp. 284–290, 2016, doi: 10.1016/j.procs.2016.02.055.

[4] J. GROVER, M. GAUR, and V. LAXMI, “Sybil Attack in VANETs Detection and Prevention,” Secur. Self-Organizing Networks, no. July, pp. 269–294, 2010, doi: 10.1201/ebk1439819197-15.

[5] W. Stallings, "Cryptography and Network Security (Principles and Practice)", Fifth edition, 2011.

[6] D. B. Rawat, V. Chaudhary, and R. Doku, “Blockchain Technology: Emerging Applications and Use Cases for Secure and Trustworthy Smart Systems,” J. Cybersecurity Priv., vol. 1, no. 1, pp. 4–18, 2020, doi: 10.3390/jcp1010002.

[7] B. K. Lee, E. H. Jeong, and I. Jung, “A DTSA (detection technique against a sybil attack) protocol using SKC (session key based certificate) on VANET,” Int. J. Secur. its Appl., vol. 7, no. 3, pp. 1–10, 2013.

[8] J. Grover, M. S. Gaur, N. Prajapati, and V. Laxmi, “RSS-based Sybil Attack Detection in VANETs,” IEEE J. Sel. AREAS Commun., no. December 2015, pp. 2278–2283, 2010.

[9] N. Dutta and S. Chellappan, “A Time-series Clustering Approach for Sybil Attack Detection in Vehicular Ad hoc Networks,” Veh. 2013 Second Int. Conf. Adv. Veh. Syst. Technol. Appl., no. February, pp. 35–40, 2013.

[10] B. Su and L. Tong, “Transmission Protocol of Emergency Messages in VANET Based on the Trust Level of Nodes,” IEEE Access, vol. 11, no. July, pp. 68243–68256, 2023, doi: 10.1109/ACCESS.2023.3292234.

[11] K. Parmar, S. Patil, D. Patel, V. Patel, B. Parikh, and P. Padaria, “Privacy-preserving Authentication Scheme for VANETs using Blockchain Technology,” Procedia Comput. Sci., vol. 220, no. 2019, pp. 40–47, 2023, doi: 10.1016/j.procs.2023.03.008.

[12] K. Bala, R. Upadhyay, S. R. Anwar, and G. Shrimal, “A blockchain-enabled, trust and location dependent - Privacy preserving system in VANET,” Meas. Sensors, vol. 30, no. May, p. 100892, 2023, doi: 10.1016/j.measen.2023.100892.

[13] T. Su, S. Shao, S. Guo, and M. Lei, “Blockchain-Based Internet of Vehicles Privacy Protection System,” Wirel. Commun. Mob. Comput., vol. 2020, 2020, doi: 10.1155/2020/8870438.

[14] R. Shrestha, R. Bajracharya, A. P. Shrestha, and S. Y. Nam, “A new type of blockchain for secure message exchange in VANET,” Digit. Commun. Networks, vol. 6, no. 2, pp. 177–186, 2020, doi: 10.1016/j.dcan.2019.04.003.

[15] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi, “A Framework for Secure Vehicular Network using Advanced Blockchain,” 2020 Int. Wirel. Commun. Mob. Comput. IWCMC 2020, no. August, pp. 1260–1265, 2020, doi: 10.1109/IWCMC48107.2020.9148201.

[16] A. S. Khan, K. Balan, Y. Javed, J. Abdullah, and S. Tarmizi, “Secure trust-based blockchain architecture to prevent attacks in VANET,” Sensors (Switzerland), vol. 19, no. 22, 2019, doi: 10.3390/s19224954.

[17] Y. T. Yang, L. Der Chou, C. W. Tseng, F. H. Tseng, and C. C. Liu, “Blockchain-Based Traffic Event Validation and Trust Verification for VANETs,” IEEE Access, vol. 7, pp. 30868–30877, 2019, doi: 10.1109/ACCESS.2019.2903202.

[18] S. Aggarwal and N. Kumar, Blockchain components and concepts☆, 1st ed., vol. 121. Elsevier Inc., 2021.