SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I11P104

Volume 13 Issue 11, 51-65, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Optimized Blockchain-based Key Management Scheme for D2D Message Authentication in 5G VANETs

S V Maruthi Rao Chinnam[1], Rama Krishna Akella[2]

[1,2]*Department of ECE, Koneru Lakshmaiah Education Foundation, Telangana, India.*

[1]*Corresponding Author: maruthichinnam@gmail.com*

Received: 03 September 2026 Revised: 13 October 2026 Accepted: 23 October 2026 Published: 27 November 2026

Abstract *- The integration of 5G technology with Vehicular Ad Hoc Networks (VANETs) enables rapid vehicle-to-vehicle and device-to-device communication but introduces critical challenges in key management, authentication, mobility, and malicious-node detection. This paper proposes lightweight blockchain based key management mechanism for cluster-based 5G-VANETs to provide secure key distribution, dynamic key updating, and seamless vehicle mobility across clusters. The proposed framework establishes dedicated public–private key pairs for D2D communication and periodically updates these credentials to strengthen message authentication and limit the impact of compromised keys. A Coati Optimization Algorithm (COA) is employed for Cluster Head (CH) selection using a fitness function derived from vehicle trust, communication distance, and transmission delay. To improve blockchain suitability for highly dynamic vehicular environments, the conventional block and transaction structures are optimized to reduce storage, processing, and communication overhead. The framework further employs trust-based identification and removal of malicious vehicles to improve network resilience. The proposed mechanism is implemented in NS-3 and evaluated under varying vehicular network conditions. Its performance is compared with HDMA and a Distributed Ledger Technology (DLT)-based approach using relevant security and network performance metrics. Simulation results demonstrate that the proposed scheme provides improved authentication efficiency, reduced communication overhead, enhanced network reliability, and stronger resistance to malicious behavior. These findings indicate that the proposed mechanism is a promising solution for secure, efficient, and scalable key management in 5G-enabled VANETs.*

Keywords *- 5G-VANETs, Blockchain, Key management, Cluster head selection, Coati Optimization Algorithm.*

1. Introduction

Vehicular Ad Hoc Networks (VANETs) have become a popular area of research in Intelligent Transportation Systems (ITS) [1]. VANETs are designed specifically for communication among various vehicles (V2V), between Vehicle and Infrastructure (V2I), Vehicle and Network (V2N) and Vehicle to Pedestrian (V2P), cooperatively called Vehicle to Everything (V2X) [2]. Mostly, VANETs are used in the exchange of information related to road safety, traffic management, like Congestion, routing & tolling, and infotainment services [3]. During communication, security and privacy are significant concerns due to the sensitivity of the information transmitted. By addressing the challenges at the physical layer level, which includes model of propagation, selection and estimation of channel and by understanding the various possible attacks, a better security solution can be achieved [4]. The mobility speeds and dynamic environments of VANETs also play a major role while addressing the security concerns. Currently, 5GAA is developing and promoting the 5 G-based V2X connectivity [5].

Device-to-Device (D2D) communication denotes direct communication amid neighboring devices without necessarily going into a network or accessing infrastructure, thereby making it almost synonymous to V2X. V2X can be implemented using D2D as shown in Figure 1.

D2D in the context of 5G V2X fits in terms of - Improved latency and reliability [6] for critical and safety applications, including collision avoidance systems, enhanced spectral efficiency by reuse and improved network capacity in congested environments, localization of services in case of ad hoc networks without relying on infrastructure, offloading network traffic onto D2D links in the proximity.

Standardization efforts are underway to define protocols and interfaces to address these issues for a secured D2D communication within the 5G V2X ecosystem shown in Figure 1. However, in our work, major concern is to ensure security in 5G VANETs by using D2D links for privacy [7].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

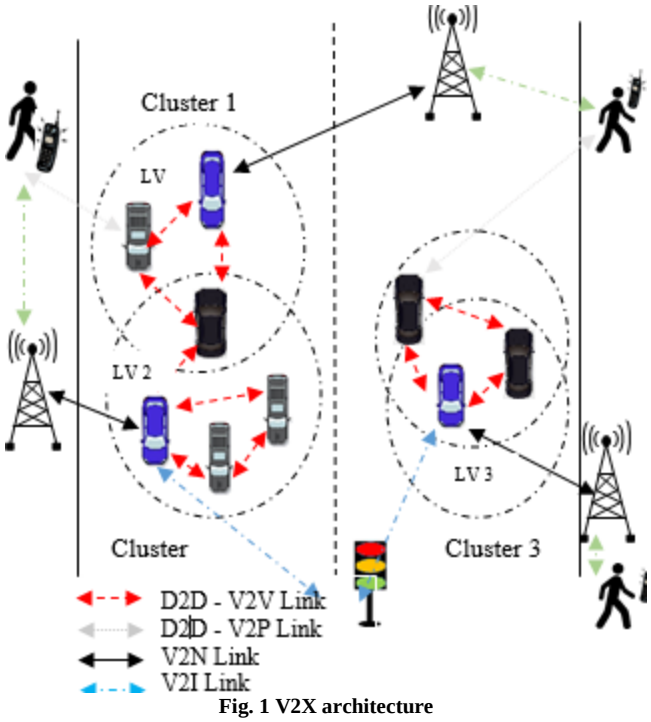

Fig. 1 V2X architecture

Security in D2D communication within 5G VANETs is paramount owing to sensitive data exchanged and possible impact on confidentiality, privacy, integrity and authenticity of vehicles and data. Authentication and Access Control is necessary to verify the identity of each communicating device through D2D. Secure protocols such as Mutual Authentication (MAC), Hash-based MAC (HMAC) and Message Source Authentication (A-MAC) in [8] and signature-based authentication in [9] guarantee that only authenticated devices can participate in communication. Access control mechanisms further restrict access for sensitive resources that have predefined policies. Here, Role-based Access Control (RAC) limits the access to resources depending on the roles of users or devices within the network and also enforces access control. Secondly, Confidentiality and data integrity transmitted over D2D links can be safeguarded with a strong end-to-end encryption with key management protocols that generate, distribute and update the crypto-keys for authentication by securing the vehicles against eavesdropping and tampering. Security protocols based on Advanced Encryption Standard (AES), Diffie-Hellman (DHE) and Elliptic Curve Cryptography (ECC) can be used for key validation. For maintaining integrity, techniques such as digital signatures or MAC can be used. RSA encryption is commonly used in combination with asymmetric cryptography for key exchange and uses digital signatures to ensure the confidentiality and integrity.

Privacy is most essential in D2D 5G VANET communication. The location of the vehicle and data exchanged between vehicles are very sensitive information, which should be known only to the authorized entities [8]. Techniques such as anonymization, ring signatures, group signatures and one way hash chain, etc., help mitigate the risk of identity disclosure and tracking. Pseudonyms are often used to protect the privacy of communicating entities. The speed requirements of 5G and low communications overhead should be addressed while designing these security protocols [10]. Secure establishment of communication channels between D2D devices involves protocols that protect against several attacks, like Man-in-the-Middle (MITM), as well as replay attacks. Establishing secure channels based on trusted parameters and cryptographic protocols helps prevent unauthorized interception or modification of communication. Research in [11] guarantees consistent and secure channels for V2I as well as V2V and is used as a benchmark for V2X. Security protocols depending on Socket Layer (SSL), along with its Transport Layer (TLS) security, can be used to authenticate the vehicles and secure communications when placed on the D2D link [12]. Adherence to security standards and guidelines is crucial for ensuring interoperability and compatibility among different vendors' equipment, and compliance with established security protocols and best practices helps build trust in the ecosystem. Regulatory bodies like 3GPP and IETF assess the vulnerabilities in the network and update the security protocols, interfaces and best industry standards [13]. Machine Learning techniques can be used to leverage performance of D2D links in VANETs [14]. It is observed that the weakest part of D2D security in 5G VANETs is key management as the keys has to be issued, refreshed and revoked periodically while the vehicles change the clusters, while the use of a central authority delays the procedure and becomes a single point of failure. The major research gap identified based on Table 1 is that infrastructure dependent schemes [8, 9] cannot update keys and/or remove malicious vehicles on their own. Cluster-based schemes [9, 11] choose cluster heads with considering trust, link lifetime and delay together. Blockchain key management schemes [15-17] carry heavy overhead and cannot handle vehicles moving between clusters. To the best of the knowledge, no existing scheme combines trust-aware, optimized CH selection with a light-weight blockchain that can manage complete key management life cycle that includes generation, update, inter-cluster movement and revocation in cluster-based 5G D2D VANETs. To address this, the proposed scheme selects CH with COA using a fitness function built from trust, link lifetime and delay and manages keys on a blockchain from which POW, Nonce and difficulty fields are removed.

2. Related Works & Major Contributions

2.1. Related Works

A summary of the various research works pertaining to securing the communication in VANETs is given in Table 1. Along with the aforementioned limitations, there are other limitations using these techniques.

Table 1. Comparison of related works

Ref	Technology	D2D Support	Cluster
[8]	Authentication & A-MAC	Limited	No
[9]	Authentication & Group Signature-Based	Hybrid	Yes
[10]	MAAC & Probabilistic Key Management	Limited	No
[11]	AAC & Third Party Certification	Within a Cluster	Yes
[12]	Certificate-based & NC weight-based TM	Yes	No
[15]	Authentication with key management & Blind Signature	Yes	No
[16]	Long lost chain & Certificate-based	Within Cluster	No
[17]	Distributed Authentication & Dynamic ledger-based key	Yes	No

[8] Relies heavily on the infrastructure and limits the applicability in areas with low network availability. As the number of vehicles increases, managing secure connections while maintaining the throughput is challenging (especially dense urban environments). [9] is a hybrid scheme that requires very careful key distribution management and can add overhead in dynamic topology environments. It is very vulnerable to insider attacks, and the use of symmetric and asymmetric cryptography increases the processing times and resource consumption. [10] Has speed optimization issues that come at the cost of security robustness. In environments that need rapid authentication, this technique is vulnerable to replay and relay attacks and is not strong in session management.

[11] is challenging in highly dynamic environments where cluster head selection becomes very complex and can potentially lead to delays. The frequent cluster reconfiguration also leads to delays, and the over-reliance on CH can create bottlenecks while authentication, and makes the system vulnerable when the CH is compromised. [12] Uses a trust management algorithm that requires extensive data analysis and storage that has a huge impact on security with a large number of vehicles. This technique has a possibility to misinterpret a legitimate node behaviour as malicious or false. [15] Adds computational complexity and has an effect on response time.

The additional steps used for preserving privacy leads to this delay. Also, the privacy-preserving technique can inadvertently allow vehicles to exploit anonymity features, making it hard to trace the adverse activities in the network. [16] Uses blockchain technique and is implemented on aerial vehicles, in which high latency and computational power is observed. [17] Like [16] face scalability issues and computational constraints that has an impact on real-time communications, where latency is increased.

It also adds the overhead that reduces performance in high-throughput environments and has data synchronization issues. This paper addresses the limitations of the above techniques and improves the security with a tradeoff between security and computational complexity.

2.2. Major Contributions

A novel distributed key management system for a 5G-enabled VANET is proposed so that the system enables vehicles to independently allocate keys to a cluster, determine pairs of public and private keys, migrate from a cluster to another and revoke any malicious vehicles. To achieve this, necessary modifications were made to block and transaction structure in blockchain and an optimal clustering model based on Coati Algorithm (COA) is used for CH selection. In the blockchain designed for key management, each vehicle possesses its own distinct transaction chain, which can record its public or private key updation in addition to cluster migration, making it easy to detect any malicious behaviors. This work is implemented using NS3, and the performance is compared with two other existing protocols. This paper is structured involving the following sections. In the introduction, features to be considered for security of D2D in 5G VANET's and a comparison of various works done are discussed. In ensuing section, the propounded system model, methodology, and implementation are detailed. In next section, simulation results are illustrated and discussed in detail by comparing with 2 other existing works [9, 17]. In the same section, security threats addressed in our work are detailed.

3. Proposed Model & Design Methodology

3.1. Proposed System Model

A 5G-enabled VANET system with multiple vehicle clusters is considered. Every cluster includes 2 kinds of vehicles. First, a lead vehicle with enough storage space, and increased computational ability and improved transmission power, maintaining a centralized approach.

Secondly, multiple-member vehicles with restricted power which can handle lightweight tasks with distributed approach. Member vehicles communicate only within their own cluster, while lead vehicles can establish communication with other vehicles to facilitate interaction among clusters. Every cluster is allocated with a fixed region which defines area of vehicle movement.

Lead vehicle is not allowed to move away from cluster, whereas member vehicles may leave their present cluster and migrate to another cluster based on task demands. The key management for this system is handled by the lead vehicles, namely Cluster Head (CHs). Here, the member vehicles are only responsible for maintaining the latest state of the blockchain, having the Latest Vehicle List (LVL) and the lead vehicle (C) information. Therefore, we have a small blockchain network with only CH's involved in the process. While executing the tasks, the Trusted Authority (TA) does not take part in key management process as CH's handle them autonomously. However, the RSU can be involved in the process at any moment. In order for our scheme to operate properly, it is assumed that the TA does not have malicious vehicles at system initialization, and CHs are mostly trustworthy with better features compared to member vehicles, which are mostly trustworthy. CH's key management process does not affect regular operations. The possibility of both external and internal threats to the system is considered. Attackers are not acceptable members that can gather information from RSU through eavesdropping that disrupts system by intercepting, replaying and modifying messages. External vehicles hijack a vehicle and access the stored information by posing as a RSU or an OBU and attack integrity of vehicle [16].

Internal threats come from valid vehicles which may maliciously act by trying to update the public key to match that of a benign vehicle or RSU in order to impersonate their identity.

3.2. Design Methodology

The proposed method contains two phases: the setup phase and blockchain based key management phase. In the setup phase, cluster formation as well as CH selection [18] based on COA [19] using a fitness function is done. In the second phase, blockchain based key management [20] is adapted for secure data transmission.

3.2.1. Setup Phase - Trust Calculation

Regional Trust (RT_{ij})

It includes Interaction Trust (IT_{ij}) and Efficiency Trust (ET_{ij}) between vehicles 'i' and 'j' (refers to trust that 'i^{th}' vehicle has on 'j^{th}' vehicle). Network operation demands data and control traffic for which distinct trust levels (IT_{ij}^{D},IT_{ij}^{C}) should be formed to guarantee effective interaction. After time interval 'Δt', every vehicle finds 'RT_{ij}' values of its nearby vehicles [21].

Interaction Trust (IT_{ij})

Trust level of vehicle 'V_i' towards 'V_j' may be determined over a time period by '$IT_{ij}(\Delta t)$'. To determine trust values for resource-limited nodes, Bayesian trust model is used [22, 23]. The model uses $E(Beta(p, v, m, n))$i.e., expectation of Beta Probability Destiny function $Beta(p|m, n)$ [23] to obtain the reputation score - p within a range [0, 1]. Here, 'm' and 'n' represent positive, negative interactions, respectively, among vehicles. The reputation interpreted as maximum likelihood of interactions in the future is given similar to Equation (1) of [23],

$$E\big(beta(p|m, n)\big) = \frac{m+1}{(m+1)+(n+1)} \quad (1)$$

A varied interval counter is used to determine the forwarding trust of vehicle's control traffic. For every time interval 'Δt', number of successful interactions ($\Psi_{ij}(\Delta t)$) and unsuccessful interactions '$\Omega_{ij}(\Delta t)$' are determined from control flow table. This includes rules where 'V_j' signifies next hop vehicle but not the destination (Dst_{Addr}). The flow entry establishes a successful interaction represented by, $R_j = \{e \vee e.next_{hop} = V_j \wedge e.Dst_{Addr} \neq V_j\}$. This is performed using Equation (2). Similarly, quantity of unsuccessful interactions '$\Omega_{ij}(\Delta t)$' can be found [23].

$$\Psi_{ij}(\Delta t) = \sum e.interval.ctrl.success\,;\ e \in R_j \quad (2)$$

By examining successful as well as unsuccessful interactions, '$IT_{i,j}^{C}$' of nearby vehicles is obtained over former 'Δt' period.

$$IT_{ij}^{C}(\Delta t) = \frac{(No\ of\ Successful\ Interactions + 1)}{(No\ of\ Successful\ Interactions + 1) + (No.of\ unSuccessful\ Interactions + 1)} \quad (3)$$

Using Equation (4), '$IT_{ij}(\Delta t)$' with latest 'Δt' is determined with 'γ' and 'φ' as weighted co-efficient with their sum equal to 1. 'γ' is assigned a value more than 'φ'. It is considered that the influence of control packets being dropped is more compared to that of data packets [23].

$$IT_{ij}(\Delta t) = \gamma \times IT_{ij}^{C}(\Delta t) + \varphi \times IT_{ij}^{D}(\Delta t);\ 0 < \varphi < \gamma < 1 \quad (4)$$

Efficiency Trust (s)

It is employed in determining forwarding performance of vehicles (V_i) neighbour (V_j) in a time interval where malevolent packets are sent as a new flow attacks. The value of trust is determined with Equation (5)

using packet-in record table, which comprises the timestamp, source, as well as destination addresses [23, 24]. Maximum amount of fresh flows that a vehicle may get from a nearby vehicle during 'Δt' is represented as 'R_v'. Smaller value of '$ET_{ij}(t)$' enables observation of more number of new flows. As interval counter (ω_j) reaches 50% of 'R_v' and 'V_j' is not seen in accepted list, security part in 'V_i' will generate a warning that will be sent to 'V_j' as a packet. If 'V_j' vehicle floods fresh flows with '$ET_{ij}(t) < 1$', then node forwards information of malicious vehicle to the CH.

$$ET_{ij}(t) = R_v/\omega_j \quad (5)$$

By combining 2 diverse kinds of forwarding trusts, '$RT_{ij}(t)$' is determined using Equation (6). Trust has a range of [0, 1] involving 4 bytes and / or a real integer within a range [0, 10] involving 1/2 byte. Nevertheless, to reduce memory and overhead, the trust is initialized as unsigned integer within a range [0, 100], which uses memory space of 1 byte while not forfeiting accuracy.

$$RT_{ij}(t) = [100 \times ET_{ij}(t) \times IT_{ij}(t)] \quad (6)$$

'$RT_{ij}(t)$' is a measure of trustworthiness of vehicle 'V_j' derived from opinion of 'V_i'.

Lifetime of a Vehicle

The lifetime of the vehicle defines how long the vehicle is available in the range of transmission and depends on vehicle velocity. For a given value of 'i',

$$LT(i) = \frac{d_i}{Vel_i} \quad (7)$$

'd_i' is the distance from the sending vehicle and 'Vel_i' is its Velocity

Delay Calculation

Delay is calculated using the function (Del (i, j)) for link between any two vehicles 'V_i' and 'V_j'. There are three types of delays [25], namely, transmission (DY_T), queuing (DY_Q) and propagation (DY_P) delays.

$$Del(i,j) = (DY_T + DY_Q + DY_P) \quad (8)$$

$$DY_T = \frac{J}{\mu} \quad (9)$$

Here, 'J' represents Packet Size in bits and 'μ' denotes link's bandwidth in bits per second

$$DY_P = \frac{PL_{ij}}{\beta} \quad (10)$$

'PL_{ij}' is length of the physical link between vehicles 'i' & 'j' and 'β' represents speed of propagation (m/s)

$$DY_Q = \frac{1}{(\sigma-\rho)} \quad (11)$$

'σ' represents service rate and 'ρ' denotes new packets entry rate. It is assumed that packet processing delay is minimal. Equation (8) is represented as the End-to-End delay for path amid source node (c) to destination node (TA / RSU).

$$D_{e2e} = \sum_{a,b\in\{c,I,bs\}} \frac{1}{\sigma-\rho} + \frac{J}{\mu} + \frac{PL_{ij}}{\beta} \quad (12)$$

Here 'I' represents number of nodes that are between 'c' and 'TA'.

3.2.2. CH Selection Phase

A comprehensive objective function is necessary rather than individual objective functions to effectively handle various features and tasks for specific QoS applications and neighbor vehicles. To support all types of data flows, a weighted parameter based approach is preferred. Objective function is actually based on delay, trust, distance and energy. Weight of 'V_j' is used to route V_i vehicle. The objective function of 'V_j' (OB_j) is given to COA as input, which selects best forwarding vehicle depending on decision taken by COA.

$$OB_j = \delta_1 \times LT_{ij} + \delta_2 \times LT(i) + \delta_3 \times D_{e2e}(c, bs) \quad (13)$$

'δ_1', 'δ_2' and 'δ_2' are relative co-efficients which are used for routing such that $\delta_1 + \delta_2 + \delta_2 = 1$ [23]. 'LT_{ij}' represents the overall trust assessment of $V_i$andV_j. 'LT_{ij}' is given by sum of regional, interaction and efficiency trusts. Generally, the vehicle with the increased residual energy (weight) has the maximum probability to be chosen as the forwarding vehicle.

COA-based CH Selection

Coatis are omnivorous animals which feed on green iguana. The way they attack iguanas and the strategies they adapt to escape from predators is quite remarkable and is used as a model for proposed CH selection method. A population of vehicles is used, which are randomly distributed in the cluster. Every vehicle has an address in cluster Search Space (SS) that matches with a collection of values of decision variables. The location is a probable solution to choose CH [25]. The opening point of COA is obtained using Equation (14) using arbitrary initialization of vehicle locations.

$$Z_i : z_{ij} = LB_j + r * (UB_j - LB_j) \quad (14)$$
$$i = 1,2,..N, j = 1,2,..M$$

Here, 'Z_i' is location of 'i^{th}' vehicle in SS, 'z_{ij}' is value of 'j^{th}' decision variable, 'N' represents quantity of vehicles, 'M' denotes amount of decision variables, 'r' signifies an arbitrary real number within range [0, 1] and 'LB_j' and 'UB_j' are lower as well as upper bounds of 'j^{th}' decision variable.

The vehicle matrix 'Z' is mathematically represented as in [13]. Vehicle Candidate Solutions (CSs) are obtained from the decision variables to solve problem's objective function, as shown in Equation (16).

$$Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_i \\ \vdots \\ Z_N \end{bmatrix}_{NxM} = \begin{bmatrix} z_{1,1} & \cdots & z_{1,j} & \cdots & z_{1,M} \\ \vdots & . & \vdots & . & \vdots \\ z_{i,1} & \cdots & z_{i,j} & \cdots & z_{i,M} \\ \vdots & . & \vdots & . & \vdots \\ z_{N,1} & \cdots & z_{N,j} & \cdots & z_{N,M} \end{bmatrix}_{NxM} \quad (15)$$

$$Q = \begin{bmatrix} Q_1 \\ \vdots \\ Q_i \\ \vdots \\ Q_N \end{bmatrix} = \begin{bmatrix} Q(Z_1) \\ \vdots \\ Q(Z_i) \\ \vdots \\ Q(Z_N) \end{bmatrix} \quad (16)$$

Here, 'Q' represents vector of objective function and 'Q_i' signifies value of objective function depending on 'i^{th}' vehicle. The quality of CS gives value of 'OB_i'. The vehicle with best value of objective function is selected as CH. As CSs are updated in each iteration, CH is also determined [19, 26]. The position of vehicles is updated, similar to the natural behavior of Coatis through exploration and exploitation.

In the exploration phase, the vehicle population is first updated in the SS, while some vehicles search for another vehicle or RSU to communicate, few vehicles / RSU wait for a vehicle to approach near them. In either case, the location of the vehicles will change, and this defines their ability to explore the SS [25]. The position of the vehicle to be communicated with (from now on referred to as prey (Tar) is assumed to be the best vehicle and is also referred to as the CH, as given in Equation (17). The location of the vehicles and nodes communicating are chosen assuming that 50% of the nodes are the members and remaining are RSU.

$$Z_i.Pos1: z_{i,j}.Pos1 = z_{i,j} + r \cdot (Tar_j - I \cdot z_{ij}); \quad i = 1,2,..\frac{N}{2}, j = 1,2,.,M \quad (17)$$

Once vehicle is obtained, its location is considered as the arbitrary location of the CH. Based on the arbitrary locations, the vehicles nearby move towards the CH in the SS as shown below.

$$Tar^G: Tar_j^G = lb_j + r.(UB_j - LB_j) \quad (18)$$

$$i = 1,2,\ldots..N, j = 1,2,\ldots..M$$
$$Z_i.Pos1 : z_{i,j}.Pos1 = \begin{cases} z_{i,j} + r \cdot (Tar_j^G - i \cdot z_{i,j}), Q_{Tar}^G < Q_i \\ z_{i,j} + r \cdot (i.z_{i,j} - Tar_j^G), Else \end{cases} \quad (19)$$

The new positions of the vehicles are subject to improvement in the objective function else they will be considered as no change in their positions. The new positions are given as

$$Z_i = \begin{cases} Z_i.Pos1; Q_i.Pos1 < Q_i \\ Same; else \end{cases} \quad (20)$$

Here '$Z_i.Pos2$' is new location of the 'i^{th}' vehicle, depending on second stage of COA, '$Z_{i,j}.Pos2$' is the 'j^{th}' dimension of '$Z_i.Pos2$'.

Here, '$Z_i.Pos1$' is the new position of the 'i^{th}' vehicle, '$z_{i,j}.Pos1$' is the'j^{th}' dimension of the '$Z_i.Pos1$' and '$Q_i.Pos1$' is its objective function.

'Tar^G' refers to location of prey vehicle in the same cluster, which is arbitrarily produced, and 'Tar_j^G' is its 'j^{th}' dimension. 'Q_{Prey}^G' gives value of objective function.

Exploitation deals updating location of vehicles in SS, where vehicle moves to a safe position near to current location in same cluster, where its resources cannot be directly utilized until it becomes a CH. The position of vehicle is considered as the arbitrary location.

$$LB_{jLoc} = \frac{LB_j}{\text{iteration count}} \text{ and } UB_{jLoc} = \frac{UB_j}{\text{iteration count}} \quad (21)$$

$$Z_i.Pos2: z_{i,j}.Pos2 = z_{i,j} + (1-2r).\left(LB_{jLoc} + r.\left(UB_{jLoc} - LB_{jLoc}\right)\right) \quad (22)$$

This new location is considered if there is an improvement in the objective function value '$Q_i.Pos2$'.

$$Z_i = \begin{cases} z_i.Pos2; Q_i.Pos2 < Q_i \\ Same; else \end{cases} \quad (23)$$

Iterations are stopped once each of vehicle's position is determined based on the first 2 stages. The number of vehicles is updated using Equation (14) to Equation (18), and process is repeated until last iteration. Once runs are complete, optimal solution is returned as shown in Figure 2.

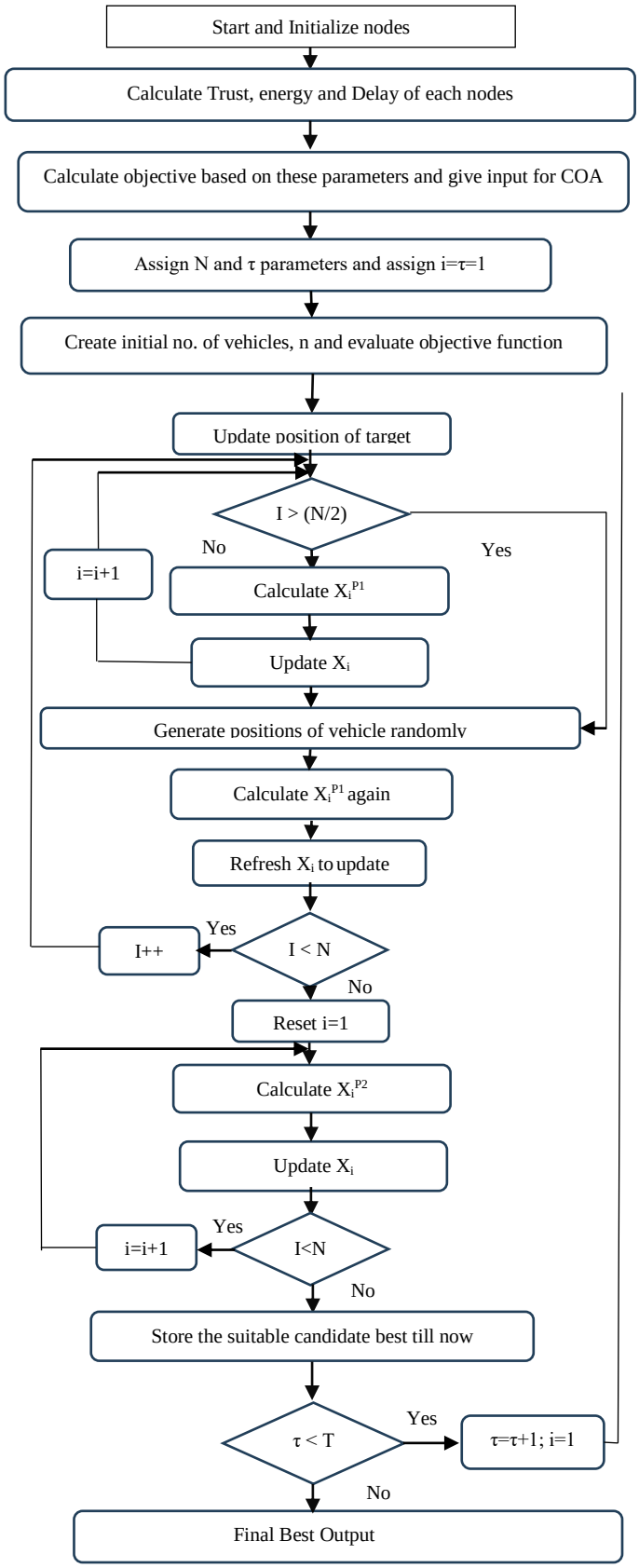

Fig. 2 Selection of Cluster Head based on COA

3.2.3. Blockchain Key Management for 5 G-enabled VANET

Blockchain is a secure decentralized ledger technology in that doesn't require third-party verification in P2P networks [27] and could be useful in secure 5G VANET's to reduce delay in communication. Nonce, along with Target Difficulty Fields in traditional blockchain described in [28] is removed, and a new miner election method, which does not demand difficult task as in Proof of Work (POW) consensus algorithm, is included [29] for better addressing key management. The proposed key management blockchain is illustrated in Figure 3.

The transaction header includes: Cluster Index (CI), Input Address (IA), generator and signature. CI gives the information of the cluster from where the transaction has originated. IA shows that present transaction is based on previous transaction. Generator represents signature of transaction data. With the help of each block hash and corresponding Merkle Root Tree [30], each transaction can be easily validated. The purpose of miner election is to select a generator that is identified globally and used for the next block through CH in each round. A small scale of highly reliable nodes is considered, and discrete Particle Swarm Optimization [31] is used for miner election scheme in our work, with an exception for RSU. It is assumed that system is in 'k^{th}' block of generation stage. CHs use a Distributed Random Generation (DRG) protocol for negotiating an unbiased arbitrary value (r_k) [31]. By merging the random value with that of every CH,

$$H(Block_k') = H(r_k \vee H(Block_k)) \quad (24)$$

Where $H(Block_k)$ represents hash value of block (K). Election score for the CH of 'P^{th}' cluster is given by,

$$Score.z_{p0} = H(Block_k' \oplus H(PK.z_{p0})) \quad (25)$$

Where, '$PK.z_{p0}$' is the public key of vehicle 'z_{p0}'. Once the calculation of all election scores are over, CH with the lowest election score will be selected as miner. Here, all CH calculate similar scores, ensuring that chosen miner is globally recognized. The DRG protocol guarantees that all CH agree on the same 'r_k'. If elected miner is compromised, CH with next lowest score will be considered to generate the next block.

Once cluster and CH are selected, we need to perform key management using Improved Hyper Elliptic Curve (HEC) cryptography over the blockchain. For this, we consider a finite field (F) with a algebraic closure of 'F' with a genus '$g > 1$' over 'F' and is a collection of solutions $(A, B) \in F \times F$ given by curve Equation [32].

$$HEC = b^2 + H(a).b = f(a) \quad (26)$$

$(A, B) \in FXF$ should satisfy partial differentiation equation $2b + h(a) = 0$ and $h'^{(a)} - f'(a) = 0$. Here, '$h(a) \in F(k)$' is a polynomial with degree (g) and is monic polynomial of degree $(2g + 1)$ [32]. Finite field (Fg) is considered with such that $g \approx 80$, and uses Diffie-Hellman (DH) key exchange.

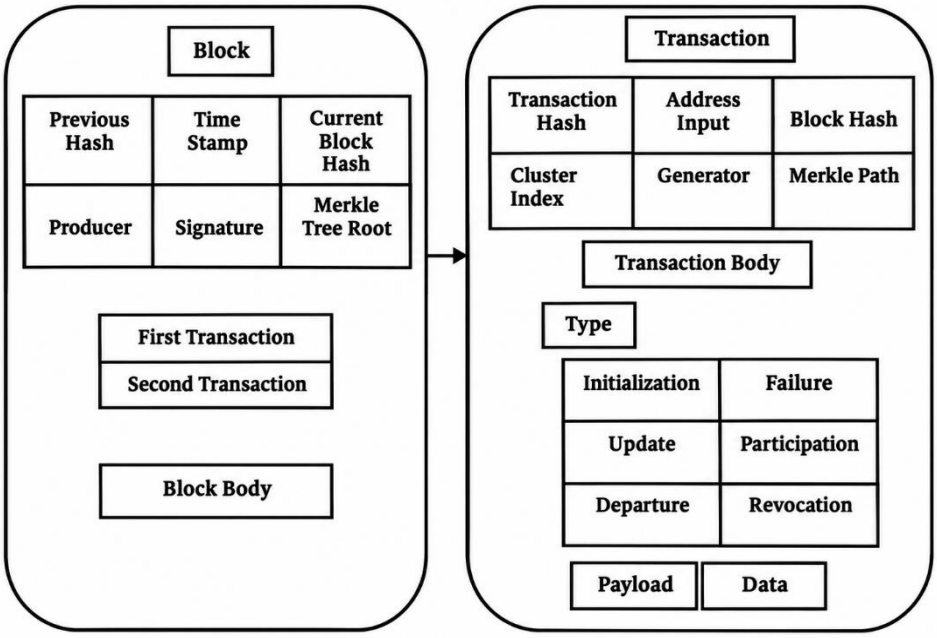

Fig. 3 Key management in blockchain

Stages in Key Management

a) System Metrics (Parameters)

TA selects the HEC key. TA chooses a private key such that $\mathrm{Pri_{Enb}} \in \mathrm{F_g}$ [31], and public key is

$$\mathrm{Pub_{Enb}} = \mathrm{Pri_{Enb}}.\mathrm{J} \tag{27}$$

Enb is Communicating node that stores '$\mathrm{Pri_{Enb}}$' in secret and broadcasts parameters ($\mathrm{g}, \mathrm{J}, \mathrm{Pri_{Enb}}$) generated by TA.

b) Vehicle ID Registration and Distribution of Cluster Key

The TA then endorses ID of each vehicle and splits genuine vehicles into numerous clusters, assigning a CH for each cluster simultaneously based on the COA. Considering that '$\mathrm{Cl_i}$' represents the '$\mathrm{i^{th}}$' cluster, '$\mathrm{V_{i,j}}$' as '$\mathrm{j^{th}}$' vehicle in '$\mathrm{Cl_i}$' and '$\mathrm{V_{i,0}}$' as head vehicle, TA distributes public or private pairs of key to members and CH's such that LVL for each cluster is produced. Each LVL consists of public keys in '$\mathrm{Cl_i}$' with '$\mathrm{n_i}$' number of vehicles in each cluster.

$$\mathrm{LVL_i}=\left\{\mathrm{Pub_{Enb}^{i_1}}, \mathrm{Pub_{Enb}^{i_2}}, \mathrm{Pub_{Enb}^{i_3}}, \ldots \mathrm{Pub_{Enb}^{i_n}}\right\} \tag{28}$$

Each vehicle in '$\mathrm{Cl_i}$' stores pairs of keys along with '$\mathrm{LVL_i}$' and conserves private key secretly. Public keys are considered as IDs. As clusters are set, the TA generates 'C', which comprises of public keys of head vehicles. TA produces Block '$\mathrm{B_0}$'that includes initial transaction of every vehicle [31] with Header and Body as given in Equation (29) and Equation (30).

$$\mathrm{Header} = \begin{cases} \mathrm{IA} = 0 \\ \mathrm{CI} = \mathrm{Cl_i} \\ \mathrm{Generator} = \mathrm{Enb_{Pub}} \end{cases} \tag{29}$$

$$\mathrm{Body} = \begin{cases} \mathrm{Type} = \mathrm{Initialise} \\ \mathrm{Payloadwithsignature} = \mathrm{Pub_{i,j}}, \mathrm{T}, \mathrm{Sign_{i,j}^{Pri}}\{\mathrm{H}(\mathrm{Pub_{i,j}}, \mathrm{T})\} \end{cases} \tag{30}$$

Where, '$\mathrm{Sign_{i,j}^{Pri}}\{\}$' is signature using TA's private key. The public key of '$\mathrm{V_{i,j}}$' and TA's signature are stored in the transaction. 'Enb' forward '$\mathrm{Block_0}$' to each CH [16, 31]. To guarantee confidentiality, CH disseminates cluster keys for encryption. Here, $\mathrm{V_{i,0}}$ distributes the cluster key$\left(\mathrm{Cl_{Key}^i}\right)$.

It first produces '$\mathrm{Cl_{Key}^i}$' and then distributes the same in the cluster using broadcast message generation algorithm '$\varphi(.)$' [16] along with '$\mathrm{LVL_i}$' to transmit $\left(\mathrm{Cl_{Key}^i}\right) \wedge \mathrm{Sign_{i,0}^{Pri}}\{\mathrm{H}(\varphi(\mathrm{Cl_{Key}^i}))\}$ to all members in '$\mathrm{Cl_i}$'. On receiving '$\mathrm{Cl_{Key}^i}$', '$\mathrm{V_{i,j}}$' verifies signature of $\mathrm{V_{i,0}}$ using inverse algorithm '$\varphi^{-1}(.)$' that includes private key ($\mathrm{Pri_{i,j}}$) to determine '$\mathrm{Cl_{Key}^i}$' from '(φ^{-1} ($\mathrm{Cl_{Key}^i}$)' [16]. The algorithm designed for key distribution ($\varphi(.)$) guarantees that only original vehicles whose public keys are on '$\mathrm{LVL_i}$' can determine '$\mathrm{Cl_{Key}}$' from '$\varphi\left(\mathrm{Cl_{Key}^i}\right)$' using private keys.

The CH establishes a group, and a miner of every round is accountable for dispensing cluster keys among CH's [31].

c) Key Updation

Vehicles must change their public and private keys often due to risk of attacks, change in the vehicles and RSU's in each cluster. '$\mathrm{V_{i,j}}$' in each '$\mathrm{Cl_i}$' should update public as well as private key pairs. '$\mathrm{V_{i,j}}$' chooses a private key, $\mathrm{Pri_{i,j}^{New}} \in \mathrm{Z_S}$ and determines conforming public key $\mathrm{Pub_{i,j}^{New}} = \mathrm{Pri_{i,j}^{New}}$. Request for key update is sent to $\mathrm{V_{i,0}}$ [9, 31].

$$\mathrm{UpdateRequest} = \mathrm{En_{Cl_{Key}}}\left\{\mathrm{Pub_{i,j}}, \mathrm{Pub_{i,j}^{New}}, \mathrm{T}, \mathrm{Sign_{i,j}^{Pri}}\{\mathrm{H}(\mathrm{Pub_{i,j}^{New}}, \mathrm{T})\}\right\} \tag{31}$$

On obtaining the request, $\mathrm{V_{i,0}}$ decrypts using '$\mathrm{Cl_{Key}^i}$' and verifies whether '$\mathrm{Pub_{i,j}}$' belongs to '$\mathrm{LVL_i}$'. If valid, it employs '$\mathrm{Pub_{i,j}}$' to guarantee signature and agree with request for update. If '$\mathrm{V_{i,j}}$' is not a valid member of '$\mathrm{Cl_i}$', '$\mathrm{V_{i,0}}$' rejects the request. On accepting the request, $\mathrm{V_{i,0}}$ searches blockchain to find vehicle ID that stores current public key of '$\mathrm{V_{i,j}}$'. $\mathrm{V_{i,0}}$ will produce an update transaction for '$\mathrm{V_{i,j}}$' with the input address changed to '$\mathrm{HB_{i,j}MerkleRootTree_{i,j}}$' and generator to '$\mathrm{Pub_{i,j}}$' with index remaining same in the header while the payload data signature is updated. $\mathrm{V_{i,0}}$ sends updated transaction to round's miner. Vehicles update '$\mathrm{LVL_i}$' based on these transactions. Public key of '$\mathrm{V_{i,j}}$' stored in '$\mathrm{LVL_i}$' is updated to '$\mathrm{Pub_{i,j}^{New}}$'. The transactions involved in the update of CH keys are confirmed by miner and updates 'C' simultaneously.

d) Vehicle movement among clusters and Revocation

The vehicles moving from one cluster to another and RSU entering or leaving a cluster are classified as leaving / joining vehicles. '$\mathrm{V_{i,j}}$' desires to leave '$\mathrm{Cl_i}$' and join '$\mathrm{Cl_k}$' due to specific task requests. It sends leaving request to $\mathrm{V_{i,0}}$ that verifies leaving request with the same steps detailed earlier. After verification, the transaction which records '$\mathrm{Pub_{i,j}}$' in blockchain is determined, and a leaving transaction is generated with only IA and generator values changed. The transaction is forwarded to miner and built into

a block. After sending a block, vehicles in 'Cl_i' determine leaving transaction about '$V_{i,j}$' and remove '$Pub_{i,j}$' from 'LVL_i'. $V_{k,0}$ gets transactions and learns about coming of '$V_{i,j}$' to 'Cl_k'. It saves '$Pub_{i,j}$' in the cache for upcoming verification. While the new vehicles / RSU are joining the cluster, i.e., '$V_{i,j}$' reaches 'Cl_k', it forwards joining request to $V_{k,0}$. On receiving participation request it checks availability of '$Pub_{i,0}$' in 'C'. If it is not available, it means $V_{i,0}$ is compromised, else, it searches for '$Pub_{i,j}$' record in cache. In case record is found, it continues with verification; else it instantly drops request. In case of successful verification, $V_{k,0}$ generates transaction for contribution of '$V_{i,j}$'. In the body, the type will be updated to participation and payload with signature data accordingly in the transaction chain and migration of '$V_{i,j}$' is recorded in transaction chain. The cluster key, along with keys of vehicles are updated every time there is a change in the number of vehicles in a cluster. As vehicles leave or join, the CH initiates a round of key distribution depending on modified LVL. For security reasons, each vehicle in a cluster periodically forwards Hello messages [15] to CH for their survival. If $V_{i,0}$ has not identified '$V_{i,j}$' signals for specific time, it means that it is compromised, a failure transaction is generated, and same is updated in header and body. Transaction is included to end of '$V_{i,j}$'s chain of transaction, vehicle is discarded and cannot re-join in this cluster. Vehicles in 'CL_i' take away '$Pub_{i,j}$' from 'LVL_i' on obtaining a new block. CH also generates Hello messages continuously to check for the state of vehicle and other CH's. If CH loses communication for a particular time period, it is considered as attacked by other CHs and miner of that round generate a failure. Each vehicle in the cluster discards the public key of CH from 'C', based on transaction, as well as vehicles of 'CL_i' must join other clusters. In order to revocate a vehicle, '$V_{i,j}$' identifies malevolent behavior of '$V_{i,M}$' and reports it. '$V_{i,j}$' will produce a report request that consists of message 'M', '$Pub_{i,j}$' and '$Sign_{i,j}^{Pri}\{H(M)\}$' and forwards to $V_{i,0}$ that verifies request for report and puts '$Sign_{i,j}^{Pri}\{H(M)\}$' to a set of messages to report. When number of these report messages associated with '$V_{i,M}$' is more than half the amount of vehicles in 'CL_i', it means that '$V_{i,M}$' shows malevolent behavior and produces a revocation transaction.

4. Results and Discussion

4.1. Design Implementation

The Proposed model is implemented with NS 3.25.

Table 2. Simulation metrics

Metric	Value
Area for simulation (Sq. m)	5000 x 5000
Time for simulation (Sec)	600
Topology	SUMO
No. of nodes / vehicles	50, 100, 150, 200, 250
Speed (Km/h)	5 to 30
Size of Data Packet	1024
Propagation Model	LDP
Frequency and Bandwidth	28GHz & 100 MHz
Traffic Type	CBR and UDP

Here LTE package chosen in NS3 is to create 5G network and to enable D2D features over the LTE package. Simulation metrics are listed in Table 2. To introduce security in NS3, OpenSSL and cryptolib packages are linked, and blocks of data are created with the help of rapidjson tool to integrate blockchain technology in NS3.

4.2. Simulation Results

The graphs Figure 4(a) to 4(d) display the various parameters evaluated using the proposed design and is compared with results of [9, 17]. The number of vehicles in each graph represents the number of requests made and communicated. Computational cost is the time taken for key generation, encryption, decryption and signature generation and is 0.23 for 50 nodes, i.e., computational cost has improved by 58.12% of that of HDMA [9] and 41.07% of D2D [17] and presented in Figure 4(a).

With increase in quantity of nodes (multiples of 50, with maximum number of nodes 250) the computational cost has improved between 41 – 48% compared to HDMA [9] and 30 – 51% compared to D2D [17]. For up to 250 vehicles, there is an average improvement in the computational cost of our design by 36.64 % compared with [17] and 46.764% compared with [9]. A minimum packet size of 512 bytes and maximum of 4K bytes is considered for within and outside the cluster communication to observe the encryption timings with respect to the data size, with a key size of 80 using SHA256.

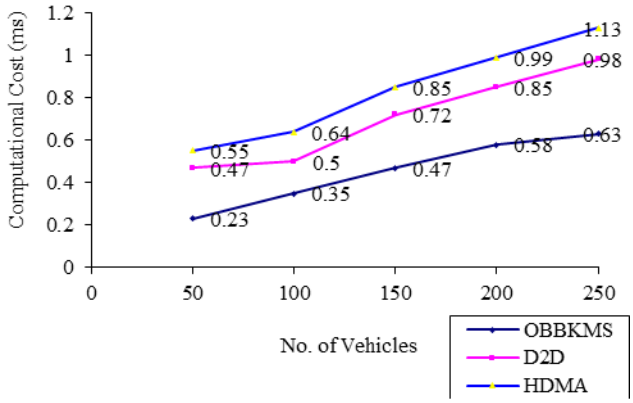

Fig. 4(a) Comparison of computational cost with benchmark protocols

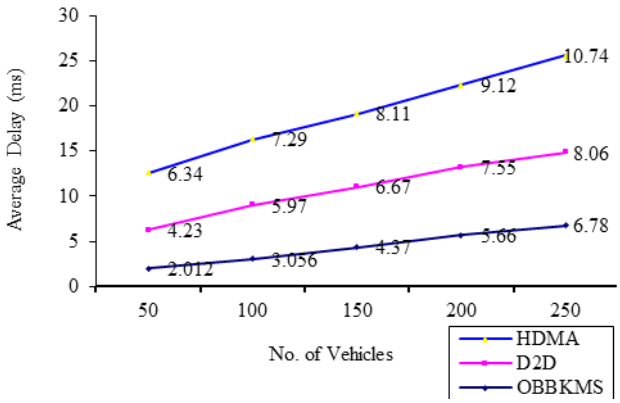

Fig. 4(b) Comparison of average delay with benchmark protocols

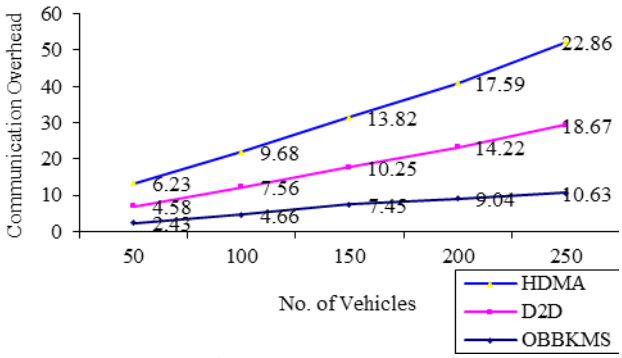

Fig. 4(c) Comparison of communication overhead with benchmark protocols

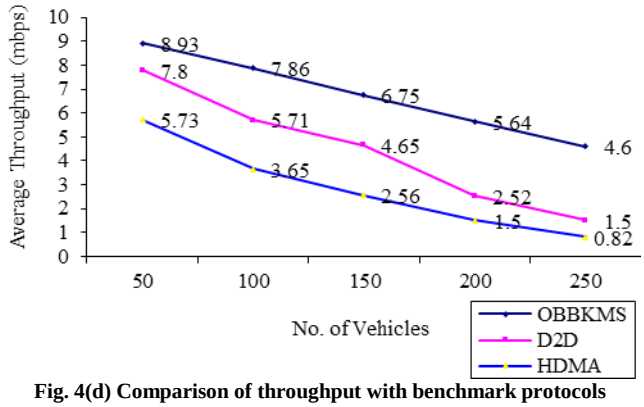

Fig. 4(d) Comparison of throughput with benchmark protocols

The broadcasted message with the key information is stored in $Block_k$. From Figure 4(c), it can be observed that communication overhead had reduced by an average of more than 4 bytes compared with [17] for total number of nodes from 50 to 250 and by 6 bytes compared with [9].

The average throughput is the ratio of the number of bits per time for communication. From Figure 4(d) it can be observed that proposed methodology average throughput is very high compared with [9, 17] by 1.523 and 2.369 times, respectively.

4.2.1. Security Analysis

Threat Model

The threat model considers both internal and external adversaries. Internal adversaries are legitimate vehicles that can exhibit malicious behaviour, trying to manipulate the public key or impersonate other vehicles. The external adversaries are non-legitimate that can eavesdrop, intercept, replay, modify or impersonate messages. This scheme assumes that TA does not register malicious vehicles during initialization and that CH are trustworthy.

Analytical Security Analysis

In this work, the Base Station (BS) has no role in the key management. In case the malicious vehicles imitate CH by changing CL_{key}, they will fail, as message of CL_{key} needs genuine signature of CH. It is challenging for invaders to counterfeit without accurate private key. Hence, it is not possible for attackers to impersonate a particular vehicle $V_{i,j}$ to malevolently update pairs of public or private keys without its genuine private key of and present cluster key using (29).

Attackers pretend to be a genuine member in the cluster CL_i and request for joining another one V_k, which is not possible, as $V_{k,0}$ checks for request legitimacy. In case $V_{i,0}$ is in normal state, malevolent request is immediately rejected as there is no conforming departure transaction. Else, $V_{k,0}$ validates the public key and signature, thereby identifying the attack as malicious vehicles do not have genuine keys.

This provides the Confidentiality of Cluster-Key distribution. This shows the principal argument against eavesdropping and unauthorized cluster access. In this methodology, the signatures are unique, and TA verifies them for validity, thereby avoiding tampering attacks.

Using timestamps [15] provided protection against replay attacks. While trying to impersonate a CH and then a member vehicle by maliciously updating its public and private key pairs, it failed due to non-accessibility of vehicle's / CH private key and the current cluster key. Each vehicle verify the authenticity of the reputation report in the blockchain and made sure that these reports are not tampered. When the attackers tried to gain access to its blockchain, it was able to find only old public keys.

The security benefit of COA is because of security – aware parameters in the CH selection objective as given by Equations (14) to (16). In this work, blockchain integrity is maintained by removing the nonce in conventional blockchain and adding a lightweight miner-election mechanism. So, when an attacker modifies a previously stored transaction, then the resulting hash becomes inconsistent with valid chain, thereby providing tamper evidence for key and identity management transactions.

a) MITM & Impersonation Attacks

A malicious node is placed between D2D pairs and tried to modify selected key. For a message M,

$M' = M + \Delta M$; Where $M' \neq M$ (32)

The signature associated with the original message remains unmodified as

$$\sigma = \text{Sign}_{i,j}^{Pri}\{H(M)\} \quad (33)$$

The receiver verifies $\text{Verify}_{i,j}^{Pri}\{H(M'), \sigma\}$ and should be zero. The average MITM attack and impersonation attack detection rates are 88.83% and 77.02%, respectively. These attack rates are calculated using (34).

$$\text{MITM Detection Rate} = \frac{\text{No.of Malicious Attacks Detected}}{\text{Total No.of Attacks}} \text{ X } 100 \quad (34)$$

Two impersonation type attacks are considered. Firstly, CH impersonation, which has failed because of COA. When the attacker tries to modify the CH public key and transmit a forged message, the receiver needs

$$\text{Verify}_{CH_{i,j}}^{pri} = ((H(M), \sigma_{CH}) = 1$$

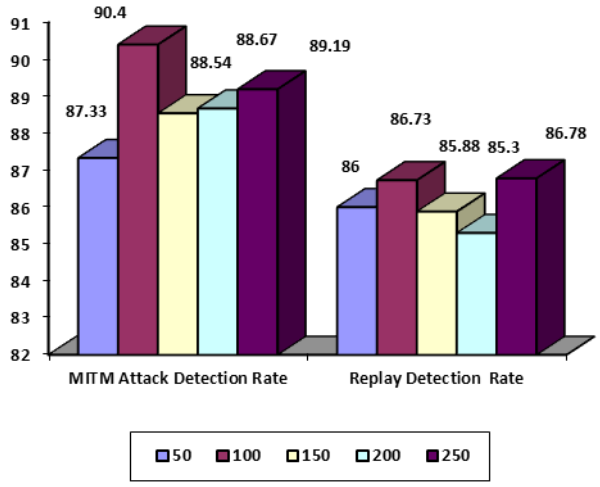

Fig. 5 Average MITM and impersonation attacks detection rates

But since the attacker doesn't possess that key, Verify returns zero. Secondly, when the attacker attempted to replace a legitimate vehicle's public key, the updated key is not available.

b) Tampering & Replay Attacks

The tampering was detected automatically when the signature verification is not successful. i.e., $\text{Verify}_{i,j}^{Pri}\{H(M'), \sigma\} = 0$. To validate for Replay attacks, the previously captured valid messages with a timestamp T_i are injected, i.e., retransmitted. The receiver compares the timestamp with the previously accepted timestamp, and the decision is made. The average detection rates of Tampering and replay attacks for up to 250 nodes is 94.93% and 86.14%, respectively. The overall detection rate of all the combined attacks is 96.58%, and the average detection delay is 9.29 ms.

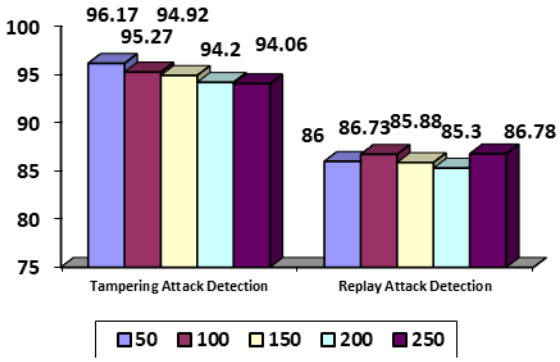

Fig. 6 Tampering and replay attacks detection rate

The four parameters taken in this analysis demonstrate the security – efficiency trade-off and the operational impact of the security mechanisms.

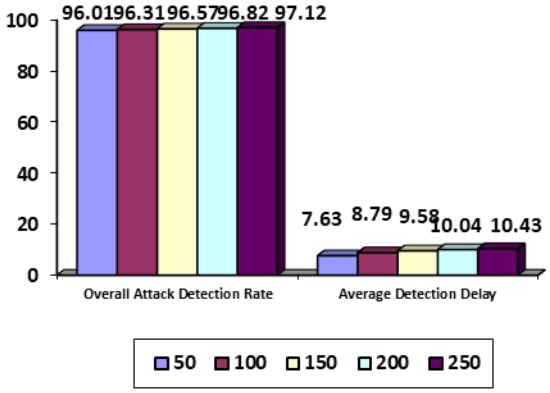

Fig. 7 Overall attack detection rate & average detection delay

The reduced computational cost and communication overhead, including the authentication cost, key generation and update, verification, and blockchain transactions / operations demonstrates the lightweight security, network efficiency and scalability of the proposed algorithm.

Similarly reduced average delay for authentication and message delivery shows the real-time nature and responsiveness of this algorithm. Finally, an increased Throughput shows the availability and scalability under security processing. Table 3 illustrates how this work is better compared to the existing models.

Table 3. Security feature comparison

Feature	HDMA [9]	D2D [17]	Proposed
Message Authentication	√	√	√
Message Integrity	√	√	√
Confidentiality	Limited	√	√
Dynamic Key Management	Limited	√	√
Cluster Key Management	Group	√	√
D2D	√	√	√
5G VANET	√	Not specific to VANET	√
Cluster based VANET	-	Hierarchical Groups	√
Optimized CH selection	-	-	√
Trust based CH selection	-	-	√
Inter-Cluster Vehicle switching	Limited	Group Join / Leave	√
Privacy	√	Partial	√
Light weight blockchain	NA	-	√
Trust Management	-	Limited	√
Automatic Key update	Limited	√	√

Unlike HDMA, which primarily targets efficient D2D message authentication in 5G-enabled VANETs, and the DLT-based architecture, which focuses on decentralized dynamic group management and key establishment, this work integrates authentication, blockchain-assisted key management, trust-aware cluster formation, optimization-based CH selection, dynamic key updating, inter-cluster mobility support, and malicious-vehicle elimination within a unified framework. The NS-3 results further demonstrate that this integrated security architecture can be operated with lower computational cost, lower communication delay, and lower communication overhead while maintaining higher throughput as vehicle density increases. The improvements are obtained under the assumptions in Section 3.1. The major reasons attributed in the improvement of parameters are tabulated in Table 4.

Table 4. Design improvements for better performance

Feature Modification	Metrics Improved	Limitations in [9, 17]
Nonce, Difficulty Field and PoW removed; Hash score miner election Equations (24) and (25).	Computational Cost and Delay	Heavy Consensus and group-wise ledger synchronization [17]
Only CH maintain consensus, and members keep LVL and C	Overhead delay, throughput	Every vehicle/device takes part in the ledger and group updates [17]
COA-based CH selection with trust, lifetime and delay	Throughput, Delay and Security	Fixed / Infrastructure-based authentication [9]
80-bit HEC keys with Diffie-Hellman key exchange	Computational Cost, Overhead	Mixed Asymmetric / Symmetric group signature processing [9]
Per-Vehicle transaction chain (CI, IA, Generator)	Overhead, Security	Group-level records that hide individual misbehavior
1-byte trust value	Overhead, Memory	4-byte real-valued trust
Cluster key broadcasted once per cluster.	Throughput, Overhead	Pairwise / Group re-keying on every change
Majority report revocation and failure transactions	Throughput, Attack detection	No trust-driven removal of malicious vehicles [9]

5. Conclusion

This paper proposes a blockchain-assisted key management mechanism for cluster-based 5G-enabled VANETs to address the challenges of secure authentication, dynamic key management, and efficient communication in highly mobile vehicular environments. The proposed framework integrates trust-, distance-, and delay-aware Cluster Head selection using the Coati Optimization Algorithm (COA), D2D-oriented public–private key management, dynamic key updating, malicious-vehicle elimination, and lightweight blockchain structures.

The resulting design aims to reduce the security-related processing and communication burden while supporting reliable authentication and inter-cluster vehicle mobility. The proposed mechanism was implemented and evaluated in NS-3 using computational cost, communication overhead, average delay, and throughput as the principal performance

indicators. Across the evaluated network sizes, this work consistently achieved lower computational cost and communication overhead than both the Distributed Ledger Technology-based architecture and HDMA. The computational cost was reduced by approximately 36.65% compared with the DLT-based approach and 46.77% compared with HDMA**,** while communication overhead was reduced by approximately 37.26% and 51.89%, respectively. Similarly, the proposed work reduced average delay by approximately 35.33% compared with DLT and 49.45% compared with HDMA based on the reported simulation results. In addition, this work maintained the highest throughput across all evaluated vehicle densities, indicating improved data-delivery capability under increasing network load. These results demonstrate that the proposed combination of optimization-driven clustering and lightweight blockchain-based key management can provide a favorable trade-off between security processing and communication performance in 5G-enabled VANETs. Future work will investigate large-scale heterogeneous vehicular networks, realistic 5G channel and mobility conditions, varying malicious-node densities, formal security verification, and prototype-based validation to further establish the scalability, robustness, and practical applicability of this work.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The authors declared that there is no funding obtained in the research work and publication of this paper.

Acknowledgments

The authors acknowledge the help provided by Mrs. Lavanya Poluboyina for her help during the research and preparation of the manuscript. Author 1 and Author 2 contributed equally to this work.

References

[1] Ing. Boris Cucor, "Outlines of Vehicular Ad-Hoc Networks," *Transportation Research Procedia*, vol. 55, pp. 1312-1319, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Eslam Farsimadan et al., "Vehicle-to-Everything (V2X) Communication Scenarios for Vehicular Ad-Hoc Networking (VANET): An Overview," *International Conference on Computational Science and its Applications*, pp. 15-30, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Weidong Yang, *Security in Vehicular Ad Hoc Networks (VANETS)*, Wirelesss Network Security: Theories and Applications, pp. 95-128, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jabar Mahmood et al., "Security in Vehicular Ad Hoc Networks: Challenges and Countermeasures," *Security and Communication Networks*, vol. 2021, no. 1, pp. 1-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Antonella Molinaro, and Claudia Campolo, *5G for V2X Communications*, 5G Italy White eBook, pp. 1-476, 2019. [Google Scholar] [Publisher Link]

[6] Ying He et al., "D2D-V2X-SDN: Taxonomy and Architecture towards 5G Mobile Communication System," *IEEE Access*, vol. 9, pp. 155507-155525, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Adeel Iqbal et al., "Cognitive D2D Communication: A Comprehensive Survey, Research Challenges, and Future Directions," *Internet of Things*, vol. 24, pp. 1-32, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nishu Gupta et al., "Authentication-based Secure Data Dissemination Protocol and Framework for 5G-Enabled VANET," *Future Internet*, vol. 12, no. 4, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Peng Wang et al., "HDMA: Hybrid D2D Message Authentication Scheme for 5G-Enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5071-5080, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Albert Wasef, and Xuemin Shen, "EMAP: Expedite Message Authentication Protocol for Vehicular Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 78-89, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[11] Rana Muhammad Amir Latif et al., "A Novel Authentication and Communication Protocol for Urban Traffic Monitoring in VANETs Based on Cluster Management," *Systems*, vol. 11, no. 7, pp. 1-27, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] R. Priya, and N. Sivakumar, "Resisting Bad Mouth Attack in Vehicular Platoon using Node-Centric Weight-based Trust Management Algorithm (NC-WTM)," *Connection Science*, vol. 34, no. 1, pp. 1807-1832, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Vishal Sharma, Ilsun You, and Nadra Guizani, "Security of 5G-V2X: Technologies, Standardization, and Research Directions," *IEEE Network*, vol. 34, no. 5, pp. 306-314, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Sherief Hashima et al., "Leveraging Machine-Learning for D2D Communications in 5G/Beyond 5G Networks," *Electronics*, vol. 10, no. 2, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Chun-Ta Li et al., "A Secure and Efficient Communication Scheme with Authenticated Key Establishment and Privacy Preserving for Vehicular Ad Hoc Networks," *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[16] Xinghua Li et al., "Blockchain-based Mutual-Healing Group Key Distribution Scheme in Unmanned Aerial Vehicles Ad-Hoc Network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11309-11322, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shu-Ping Lu et al., "Distributed Ledger Technology Based Architecture for Decentralized Device-to-Device Communication Network," *IEEE Access*, vol. 10, pp. 92006-92022, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Bryan Raj et al., "A Survey on Cluster Head Selection and Cluster Formation Methods in Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, pp. 1-53, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mohammad Dehghani et al., "Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems," *Knowledge-Based Systems*, vol. 259, pp. 1-43, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mahmoud A. Shawky et al., "Blockchain-based Secret Key Extraction for Efficient and Secure Authentication in VANETs," *Journal of Information Security and Applications*, vol. 74, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jyoti Grover, Manoj Singh Gaur, and Vijay Laxmi, *Trust Establishment Techniques in VANET*, Wireless Networks and Security: Issues, Challenges and Research Trends, pp. 273-301, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[22] Said Talbi et al., "Adaptive and Dual Data-Communication Trust Scheme for Clustered Wireless Sensor Networks," *Telecommunication Systems*, vol. 65, pp. 605-619, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Rui Wang et al., "ETMRM: An Energy-Efficient Trust Management and Routing Mechanism for SDWSNs," *Computer Networks*, vol. 139, pp. 119-135, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Na Fan, and Chase Q. Wu, "On Trust Models for Communication Security in Vehicular Ad-Hoc Networks," *Ad Hoc Networks*, vol. 90, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Zsombor Pethő, Zsolt Szalay, and Árpád Török, "Safety Risk Focused Analysis of V2V Communication Especially Considering Cyberattack Sensitive Network Performance and Vehicle Dynamics Factors," *Vehicular Communications*, vol. 37, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Ramasubbareddy Somula, Yongyun Cho, and Bhabendu Kumar Mohanta, "EACH-COA: An Energy-Aware Cluster Head Selection for the Internet of Things Using the Coati Optimization Algorithm," *Information*, vol. 14, no. 11, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Om Pal et al., "Key Management for Blockchain Technology," *ICT Express*, vol. 7, no. 1, pp. 76-80, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Nakamoto, Satoshi, and A. Bitcoin, "A Peer-to-Peer Electronic Cash System," *SSRN*, pp. 1-9, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[29] Ziad Hussein, May A. Salama, and Sahar A. El-Rahman, "Evolution of Blockchain Consensus Algorithms: A Review on the Latest Milestones of Blockchain Consensus Algorithms," *Cybersecurity*, vol. 6, no. 1, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[30] HS de Ocáriz Borde, "*An Overview of Trees in Blockchain Technology: Merkle Trees and Merkle Patricia Tries*," University of Cambridge: Cambridge, UK, 2022. [Google Scholar]

[31] Yawen Tan, Jiajia Liu, and Nei Kato, "Blockchain-based Key Management for Heterogeneous Flying Ad Hoc Network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7629-7638, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] Debajyoti Mukhopadhyay et al., "Securing the Data in Clouds with Hyperelliptic Curve Cryptography," *2014 International Conference on Information Technology*, Bhubaneswar, India, pp. 201-205, 2014. [CrossRef] [Google Scholar] [Publisher Link]