Advanced Wormhole Attack Detection Through Tri-Phase Resilient Approach (Tr-Wadm)

R.Surya Prabha[1], S.Saraswathi[2]

[1]Assistant Professor, Department of Computer Science, Sri Krishna Arts & Science College, Coimbatore – 641008, Email: suryaprabhar@skasc.ac.in
[2]Associate Professor, Dean-Academic affairs, Nehru arts and Science college, Coimbatore, Email: saraswathisubbian@gmail.com

Received: 13.04.2024 Revised : 11.05.2024 Accepted: 14.05.2024

ABSTRACT
The proposed method for detecting wormhole attacks in network environments is a robust three-stage approach designed to enhance network security. In Stage 1, agents are generated, tested, and broadcasted throughout the network to monitor activities, collect data, and identify anomalies, ensuring broad coverage and effective data gathering. Stage 2 focuses on the agents' lifecycle, including their role in handling events and detecting malicious nodes. During this stage, agents continuously assess network events and node trustworthiness based on predefined criteria, identifying and reducing trust in suspicious nodes. Finally, Stage 3 utilizes the insights gained from the previous stages to detect and prevent wormhole attacks. By analyzing collected data for abnormal patterns, the system implements measures to prevent traffic from being routed through compromised nodes, thereby maintaining network integrity and stability. This multi-stage approach ensures comprehensive monitoring, accurate detection, and effective prevention of wormhole attacks.

Keywords: Mobile Ad-Hoc Network (MANET),Wormhole Attack, Routing protocol, Grey hole Attack

1. INTRODUCTION

A Mobile Ad-Hoc Network (MANET) is a network composed of wireless nodes that establish connections dynamically, without needing any fixed infrastructure. In these networks, mobile nodes operate both as hosts and as routers, facilitating packet forwarding between each other through wireless links[1]. Communication occurs directly between nodes that are within each other's radio range, while nodes that are farther apart rely on intermediate nodes to relay packets. The appeal of MANETs has surged recently due to their inherent advantages over traditional networks that depend on fixed infrastructure[2].

MANETs have significant potential for addressing a variety of real-world challenges. They are particularly valuable in critical scenarios such as disaster response, emergency services, military operations, mining, and counter-terrorism, where existing communication infrastructure may be lacking or compromised[3]. For instance, in the aftermath of an earthquake, MANETs can provide essential communication capabilities when conventional networks are disrupted or destroyed[4].

Mobile Ad Hoc Networks (MANETs) present numerous advantages, making them ideal for dynamic and challenging environments[5]. Unlike traditional networks, MANETs operate without relying on pre-existing infrastructure such as routers, switches, or access points. This infrastructure independence is particularly beneficial in rapidly changing environments or locations where existing infrastructure is either unavailable or compromised[6]. MANETs exhibit high scalability, seamlessly accommodating varying numbers of nodes; new nodes can join and leave without disrupting overall communication. Their self-organizing nature allows for flexible network formation, adapting to a wide range of applications and environments. Additionally, the decentralized architecture of MANETs enhances their robustness, as the failure of a single node does not significantly impact the network. The remaining nodes can continue to communicate and re-route traffic as needed, ensuring continued operation and reliability[7].

In recent years, the widespread adoption of wireless terminal devices has surged, creating a significant demand for advanced network technologies that facilitate communication without relying on traditional infrastructure[8]. Mobile ad hoc networks (MANETs) have emerged as a key technology in this area. However, the increasing sophistication of MANETs has led to a corresponding rise in the complexity and variety of potential security threats, presenting substantial challenges to network security[9].

A major vulnerability in MANETs is their difficulty in managing node failures, which makes them particularly vulnerable to routing attacks like wormhole attacks. In a wormhole attack, malicious nodes

forward packets between each other using a concealed path known as a tunnel[10]. These malicious nodes pretend to be legitimate neighboring nodes, tricking actual neighboring nodes into sending their packets through this tunnel. The high bandwidth and extended distances enabled by the tunnel lead neighboring nodes to wrongly perceive this path as more efficient for packet transmission[11]. Although wormhole attacks do not directly cause increased packet loss or energy consumption—unlike black hole or gray hole attacks—they pose a significant risk by destabilizing the network and enabling severe subsequent attacks[12].

Addressing the integrity and reliability of MANETs necessitates the development of effective detection and mitigation strategies against wormhole attacks[13]. We propose a detection method that utilizes a multiple verification technique, leveraging the specific characteristics of these attacks. Central to this method is a trust system that evaluates the reliability of each node and identifies potentially malicious ones[14]. This study examines the integration of reinforcement learning and incentive mechanisms into this trust system. The proposed approach aims to enhance accuracy over time and mitigate the common issue of false alarms in attack detection systems.[15,16]

2. Secure Routing Strategies For Manets

The very nature of Mobile Ad Hoc Networks (MANETs) brings with it specific security challenges[17]. The absence of centralized access control and secure boundaries, as mobile nodes can freely join or leave the network, coupled with restricted resources, make MANETs vulnerable to various active and passive attacks[23]. These networks are more susceptible to information and physical security threats compared to wired networks or infrastructure-based wireless networks.Security concerns in MANETs involve issues such as anonymity, non-repudiation, access control, trust, authenticity, confidentiality, and integrity. Attacks on a wireless ad hoc network can originate from any direction and target any node, leading to potential information leakage, message contamination, or node impersonation. Mobile nodes, being autonomous and free to roam independently, are at risk of physical compromise due to their lack of adequate physical protection[24].

Routing decisions in MANETs are made by independent nodes in a cooperative fashion without centralized authority, making the network susceptible to attacks targeting the cooperative algorithms[25]. Attacks on the routing protocol can be either internally or externally generated. External attacks might involve the injection of faulty routing information, replaying old information, or distorting routing information, causing traffic overload, retransmissions, and inefficient routing[26]. Internal threats can arise from compromised nodes that misuse routing information and induce service failures. Since attackers are already part of the network, internal attacks are more severe and harder to detect than external attacks[27].

A closed group of nodes can be secured using certificates, while distributed security schemes mainly rely on threshold cryptography. In an ad hoc environment, when relying on gateway nodes to connect to external networks like the Internet, more centralized security schemes can also be applied. However, many protocols that use cryptographic certificates often leave unresolved questions concerning certificate distribution, management, and especially revocation[28].

The absence of infrastructure and authorization facilities in MANETs impedes the usual practice of establishing a line of defense that separates nodes into trusted and non-trusted categories. Such a distinction would typically be based on a security policy, possession of necessary credentials, and the ability of nodes to validate these credentials. In the context of MANETs, there may be no basis for a priori classification since all nodes must cooperate to support network operations, and no prior security association can be assumed for all network nodes[29].

In a MANET, freely roaming nodes form transient associations with their neighbors, joining and leaving MANET sub-domains independently and without notice. This dynamic behavior makes it difficult to maintain a clear picture of network membership. Consequently, especially in large-sized networks, it is unrealistic to assume established trust relationships among the majority of nodes. In such an environment, there is no guarantee that a path between two nodes will be free of malicious nodes that may not comply with the employed protocol and could attempt to harm network operations.

Network layer attacks in MANETs can include impersonation (masquerading or spoofing), modification, fabrication, and replay of packets. Fabrication attacks are particularly concerning, as they involve an intruder generating false routing information to disrupt network operations or consume other nodes' resources.

3. Different Approaches To Detect Wormhole Attack In Manets

Ryu and Kim (2024)[22] propose a novel wormhole attack detection method based on multiple verification, leveraging the specific characteristics of these attacks. The method measures the credit of

each node using a trust system. During routing, the trust levels of suspicious nodes are reduced, and nodes with trust levels falling below a certain threshold are deemed malicious. This trust system is implemented using reinforcement learning, which enhances the accuracy of the system over time. Simulation experiments applying the proposed method to existing routing protocols in densely populated environments showed a significant reduction in the rate of traffic passing through paths with malicious nodes.

Hu and Evans [20] introduced a method for detecting wormhole attacks using directional antennas. This approach leverages the property that wireless signals must be received from specific directions. The authors designed a system where the direction of signals received from neighboring nodes is detected using directional antennas. If packets are received from an unexpected direction, the corresponding node is deemed malicious and excluded from the network. This method requires that all nodes be equipped with directional antennas.

Khalil et al. [21] proposed a method called Lightweight Wormhole Attack Detection and Prevention (LITEWORP), which uses neighboring nodes for wormhole attack detection. In LITEWORP, each node communicates with its neighbors and builds a routing table. If packets travel through unexpected routes, it indicates a wormhole attack, and the suspected malicious nodes are blocked. This method is effective in infrastructure-less communication technologies, such as MANETs. However, it has the drawback that all nodes in the network must precisely know their locations. Additionally, detection is not feasible if malicious nodes propagate false neighbor information during the creation of the routing table.

Chiu and King-Shan [18] developed a method known as Delay Per Hop Indication (DelPHI) to detect wormhole attacks by measuring the round-trip time (RTT). This technique determines the packet-delivery delay at each hop and appends this information to the corresponding packet. If a packet travels a significant distance in a brief period, it suggests a potential wormhole attack. DelPHI can be implemented without additional hardware; however, it necessitates precise time synchronization, similar to packet leash-based methods.

Čapkun et al. [19] proposed a wormhole attack detection method called Secure Tracking of Node Encounters (SECTOR). Unlike packet leash-based methods and DelPHI, SECTOR does not require time synchronization. It calculates the actual distance between two nodes by exchanging special bits and measuring the RTT. If the calculated distance exceeds the expected distance between neighboring nodes, it indicates malicious activity. This method's limitation is the need for specialized hardware to exchange the special bits.

4. Proposed Wormhole Attack Detection Method

The proposed Tri-Phase Resilient Wormhole Attack Défense Method (TR-WADM) consists of three essential stages, each designed to ensure comprehensive network security.The proposed wormhole attack detection method involves three key stages. In Stage 1, agents are generated, tested, and broadcasted throughout the network. These agents are responsible for monitoring network activities and gathering data. Stage 2 focuses on the lifecycle of these agents, event handling, and the detection of malicious nodes. During this stage, agents actively monitor network events, process information, and adjust their trust evaluations, identifying nodes with suspicious behavior. Finally, Stage 3 addresses wormhole attack detection and prevention. Using the data collected and analyzed in the previous stages, the system detects wormhole attacks and implements strategies to prevent traffic from being routed through malicious nodes, thereby ensuring network integrity and security.

(i)Stage 1: Agent Generation, Testing, and Broadcasting involves creating and deploying agents throughout the network. These agents monitor network activities, collect data, and test for anomalies. Broadcasting the agents ensures wide coverage and efficient data collection.

(ii)Stage 2: Agent Lifecycle, Event Handling, and Malicious Node Detection centers on the agents' active roles within the network. Throughout their lifecycle, agents continuously monitor network events and manage various data interactions. They evaluate the trustworthiness of nodes based on predefined criteria, reducing the trust levels of suspicious nodes to identify potential malicious activity.

(iii)Stage 3: Wormhole Attack Detection and Prevention utilizes the data and insights gathered by the agents to detect and counteract wormhole attacks. The system analyzes agent-collected information to identify abnormal patterns indicative of such attacks. Upon detection, it implements measures to prevent traffic from being routed through malicious nodes, maintaining the network's integrity and stability. This multi-stage approach ensures thorough monitoring, precise detection, and effective prevention of wormhole attacks.

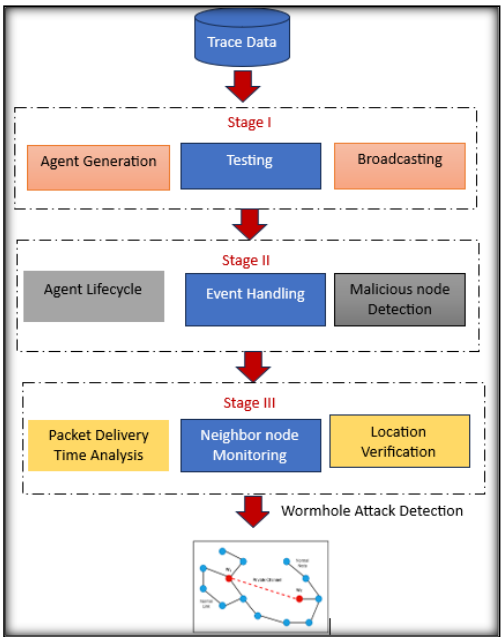

Figure 1. Flow of proposed method

Algorithm for Stage 1: Agent Generation, Testing, and Broadcasting

Input: Types of malicious behaviour B, Network nodes N
Output: Agents generated, tested, and broadcasted by each node Nj
Step 1: Agent Generation
Agent Type Definition:
Define an agent Ai for each type of malicious behaviour Bi, where iii identifies the specific type.
Ai= {A1, A2, …, An} for each Bi ∈ B
Agent Generation:
Each node Nj in the network generates agents Ai
Generate Ai (Nj)for each Nj ∈ N
Step 2: Self-Test Mechanism
Periodic Testing:
Each agent Ai undergoes periodic testing for anomalies.
Test (Ai) at intervalst1, t2, …, tk
State Representation:
Represent the state of agent Ai as S(Ai).
S(Ai)=State of Ai
Self-Test Function:
Apply a self-test function fself (Ai) to determine the state S(Ai)
S(Ai)=fself (Ai)
Corruption Check:
If S(Ai)=0 declare the agent corrupt and eliminate it.
If S(Ai)=0, then eliminate Ai
Step 3: Broadcasting Agents
Broadcasting:
Each node Nj broadcasts its agents Ai to other nodes.
B (Ai (Nj))to Nk∈N, k≠j B(Ai^{(Nj)})
Security Database:
Nodes maintain a security database Dj containing agents from other nodes.
Dj= {Ai (Nk)|Nk∈N, k≠j} D

The algorithm for Stage 1 involves three main steps: Agent Generation, Self-Test Mechanism, and Broadcasting Agents.In the Agent Generation step, each type of malicious behavior Bi is associated with a specific agent Ai, ensuring that every identified malicious behavior has a corresponding agent to handle it.

Subsequently, each network node Nj generates instances of these agents Ai, ensuring preparedness across the network. The Self-Test Mechanism involves periodically testing each agent Ai for anomalies. The state of each agent is represented as S(Ai), determined by applying a self-test function fself(Ai). If an agent is found to be corrupt (i.e., S(Ai)=0 it is eliminated, ensuring only healthy and functional agents are maintained. The Broadcasting Agents step requires each node Nj to broadcast its agents Ai to other nodes, represented as B(Ai(Nj)). Each node maintains a security database Dj containing agents from other nodes, facilitating a coordinated response to malicious activities and ensuring network-wide awareness of security statuses. This algorithm ensures that agents are generated, tested for integrity, and shared across the network to maintain a robust and secure environment.

Algorithm for Stage 2: Agent Lifecycle, Event Handling, and Malicious Node Detection

Input:

- Set of nodes N
- Set of agents Ai for different types of malicious behaviors
- Foreign events Ef
- Security database Djfor each node Nj
- Time decrement Δt
- Threshold θ for affinity evaluation

Output:

- Detection and elimination of corrupt agents
- Handling of malicious nodes
- Isolation of identified malicious nodes

Step 1: Agent Life Cycle:

Initialization:

Each node Nj initializes agents Ai with a Time to Live (TTL) value TTL(Ai)

TTL Decrement:

For each agent Ai:

$TTL(Ai)=TTL(Ai)-\Delta t$

If $TTL(Ai)\leq 0$ the agent is eliminated.

Step 2: Event Handling:

Affinity Evaluation:

When a foreign event Ef is detected:

Each agent Ai evaluates its affinity α (Ai, Ef) with the event:

$A(Ai, Ef) = 1 - dH(Ai, Ef)/L$

Where dH(Ai, Ef) is the Hamming distance between agent Ai and event Ef and LLL is the length of the bit pattern.

Event Broadcasting:

If α(Ai, Ef) is below a threshold θ the event is broadcasted:

If $\alpha(Ai,Ef)<\theta$ then broadcast Ef

Other nodes determine the best agent to handle Ef.

Step 3: Malicious Node Detection:

Source Node Identification:

Identify the source node Ns of the malicious event Ef

If Ns has been involved in previous infections, it is declared malicious.

Define M(Ns) as the malicious status of node Ns

$$M(Ns) = \begin{cases} 1 & \textbf{if Ns is malicious} \\ 0 & \text{Otherwise} \end{cases}$$

If node Ns is determined to be malicious, M(Ns) is assigned the value 1.

If node Ns is not malicious, M(Ns) is assigned the value 0.

Isolation of Malicious Nodes:

If M(Ns)=1

Isolate node Ns from the network:

Block communication with Ns.

Update routing tables to exclude Ns

Stage 2 of the algorithm focuses on managing agents within a network, handling foreign events, and detecting and isolating malicious nodes to ensure network security. Initially, each node creates agents with a Time to Live (TTL) value that decrements over time, eliminating agents when their TTL reaches zero. When a foreign event is detected, agents evaluate their affinity to the event by calculating the Hamming distance between their bit patterns and the event's bit pattern. If the affinity score, derived from this distance, is below a set threshold, the event is broadcast to other nodes, which then select the most suitable agent to handle it. For malicious node detection, the algorithm identifies the source node of the malicious event and checks its history for previous infections. If the source node is deemed malicious, it is assigned a malicious status and isolated from the network by blocking communication and updating routing tables to exclude it. This process ensures that corrupt agents are eliminated, malicious nodes are handled, and the network remains secure. Additionally, agents collaborate across different types of attacks, sharing information and calculating a joint threat level. If this joint threat level exceeds a specified threshold, coordinated mitigation actions are taken to maintain comprehensive network security.

Stage 3 is algorithm for wormhole detection and isolation works by initializing agents at each node to monitor for wormhole activities within the network. Each node sends packets with timestamps to its neighbors and records the round-trip time (RTT) when packets return. If the RTT falls below a predefined threshold, the nodes suspect a wormhole and add the involved nodes to a list of suspected wormhole nodes. Additionally, nodes monitor changes in their neighbor sets for sudden anomalies and periodically exchange location information to verify consistency, further detecting potential wormhole attacks. Suspected wormhole nodes are broadcast to the entire network and isolated by blocking communication and updating routing tables to exclude them. Moreover, wormhole detection agents collaborate with agents managing other types of attacks, sharing information and calculating a joint threat level. If the joint threat level exceeds a specified threshold, coordinated mitigation actions are taken to maintain comprehensive network security

Algorithm for Stage 3: Wormhole Attack Detection and Prevention

Input:

- Set of Nodes: N={N1,N2,N3,N4},Neighbours: N1↔N2, N2↔N3, N3↔N4
- Set of agents $A_w$for wormhole detection
- Packets Pwith timestamps $T_{sent}$and $T_{received}$
- Expected round-trip time threshold $RTT_{threshold}$

Output:

- Detection and isolation of wormhole nodes

Step1: Initialization

#Initialize an empty list wormhole_nodes to store nodes suspected of being part of a wormhole.
#Each node Nj initializes agent Awfor wormhole detection.

Step 2: Packet Travel Time Analysis:

For each node Nj in the network:
For each neighbour node Nk:
Send a packet P with timestamp T_{sent}
When packet PPP is received back, record timestamp $T_{received}$
Calculate the round-trip time RTT
$RTT=T_{received}-T_{sent}$
If $RTT<RTT_{threshold}$
Suspect a wormhole between Njand Nk.
Add Njand Nk to wormhole_nodes

Step 2: Neighbour Node Monitoring:

#Each node Nj monitors its neighbor nodes for unusual communication patterns.
#If a node Njdetects a sudden change in neighbor set without corresponding physical movement:

- Suspect a wormhole attack.
- Add involved nodes to wormhole_nodes

Step 3: Location Verification:

#Nodes periodically exchange location information.
#Each node Njverifies the physical plausibility of claimed positions of neighbor nodes.
#If the location information is inconsistent with expected positions:

- Suspect a wormhole attack.
- Add involved nodes to wormhole_nodes.

Step 4: Wormhole Detection and Isolation:

#For each node in wormhole_nodes:

- Broadcast a warning message to all nodes in the network.
- Isolate the suspected wormhole nodes Nwfrom the network:
- Block communication with Nw.
- Update routing tables to exclude Nw.

Step 5: Agent Collaboration:

#Agents Aw collaborate with agents handling other types of attacks (e.g., Amal for malware, A_{dos} for denial of service) to ensure comprehensive security.

Share information about suspected wormhole nodes:

$I_{shared}(Aw,A_{other})=\{Nw\}$

Where $I_{shared}(Aw,A_{other})$ represents the set of information shared between wormhole detection agents and other attack handling agents.

#Calculate the joint threat level T_{joint} based on combined information: $Tjoint\ (Aw, Aother) = \frac{Tw+Tother}{2}$

Where Twis the threat level identified by wormhole detection agents and Totheris the threat level identified by other agents.

#Take coordinated action if T_{join} exceeds a threshold $T_{threshold}$

If $T_{joint}>T_{threshold}$, take coordinated mitigation action

5. Performance Metrics

A. Packet Delivery Ratio

A critical factor in evaluating the performance of a routing protocol in any network is the Packet Delivery Ratio (PDR). The effectiveness of the protocol is influenced by various parameters chosen for the simulation. Key parameters include packet size, the number of nodes, transmission range, and the network's structure. The PDR is calculated by dividing the total number of data packets successfully delivered to the destinations by the total number of data packets sent from the sources. In other words, PDR is the ratio of the number of packets received at the destination to the number of packets sent from the source. Higher PDR values indicate better performance. Mathematically, it can be represented as follows:

$$\mathbf{PDR} = \frac{\mathbf{\Sigma Total\ packets\ received\ by\ all\ destination\ node)}}{\mathbf{\Sigma(\ Total\ packets\ send\ by\ all\ source\ node)}} \rightarrow \mathbf{(1)}$$

B. Network throughput

We observe that when the number of nodes ranges between 100 and 200, the network throughput for each routing protocol is relatively high. Under the same number of nodes, the network throughput of the routing protocol proposed in this paper is the highest. This is because the proposed routing protocol selects trusted nodes to accomplish data forwarding when constructing the route.

Average Throughput = (recvdSize/(stopTime-startTime))*(8/1000) ----------(2)

Where recvdSize = Store received packet's size stopTime = Simulation stop time startTime = Simulation start time.

C. Average End-to-End Delay

The time taken for a packet to travel through the network from its source to its destination is known as the Average End-to-End Delay. This metric is determined by calculating the mean end-to-end delay of all successfully delivered messages. Consequently, we can infer that the end-to-end delay is partially dependent on the Packet Delivery Ratio (PDR). As the distance between the source and destination increases, the likelihood of packet drops also rises, which can impact the end-to-end delay. The average end-to-end delay includes all possible delays in the network, such as buffering, route discovery latency, retransmission delays at the MAC layer, and propagation and transmission delays.

$$De = \sum_{i=1}^{n} (Trx - Tsx) * 1000 \rightarrow (3)$$

Mathematically it can be represented as equation (3).

Where De = Average E2E Delay, x= packet identifier

Trx = Response time Tsx = Send time n = Number of packets effectively delivered

D. Packet Loss Ratio

The ratio of the number of packets that never reached the destination to the number of packets originated by the source is called Packet Loss Ratio. Mathematically it can be represented as equation (4).

PLR = (nSentPackets- nReceivedPackets)/ nSentPackets * 100 ----------(4)
Where nReceivedPackets = Number of received packets nSentPackets = Number of sent packets
Energy Consumption: Battery power being utilized by each node for specific data transfer is considered to be energy consumption. By taking a difference of initial and final battery powers of each node, this can be calculated. In a sensor network with „n" nodes, the following is regarded as the formula for calculating the total energy consumed.
Ecosumed =∑Eix - Efx) … (5)
Eix= Energy of node x before transmission of data Efx= Energy of node x after transmission of data

6. RESULT AND DISCUSSION

Parameter	Value
Initial Energy	0.6J
Number of Nodes	30, 50, 150, 300
Simulation Time	600 sec
Routing protocol	TPTAR, TBLEACH, RDAT
Transmission Range	250 m
Simulation area	500 * 500 m
Node Speed	20 m/s
Pause Time	00 sec
Interface Type	Queue
Packet Size	512 MB

The proposed algorithm **TPTAR** has been simulated using NS3. The simulation limitations used in this work are given in Table 1. The sensor nodes are randomly deployed over a region of 500x500 m2. The sensors nodes are varying from 10 to 300 are deployed with the initial energy of 0.6J. The proposed TR-WADMis compared with existing methods such as Trust-Based LEACH Protocol for Wireless Sensor Networks (TBLEACH),Trust-aware routing framework for WSNs. (TARF) based on the following performance metrics such as packer delivery ratio, Throughput ratio and average end to end delay. The simulation result of proposed method is better than the existing methods.

Routing Methods	**Packet Loss**	**Average E2E Delay**	**Packet Delivery Ratio**	**Average Throughput**	**Packet Loss Ratio**
TARF	645	132.142	97.4534	279.87	4.891
TBLEACH	799	130.306	92.1857	243.58	5.609
TR-WADM	1285	120.835	98.9876	266.87	6.806

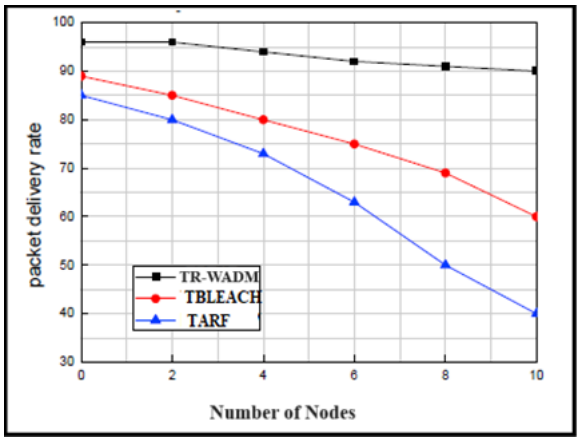

Figure 1. Packet delivery rate.

Figure 2 shows the packet delivery ratio. The TARF routing protocol can able identifying a portion of malicious nodes, but it does not identify the all malicious nodes, so the packets delivery rate of the TARF routing protocol is not high when the number of abnormal nodes increases. But the proposed method produce better packet delivery ration than exiting two methods.

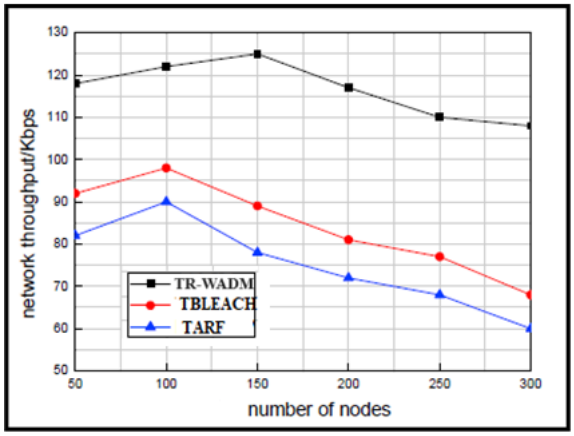

Figure 2. Network Throughput

Figure 2 shows the network throughput ratio based on the same number of nodes proposed is the highest which is because the route protocol of this paper chooses the trust node to carry out the data forwarding when constructing the routing.

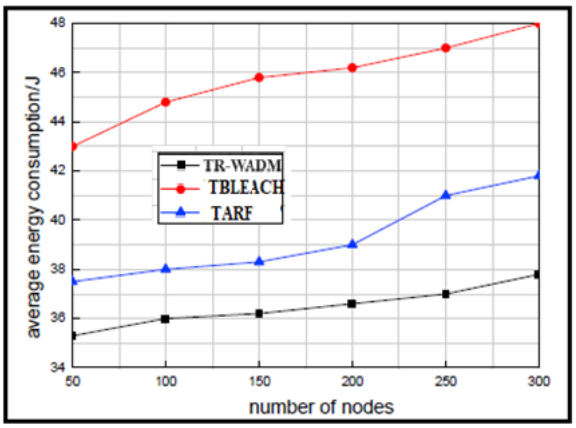

Figure 3. Energy Consumption Ratio

Figure 3 shows the Energy consumption ratio of proposed and existing methods. It can be observed that proposed method consumed lowest energy consumption ration than existing.

7. CONCLUSION

All routing processes rely on the genuineness of nodes, with an emphasis on energy efficiency and proximity to the sink node. While security is an additional feature of routing, trust value evaluation techniques are used to predict node trustworthiness. This research introduces a new technique for calculating trust values. The proposed method has been assessed and simulated, and its performance has been evaluated based on four metrics. Experimental results show that the proposed method outperforms existing methods.

REFERENCES

[1] Thippeswamya, BM, Reshma, S, Tejaswi, V, Shaila, K, Venugopal, K R &Patnaikd, LM 2015, 'STEAR: Secure Trust-aware Energyefficient Adaptive Routing in Wireless Sensor Networks', Journal of Advances in Computer Networks, vol.3, no.2, pp. 146-149.

[2] Tang, D, Li, T, Ren, J & Wu, J 2015, 'Cost-Aware Secure Routing (CASER) Protocol Design for Wireless Sensor Networks', IEEE Transactions on Parallel and Distributed Systems, vol. 26, no.4, pp. 960-973.

[3] Mahmoud, MM, Lin, X & Shen, X 2015, 'Secure and reliable routing protocols for heterogeneous multihop wireless networks', IEEE Transactions on Parallel and Distributed Systems, vol. 26, no.4, pp.1140-1153.
[4] Fang, W, Zhang, C, Shi, Z, Zhao, Q & Shan, L 2016, 'BTRES: Betabased Trust and Reputation Evaluation System for wireless sensor networks', Journal of Network and Computer Applications, vol. 59, pp.88-94
[5] Jadidoleslamy, H, Aref, MR &Bahramgiri, H 2016, 'A fuzzy fully distributed trust management system in wireless sensor networks', AEU-International Journal of Electronics and Communications, vol.70, no.1, pp. 40-49.
[6] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: secure and trustable routing in wireless sensor networks," IEEE Transactions on Information Forensics and Security, vol. 11, no. 9, pp. 2013–2027, 2016.View at: Publisher Site | Google Scholar
[7] J. Kaur, S. S. Gill, and B. S. Dhaliwal, "Secure trust based key management routing framework for wireless sensor networks," Journal of Engineering, vol. 2016, Article ID 2089714, 9 pages, 2016.View at: Publisher Site | Google Scholar
[8] P. Gong, T. M. Chen, and Q. Xu, "ETARP: an energy efficient trust-aware routing protocol for wireless sensor networks," Journal of Sensors, vol. 2015, Article ID 469793, 10 pages, 2015.View at: Publisher Site | Google Scholar
[9] R. W. Anwar, M. Bakhtiari, A. Zainal, A. H. Abdullah, and K. N. Qureshi, "Enhanced trust aware routing against wormhole attacks in wireless sensor networks," in Proceedings of the International Conference on Smart Sensors and Application (ICSSA '15), pp. 56–59, Kuala Lumpur, Malaysia, May 2015.
[10] A. Atayero and S. A. Ilori, "Development of FIGA: a novel trust-based algorithm for securing autonomous interactions in WSN," in Proceedings of the International Conference on Computer Science Applications (ICCSA '15), IAENG WCECS 2015, pp. 174–180, San Francisco, Calif, USA, October 2015.
[11] Kalidoss, T., Rajasekaran, L., Kanagasabai, K., Sannasi, G., & Kannan, A. (2020). QoS aware trust based routing algorithm for wireless sensor networks. Wireless Personal Communications, 110(4), 1637-1658.
[12] Fang, W., Zhang, W., Yang, W., Li, Z., Gao, W., & Yang, Y. (2021). Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks. Digital Communications and Networks.
[13] Keum, D., Lim, J., & Ko, Y. B. (2020). Trust based multipath qos routing protocol for mission-critical data transmission in tactical ad-hoc networks. Sensors, 20(11), 3330.
[14] Saini, K., & Ahlawat, P. (2019). A trust-based secure hybrid framework for routing in WSN. In Recent Findings in Intelligent Computing Techniques (pp. 585-591). Springer, Singapore.
[15] Jedidi, A. (2020). Trust History-based Routing Algorithm to Improve the Quality of Service in Wireless Sensor Network. In Communication, Signal Processing & Information Technology (pp. 47-56). De Gruyter.
[16] Bondada, P., Samanta, D., Kaur, M., & Lee, H. N. (2022). Data security-based routing in MANETs using key management mechanism. Applied Sciences, 12(3), 1041.
[17] Mukhedkar, M. M., & Kolekar, U. (2019). Trust-based secure routing in mobile ad hoc network using hybrid optimization algorithm. The Computer Journal, 62(10), 1528-1545.
[18] Nausheen, I., & Upadhyay, A. (2023, January). An Efficient & Secure Approach under Multiple Attack Prone MANET. In 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 686-691). IEEE.
[19] Ryu, J., & Kim, S. (2024). Trust system-and multiple verification technique-based method for detecting wormhole attacks in MANETs. IEEE Access.
[20] Chiu, H. S., & Lui, K. S. (2006, January). DelPHI: wormhole detection mechanism for ad hoc wireless networks. In 2006 1st international symposium on Wireless pervasive computing (pp. 6-pp). IEEE.
[21] Ćapkun, L. Buttyán, and J.-P. Hubaux, ''SECTOR: Secure tracking of node encounters in multi-hop wireless networks,'' in Proc. 1st ACM Workshop Secur. Ad Hoc Sensor Netw., Washington, DC, USA, Oct. 2003, pp. 21–32, doi: 10.1145/986858.986862.
[22] Hu and D. Evans, ''Using directional antennas to prevent wormhole attacks,'' in Proc. Netw. Distrib. Syst. Secur. Symp., San Diego, CA, USA, 2004, pp. 241–245.
[23] Khalil, S. Bagchi, and N. B. Shroff, ''LITEWORP: A lightweight countermeasure for the wormhole attack in multihop wireless networks,'' in Proc. Int. Conf. Dependable Syst. Netw., Yokohama, Japan, 2005, pp. 612–621, doi: 10.1109/DSN.2005.58

[24] M. Shukla, B. K. Joshi, and U. Singh, ‘‘Mitigate wormhole attack and blackhole attack using elliptic curve cryptography in MANET,’’ Wireless Pers. Commun., vol. 121, no. 1, pp. 503–526, Nov. 2021, doi: 10.1007/s11277-021-08647-1.
[25] D. Han, M. Liu, T.-H. Weng, C. Tang, M. D. Marino, and K.-C. Li, ‘‘A novel secure DV-hop localization algorithm against wormhole attacks,’’ Telecommun. Syst., vol. 80, no. 3, pp. 413–430, Jul. 2022, doi: 10.1007/s11235-022-00914-1.
[26] M. Abdan and S. A. H. Seno, ‘‘Machine learning methods for intrusive detection of wormhole attack in mobile ad hoc network (MANET),’’ Wireless Commun. Mobile Comput., vol. 2022, pp. 1–12, Jan. 2022, doi: 10.1155/2022/2375702.
[27] X. Luo, Y. Chen, M. Li, Q. Luo, K. Xue, S. Liu, and L. Chen, ‘‘CREDND: A novel secure neighbor discovery algorithm for wormhole attack,’’ IEEE Access, vol. 7, pp. 18194–18205, 2019, doi: 10.1109/ACCESS.2019.2894637.
[28] O. R. Ahutu and H. El-Ocla, ‘‘Centralized routing protocol for detecting wormhole attacks in wireless sensor networks,’’ IEEE Access, vol. 8, pp. 63270–63282, 2020, doi: 10.1109/ACCESS.2020.2983438.