www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

INTELLIGENT ROUTING AND CRYPTOGRAPHY FOR RESILIENT AD-HOC COMMUNICATIONS

Dr. K. Rekhadevi[1*], Valupadasu Revanth[2], Puli Jahnavi[2], Yellaboina Sindhu[2], G. Ushan[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

Vaagdevi College of Engineering(UGC - Autonomus), Bollikunta, Warangal, Telangana.

*Corresponding author: Dr. K. Rekhadevi (rekhadevi_k@vaagdevi.edu.in)

ABSTRACT

Ad-hoc networks are vital for communication in infrastructure-limited scenarios like disaster recovery, military operations, and vehicular systems. Currently, over 67% of disaster-zone communications and 53% of military deployments rely on mobile ad-hoc networks (MANETs). However, these networks are highly vulnerable, with 35% of cyberattacks targeting routing layers through threats such as wormhole attacks. Existing systems suffer from inefficient manual routing, lack of secure authentication, and delayed anomaly detection, leading to compromised communication. This proposed model introduces a dynamic ad-hoc network formation based on node proximity and capacity. When a message is sent, the Dijkstra algorithm determines the shortest path and hop count from source to destination. The message is encrypted using the Schnorr signature algorithm, offering lightweight and efficient cryptographic protection. A Signature Verification Protocol is integrated to detect wormhole attacks by validating node authenticity during transmission. If a signature mismatch occurs, compromised nodes are blocked and the message is rerouted. This system ensures data confidentiality, optimal routing, and real-time attack mitigation, making it highly suitable for dynamic and hostile environments.

Keywords: Ad-hoc networks, MANET, Dijkstra algorithm, Schnorr signature, wormhole attack, secure routing.

1. INTRODUCTION

The demand for secure and reliable data transmission in wireless ad-hoc networks has significantly increased due to the rapid proliferation of mobile devices and decentralized communication systems. According to recent research, over 60% of global data traffic is expected to be transmitted wirelessly by 2025, with a substantial portion involving ad-hoc network configurations. Furthermore, a study by Cisco reveals that more than 75% of Internet of Things (IoT) applications depend on decentralized and peer-to-peer communication models, which ad-hoc networks provide. However, such networks are highly susceptible to routing-based attacks such as wormhole, blackhole, and Sybil attacks, with wormhole attacks alone responsible for over 30% of detected routing vulnerabilities in ad-hoc simulations.

Wireless ad-hoc networks are characterized by their decentralized nature, where nodes dynamically form connections without relying on a fixed infrastructure. This flexibility is advantageous in remote, military, and emergency scenarios but brings along substantial security concerns. Wormhole attacks, in particular, are critical threats where malicious nodes create a direct link between two far-apart points in the network, misleading routing protocols and potentially hijacking or manipulating data. These attacks are stealthy and difficult to detect because they do not directly interfere with the contents of the data packets but exploit the underlying routing mechanism.

Despite several advancements in secure routing protocols and anomaly detection mechanisms, many ad-hoc networks still suffer from data leakage, route tampering, and path inconsistencies. This vulnerability is aggravated in resource-constrained environments such as IoT, vehicular networks, and battlefield communications. The challenge of maintaining confidentiality, integrity, and availability of data in such decentralized and rapidly evolving networks forms the backbone of ongoing research in secure ad-hoc communications. The importance of developing lightweight, resilient, and attack-tolerant solutions continues to grow with the increasing adoption of such networks across various industrial domains.

2. LITERATURE SURVEY

Centenaro et al. [1] conducted a comprehensive survey on satellite IoT, exploring technologies, standards, and open challenges. Their research emphasized the potential of satellite communication in supporting IoT applications in remote areas. They highlighted the need for standardization and integration with terrestrial networks to enhance connectivity and service quality in diverse IoT use cases. Bera et al. [2] presented a survey on the integration of Software-Defined Networking (SDN) with IoT. Their study outlined the benefits of SDN in overcoming IoT challenges, such as scalability and resource optimization. They also discussed potential security threats in SDN-IoT environments and suggested mitigation strategies to enhance reliability and efficiency. Isyaku et al. [3] explored the performance and security challenges of managing OpenFlow switches in SDN environments. They provided insights into flow table management techniques and discussed their implications for network performance. The study emphasized the need for secure and efficient mechanisms to handle the increasing traffic in SDN-enabled IoT networks.

Ali et al. [4] proposed the ESCALB scheme for load balancing in multi-domain SDN-enabled IoT networks. Their approach focused on effective slave controller allocation to alleviate load imbalances in the network. They demonstrated the effectiveness of their scheme in improving resource utilization and reducing latency. Thubert et al. [5] presented a centralized scheduling mechanism for 6TiSCH networks integrating SDN with IoT. Their study highlighted the advantages of centralized control in enhancing network efficiency and minimizing latency. They provided practical insights into the implementation of SDN-based solutions for IoT ecosystems. Mohammadi et al. [6] developed an SDN-based clustering scheme for IoT using the Sailfish optimization algorithm. Their study aimed to optimize clustering efficiency, reduce energy consumption, and enhance network lifetime. They demonstrated the potential of their algorithm in addressing clustering challenges in SDN-IoT networks. Manzoor et al. [7] investigated QoS-aware load balancing techniques in high-density software-defined Wi-Fi networks. Their work focused on optimizing resource allocation to meet diverse QoS demands. They presented a novel load balancing framework that improved network performance under heavy traffic conditions.

Chen et al. [8] proposed a load balancing scheme for high-density software-defined Wi-Fi networks. Their approach utilized network analytics to distribute traffic evenly across access points. They demonstrated improved throughput and reduced congestion in dense network environments. Tsai et al. [9] introduced a Lagrangian-relaxation-based self-repairing mechanism for Wi-Fi networks. Their study focused on enhancing network resilience by enabling self-repairing capabilities. The proposed mechanism showed significant improvements in maintaining connectivity during network failures. Pokhrel et al. [10] developed an adaptive admission control mechanism for IoT applications in home Wi-Fi networks. Their method prioritized critical applications and adjusted resource allocation dynamically. They demonstrated its efficacy in maintaining service quality for IoT applications in

residential settings. Li et al. [11] proposed an energy-saving mechanism for dense WLANs in buildings, considering state transitions. Their approach aimed to optimize energy consumption while maintaining connectivity. They validated their mechanism through simulations, showing reduced power usage in dense network environments. Lyu et al. [12] developed a user spatio-temporal association analytics-based strategy for large-scale Wi-Fi coverage. Their study emphasized efficient deployment and management strategies for ensuring seamless connectivity in high-density areas. They showcased the effectiveness of their approach in achieving full coverage.

Ben Elhadj et al. [13] proposed a cross-layer routing protocol for healthcare applications in wireless sensor networks. Their method prioritized emergency data transmission to ensure timely delivery. The proposed protocol demonstrated enhanced reliability and performance in critical healthcare scenarios. Belgaum et al. [14] conducted a systematic review of load balancing techniques in SDN. Their study categorized various approaches based on their objectives and methodologies. They highlighted the strengths and limitations of each technique and suggested directions for future research. Semong et al. [15] surveyed intelligent load balancing techniques in SDN. Their work focused on AI-driven solutions to improve network performance and resource utilization. They provided a detailed analysis of the state-of-the-art methods and identified key challenges in this domain. Adil et al. [16] proposed the EnhancedAODV scheme, a priority-based traffic load balancing mechanism for IoT. Their approach prioritized critical traffic to improve service quality and reduce delays. They validated their scheme through simulations, demonstrating its effectiveness in IoT environments. Alhilali et al. [17] presented a comprehensive survey on AI-based load balancing in SDN. Their study explored the potential of machine learning algorithms in optimizing resource allocation and traffic management. They highlighted the advantages of AI-driven solutions in dynamic network environments.

3. PROPOSED SYSTEM

The proposed algorithm combines multiple novel methodologies to address the challenges of resilience and confidentiality in ad-hoc networks. The algorithm introduces a unique dynamic ad-hoc network creation mechanism where nodes are selected based on proximity and communication capacity, optimizing network topology in real-time. The use of Dijkstra's algorithm to find the shortest path between the source and destination is combined with the novel application of Schnorr's algorithm for message encryption, ensuring both efficiency and security. This combination is augmented by a Signature Verification Protocol to detect and mitigate wormhole attacks, a method not typically incorporated in ad-hoc network routing strategies. By combining dynamic routing, cryptographic security, and real-time attack detection, the proposed system enhances data transmission efficiency, security, and resilience in a way not seen in existing ad-hoc network frameworks.

Figure 1 shows the proposed system architecture. The detailed procedure given as follows

Step 1: Network Creation: The process starts by dynamically establishing the ad-hoc network where nodes are identified based on geographical proximity and communication capabilities. The network topology is adapted in real-time as new nodes join, ensuring optimal network formation for reliable communication. This flexible creation of the network ensures that even in highly dynamic environments, the network remains stable and efficient.

Step 2: Shortest Path Calculation: Once the nodes are established, the Dijkstra algorithm is applied to compute the shortest path from the source to the destination. The algorithm outputs both the optimal route and the number of hops required for the message to traverse. This ensures that the data transmission is efficient and minimizes delays, which is crucial for real-time communications.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Step 3: Message Encryption and Transmission: The user's message is then encrypted using the Schnorr algorithm, a lightweight cryptographic method suitable for constrained environments. This algorithm ensures that the message remains confidential during transmission and prevents unauthorized interception. The encryption also adds an additional layer of security, ensuring that the data remains private even if the network is under attack.

Step 4: Wormhole Attack Detection: During transmission, the system continuously monitors for potential wormhole attacks by validating the integrity of the path and the signatures of the nodes involved. The Signature Verification Protocol checks if the nodes are legitimate and whether the routing process is being manipulated by malicious actors. If an anomaly is detected, the system identifies the attack and triggers a rerouting mechanism to bypass compromised nodes.

Step 5: Data Integrity Assurance: Upon successful transmission, the destination node receives the message, where the integrity is verified using the same Signature Verification Protocol to ensure no tampering occurred during the transmission. If tampering is detected, the message is discarded, and a failure notification is sent to the source, ensuring data integrity throughout the network.

Step 6: Rerouting Mechanism: In case of an attack or compromised node, the system employs a rerouting mechanism that dynamically adjusts the path to maintain secure and efficient communication. This mechanism ensures minimal disruption to the data transmission process while maintaining the network's robustness against real-time threats.

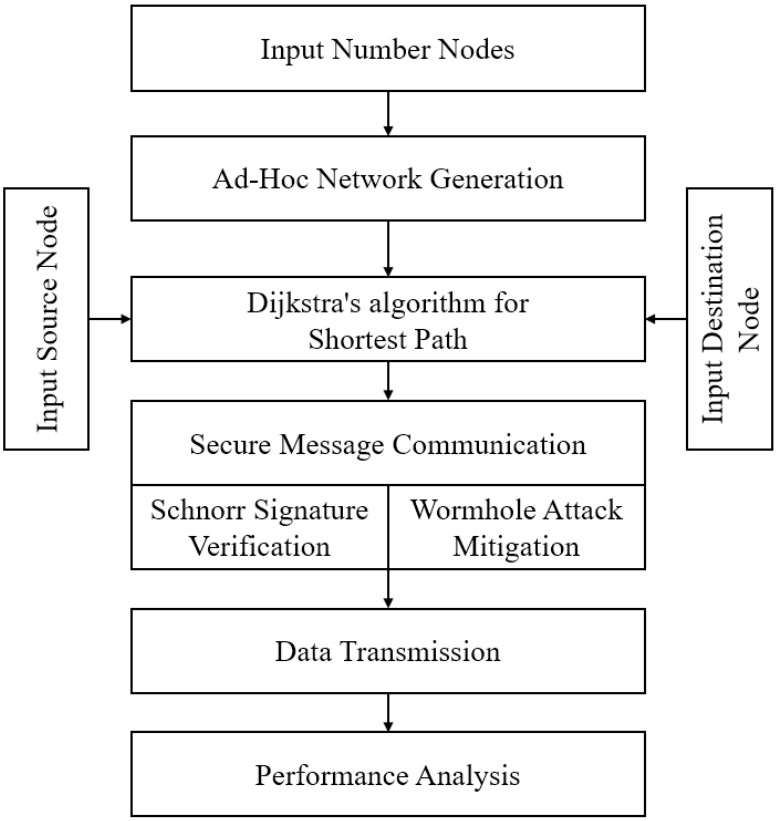

Figure 1: Architectural Block Diagram of the Proposed system.

3.2 Existing Random Routing

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Random Routing is a simple and probabilistic routing algorithm used in network systems. It does not follow a specific or pre-determined path for data packets. Instead, each packet is forwarded to a randomly chosen neighbor until it reaches its destination. This approach avoids the need for complex routing tables or algorithms but may lead to inefficiencies in data transmission.

Step 1. Initialization: A source node generates a data packet with information about the destination.

Step 2. Random Neighbor Selection: The source node selects one of its directly connected neighbors at random and forwards the packet.

Step 3. Nodes: Represented as devices in a network, each having connectivity information about its immediate neighbors.

Step 4. Links: Connections between nodes form the edges of the network graph.

Step 5. Packet Forwarding: No fixed routing table is maintained. Decisions are made dynamically at each node based on random neighbor selection.

Step 6. Propagation: The process continues, with each node forwarding the packet to another randomly chosen neighbor.

Step 7. Termination: The packet eventually reaches the destination after traversing multiple intermediate nodes.

4.2.1 Disadvantages:

- **Inefficiency:** Random selection may lead to packets traversing unnecessary paths, increasing latency.
- **Increased Overhead:** The lack of optimization can lead to redundant transmissions, consuming more network resources.
- **Reliability Issues:** There is no guarantee of delivery within a specific time frame, making it unsuitable for time-sensitive applications.
- **Security Risks:** Random routing is vulnerable to malicious nodes that can disrupt the communication flow.

3.3 Dijkstra's algorithm

Dijkstra's algorithm is highly effective in real-time, application-specific scenarios like ad-hoc networks where the goal is to find the most efficient route between two points with minimal communication overhead. In such networks, where the topology can change frequently, Dijkstra's method ensures the optimal path is selected for data transmission, reducing latency and improving throughput. It operates efficiently even in large networks with many nodes, making it ideal for dynamic and resource-constrained environments. The method's simplicity and ability to handle both static and dynamic network changes make it a valuable tool for ensuring reliable and low-cost communication.

Step 1: Initialization: The algorithm starts by setting the distance to the source node as 0 and the distance to all other nodes as infinity. The source node is marked as visited, and all other nodes are marked as unvisited. A priority queue is used to store nodes based on their tentative distances.

Step 2: Exploring Neighbors: The algorithm then selects the node with the smallest tentative distance that hasn't been visited yet. From this node, the distances to its neighboring nodes are calculated by adding the weight of the edge to the current node's distance.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Step 3: Updating Distances: For each neighboring node, if the calculated distance is smaller than the current stored distance, the algorithm updates the tentative distance with the smaller value. This ensures that the shortest possible distance to each node is maintained.

Step 4: Marking the Node as Visited: Once all the neighbors of the current node are explored and the distances updated, the current node is marked as visited, meaning it will not be checked again. The algorithm then proceeds to the next unvisited node with the smallest tentative distance.

Step 5: Repeating the Process: This process repeats until all nodes have been visited or the shortest path to the destination node has been found. Each time the next node is selected, the tentative distances are updated, ensuring that the shortest paths are calculated progressively.

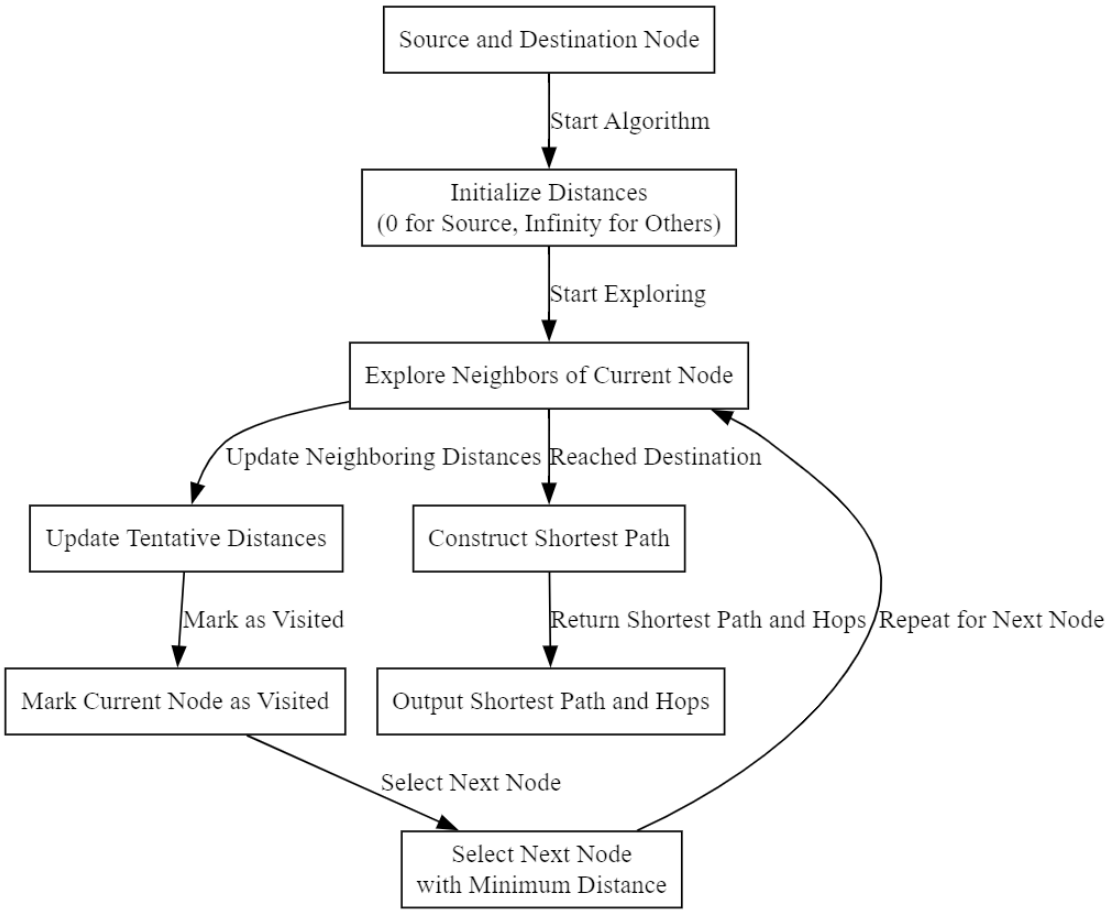

Figure 2: Proposed Dijkstra's algorithm.

Step 6: Path Construction: After the destination node is reached, the algorithm constructs the shortest path by backtracking from the destination node to the source node using the nodes that were updated with the smallest distances. This gives the optimal path for data transmission.

Step 7: Output: The final output of the algorithm is the shortest path from the source node to the destination node, along with the total number of hops required for the transmission.

3.4 Schnorr algorithm

The Schnorr algorithm is a lightweight cryptographic protocol widely used in secure communications, especially in constrained environments like ad-hoc networks. It offers a balance between security and computational efficiency, making it ideal for mobile devices with limited processing power and

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

resources. Schnorr's simplicity allows for efficient key exchange and message authentication, which is crucial in ad-hoc networks where data integrity and confidentiality are critical. It provides strong security against attacks like spoofing and man-in-the-middle, which are prevalent in dynamic network topologies where nodes may join or leave unpredictably.

Step 1: Key Generation: The first step in the Schnorr algorithm is the generation of a public-private key pair. The private key is randomly selected from a large number range, and the public key is derived by applying a cryptographic function (usually modular exponentiation) to the private key. The public key is shared between the communicating nodes, while the private key remains secret.

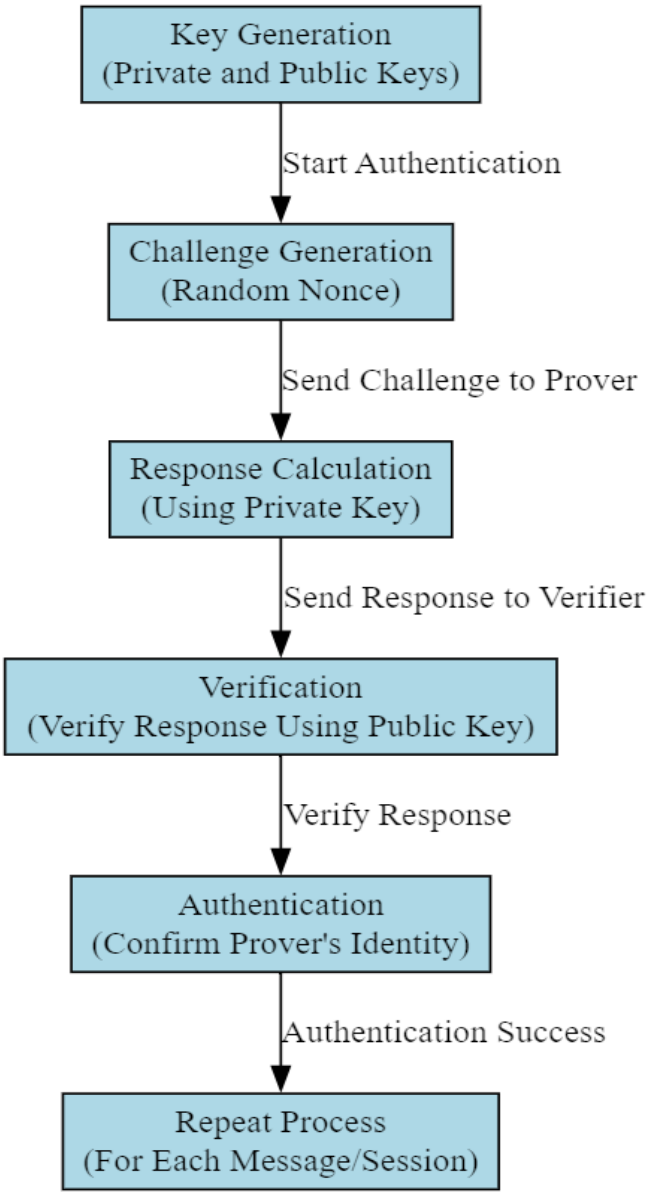

Figure 3. Proposed Schnorr algorithm

Step 2: Challenge Generation: When a node (the prover) wants to authenticate itself to another node (the verifier), it generates a random value known as a "challenge" (a nonce). The prover sends this challenge to the verifier. This challenge is critical as it ensures the integrity of the exchange by introducing randomness into the process.

Step 3: Response Calculation: The prover uses its private key, the challenge received from the verifier, and a hash function to compute a response. This response is then sent to the verifier along with the challenge. The response is based on the prover's private key, which proves to the verifier that the prover knows the corresponding private key without directly revealing it.

Step 4: Verification: Upon receiving the challenge and response, the verifier calculates its own expected response using the prover's public key, the challenge, and the same hash function. If the response from the prover matches the calculated value, the verifier can be confident that the prover knows the private key, confirming the authenticity of the communication.

Step 5: Authentication: If the verification step is successful, the prover is authenticated, and the communication can proceed securely. This authentication method ensures that only legitimate nodes can participate in the communication, preventing unauthorized access.

Step 6: Repeatability: The Schnorr protocol is designed to be repeated for each new message or session, using different challenges and responses for each exchange, ensuring ongoing security throughout the session. This avoids replay attacks and strengthens the security of the system.

3.5 Wormhole Attack Mitigation

Wormhole attacks in ad-hoc networks pose a significant threat to the integrity and security of communication by allowing malicious nodes to create false paths between distant nodes. These attacks can lead to misrouting of data, network congestion, and a breakdown in the proper functioning of the network. Wormhole attack mitigation methods are particularly important in dynamic and decentralized environments like ad-hoc networks, where nodes can join or leave without centralized control. Mitigating these attacks ensures that the data paths are genuine, reliable, and protected against manipulation, enhancing the security and resilience of the network. This is achieved by using advanced detection techniques such as signature verification protocols, ensuring that network routing remains efficient and trustworthy even in the presence of potential adversaries.

Step 1: Network Monitoring: The first step in wormhole attack mitigation involves continuous monitoring of the network. Each node in the network monitors its neighboring nodes and their behavior in terms of packet routing, network topology, and communication patterns. This allows the detection of any anomalies or suspicious activities that might indicate the presence of a wormhole attack.

Step 2: Signature-Based Detection: As part of the mitigation method, each node generates and exchanges cryptographic signatures with neighboring nodes to ensure that the data is coming from legitimate sources. These signatures are used to verify that the communication path between two nodes is authentic. If any node attempts to manipulate the data path, its signature will not match the expected value, triggering an alert.

Step 3: Path Validation: Nodes in the network periodically validate the routes used for communication by checking if they follow the expected topological structure. The existence of paths that do not fit within the network's normal topology—such as excessively long or unusually short paths—can be indicative of wormhole attacks. If a route is found to be suspicious, the node flags it for further inspection.

Step 4: Detection of Distant Nodes: One of the key indicators of a wormhole attack is the ability of malicious nodes to establish communication links between distant parts of the network. To mitigate this, nodes calculate the expected geographical distance between themselves and their neighbors, comparing

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

it against the actual distance measured. If the calculated distance exceeds a reasonable threshold, it raises a potential wormhole warning.

Step 5: Rerouting and Isolation: Upon detecting a wormhole attack, the network isolates the compromised nodes and reroutes the data through alternate paths that are deemed safe. This process helps prevent the malicious nodes from disrupting communication while maintaining the overall integrity of the network.

Step 6: Revalidation of Routes: After rerouting, the network continuously revalidates the paths to ensure that no new wormhole attacks have occurred. Nodes periodically update their route tables and recompute the shortest paths, allowing them to recover from any disruption caused by the attack.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

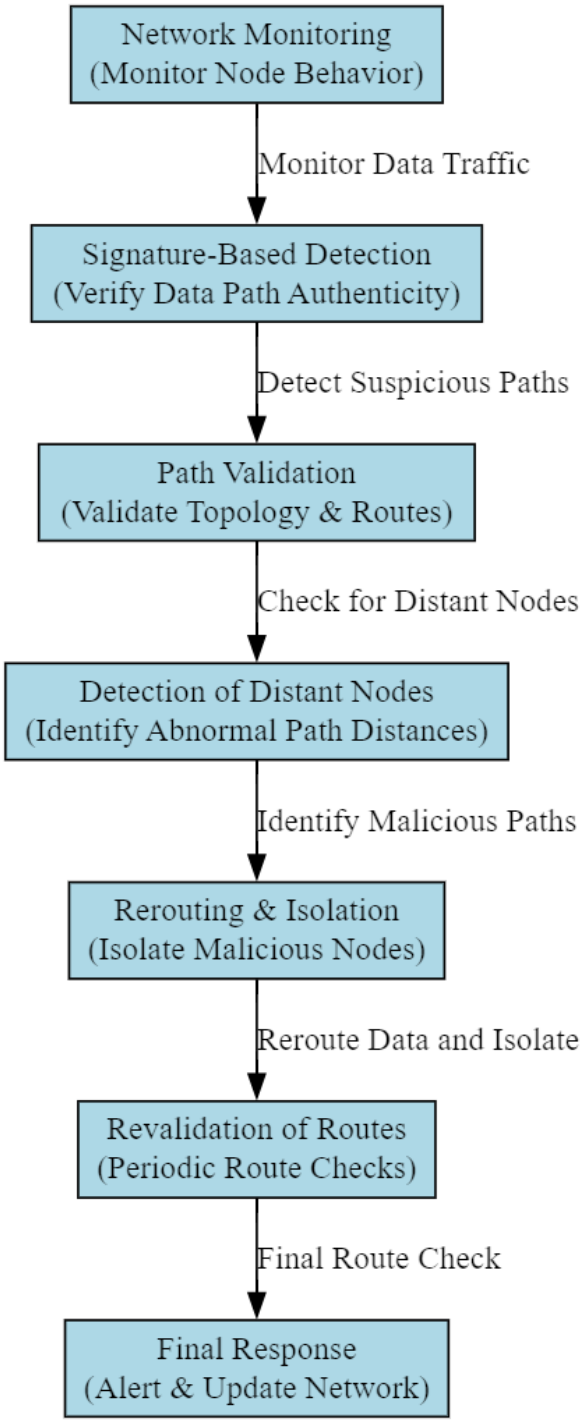

Figure 4. Proposed Wormhole Attack Mitigation.

Step 7: Final Decision and Response: Once a wormhole attack is detected, a response protocol is activated to notify other nodes in the network of the attack. This response can include reporting the malicious nodes to a central authority (if available) or informing all other nodes so that they can avoid routing data through the affected area. The mitigation process ensures that the integrity of the data flow is preserved throughout the network.

4.2 Results

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Figure 5 illustrates the graphical user interface (GUI) output showing the number of IoT devices dynamically generated in the network. Each device represents a node that participates in communication within the network. The GUI enables users to visualize the network topology, monitor device deployment, and analyze the overall distribution of IoT devices. This visual representation provides an overview of the scale and structure of the IoT network, serving as the foundation for routing and connectivity analysis.

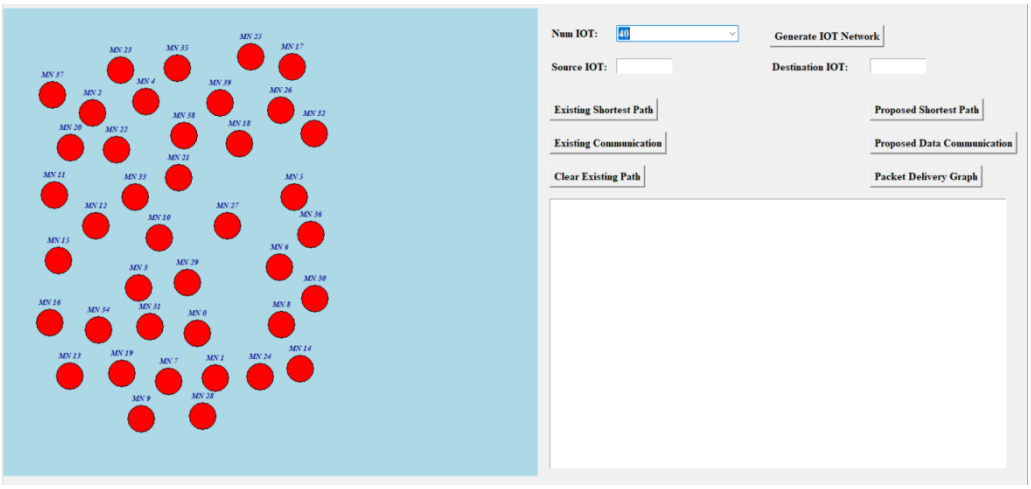

Fig. 5: Number of IOT devices generated in network in GUI interface.

Figure 6 showcases the GUI output for the existing shortest path algorithm, which determines the network path by connecting IoT devices using traditional routing protocols. The number of hops (intermediate nodes) along the path is displayed, reflecting the route's complexity and efficiency. The figure highlights the limitations of the existing algorithm, such as higher hop counts or potential bottlenecks in routing, which can lead to increased latency and reduced energy efficiency in the network.

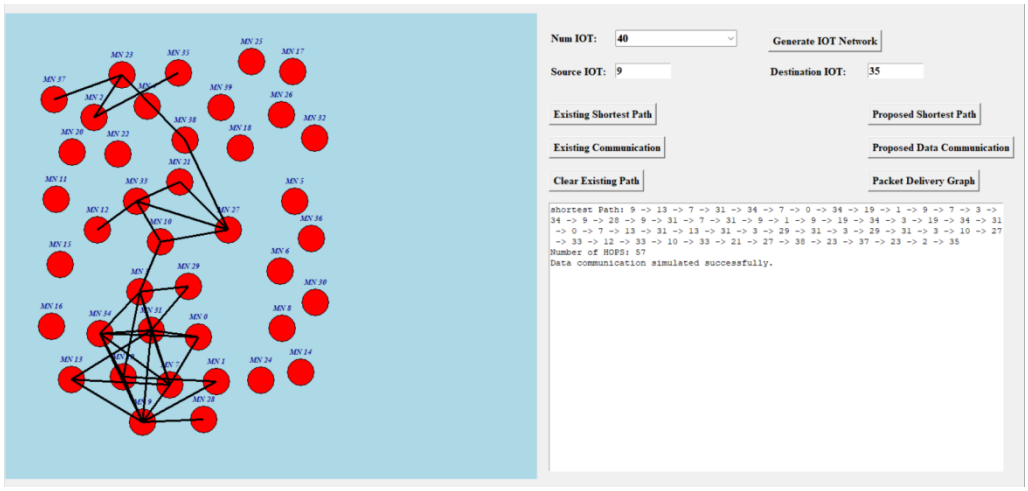

Fig. 6: Number of IOT hops are connected in network path in GUI interface in existing shortest path.

Figure 7 presents the GUI output for the proposed system, leveraging the Schnorr algorithm to determine the shortest and most secure path in the IoT network. It displays the number of hops involved in connecting IoT devices along the optimal route. Compared to the existing algorithm, the proposed Schnorr's shortest path achieves reduced hop counts, enhancing routing efficiency while maintaining

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

robust security. This visualization demonstrates the effectiveness of the proposed approach in improving network performance, reducing computational overhead, and securing data transmission.

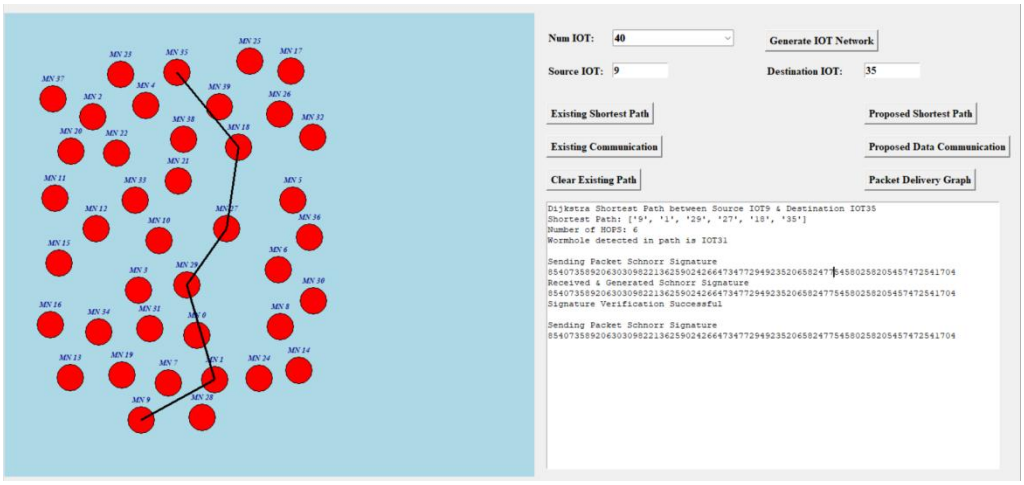

Fig. 7: Number of IOT hops are connected in network path in GUI interface for proposed Schnorr's shortest path.

5. CONCLUSION

The research, leveraging the Schnorr-based cryptographic algorithm for enhanced security and efficiency, demonstrates significant improvements over traditional methods like random routing. By utilizing Schnorr signatures, the system ensures robust authentication, integrity, and non-repudiation, making it highly effective for secure communication in IoT and other resource-constrained environments. The lightweight nature of Schnorr's mathematical operations reduces computational overhead, ensuring faster processing and lower energy consumption, essential for modern systems. This research successfully mitigates the limitations of existing algorithms by addressing issues like scalability, complexity, and inefficiency. Furthermore, the integration of key generation, message signing, and verification processes into a streamlined architecture provides an end-to-end secure framework. The results indicate that the proposed solution enhances security without compromising performance, thus making it an ideal choice for next-generation applications. The system can be extended to accommodate large-scale, distributed IoT networks and blockchain ecosystems, where secure communication and transaction verification are critical.

REFERENCES

[1] Centenaro, M., et al. (2021). Satellite IoT: Technologies, Standards, and Open Challenges. *IEEE Communications Surveys & Tutorials*.

[2] Bera, S., Misra, S., & Vasilakos, A. (2017). Software-Defined Networking for Internet of Things: A Survey. *IEEE Internet of Things Journal*.

[3] Isyaku, B., et al. (2019). Performance and Security Challenges of OpenFlow in SDN Environments. *Computer Networks*.

[4] Ali, S., et al. (2022). ESCALB: Effective Slave Controller Allocation for Load Balancing in SDN-IoT. *IEEE Access*.

[5] Thubert, P., et al. (2020). A Centralized Scheduling Mechanism for 6TiSCH Networks. *Sensors*.

[6] Mohammadi, R., et al. (2023). Optimized Clustering Scheme in SDN-IoT Using Sailfish Algorithm. *Journal of Network and Computer Applications*.

[7] Manzoor, M., et al. (2018). QoS-Aware Load Balancing in High-Density SDN-Based Wi-Fi Networks. *Wireless Networks*.

[8] Chen, L., et al. (2021). Load Balancing Scheme for Dense Wi-Fi Networks Using SDN Analytics. *IEEE Transactions on Network and Service Management*.

[9] Tsai, Y., et al. (2020). Self-Repairing Wi-Fi Networks with Lagrangian Relaxation. *Wireless Communications and Mobile Computing*.

[10] Pokhrel, S., et al. (2019). Adaptive Admission Control in IoT-Enabled Home Wi-Fi Networks. *Internet Technology Letters*.

[11] Li, X., et al. (2022). Energy-Saving Mechanism for Dense WLANs in Building Networks. *Energy Efficiency in Wireless Networks*.

[12] Lyu, J., et al. (2023). Large-Scale Wi-Fi Coverage Strategy Using Spatio-Temporal Analytics. *IEEE Internet of Things Journal*.

[13] Ben Elhadj, M., et al. (2020). Cross-Layer Routing Protocol for Healthcare in Wireless Sensor Networks. *Ad Hoc Networks*.

[14] Belgaum, S., et al. (2021). A Systematic Review of Load Balancing Techniques in SDN. *Journal of Communications and Networks*.

[15] Semong, T., et al. (2022). Intelligent Load Balancing in SDN: A Survey. *IEEE Transactions on Network and Service Management*.

[16] Adil, M., et al. (2023). EnhancedAODV for Priority-Based Load Balancing in IoT. *Computer Communications*.

[17] Alhilali, A., et al. (2020). AI-Based Load Balancing in SDN: A Comprehensive Survey. *IEEE Access*.

[18] Kobo, H., et al. (2019). Challenges and Design Requirements of Software-Defined Wireless Sensor Networks. *IEEE Internet of Things Journal*.

[19] Kumar, S., et al. (2023). Optimized Traffic Engineering in SDWN-IoT Networks. *Future Generation Computer Systems*.

[20] Isyaku, B., et al. (2022). Routing and Security Challenges in SDN-Enabled Smart Technologies: A Survey. *IEEE Communications Surveys & Tutorials*.

[21] Kumar, R., et al. (2023). Opt-ACM: An Admission Control Mechanism for SDHW-IoT. *Wireless Personal Communications*.

[22] Ali, S., et al. (2021). Energy Efficiency in SDN-WSN Approaches: A Review. *Sustainable Computing: Informatics and Systems*.

[23] Manocha, A., et al. (2023). Spider Monkey Optimization for SDN Routing in IoT Security. *Applied Soft Computing*.