

NAPLAM: a novel ledger-based algorithm for detection and mitigation of sinkhole attacks in routing protocol for low power and lossy networks-based Internet of things

Akshaya Dhingra¹, Vikas Sindhu¹, Lakshay Dhingra²

¹Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, India

²Indian Institute of Foreign Trade (IIFT), New Delhi, India

Article Info

Article history:

Received Sep 4, 2024

Revised Feb 14, 2025

Accepted Mar 5, 2025

Keywords:

Internet of Things (IoT)

Low power and lossy networks (LLNs)

Network and packet ledger to ascertain malicious devices/nodes (NAPLAM)

Routing protocol for low power and lossy networks (RPL)

Sinkhole attack

ABSTRACT

The Internet of Things (IoT) is a network of connected physical objects that collect and share data over the Internet. However, routing attacks can disrupt data exchange, especially multi-node sinkhole attacks in low power and lossy IoT networks (LLNs). To support communication in LLN IoT, the IPv6-based routing protocol for LLNs (RPL) is used. Despite having several advantages, RPL also faces challenges like being vulnerable to attacks, having limited resources, compatibility, and scalability issues. Additionally, traditional security methods often do not work well for LLN-IoT devices because they lack the necessary computing power. To overcome these challenges, we have proposed a novel ledger-based framework called network and packet ledger to ascertain malicious devices using routing protocol for LLN (NAPLAM-RPL). This framework can effectively detect and mitigate multi-node sinkhole attacks in IoT networks. This paper also compares NAPLAM-RPL with similar protocols using the NetSim Simulator. The experimental analysis shows that NAPLAM-RPL improves network performance and outperforms existing methods like RF-trust, SoS-RPL, INTI, C-TRUST, and heartbeat algorithm in crucial areas, including packet delivery rate (PDR), throughput, End-to-End (E2E) delay, energy consumed, and detection accuracy.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Akshaya Dhingra

Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University

Rohtak, Haryana, India

Email: akshaya.rs.uiet@mdurohtak.ac.in

1. INTRODUCTION

The Internet of Things (IoT) technology integrates networking, sensing, data processing, and machine learning technologies to solve problems [1], [2]. The term IoT combines “Internet,” meaning worldwide connection of computer networks, and “things,” which can be any uniquely identified objects interconnected to this global network. Overall, IoT is an IPv6-based network that integrates software/hardware to exchange data over the internet [3]. Today, IoT is being deployed in multiple applications like transport, health, industry, agriculture, homes, and military, to make our lives easier and better.

Despite having significant benefits, IoT also has some severe challenges. Here are a few considerable difficulties in IoT. The devices used in IoT environments are public and utilize wireless communication technology, making systems more susceptible to routing attacks. Secondly, IoT connects many embedded mobile devices and systems that need help with scaling, dynamic flexibility, and

compatibility concerns [4]. The critical aspect of the IoT is that most attacks have occurred through the Internet. The IoT devices are resource-constrained and have little memory, limited computing resources, and power. Additionally, more challenges arise in IoT due to the mobile nature of devices and systems [5].

IoT devices cannot be secured using traditional methods because these techniques require more computation and processing capabilities. Apart from these issues, many devices are being used in IoT that can create a new problem called scalability in IoT. Therefore, a reliable algorithm or detection technique is required to efficiently address these issues and detect and mitigate internal and external routing attacks in low-power and lossy network (LLN) IoT networks [6]. Our proposed scheme, namely NAPLAM-RPL (network and packet ledger to ascertain malicious devices using routing protocol for LLN), is a ledger-based technique that stores the node's information (receiver ID, App ID, and number of Packet Received attributes when the data packets are in transit) in the form of intricate sets whenever the IoT network is created and initialized. To detect malicious nodes or attackers in the network, all the data packets are tracked, and the throughput of each node is calculated. Secondly, it calculates the real-time packet drop rate based on the pre-set threshold value. If the packet drop rate of a node/device is more than the threshold value, it marks it as a malicious or intruder node. Lastly, it removes the intruder from the network and actively projects the worst rank possible to prevent the node from entering the network again.

The overview of previous literature related to insider and outsider routing attacks occurring in IoT networks and their detection methods are explained in Table 1 (in Appendix) [6]–[13], [14]–[23].

This research aims to describe a ledger-based secured protocol (NAPLAM) that protects RPL-based IoT networks against Multi-Node Sinkhole attacks. The primary contributions of NAPLAM-RPL are:

- a. Ledger-based HashMap is joined with the RPL protocol for efficient data exchange and control packets between nodes.
- b. The NAPLAM is embedded in the RPL protocol source code using C++ language in Netsim Simulator.
- c. This scheme can detect and mitigate sinkhole attacks occurring in RPL IoT using parameters like packet delivery rate (PDR), throughput, end-to-end (E2E) delay, energy consumed (EC), and detection accuracy, respectively.
- d. The proposed algorithm NAPLAM-RPL efficiency is evaluated by comparing its results with existing security schemes.

2. PROPOSED ALGORITHM

This section gives an overview of the proposed algorithm NAPLAM-RPL, which we have designed based on the ledger data structure. This algorithm detects and mitigates Sinkhole attacks in RPL-IoT networks using HashMap [24] and HashSet [25] structure codes to safely store the data packet information and detect malicious nodes once the network is created. The steps involved in the algorithm are described in Figure 1.

- a. Step 1: Ledger creation, initialization

In Sinkhole attacks, the intruder node acts as the genuine node and advertises false rank in the network using control messages. Whenever a new IoT network is initialized and configured, an event will occur in the network. So, as soon as an event happens in the network, a ledger is created and initialized to store information about all the lost packets using the TrackPacketsInit() function of the Track_Packet.c file.

- b. Step 2: Structure of the ledger

In this step, a ledger is created to store the node's information (Receiver ID, App ID, and number of Packet Received attributes when the data packets are in transit) in the form of nested HashMap and HashSet, namely PacketTrackerMap and Malicious Nodes.

PacketTrackerMap: It is a nested HashMap that stores the ReceiverId of the packet as its key and stores other related information using multiple layers of objects. The detailed structure of this map is shown in Figure 2. The HashMap, namely "PacketTrackerMap," was first created. Each object/entity/entry to this HashMap is called a "Packet Tracker" object. This HashMap has two fields. The first is the ReceiverId, which stores the receiver node number as a Key Value. This HashMap's second field is a pointer pointing to the next HashMap, "ReceivedPacketMap," only when it has the correct key value. Similarly, ReceivedPacketMap has two fields, namely AppID in the form of key-value and Packet Record pointer, which points to a pair object. Each entity stored in ReceivedPacketMap is called a "ReceivedPacket" object. The pair object Packet Record contains only two fields, i.e., no. of packets received and packet set. The field packet set points to a hash set that stores the packetID of packets successfully transmitted but not yet retransmitted by the receiving node. By having this hash set record, we can track packets between hops.

Malicious Nodes: It is a kind of blacklist for suspicious nodes. This HashSet stores the ID of the already identified malicious nodes by the algorithm so that sinkhole attacks can be mitigated by reducing the priority of these nodes in the destination oriented directed acyclic graph (DODAG). The structure of this HashSet is shown in Figure 3. The ledger uses HashMaps and HashSets to minimize the time and energy needed to store, update, and access ledger information.

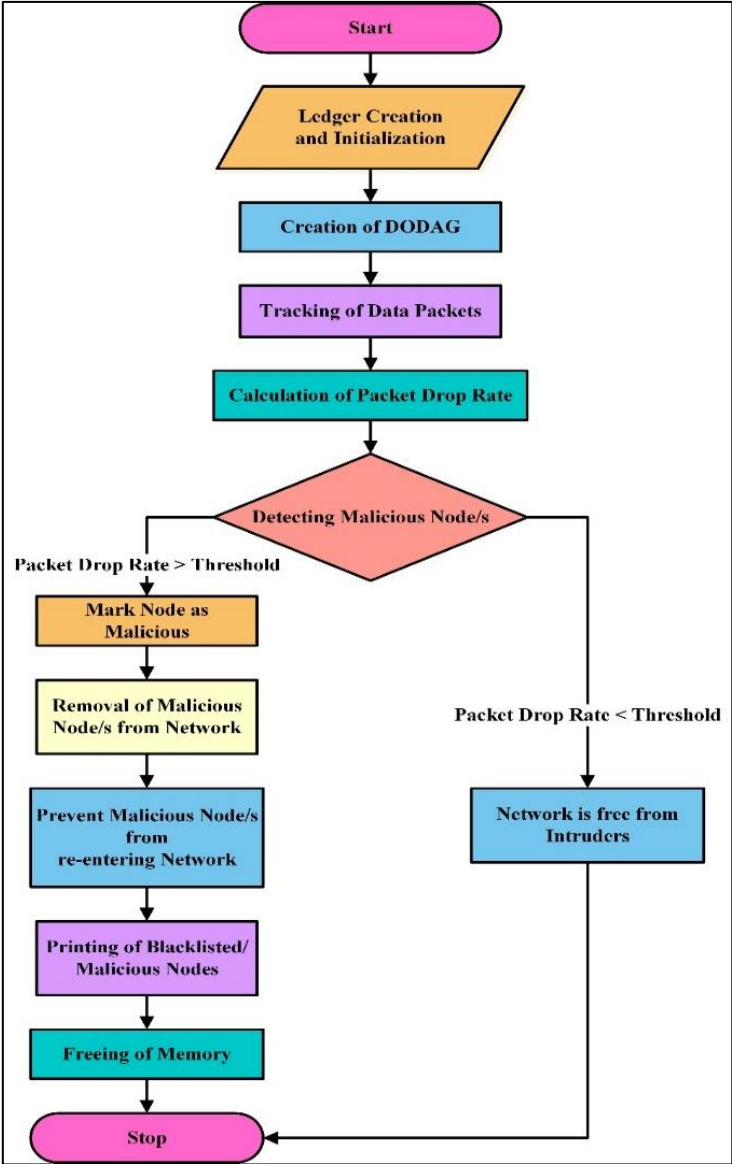


Figure 1. NAPLAM-RPL algorithm flowchart

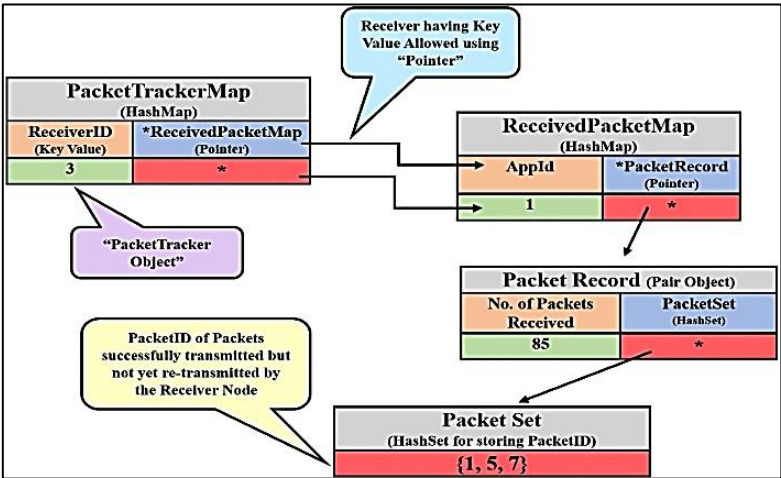


Figure 2. Structure of PacketTrackerMap in NAPLAM

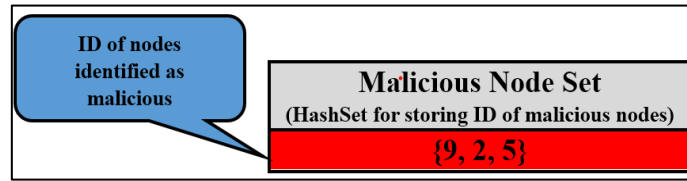


Figure 3. Structure of malicious node set

c. Step 3: Creation of DODAG

Initially, the algorithm allows all the nodes to exchange DAO and DIO messages freely to form a DODAG to send application data. During this time, malicious nodes act the same way as other nodes in the network and project an excellent rank to lure their neighbors so that other nodes choose it as their preferred parent because it is not possible to know if the node really has a good rank (due to its location and network strength) or is falsely projecting a good rank (due to its malicious nature).

d. Step 4: Tracking of application data packets

As soon as any event involving the transmission and reception of a packet occurs, the ledger stores the record of successfully transmitted packets and removes the record of packets successfully received. However, during this process, it still records the total packets a node gets to calculate throughput.

e. Step 5: Calculation of packet drop rate

After every event, the ledger is updated, and a real-time packet drop rate is also calculated using the ledger data.

f. Step 6: Detecting a malicious node

That node is identified as malicious if the packet drop rate is higher than a certain threshold percentage (say 80%), along with other criteria. Specific provisions are made if the number of packets sent till now is tiny and for packets that might be in transit to minimize false positive detection of malicious nodes.

g. Step 7: Marking a node as malicious

Once detected, we add such a node to the malicious node HashSet (which is a blacklist) to mark it as a malicious node.

h. Step 8: Removing the malicious nodes from the DODAG

As soon as malicious nodes are detected, NAPLAM IDS changes the preferred parent of all the child nodes of the malicious node to some other parent and removes the malicious one from the network.

i. Step 9: Preventing malicious nodes from re-entering the network

Now, using our Malicious Nodes hash set, NAPLAM IDS actively projects the worst rank possible for the malicious nodes against their try to project a fake good rank to enter the network. This prevents malicious nodes from entering the network again.

j. Step 10: Printing of detected malicious nodes

All the nodes that are detected as malicious ones (or blacklisted) by NAPLAM are printed with the number of packets that successfully reach the destination.

k. Step 11: Freeing of Memory

Memory is freed at the IoT network's closing or the simulation end.

3. RESULTS AND DISCUSSION

3.1. Performance evaluation

This section estimates the performance of the proposed NAPLAM-IDS using the NetSim v13.3 simulator and visual studio. A simulation model consisting of 100 sensor nodes, a 6LoWPAN, a router, and a wired node (acting as receiver) is created in the NetSim simulator to evaluate NAPLAM-RPL, as shown in Figure 4. The simulation is performed by increasing the number of sinkhole nodes from 10 to 50 in each round to assess the performance of the proposed scheme. Table 2 shows the simulation parameters selected for the performance evaluation of the proposed scheme. The performance of the proposed NAPLAM-RPL scheme is compared with that of existing schemes RF-Trust [6], SoS-RPL [7], INTI [20], C-TRUST [17] and Heartbeat algorithm [9] against multi-node sinkhole attacks. The results of the network are evaluated based on five performance parameters explained below.

PDR is an essential metric for evaluating the reliability and efficiency of a network. It represents the ratio of packets successfully delivered to their intended destination versus the total number of packets transmitted and given by (1).

$$\text{Packet Delivery Rate (in Kbps)} = \frac{\text{Number of Packets Successfully Received}}{\text{Total Number of Packets Transmitted}} \quad (1)$$

Throughput measures the data or information processed and transmitted by an application within a specific time frame. It indicates the capability of an application to handle a particular no. of transactions or requests per unit of time. The application throughput can be calculated using (2).

$$\text{Throughput (in Kbps)} = \frac{\text{Total Payload delivered to destination (in Bytes)} \times 8 \times 1000}{\text{Simulation Time (in } \mu\text{sec)}} \quad (2)$$

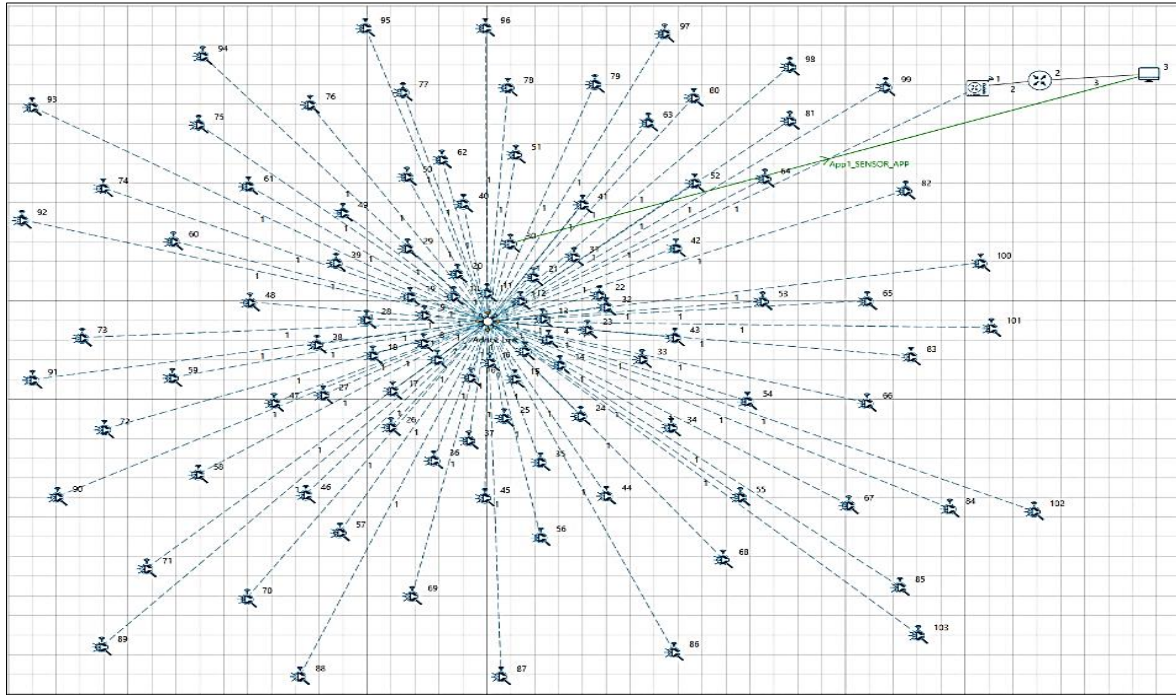


Figure 4. Simulation setup

Table 2. Simulation parameters	
Simulation parameters	Values
Simulator	Netsim v13.3
Network coverage area	500m * 500m
Network protocol	RPL
Number of nodes	100
Node type	Wireless Sensor
Number of malicious nodes	10, 20, 30, 40, 50
Simulation time	3600 seconds
MAC_layer protocol	IEEE 802.15.4
Transport protocol	UDP
Traffic	Sensor App
Data rate	3072 bps
Packet size	64 Bytes
Inter-arrival time	166666 μseconds
Transmission range	50 m

Average E2E delay measures the time it takes for data to be transmitted from a source to a destination across a network, including all delays incurred along the way. In NetSim, the average E2E delay is the time packets take between Application_{Out} and Application_{In} time and is given (3).

$$\text{Average E2E delay (in ms)} = \frac{(\text{Application}_{\text{Out}} - \text{Application}_{\text{In}}) \times 1000}{\text{Total Number of Packets Transmitted}} \quad (3)$$

Energy consumed (EC) refers to the ratio of energy consumed by an individual wireless node to the total number of nodes in the network. It is measured in milli-joules (mJ). The average power consumed in a network depends on various factors, such as the network topology, the number of nodes, the type of sensors used, the application, and the communication protocol used, and is given by (4).

$$\text{Average Energy Consumed (in mJ)} = \frac{\sum_{i=1}^n \text{Energy Consumed by WSN}_i}{\text{Total Number of WSNs}} \quad (4)$$

Detection Accuracy is the ratio of the number of nodes correctly identified as illegitimate or malicious nodes to the total number of malicious nodes in the network. It can be calculated using (5).

$$\text{Detection Accuracy} = \frac{\text{Number of nodes that are correctly identified as illegitimate or malicious}}{\text{Total number of malicious nodes in the network}} \quad (5)$$

3.2. Comparative analysis

This section performs a comparative analysis of simulation results for proposed NAPLAM-RPL, RF-Trust, and SoS-RPL schemes. This evaluation of results is done based on performance metrics, i.e., PDR, throughput, E2E delay, energy consumption, and detection accuracy.

Figure 5 depicts the packet delivery rate (PDR) for NAPLAM-RPL, RF-Trust, SoS-RPL, C-Trust, INTI, and heartbeat algorithm at varying malicious nodes from 10 to 50. It is observed that the maximum PDR for the proposed NAPLAM-RPL scheme is 96.69% in comparison with 93.2% in the case of RF-Trust, 86.18% for SoS-RPL, 87.7% for C-Trust, 83.19% for INTI and 79.8% for Heartbeat algorithm respectively. The proposed approach blacklists the malicious nodes from the RPL-DODAG formation. Therefore, the average calculated PDR in the case of NAPLAM-RPL is 87.43%, while the average PDRs for RF-Trust, SoS-RPL, C-TRUST, INTI, and Heartbeat algorithm are 81.72%, 74.27%, 74.97%, 71.47% and 68.62%. Hence, NAPLAM-RPL delivers 5.71%, 13.16%, 12.46%, 15.96 and 18.81% more average packets in comparison to RF-Trust, SoS-RPL, C-Trust, INTI, and Heartbeat algorithm, respectively.

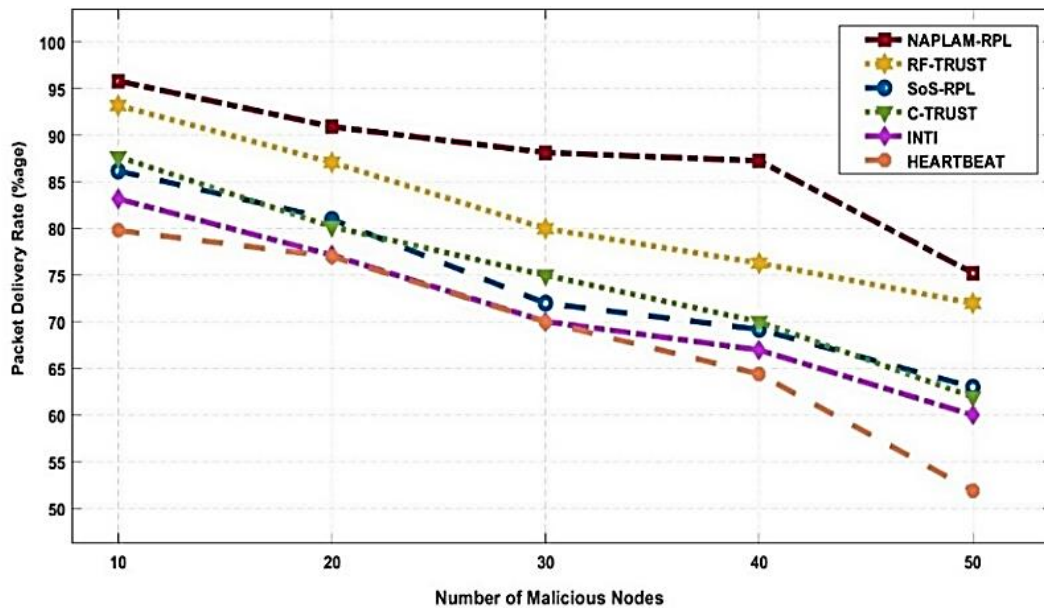


Figure 5. PDR v/s number of malicious nodes

Figure 6 depicts the throughput (in Kbps) w.r.t. number of malicious nodes for the proposed scheme, RF-Trust, SoS-RPL, INTI, C-TRUST, and Heartbeat algorithm. It is observed from the figure that throughput for all the scenarios decreases gradually with an increase in the number of malicious nodes. As NAPLAM-RPL uses a rank-based prediction scheme to detect sinkhole nodes, this scheme can achieve a maximum throughput of 3032 Kbps for a network with ten sinkhole nodes. Meanwhile, the RF-Trust SoS-RPL, C-TRUST, INTI and heartbeat algorithm lags behind, having throughput rates of 2877, 2520, 2608, 2376, 2103 Kbps, respectively. Therefore, the proposed scheme delivers a maximum payload to the destination compared to other schemes.

The impact of the average E2E delay (in microseconds) w.r.t number of malicious nodes is illustrated in Figure 7. It is observed that the proposed scheme has the lowest delay as it can detect and blacklist sinkhole nodes earlier than other models. The average delay for the NAPLAM-RPL lies in the range of (1729.5 to 2809 μ sec) in comparison with (2647 to 3120 μ sec) for RF-Trust, (2789-3261 μ sec) for SoS-RPL, (2879-3261 μ sec) for INTI, (3012-3464 μ sec) for C-TRUST and (3204 to 3550 μ sec) for Heartbeat algorithm. Therefore, the proposed algorithm takes less time to detect sinkhole nodes than other schemes.

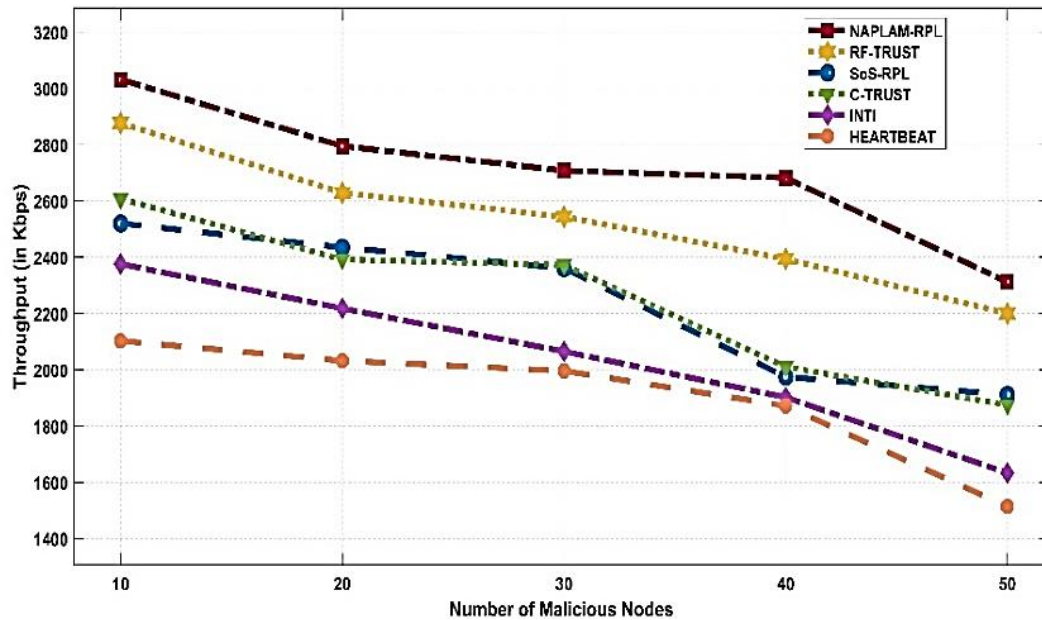


Figure 6. Throughput v/s number of malicious nodes

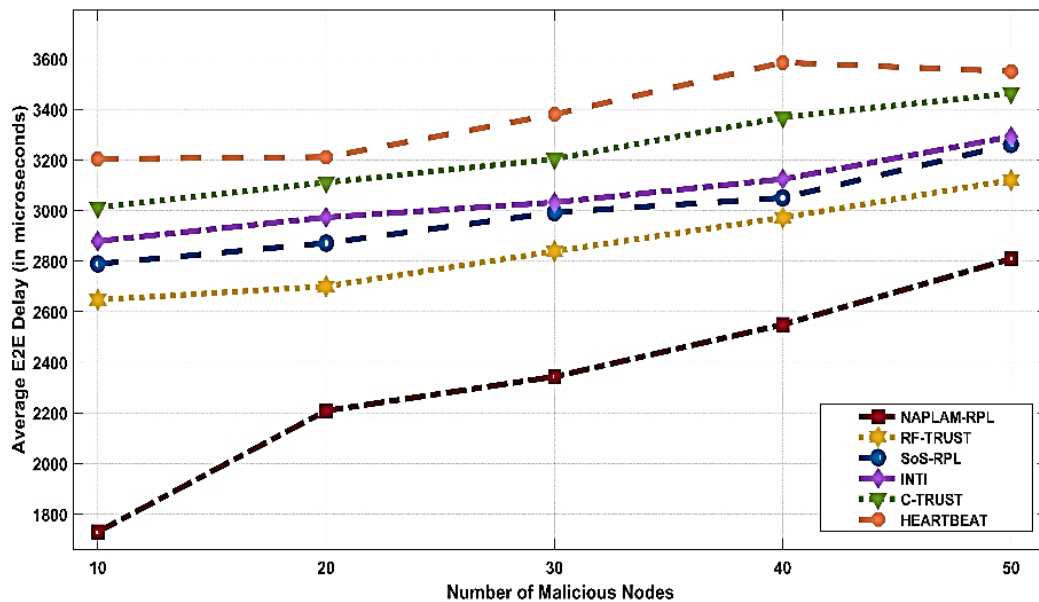


Figure 7. Average E2E delay v/s number of malicious nodes

The energy consumed by all the schemes w.r.t. and the number of malicious nodes is illustrated in Figure 8. The figure shows that the energy consumed (in millijoules(mJ)) by the sensor nodes in each scheme increases proportionally with the number of malicious nodes from 10 to 50. The proposed NAPLAM-RPL

algorithm is energy-efficient as it uses energy in the range of 44.2 to 53.85 mJ, in comparison with RF-Trust, SoS-RPL, INTI, C-Trust and Heartbeat algorithm which consumes energy ranging from (63.57 to 76.99 mJ), (66.36 to 85.8 mJ), (122.65 to 126.85 mJ), (131.54 to 161.39 mJ) and (135.25 to 181.22mJ). The average energy consumed by sensor nodes for NAPLAM-RPL is (48.41 mJ), RF-Trust is (69.27 mJ), SoS-RPL is (73.57 mJ), INTI is (124.77 mJ), C-TRUST is (145.88 mJ) and Heartbeat algorithm is (154.636 mJ) respectively. Therefore, it is observed that the proposed NAPLAM-RPL consumes 30%, 34.85%, 61.2%, 66.8% and 68.69% less energy than RF-Trust, SoS-RPL, INTI, C-Trust and Heartbeat algorithms.

Figure 9 shows the detection accuracy (%) of all the algorithms, i.e., proposed NAPLAM-RPL, RF-Trust, SoS-RPL, INTI, C-Trust, and Heartbeat algorithm w.r.t. the number of malicious nodes. The graph shows that detection accuracy reduces proportionally with the increase in sinkhole nodes. The maximum detection accuracy of the proposed NAPLAM-RPL is 98.5% in a network with ten malicious nodes. Meanwhile, the other schemes, i.e., RF-Trust, SoS-RPL, INTI, C-Trust, and Heartbeat algorithm, were 97.2%, 96.5%, 95.35%, 92.9%, and 86.32%, respectively. Therefore, the overall detection accuracy of the proposed scheme is higher than that of the other schemes.

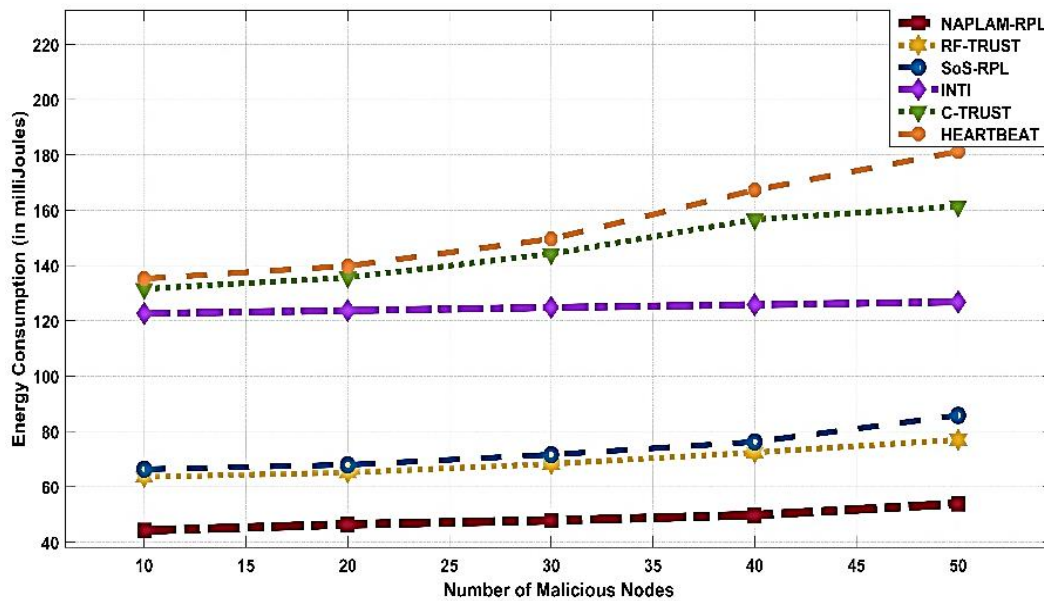


Figure 8. Energy consumption v/s number of malicious nodes

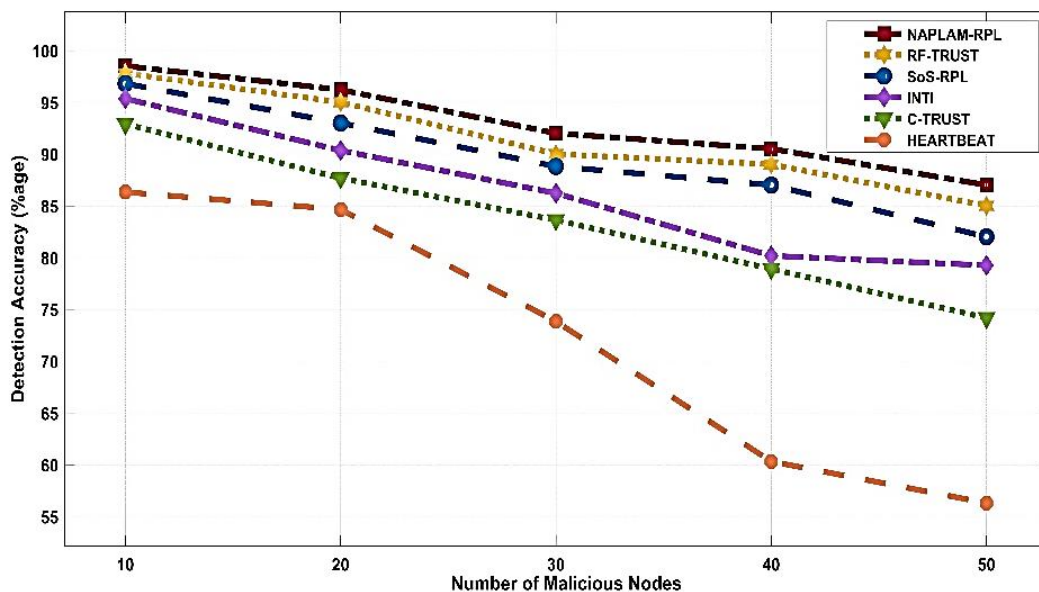


Figure 9. Detection accuracy v/s number of malicious nodes

The experimental results show that the proposed NAPLAM-RPL framework outperforms other schemes, i.e., RF-Trust, SoS-RPL, INTI, C-Trust, and Heartbeat algorithm regarding PDR when the number of malicious nodes is varied between 10 and 50. The NAPLAM-RPL framework achieved a maximum PDR of 96.69%, which is higher than the PDRs of RF-Trust (93.2%), SoS-RPL (86.18%), C-Trust (87.7%), INTI (83.19%) and Heartbeat algorithm (79.8%) respectively. In addition, the proposed framework also showed higher throughput, lower end-to-end delay, and lower energy consumption compared to the other schemes. The detection accuracy of NAPLAM-RPL was also higher than other schemes, which indicates its effectiveness in identifying and blacklisting malicious sinkhole nodes from RPL-DODAG formation. The experimental results prove that the NAPLAM-RPL framework is a promising solution for providing secure and reliable communications in IoT networks.

4. CONCLUSION

This article presents a novel NAPLAM-RPL framework for detecting and mitigating multi-node sinkhole attacks in resource-constrained IoT networks. The proposed framework is based on ledger creation and provides a trust-based mechanism to identify and blacklist malicious nodes from the RPL-DODAG formation. The results show that the proposed NAPLAM-RPL framework provides the highest detection accuracy (98.5%), PDR (95.83%), and throughput (3032 Kbps) in comparison with the existing schemes, i.e., RF-Trust, SoS-RPL, INTI, C-Trust and heartbeat algorithm. The article also provides valuable insights into the current state-of-the-art sinkhole attack detection and mitigation in IoT networks and contributes significantly to advancing security in IoT networks. The NAPLAM-RPL framework is a promising solution for providing secure and reliable communications in IoT networks vulnerable to sinkhole attacks. As a future extension of this work, we will further enhance the robustness and applicability of the NAPLAM-RPL algorithm in diverse IoT environments, ultimately leading to better protection against a broader range of security threats.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to all those who have contributed to the success of this research. I express my heartfelt gratitude to Dr. Anil Sangwan, for his invaluable guidance, in writing the comparative analysis section of this research paper. I would also like to acknowledge the editorial team's and reviewer's contributions, which strengthened the clarity and consistency of this research paper.

FUNDING INFORMATION

No funding was received to assist with preparing this manuscript.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Akshaya Dhingra	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Vikas Sindhu					✓	✓				✓		✓		
Lakshay Dhingra		✓	✓	✓	✓				✓		✓			

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data supporting this study's findings are available from the corresponding author, Akshaya Dhingra, upon reasonable request.

REFERENCES

- [1] S. Shanmugam, V. Muthu Ganeshan, K. Prathapchandran, and T. Janani, "Mitigating black hole attacks in routing protocols using a machine learning-based trust model," *International Journal of Sociotechnology and Knowledge Development*, vol. 14, no. 1, pp. 1–23, Oct. 2022, doi: 10.4018/IJSKD.310067.
- [2] A. Seyfollahi and A. Ghaffari, "A review of intrusion detection systems in RPL routing protocol based on machine learning for Internet of Things applications," *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/8414503.
- [3] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, Dec. 2018, doi: 10.1186/s13677-018-0123-6.
- [4] G. Sharma, J. Grover, and A. Verma, "Performance evaluation of mobile RPL-based IoT networks under version number attack," *Computer Communications*, vol. 197, pp. 12–22, Jan. 2023, doi: 10.1016/j.comcom.2022.10.014.
- [5] J. C. S. Sicato, S. K. Singh, S. Rathore, and J. H. Park, "A comprehensive analyses of intrusion detection system for IoT environment," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 975–990, 2020, doi: 10.3745/JIPS.03.0144.
- [6] K. Prathapchandran and T. Janani, "A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest – RFTRUST," *Computer Networks*, vol. 198, p. 108413, Oct. 2021, doi: 10.1016/j.comnet.2021.108413.
- [7] M. Zaminkar and R. Fotuhi, "SoS-RPL: securing internet of things against sinkhole attack using RPL protocol-based node rating and ranking mechanism," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1287–1312, May 2020, doi: 10.1007/s11277-020-07421-z.
- [8] J. Pacheco and S. Hariri, "Anomaly behavior analysis for IoT sensors," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, May 2018, doi: 10.1002/ett.3188.
- [9] E. Garcia Ribera, B. Martinez Alvarez, C. Samuel, P. P. Ioulianiou, and V. G. Vassilakis, "An intrusion detection system for RPL-based IoT networks," *Electronics (Switzerland)*, vol. 11, no. 23, p. 4041, Dec. 2022, doi: 10.3390/electronics11234041.
- [10] P. S. Nandhini, S. Kuppuswami, S. Malliga, and R. DeviPriya, "A lightweight energy-efficient algorithm for mitigation and isolation of internal rank attackers in RPL based Internet of Things," *Computer Networks*, vol. 218, p. 109391, Dec. 2022, doi: 10.1016/j.comnet.2022.109391.
- [11] S. Deshmukh-Bhosale and S. S. Sonavane, "A real-time intrusion detection system for wormhole attack in the RPL based internet of things," *Procedia Manufacturing*, vol. 32, pp. 840–847, 2019, doi: 10.1016/j.promfg.2019.02.292.
- [12] A. Le, J. Loo, K. K. Chai, and M. Aiash, "A specification-based IDS for detecting attacks on RPL-based network topology," *Information (Switzerland)*, vol. 7, no. 2, p. 25, May 2016, doi: 10.3390/info7020025.
- [13] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, M. Humayun, A. O. Ibrahim, and A. Abdelmaboud, "A trust-based model for secure routing against RPL attacks in Internet of Things," *Sensors*, vol. 22, no. 18, p. 7052, Sep. 2022, doi: 10.3390/s22187052.
- [14] M. D. Alshehri and F. K. Hussain, "A fuzzy security protocol for trust management in the internet of things (Fuzzy-IoT)," *Computing*, vol. 101, no. 7, pp. 791–818, Dec. 2019, doi: 10.1007/s00607-018-0685-7.
- [15] E. V. Abhinaya and B. Sudhakar, "A secure routing protocol for low power and lossy networks based 6LoWPAN networks to mitigate DIS flooding attacks," *Journal of Ambient Intelligence and Humanized Computing*, Jan. 2021, doi: 10.1007/s12652-020-02804-3.
- [16] S. Tahir, S. T. Bakhsh, and R. A. Alsemmeari, "An intrusion detection system for the prevention of an active sinkhole routing attack in Internet of things," *International Journal of Distributed Sensor Networks*, vol. 15, no. 11, p. 155014771988990, Nov. 2019, doi: 10.1177/1550147719889901.
- [17] T. ul Hassan, M. Asim, T. Baker, J. Hassan, and N. Tariq, "CTrust-RPL: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based Internet of Things applications," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, Jan. 2021, doi: 10.1002/ett.4224.
- [18] M. Surendar and A. Umamakeswari, "InDRoS: An intrusion detection and response system for Internet of Things with 6LoWPAN," in *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, Mar. 2016, pp. 1903–1908. doi: 10.1109/WiSPNET.2016.7566473.
- [19] P. Bhale, S. Dey, S. Biswas, and S. Nandi, "Energy efficient approach to detect sinkhole attack using roving IDS in 6LoWPAN network," in *Communications in Computer and Information Science*, vol. 1139 CCIS, Springer International Publishing, 2020, pp. 187–207. doi: 10.1007/978-3-030-37484-6_11.
- [20] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, May 2015, pp. 606–611. doi: 10.1109/INM.2015.7140344.
- [21] M. M. Iqbal, A. Ahmed, and U. Khadam, "Sinkhole attack in multi-sink paradigm: Detection and performance evaluation in RPL based IoT," in *2020 International Conference on Computing and Information Technology, ICCIT 2020*, Sep. 2020, pp. 1–5. doi: 10.1109/ICCIT-144147971.2020.9213797.
- [22] D. Airehrour, J. A. Gutierrez, and S. K. Ray, "SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things," *Future Generation Computer Systems*, vol. 93, pp. 860–876, Apr. 2019, doi: 10.1016/j.future.2018.03.021.
- [23] S. Sahraoui and N. Henni, "SAMP-RPL: secure and adaptive multipath RPL for enhanced security and reliability in heterogeneous IoT-connected low power and lossy networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 409–429, May 2023, doi: 10.1007/s12652-021-03303-9.
- [24] J. Baker, S. Cossu, S. Ford, C. Schoepp, T. Copeland, and B. Milo, "tidwall/hashmap.c: Hash map implementation in C," *GitHub*. <https://github.com/tidwall/hashmap.c> (accessed Jan. 03, 2024).
- [25] S. Avseyev, R. Steenkamp, M. M. Pedersen, and Yangke., "avsej/hashset.c: hash set C implementation," *GitHub*. <https://github.com/avsej/hashset.c> (accessed Jan. 03, 2024).

APPENDIX

Table 1. Previous literature related to IoT





Ref.	Issue Addressed	Technique used	Detection Scheme	Protection against attack	Validation Scheme/Software	Enhanced QoS parameters	Research Gap
[6]	To detect attacks in Scalable and Dynamic IoT environment	RF (Random Forest) Trust	Trust based	Sinkhole	Cooja-Contiki 3.0	Accuracy, FNR, EC, Delay, Throughput	Can only detect sinkhole attacks
[7]	Protection of RPL protocol based on rating and ranking	SoS-RPL algorithm	Specification Based	Sinkhole	NS-3	DF, FNR, FPR, DR, Maximum Throughput, PDR.	IDS is designed assuming that the ICMPv6 message is safe.
[8]	Authentication of smart infrastructure IoT	ABA-IDS	Anomaly Based	DoS, Flooding, Replay, Man-in-the-Middle	Arduino and Raspberry PI 3	DF, FNR and Accuracy	Behavioral drift can affect the working of ABA-IDS
[9]	An IDS for an RPL-based IoT network	Heartbeat Algorithm	Hybrid IDS (threshold, signature, heart-beat messages)	DoS Attack and types	Contiki-NG	CPU usage, EC, Overheads	Rank attack detection is not included
[10]	Light-weight algorithm for detection and isolation of rank attacks	RAD algorithm	Cryptographic hash algorithm	Rank	Cooja	PDR, Delay, Accuracy 96%	Random Sampling is used to detect errors
[11]	Detection of wormhole attack and attacker	Real-Time IDS	Centralized Approach	Wormhole	Contiki-Cooja	RSSI (Received Signal Strength Indicator)	The analysis is done based on DF only.
[12]	Detection of topology attack in RPL IoT	SAPT	Specification-based IDS	Topology	Contiki 2.6	EC, DF	The proposed IDS cannot be used for performance-type internal threats
[13]	Address rank and blackhole attack in static and mobile RPL-IoT	SM-Trust	Trust based IDS	Rank and blackhole	Instant Contiki 2.7/ Cooja	Improved stability PDR, Throughput, and EC	EC by the model is more. Testbed implementation was not feasible for colliding attacks.
[14]	Detection of Malicious Nodes in IoT	CBFL Approach	Fuzzy-Trust based IDS	ON-OFF	Cooja-Contiki	DF, Number of attacks, Trust Score, Contradictory Attacks	The model is not compared with others
[15]	Prevention of insider and outsider attacks	ALBRD technique	Threshold-based	DIS-Flooding	Cooja	PDR, Delay, EC, Throughput, Control Overheads	Can only detect DIS-Flooding attacks
[16]	Protect integrity, confidentiality, and authentication against intrusions.	PASR	Clustering Technique	Active Sinkhole	NS-2 Simulator	PDR, EC, DF, Routing overheads	Can only predict sinkhole attacks
[17]	Detection of internal attacks	C-Trust	Hierarchical Trust-based	Blackhole	Cooja 2.7 M-2-P traffic	DF, PDR, EC, No. of malicious nodes, residual energy	Mechanism can only work on control-layer
[18]	Resistance against attacks	INDReS	Specification-based	Sinkhole	NS-2	PDR, Throughput, Overheads	Designed only to overcome SVELTE and INTI shortcomings

Table 1. Previous literature related to IoT (continued)





Ref.	Issue Addressed	Technique used	Detection Scheme	Protection against attack	Validation Scheme/Software	Enhanced QoS parameters	Research Gap
[19]	Target Routing Attacks	USER IDS	Profile-based clustering mechanism	Sinkhole	Cooja-Contiki	TPR, TNR, EC, Minimum Memory (RAM/ROM) FNR, FPR, DF	Can only recognize sinkhole attacks
[20]	Detecting Attacks in Mobile and Static RPL-IoT	INTI	Watchdog Timer + Trust-based	Sinkhole	Cooja		Cannot detect other attacks
[21]	To detect multi-sink node attacks	SDMSI	Trust based	Rank and Sinkhole	Cooja-Contiki MAC layer	DF, EC, FPR, TPR, PDR	Inefficient as it takes more overhead for detection of multi-path attacks
[22]	Secure network from forcing attacks	SecTrust-RPL	Trust based	Sybil and Rank	Cooja-Contiki 3.0 XM 1000 motes	DF, PDR	It can detect attacks based on power consumed by selfish node
[23]	Protect the network from internal and external cyber-attacks	SAMP-RPL	Hash function	Multi-path routing	Cooja-Contiki	PDR, DF, EC, loss rate	The idea relies on heterogenous RPL-IoT

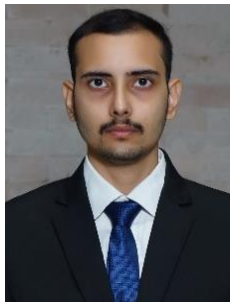
BIOGRAPHIES OF AUTHORS







Akshaya Dhingra     is pursuing Ph.D. degree from the Department of Electronics and Communication Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University Rohtak, Haryana, India. She received an M.Tech. degree in Electronics and Communication Engineering in 2019 and a B.Tech. in Electronics and Communication Engineering in 2017 from Maharshi Dayanand University, Rohtak, Haryana, India. She has published more than 12 articles in reputable journals and conferences. Her areas of interest are communication networks, the Internet of Things, and security. She can be contacted at akshaya.rs.uiet@mdurohtak.ac.in.



Vikas Sindhu     is an associate professor at Maharshi Dayanand University, Rohtak. He received a B.Tech. degree in electronics and communication engineering in 2004, an M.Tech. degree in electronics instruments and control engineering in 2006, and Ph.D. from Maharshi Dayanand University Rohtak, Haryana, India. He has published around 30 research papers in national and international journals and conferences. His areas of interest are electronic devices, cognitive radio, electric vehicles, and the Internet of Things. He can be contacted at vikassindhu.uiet@mdurohtak.ac.in.



Lakshay Dhingra     is pursuing an MBA in business analytics from the Indian Institute of Foreign Trade, New Delhi, Delhi, India. He received a B.Tech. degree in Computer Science and Engineering in 2021 from Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonapat, Haryana, India. He has 11 months of full-time professional experience as a specialist programmer at Infosys Ltd., Bengaluru, India. His areas of interest include computational theory, analytics, marketing, and international trade. He can be contacted at lakshay_ba25@iift.edu.