A Defense Solution to Secure Low-Power and Lossy Networks Against DAO Insider Attacks

Anil Kumar Prajapati[1], Emmanuel S. Pilli*[1], Ramesh Babu Battula[1] and Abhishek Verma[2]

Abstract— **The Low-Power and Lossy Network (LLN) is the most important building block in the Internet of Things (IoT), comprising numerous tiny sensor nodes connected together. The Routing Protocol for Low-Power and Lossy Networks (RPL) is an IPv6-based protocol developed by the Internet Engineering Task Force (IETF) to facilitate routing for LLN devices. The Destination Advertisement Objects (DAOs) are transmitted from RPL nodes in the network toward the root node to construct downward routes. The malicious node exploits the DAO transmission mechanism to replay the DAO with a fixed time interval in the network in order to launch the DAO Insider attack. The DAO Insider attack causes a large number of DAO, which contributes to network congestion; as a result, data packets are delayed, and network performance is degraded. This paper proposes a defense solution that monitors DAO timestamps between child and parent nodes, flagging suspicious nodes that exceed a threshold within a time interval, blacklisting, and discarding DAOs from identified malicious nodes. Moreover, it limits the number of DAO transmitted by a child node within a specified time interval to mitigate the impact of an attack. The experiments show that the DAO insider attack has a negative impact on network performance (packet delivery ratio, average end-to-end delay, and throughput) at various DAO replay intervals. The proposed defense solution restores optimal network performance with a high detection rate.**

Index Terms— **IoT, LLN, IPv6, RPL Security, DAO Insider Attack.**

I. INTRODUCTION

The Internet of Things (IoT) plays a crucial role in connecting and exchanging information with other things (physical objects, sensors, computing devices) and facilitating the transfer of information or data over wireless links without the intervention of a human. The Internet of Things can be conceptualised as a system in which sensors collect data, gateways transfer data, and back-end systems make intelligent decisions. Low power and lossy links, which result in high packet loss and reduced throughput, are the primary constraints for IoT devices. Internet of Things has extensive applications in the fields of agriculture, healthcare, industry, market, transportation, vehicles, and smart homes [1].

The IETF ROLL working group addressed the issue of routing between LLN by standardising the RPL (Routing Protocol for Low Power and Lossy Network) IPv6-based routing protocol for LLN in RFC 6550 [2]. Since the development of RPL for LLN and the emergence of IoT, which connects billions of devices worldwide, RPL has emerged as the routing protocol for IoT. IPv6 is an essential feature for LLN, and RPL significantly overlaps it.

RPL protocol has security features and modes, but they still need to be fully implemented, making the protocol more susceptible to routing attacks [3]. An intruder can utilize the functionality of routing to launch attacks and disrupt the operation of a network. Mayzaud *et al.* [4] proposed a comprehensive classification of RPL attacks based on resource, topology, and traffic. One of the most destructive attacks is the DAO Insider attack, in which an attacker node continuously replays a specified number of DAOs to the parent node of the RPL network in order to degrade performance. The DAO message traveled through intermediate nodes in the network until it reached the root, and the DAO-ACK message was sent back to the child node as a response. An adversary node takes advantage of the situation and floods the network's with DAO, decreasing the packet delivery ratio, throughput, and average power consumption and increasing the average end-to-end delay.

A lightweight defense solution is required to counteract this type of attack, which has motivated us to design this defense solution. The main idea behind our proposed approach is to allow a maximum of N DAO by child node within a specific time interval. When a DAO Insider attack occurs, the malicious node floods the RPL network's DAOs. Our proposed defense solution keeps track of the timestamps of all DAOs sent from the child node to the parent node. If a child node's DAOs count exceeds a predefined threshold within a predefined time interval, the child node is classified as suspicious. When a node is classified as suspicious multiple times, it is moved to the blacklist table and flagged as malicious. When the malicious node transmits DAOs again and is found in the blacklist table, the received DAO is discarded. This step reduces resource consumption and indicates a quick response to the attack.

The primary contribution of this work is as follows:

- Defense against the DAO Insider attack that provides quick mitigation and requires low resource computation.
- The effectiveness of the proposed solution is evaluated in terms of packet delivery ratio, average end-to-end delay, throughput, and the number of DAO received.

The structure of this paper is as follows: Section II highlights the overview of RPL (II-A) and its vulnerabilities (II-B), further its also explains the DAO Insider attack (II-C) and related work (II-D). Section III explains the proposed defense solution and its working with the help of algorithm. Section IV presents the simulation setup (IV-A), performance evaluation of proposed defense solution (IV-B). Section V concludes our work.

[1]Anil Kumar Prajapati, Emmanuel S. Pilli* (Corresponding Author), and Ramesh Babu Battula is with Department of Computer Science and Engineering, Malaviya National Institute of Technology Jaipur 302017, India. 2018rcp9156, espilli.cse, rbbattula.cse}@mnit.ac.in

[2]Abhishek Verma with the Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow 226025, India. abhishekverma@ieee.org

II. BACKGROUND AND RELATED WORK

This section provides an overview of RPL and discusses various security vulnerabilities. The DAO Insider attack is explained in detail and a related review of work on RPL attacks is provided.

A. Overview of RPL

RPL is a distance routing protocol for resource-constrained nodes which have limited resources (computation, memory) and high packet loss. RPL periodically makes DODAG (Destination Oriented Directed Acyclic Graph) [5] where all the traffic moves towards root nodes called DAG Root. RPL can have multiple DODAG's and each DODAG's identified with a unique ID known as DODAG ID. RPL supports three types of traffic: Point to Point (P2P), Point to Multi-point (P2MP), and Multi-point to Point (MP2P). RPL has a feature like auto-configuration (dynamic discovery of nodes), self-healing (adaption of change in topology when failures occur), loop avoidance and detection (DAG mechanism to avoid loop and repair mechanism), independence and transparency (operate with constrained devices and network irrespective of Link layer), multiple edge routers (creates multiple DODAG's when required). The RPL topology construction procedure employs four distinct types of control messages: DIS (DODAG Information Solicitation), DIO (DODAG Information Object), DAO (DODAG Advertisement Object), and DAO-ACK (DAO Acknowledgment). First DIO message is sent to other nodes to advertise information about existing DODAG; a DIS message is used to know about existing DODAG, a DAO message send when a node wants to join the DODAG, and DAO-ACK send as an acknowledgment for accepting or rejecting when a node joining the DODAG.

B. Security Vulnerabilities in RPL

Since RPL is a popular protocol for the Internet of Things, it is vulnerable to a wide range of security attacks. Wallgren *et al.* [3] discussed a few routing attacks and countermeasures for attacks on RPL protocol. Later on, Pongle *et al.* [6] introduced some RPL Specific attack Version Numbers, Rank, DIS attacks, and Local Repair in the survey. Mayzaud *et al.* [4] introduced a detailed taxonomy of RPL attacks in RPL-based Internet of Things which focused on three main categories (1) Covers attacks targeting the exhaustion of network resources (energy, memory, and power), (2) Attacks targeting to RPL network topology, (3) attack against network traffic such as eavesdropping and misappropriation. The classification is according to the attackers goals and means considering the specific properties of RPL network. Verma *et al.* [7] introduced the Copycat attack, which is a replay based attack on RPL.

C. The DAO Insider Attack

The routing among the LLN devices is constructed with RPL protocol with the help of ICMPv6-based control messages. The control messages used for the establishment and maintenance of routing paths and RPL uses four distinct control message types: DIS for Discover DODAG and find the potential parent, DIO for disseminating network information by the root node, DAO for register node in topology and making the downward route from root to leaf node and last DAO-ACK for the response for DAO message as acknowledgment for successful route registration by the root node. As we discussed, the main role of DAO messages is to construct the downward path to facilitate the bi-directional communication between the root and respective nodes present in the topology. The specification available in RFC 6550 [2] does not have any standard rules to handle the transmission of DAO control messages. It depends on the available implementation of RPL in different platforms such as Contiki-NG, OMNeT++, RIOTS, and OpenWSN. One of the implementations in [8] transmits the DAO messages regularly based on the specified time interval whereas the ContikiRPL [9] specified the trickle timer mechanism to transmit the DAO messages. The DAO messages are transmitted in unicast fashion by the child node in the following situations:

- When a parent node transmits the Unicast-DIO messages to the child node.
- When the repair mechanism is run and the topology is reconstructed to change its preferred parent node.
- When a node receives error messages due to routing error.

The transmission of DAO messages is carried away through the per hope basis means when a child node transmits the DAO message to register itself in the network, the DAO message is traveled through all the hops as intermediate parent nodes between the node itself and root node. The whole process leads to the transmission of several DAO control messages toward the root node. The malicious node exploits this mechanism to transmit excessive DAO control messages to the root node, thereby increasing the control message overhead in terms of DAO messages. This action effects the performance significantly in terms of reducing the Packet Delivery Ratio (PDR), Average Power Consumption (APC), Throughput (TH) and increasing the Average End to End Delay (AE2ED).

There are several ways to launch a DAO attack, including the transmission of eavesdropped DAO messages captured from a genuine node to the root node by an outsider node, known as a DAO outsider attack, or the replaying of DAO control messages to the root node, known as a DAO insider

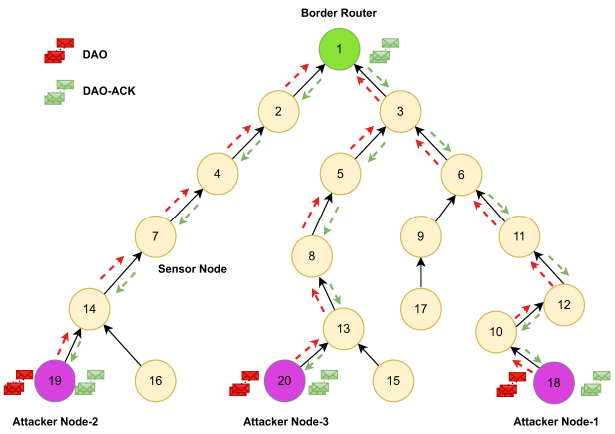

Fig. 1. The DAO Insider Attack (Attacker Node-1,2,3 replay the DAO, and the path is flooded with DAO messages).

attack. The straightforward process of a DAO attack is depicted in Fig. 1. The Attacker Nodes-1, 2, 3 (Node ID-17, 18, 19) transmit N DAO control messages to the root node and, in response, receive DAO-ACK control messages. However, this type of attack is easily mitigated by cryptographic algorithms or link-layer encryption, but they require additional overhead, so lightweight solutions must be designed.

D. Related Work

Perazzo *et al.* [10] proposed a novel attack known as DIO Suppression attack which suppress the DIO message of other node in RPL-based IoT network leads to degradation of route or network partitions. Ghaleb *et al.* [11] addressed the novel attack known as DIO Insider attack in RPL-based IoT. They also proposed a mitigation mechanism SecRPL by applying the restriction on number of DAO forwarded by parent node. Patel *et al*. [12] designed a watchdog-based blackhole attack IDS that filters out suspicious nodes before monitoring behaviour for attack detection. Verma *et al*. [13] addressed a DIS flooding attack in RPL, which leads to high control packet overhead and increases energy consumption. Verma *et al.* [14] proposed a defense mechanism also proposed to mitigate the DIS flooding attack based on the DIS safety threshold, which limits the transmission of DIS messages. Wadhaj *et al.* [15] proposed two different defense mechanisms SecRPL1 and SecRPL2, against the DIO insider attack. SecRPL1 is based on the concept of restricting the DAO message per child node, whereas SecRPL2 restrict the entire number of DAO message transmitted by any node. Guo *et al.* [16] proposed a adaptive adjusted threshold based lightweight defense scheme against the DIS flooding attack in RPL. Baghani *et al.* [17] performed a DAO Induction attack analysis in the RPL network and a defence mechanism that monitors the DTSN update to detect attacks. Verma *et al.* [18] introduced the mitigation scheme for DAO Insider attacks based on the DAO threshold which put the restriction on the number of DAO and blacklist table used to store the blacklisted node. Sheibani *et al.* [19] proposed a lightweight IDS against Dropped DAO attack based on the DAO packet forwarding behavior of the parent node. There are only a few approaches that defend the DAO insider attack that motivates to design the lightweight solution for RPL-based IoT. We proposed a DAO time interval based defense mechanism that used blacklisting to mitigate the DAO insider attack.

III. Proposed Defense Solution

This section describes the operation of the proposed defense against the DAO Insider attack in RPL-based IoT networks. To develop an effective defense solution, we analyzed the working of the transmission of DAO control messages in the RPL network under the normal scenario with the help of a number of experiments. It is evident that the DAO transmission occurs in three cases discussed in Section II-C under the normal scenario. The RPL incorporates the DAO transmission mechanism to control the transmission of DAO through adding the delay. The delay is added in terms of time interval between the two consecutive DAO control messages transmitted by a node at the time of route establishment and it is increased after the successful route registration. The malicious node exploits this mechanism and transmits a vast number of DAO messages to degrade the performance of a network.

Here, we propose a defense solution which imposes a restriction on number of DAO is transmitted by child nodes on specified time interval. The two threshold *DAORecvTh* and *SuspendTh* are used in defense solution. The *DAORecvTh* restricts the number of DAO transmissions whereas *SuspendTh* is used to decide the node is malicious. The *TimeWindow* is used for early detection of aggressive attacker. The main idea behind our approach is to allow the maximum N number of DAO transmitted within a time interval by a child node. The defense solution is deployed to all of the nodes in the network and executed in a decentralised fashion. The algorithm 1 presents the pseudo-code implementation of the proposed defense scheme.

Algorithm 1 Defense Solution for DAO Insider Attack

```
1:  NodeTable                                          ▷ Table to store node information
2:  BlackListTable                                     ▷ Table to store blacklist node information
3:  DAORecvTh ← Threshold value for receive DAO
4:  TimeWindow ← Time window in which DAO message observed
5:  SuspendTh ← Threshold value to move node into blacklist
6:  procedure InitializeNode(node)
7:      NodeTable[node] ← {}
8:      BlackListTable[node] ← {}
9:  end procedure
10: procedure ProcessDAO(dao_message)
11:     sender ← dao_message.sender
12:     if sender ∈ BlackListTable then
13:         return                                     ▷ Sender is already blacklisted
14:     end if
15:     for node ∈ NodeTable do
16:         if node.sender = sender & node.prefix = global_id then
17:             if node.dao_count < DAORecvTh then
18:                 node.dao_time.append(dao_message.time)
19:                 node.dao_count ← node.dao_count + 1
20:             else if node.dao_count = DAORecvTh then
21:                 time_diff ← (T_n − T_1)
22:                 if time_diff < TimeW then
23:                     node.susp ← node.susp + 1
24:                     if node.susp = SuspendTh then
25:                         BlackListTable.add(node)
26:                     else
27:                         reset(node.dao_time, node.dao)
28:                     end if
29:                 else
30:                     forward_dao(dao_message)
31:                     reset(node.dao_time, node.dao)
32:                 end if
33:             end if
34:             return
35:         end if
36:     end for
37:     NodeTable.add(new_entry(sender))
38: end procedure
```

The approach has two important procedures *InitializeNode* and *ProcessDAO* that are also executed upon receiving DAO message. The *NodeTable* and *BlackListTable* are the two structures used in the defense scheme to maintain information about the node of DAO senders in line 1–2. The *NodeTable* stores the sender address of a node, its global address, timestamp of different DAOs within a specified time window, DAO count, and suspend count. The *BlackListTable* stores the address of the blacklisted node and status value. The various thresholds used in the defense solution are declared and initialized with default values in lines 3–5. The defense mechanism is initiated when a node receives a DAO control message from its child node. First, the *InitializeNode* procedure (lines 6–8) runs which initializes the empty *NodeTable* and *BlackListTable* for node. The *ProcessDAO* procedure (line 10–38) which is defined in Contiki-RPL implementation

in the "dao_input_storing" is modified to implement the defense solution.

The address of the DAO sender (line 11) node is stored in sender variable and checked into the *BlackListTable* (line 12–14). Suppose the address is present in the *BlackListTable* for the node, indicating that the DAO sender is a malicious node already detected. In such a situation, the DAO message is discarded without further processing, thereby preserving the node's energy. It is also a sign of quick defense against the DAO attack. If the node does not present in the *BlackListTable* then two cases may be possible for a node. In the first case, if a child node is sending the DAO the first time to the parent node, then a new entry (line 37) is created in the *NodeTable*. In the second case (lines 16–19), if a node is already present in the *NodeTable*, its prefix value is compared with the global IP address, and the DAO count is less than the specified *DAORecvTh*. If a match is found, the node is the originator of DAO, so the corresponding DAO count is incremented, and the time stamp of DAO messages is stored. Whenever the DAO count of the sender address reaches to *DAORecvTh* (line 20) the time difference is calculated between the timestamp of the last DAO message and the first DAO message (line 21). If the value is less than the time window (line 22) it means the DAO sender node has transmitted more DAO messages so we increment the suspend count by 1 (line 23). When suspend count (line 24) is equal to *SuspendTh*, then DAO sender is moved into *BlackListTable* (line 25) otherwise resetting the value of DAO count and time value (line 27) in *NodeTable*. When compared to the complex computation involved in encryption and hashing-based cryptographic solutions, the proposed solution consumes less resources.

IV. Performance Analysis

This section describes the simulation setup and performance metrics. Also, a thorough analysis of the effects of these performance metrics on the DAO attack and proposed defense solution is discussed.

A. Simulation Setup

The effect of the DAO Insider attack on performance metrics and performance achieved by our proposed defense solution against the attack using a number of experiments on the Contiki-NG [9] operating system, which has extensive support for IoT protocol stack and associated protocols such as RPL, 6LoWPAN, IPv6, and CoAP. The Cooja simulator is a cross-layer emulator in Contiki-NG OS for running experiments with low-power sensor devices. Table I has all the parameters used to carry out the simulations. This paper considers three scenarios: RPL_normal (no attack), RPL_attack (with DAO Insider attack), and RPL_defense (with DAO Insider attack and proposed defense solution). The RPL_attack and RPL_defense are analyzed in four different DAO replay intervals of 250, 500, 1000, and 2000 ms for ten experiments with random seeds, and simulation logs for each experiment are collected. The simulation results are analyzed in terms of packet delivery ratio, average end-to-end delay, number of received DAOs, and throughput. The parameters are analyzed on mean values while considering the computed errors 95% confidence interval.

TABLE I
Simulation Parameters and its Values

Simulation Parameter	Value
Simulation Area	150 m × 150 m
Simulation Time	1800 seconds
Number of Nodes	21 (1 Server, 20 Client)
Simulation Mote	Zolertia Z1
DAO Replay Interval	250, 500, 1000, 2000 ms
Mode of Operation	Storing Mode
Packet Sending Interval	60 seconds
Objective Function	MRHOF
Radio Medium	UDGM

B. Performance Evaluation

The performance evaluation of proposed defense solution on the following performance metrics mentioned below:

- Packet Delivery Ratio (PDR): It is expressed as the proportion of packets that have reached the root node out of the total number of packets sent from all nodes, usually measured in percentage (%).
- Average End-to-End Delay (AE2ED): The average time in seconds it takes for all data packets to travel from the initial node to the destination node.
- Number of Received DAO's: This performance metric represents the number of received DAO's by parent node from all child nodes in DODAG.
- Throughput (TH): The amount of data packet successfully delivered towards the root node from other nodes within of time interval, usually measured in bit per second (bps).
- Implementation Overhead: The additional ROM and RAM requirement to implement the proposed solution on the low-power sensor nodes, usually measured in Kilobytes (KB).

The above described performance metrics are measured under the RPL_normal, RPL_attack, and RPL_defense scenario. The influences of the performance metrics on the RPL-based IoT network are discussed as follows:

1) Influence on the Packet Delivery Ratio: The PDR under RPL_normal, RPL_attack, and RPL_defense scenario is depicted in Fig. 2 and it is clearly visible that performance is degraded in RPL_attack scenario. In the RPL_attack scenario, an attacker chooses the DAO replay interval of 250, 500, 1000, and 2000 ms respectively to transmit the huge number of DAO's towards the parent node in the RPL network. This simulation leads to the higher control message overhead as compared to RPL_normal and all the parents that receive the DAO's also require to process the DAO's, and need to reply with an equal amount of DAO-ACK's. The important observation here is, the processing of overhead introduced due to DAO's leads to the huge amount of data packets in the network due to this the PDR is lowered in RPL_attack scenario. Furthermore, it can be seen in Fig. 2 that an attacker node that sends DAO's in the lower DAO replay interval (250ms) in the RPL_attack scenario has a large influence on PDR and is classified as an aggressive attacker. However, the proposed RPL_defense solution against the DAO Insider attacker fully justifies its effectiveness and reduces the influences of the RPL_attack scenario to improve the PDR as network performance.

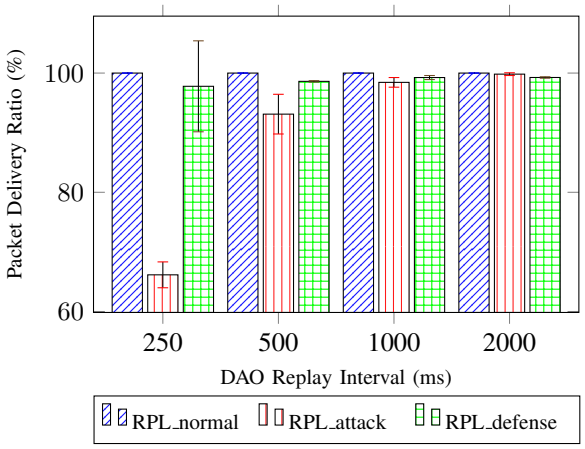

Fig. 2. Packet Delivery Ratio for scenario of RPL_normal, RPL_attack and RPL_defense

.

2) Influence on Average End to End Delay: The AE2ED under RPL_normal, RPL_attack, and RPL_defense scenario is depicted in Fig. 3 and it is evident that performance is degraded in RPL_attack scenario. Here is an important observation about increasing in the AE2ED in RPL_attack due to congestion at the parent node due to receive the huge amount of DAO's received from the child nodes. The parent nodes engaged themselves to process the DAO's which require a huge amount of time and the acknowledgment of data packets gets delayed, hence AE2ED delay is increased. As we observed in the case of PDR, the aggressive attacker has major influence on AE2ED as compared to a non-aggressive attacker in the RPL_attack scenario. However, the proposed defense solution RPL_defense reduces the influences of RPL_attack scenario to decrease the AE2ED, and improvement in the network performance is clearly visible from Fig. 3 because of reducing the DAO's received from child attacker nodes.

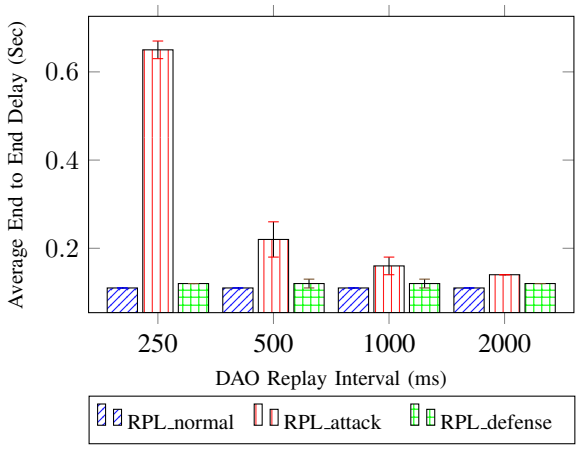

Fig. 3. Average End to End Delay for scenario of RPL_normal, RPL_attack and RPL_defense

.

3) Influence on Number of Received DAO: The number of received DAO's by parent nodes under RPL_normal, RPL_attack, and RPL_defense scenario is depicted in Fig. 4 and it is evident that a huge number of DAO's received in RPL_attack scenario. As we seen in Fig. 4, an aggressive attacker node transmit a huge number of DAO's to their

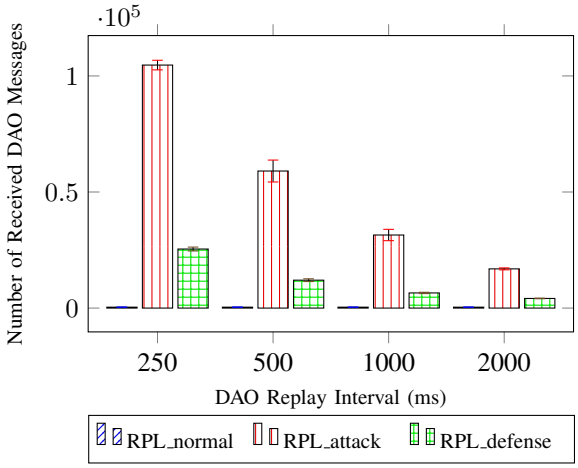

Fig. 4. Number of Received DAO for scenario of RPL_normal, RPL_attack and RPL_defense

.

parent node, and the DAO's further transmitted till they reached to the root node. This situation raises the control packet overhead within the RPL network and influences the network performance parameter like packet delivery ratio, throughput and end-to-end delay. However, the proposed defense solution RPL_defense here significantly reduces the number of received DAO's by parent node with the help of *BlackListTable*. It discards DAO's received from a blacklisted child node to improve the performance of the RPL network.

4) Influence on Throughput: The Throughput measured under RPL_normal, RPL_attack, and RPL_defense scenario is depicted in Fig. 5 and it makes a huge influence in the case of RPL_attack scenario. As we mentioned previously, an aggressive attacker node transmits more number of DAO's that add congestion in the RPL network which delayed the data packet delivery at the root node. Due to this, the throughput of the network is decreased mostly in case of aggressive DAO insider attackers. The RPL_defense against DAO insider attack reduces the influence of RPL_attack by rejecting the DAO's within a time window and improving the RPL network throughput.

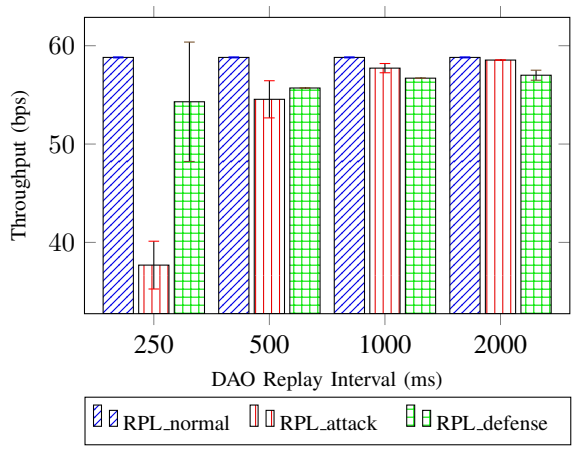

Fig. 5. Throughput for scenario of RPL_normal, RPL_attack and RPL_defense

.

5) Influence on Implementation Overhead: RAM and ROM overhead is measured for Standard Z1 motes, Normal

Z1 motes under RPL_normal, and Defense Z1 motes under RPL_defense. Standard Z1 motes have 92KB ROM and 8KB RAM. Under the RPL_normal scenario in standard ContikiRPL implementation, Normal Z1 motes need 57.5KB of ROM and 7KB of RAM. The proposed defense solution uses Defense Z1 motes and requires 60.2KB of ROM and 7.5KB of RAM, equivalent to Normal Z1 motes in RPL_normal. This demonstrates that the proposed RPL_defense solution imposes no overhead. RPL_defense's low memory requirements make it a lightweight defense solution.

V. Conclusion

In this paper, the DAO Insider attack is observed where a malicious node exploits the DAO transmission mechanism to replay the DAO with a fixed time interval towards the root node. Different DAO replay intervals are used to launch the attack by the malicious node. This attack significantly degrades the RPL's network performance regarding packet delivery ratio, average end-to-end delay, and throughput. Additionally, a defense solution is proposed to mitigate the DAO Insider attack by limiting the number of DAO sent by nodes in the network within a fixed time interval and blacklisting malicious nodes for quick defense. A significant advantage of this solution is the minimal implementation requirement and computational overhead, which makes it lightweight and well-suited for LLN devices. The simulation results show that the proposed approach reduces the impact of an attack while improving network performance with a high detection rate. Future efforts will include evaluating mobile network performance and testbed implementation of the proposed defense solution.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] T. Winter et al., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," Internet Engineering Task Force, Fremont, CA, USA, RFC 6550, March 2012, [Online]. Available:https://www.rfc-editor.org/info/rfc6550.

[3] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-Based Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, p. 794326, 2013.

[4] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.

[5] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[6] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *2015 International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–6.

[7] A. Verma and V. Ranga, "Addressing Copycat Attacks in IPv6-Based Low Power and Lossy Networks," in *Science and Information Conference*. Springer, 2020, pp. 415–426.

[8] U. Herberg and T. Clausen, "A Comparative Performance Study of the Routing Protocols LOAD and RPL with Bi-Directional Traffic in Low-power and Lossy Networks (LLN)," in *8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, 2011, pp. 73–80.

[9] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, p. 101089, 2022.

[10] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO Suppression Attack Against Routing in the Internet of Things," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2524–2527, 2017.

[11] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, and L. Mackenzie, "Addressing the DAO Insider Attack in RPL's Internet of Things Networks," *IEEE Communications Letters*, vol. 23, no. 1, pp. 68–71, 2018.

[12] H. B. Patel and D. C. Jinwala, "Blackhole Detection in 6LoWPAN Based Internet of Things: An Anomaly Based Approach," in *IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 947–954.

[13] A. Verma and V. Ranga, "Addressing Flooding Attacks in IPv6-based Low Power and Lossy Networks," in *IEEE Region 10 Conference (TENCON)*, Kochi, India, 2019, pp. 552–557.

[14] ——, "Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3802, 2020.

[15] I. Wadhaj, B. Ghaleb, C. Thomson, A. Al-Dubai, and W. J. Buchanan, "Mitigation Mechanisms Against the DAO Attack on the Routing Protocol for Low Power and Lossy Networks (RPL)," *IEEE Access*, vol. 8, pp. 43 665–43 675, 2020.

[16] G. Guo, "A Lightweight Countermeasure to DIS Attack in RPL Routing Protocol," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021, pp. 0753–0758.

[17] A. S. Baghani, S. Rahimpour, and M. Khabbazian, "The DAO Induction Attack: Analysis and Countermeasure," *IEEE Internet of Things Journal*, 2021.

[18] S. K. Verma, A. Verma, and A. C. Pandey, "Addressing DAO Insider Attacks in IPv6-Based Low-Power and Lossy Networks," in *IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2022, pp. 1–6.

[19] M. Sheibani, B. Barekatein, and E. Arvan, "A lightweight distributed detection algorithm for DDAO attack on RPL routing protocol in Internet of Things," *Pervasive and Mobile Computing*, p. 101525, 2022.