A Lightweight Security Solution for Mitigation of Hatchetman Attack in RPL-based 6LoWPAN

Girish Sharma^{*,a,b}, Jyoti Grover^a, *Senior Member, IEEE*, Abhishek Verma^c, *Member, IEEE,*

^{a} Malaviya National Institute of Technology Jaipur, JLN Marg, Jaipur, Rajasthan, India 302017
^{b} Manipal University Jaipur, Dehmi Kalan, Jaipur, Rajasthan, India 303007
^{c} Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Uttar Pradesh, India

Abstract—In recent times, the Internet of Things (IoT) has a significant rise in industries, and we live in the era of Industry 4.0, where each device is connected to the Internet from small to big. These devices are Artificial Intelligence (AI) enabled and are capable of perspective analytics. By 2023, it's anticipated that over 14 billion smart devices will be available on the Internet. These applications operate in a wireless environment where memory, power, and other resource limitations apply to the nodes. In addition, the conventional routing method is ineffective in networks with limited resource devices, lossy links, and slow data rates. Routing Protocol for Low Power and Lossy Networks (RPL), a new routing protocol for such networks, was proposed by the IETF's ROLL group. RPL operates in two modes: Storing and Non-Storing. In Storing mode, each node have the information to reach to other node. In Non-Storing mode, the routing information lies with the root node only. The attacker may exploit the Non-Storing feature of the RPL. When the root node transmits User Datagram Protocol (UDP) or control message packet to the child nodes, the routing information is stored in the extended header of the IPv6 packet. The attacker may modify the address from the source routing header which leads to Denial of Service (DoS) attack. This attack is RPL specific which is known as Hatchetman attack. This paper shows significant degradation in terms of network performance when an attacker exploits this feature. We also propose a lightweight mitigation of Hatchetman attack using game theoretic approach to detect the Hatchetman attack in IoT.[1]

Index Terms—Industry 4.0, IoT, LLN, Constrained Devices, RPL, Hatchetman, Game Theory.

I. INTRODUCTION

Internet of Things is one of the key technology for Industry 4.0. IoT and Industrial IoT (IIoT) enable many applications in the consumer world [1]. It is expected that around 27 billion IoT devices will be connected to the Internet [2] by the year 2025. IoT is connecting different devices ranging from Radio Frequency Identification (RFID), smart grids, Wireless Sensor Network (WSN) to the Internet. The different devices connected to the Internet makes IoT networks heterogeneous [3] and it uses different protocol standards as compared to the traditional protocol stack. One of the protocols that an IoT protocol stack uses is 6LoWPAN [4] which works as an adaptation layer. This layer supports the constrained nodes and the networks with low data rates. 6LoWPAN was designed to optimize the IPv6 packets over the lossy and low data rate constrained networks such as IEEE 802.15.4. The adaptation layer handles header compression, fragmentation and mesh addressing i.e. mutiple hops forwarding of IPv6 packets.The IETF Routing Over Low-Power and Lossy Networks (ROLL) working group proposed a new protocol commonly known as Routing Protocol for Low Power and Lossy Networks (RPL) for the constrained networks. RPL, which is based on distance vector routing protocol optimizes the resource requirements along the routing path.

Applications for the Internet of Things are growing every day, and RPL was created specifically for them. So there is a need to address different security aspects of RPL. Routing attacks on RPL have gained lot of attention by researchers since some are inherited from WSN and some are RPL specific. One such attack is Hatchetman attack which has drastic impact in terms of throughput and Packet Delivery Ratio (PDR). This attack was first addressed by Cong Pu *et al.* in their article [5]. In their paper, the performance evaluation is carried out in terms of PDR, Energy consumption by changing the source routing header. Latency in packet delivery, energy usage, and throughput are all impacted by a hatchetman attack. During packet forwarding, this attack dynamically changes the source routing header, leading to a situation where legitimate nodes are unable to forward the packet. Consequently, this gives rise to a denial of service attack. RPL supports non-storing mode for downward packet delivery which is useful for LLNs because of memory constraint [6]. The root node stores the IPv6 address of each hop between the source and destination. The attacker may take advantage of this information and fills the Source Route Header (SRH) with the fake IPv6 address. This results in the propagation of ICMPv6 error messages to the SRH generator. In this paper we analyze the impact of hatchetman attack on the network and also proposes the mitigation of the attack for RPL based LLNs.

A. *Contributions*

The contributions of our paper are listed below:

1) Implementation and analysis of the hatchetman attack in RPL-based static and mobile IoT networks.

G. Sharma, J. Grover are with the Department of Computer Science & Engineering, Malaviya National Insitiute of Technology Jaipur, Rajasthan, India, 302017. G. Sharma is also with Manipal University, Dehmi Kalan, Jaipur, Rajasthan 303007. A. Verma is with the Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow, Uttar Pradesh, India, e-mails: jgrover.cse@mnit.ac.in, 2020rcp9012@mnit.ac.in, abhiverma@iiitdmj.ac.in,
*Corresponding Author
[1]https://ieeexplore.ieee.org/abstract/document/10469481/

2) Lightweight mitigation of hatchetman attack using game theoretic approach.

B. Organization

The remaining paper is organized as follows: Section II overview of RPL protocol. This section also outlines the functioning of the hatchetman attack. The implementation of hatchetman attack is discussed in section III. In the section IV, we discuss the detection approach and mathematical formulation of the performance parameters. We also propose an algorithm for the detection of the attack in the section V. At last, Section VI concludes some useful insights and presents future study directions.

II. BACKGROUND

A. Overview of RPL Protocol

Using the distance vector routing concept, RPL sets up networks as Directed Acyclic Graphs (DAGs) and serves Low Power Lossy Networks (LLN). The DAG topology has one or more root nodes, which is generally a border router that connects the outside world to the other nodes in the DAG. RPL works on two modes to implement the resource-constrained nature of the nodes: a) Storing b) Non-Storing. a) In storing mode, each node has a complete path to route a packet to the sink node or the other nodes. b) In non-storing mode, only the border router has complete information for the nodes, i.e., the downward path. In this mode, the nodes only have a parents' list so they can forward the packets to the border router.

RPL, based on the destination-oriented source routing protocol, implements five control messages: a) *DODAG Information Object (DIO)*: This is used to maintain and configure the upward routes. The nodes listen to the DIO messages to configure as per the changes in the topology and determines the parent and the route to the sink using the information provided in DIO message [7]. b) *DODAG Information Solicitation (DIS)*: DIS is used to probe the network when the node wants to join the DODAG. If the DIS was sent as a unicast, the receiving node would transmit the DIO message containing the DODAG configuration. In the case of multicast, the receiving nodes would reset the trickle timer and broadcast the DIO messages. c,d) *Destination Advertisement Object (DAO) and Acknowledgment (DAO- ACK) Messages*: The nodes maintain the downward routes by sending the parent information toward the border router using the DAO message. The root maintains this information to send the datagram to the nodes by determining the source route. The receiver responds to the sender by sending DAO-ACK depending on the flag field of the DAO message [8]. e) Consistency Check (CC) messages: These are used for coordinating time and security measurements between any two nodes. These messages are only available in RPL' secure mode.

RPL's objective function (OF) is used to determine the node's rank. Various OFs in RPL include ETX Objective function (ETXOF), Minimum Rank with Hysteresis Objective Function (MRHOF), and Objective Function Zero (OF0). RFC 6551 defines some of the routing metrics such as Expected Transmission Count (ETX), Hop Count, Latency, Link Quality

Level, Node Energy and Throughput. The rank of a node determines how close it is to the root node. Rank helps the DODAG to remove count-to-infinity and loops in the network. The node can decrease its rank if it finds a lower-cost route to the root [9].

In summary, RPL is a new proposed standard that facilitates applications based on the IPv6 protocol. With the advent of IoT as a new industry standard, RPL has become one of the most important network layer protocols in sensor networks due to the resource-constrained nature of nodes.

B. Hatchetman Attack in RPL

This section discusses implementation of Hatchetman attack using the *Contiki* Operating System. RPL protocol is based on distance vector source routing mechanism. The attacker can exploit the extended header of IPv6. IPv6 packet has fixed base header of size 40 bytes. The extension headers contains different options depending on the requirement. The source fills all these extension header information [10]. One such option used in case of source routing is **Routing Extension Header (REH)**. The source node stores the complete routing path. The intermediate nodes transmit the packet to the subsequent destination, and as the packet follows its path, it ultimately reaches the intended destination.

Fig. 1: Normal Scenario: Packet Delivers Successfully (Downward PDR)

Figure 1 shows a simple line topology. The source is connected to all the nodes through the intermediate nodes. The **strict source routing** mechanism sends the packet from the source to destination when there is no attack. The **Segments Left** field of the SRH shows that number of intermediate nodes to be visited to reach to the destination. Hatchetman attack scenario is shown in the Fig. 2 But if an attacker changes the IPv6 address in the source route path with an unreachable IPv6 address, the packet will not reach the destination, and this will lead to the Denial of Service Attack (DoS).

This attack reduces the downward PDR and also increases the ICMPv6 error messages as shown in section III. The node's energy consumption rises as it repeatedly tries to send the packet on to the next hop.

III. HATCHETMAN ATTACK IMPLEMENTATION

This section shows the impact of the hatchetman attack on static and mobile RPL based IoT. In contrast to the vast

Fig. 2: **Attack Scenario: Packets do not reach node n_4 onwards**

majority of research, which focus on static networks, this paper evaluates performance metrics for mobile IoT.

A. Experimental Setup

We use cross level COOJA simulator for the performance evaluation. COOJA simulator which was developed specifically for IoT supports RFC 6550, uses Contiki as underlying operating system for sensor nodes. Contiki 3.0 optimizes the IoT standards proposed in RFC 6550. Wireless technology for the media access control and physical layers in constrained networks is also supported by Contiki as it implements IEEE 802.15.4. At network layer Contiki implements RPL as the routing protocol which is based on distance vector routing protocol.

The Z1 platform, a low-power microcontroller, is used in the experiment since its transceiver can communicate at 2.4GHz. Z1 is compliant with IEEE 802.15.4 and can handle low data rates.

We use COOJA simulator for the normal and attack scenarios simulation. It has a built-in hardware simulator called MSPsim that replicates the same binary code as sensor devices. This makes the models more realistic. We use the Z1 platform, which is a node for 6LoWPAN. In this simulation, the Unit Disc Graph Medium (UGDM) radio model is used. The test is run on a grid that was 200m x 200m and had between 10 and 50 nodes. We also use Random Waypoint Mobility Model to simulate the mobile networks where speeds of nodes varies from $1-2\ m/sec$.

TABLE I: **Experimental simulation parameters**

Parameters	Value
Grid Size	$200m\ X\ 200m$
Sensor Nodes	$10\ to\ 30$
Gateway Nodes	1
Radio Medium	Unit Disk Graph Medium
Physical and Medium Access Control Layer	IEEE 802.15.4
Transmission Range	$50\ m$
Interference Range	$100\ m$
Number of Attacker nodes	1
Speed of Node	$1-2\ m/sec$
Data Packet Size	$30\ bytes$
Data packet sending interval	$60\ secs$

B. Hatchetman Attack Algorithm

To implement the attack we make changes in *rpl-ext-header.c* of Contiki operating system. The attacker node modifies the IPv6 address in the Source Routing Header (SRH), thereby preventing the next hop from successfully forwarding the packet to its intended destination. The Next Header bits of SRH provides the type of the next header as in IPv6 Next Header field. The Routing Type value is 3 in SRH. The Segments Left field is decremented when the packet moves from one hop to next hop. The 4 bits CmprI fields stores number of prefix octets for the segments except the last one and 4 bits CmprE bits specifies the prefix octets for the last segment. The address field stores the IPv6 address of all the hops through which the packet will move towards the destination.

Algorithm 1 shows the implementation of hatchetman attack by changing *rpl-ext-header.c* file of Contiki. This algorithm processes SRH as per the rfc 6554. The algorithm works as follows:

i The attacker node fetches the SRH and finds the index of next to next destination by checking the number of Segments Left.

ii If the IPv6 address is available, the attacker modifies using the random() fucntion.

iii When the packet reaches to the next hop, it can not forward the packet because of random address.

iv The node tries to send the packet multiple times but fails and this generate ICMP error message to be propagated to SRH generator.

v The attack causes a denial of service attack. The critical aspect of this attack is that the attacker forwards the packet to the next hop, and it is difficult to find the attacker.

vi This algorithm takes constant amount of time ($O(1)$) as there is no loop involved.

This attack process is shown in the Fig. 3. If the attacker changes the SRH, the node next to it will not be able to forward the packet to the next node reducing the downward PDR. Figure 3 also shows that the attacker's position is vital.

Fig. 3: **Hatchetman Attack Process**

Algorithm 1 Hatchetman Attack

    ▷ Processing of source routing Header by the attacker node

    if $Segments\ Left == 0$ **then**

2:    Read Routing header's Next Header field and process the next header

3: **else**

4:    Find out n how many addresses are in the Routing header

    ▷ n = (((Hdr Ext Len * 8) - Pad - (16 - CmprE)) / (16 - CmprI)) + 1 The values are taken from source routing header

5:    **if** $Segments\ Left > n$ **then**

6:      Discard the packet. ICMP Error

7:    **else**

8:      $Segments\ Left = Segments Left - 1$

9:      Compute $index$ of the next address to be visited in the Address[1..n] of SRH

10:    Compute the $index\_Next$ of the next to next address to be visited in the Address[1..n] of SRH

11:    Store a random address at the next to next in the vector Address[1..n]   ▷ This causes that the attacker forwards the packet to next hop. But the next hop can not forward the packet because SRH has been illegitimately modified.

12:    Swap the IPv6 Current Destination Address and next address computed in previous step

13:    Decrement hop limit

14: **end if**

15: **end if**

16: **end procedure**

IV. SYSTEM MODEL

This section shows our lightweight Non-cooperative Game Theory based approach for detecting the Hatchetman attack [11]. Mathematical models of cooperation and conflict between rational, intelligent agents are the primary objective of game theory. It has played a significant role in Wireless Sensor Networks for detecting the attacker nodes [12, 13]. We propose the following matix game for detection of Hatchetman attack. The game consists of strategies, players and their actions.

TABLE II: **Notations for Hatchetman Attack Detection Game**

Description	Notation
Number of finite nodes (Agents)	n
Strategy-1 Node forwards packet	Fp
Strategy-2 Node does not forward packets	Dfp
Payoff function of the player (node) i	$u_i(i,j); u_i : S \rightarrow \mathbb{R}$
Average Power Consumption at node i	ϵ_i
End to End Delay at node i	δ_i
Downward Packet Delivery Ratio node i	μ_i
Number of Packets	$\mathcal{N}$
clock_ticks: number of ticks	c_t
Duration of ticks	c_d
Current consumption	c_c
Number of packets sent by sink to sensor node	Si_n
Number of packets received by sensor nodes	Sn_n

The game can be represented by the matrix displayed in Fig. 4. Rows and columns make up a game's matrix. Player's tactics is depicted by the rows and columns. Player's expected returns on various strategies are represented by rows and columns in the matrix. To understand the concept of matrix games let us assume that there are 2 players. In non-storing mode of RPL, each node is either forwarder or the recipient of the packet. So we can model the two player matrix game as in Fig. 4.

Fig. 4: **Two Players Game for Hatchetman Attack Detection**

  i The Fig. 4 shows the payoff of each player (node) depending on whether it forwards the packet or not.

 ii This matrix is maintained for each player where player i represents the node itself and player j is the parent of node i.

iii If the node i is not able to forward the packet due to modifications by the parent j; this signifies the payoff $(0, -1)$.

iv If the parent is not able to forward the packet; this situation already leads to attack for the node previous to the parent.

Fig. 4 illustrates the pure strategy Nash Equilibrium (PSNE) in which both players choose strategy (Dfp, Dfp), resulting in player i being the attacker node.

A. Mathematical Formulation

To solve the non-cooperating games as depicted in Fig. 4, we use the dominating strategies to eliminate the rows and columns to get the solution. In our proposed approach the attacker node payoff will be the dominant player.

Attacker Prediction: Here we find the attacker mathematically by using dominant strategy. A strategy $s'_i \in S_i$ of a node is dominated by a strategy $s_i \in S_i$ if the payoff function of the strategy exceeds that of the node 1.

$$u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}), \forall s_{-i} \in S_{-i} \qquad (1)$$

In the Fig. 4 we can see that Fp is dominated by Dfp for both the nodes which reduces the number of possibilities and lead to outcome of the game. In our proposed approach, we can find the attacker node by eliminating the rows non-dominating rows and columns.

Delay: Average time required for all packets to reach the destination application layer from the moment the packet is sent from the source's application. This is calculated as per equation 2

$$Delay = \frac{\sum \delta_i}{\mathcal{N}} \qquad (2)$$

Packet Delivery Ratio: This is the average number

$$\mu_i = \frac{\sum S_i}{Sn_n} \quad (3)$$

Average Power Consumption: Average of sum of power that the sensor node used.

$$\epsilon_i = avg\_current * voltage \; mW \quad (4)$$

where $voltage$ is which the system provides to the components and $avg\_current$ is $(c_t * c_c)/c_d \; mA$

The performance parameters resuts are shown in the section V.

B. Model Explanation

The matrix game between node i and its parent j is depicted in Fig. 6. The table is initialised without any Pure Strategy Nash Equilibria (PSNE) for the matrix game. In the event of an attack, node i will be capable of detecting the intrusion due to its inability to send the packet. The proposed solution for the PSNE is represented by the tuple (Dfp, Fp).

V. SOLUTION APPROACH

The solution to the hatchetman attack is based on the game theory. Game theoretic approach provides a good formulation for network attacks mitigation. The Algo. 2 shows how we can detect and mitigate the hatchetman attack in IoT. The notation $u(i,j)$ represents the payoff matrix, $B_l(k)$ blacklist nodes, ch_i, ch_n represents the initially calculated checksum and current checksum respectively. SN_{ip} is IPv6 control message to add the sensor node.

Algorithm 2 Hatchetman Attack Detection Process

Require: $u(i,j)$, SRH   ▷ Payoff Matrix
Ensure: $B_l[k]$ ▷ Black List Nodes
 1: Initialize the payoff matrix at each sensor node
 2: **for** each downward packet at sensor node **do**
 3:    Read SRH to get the next address
 4:    **if** Address is available **then**
 5:      Forward the packet to next node
 6:    **else if** $Node_i$ generates SN_{ip} & $ch_i ! = ch_n$ **then**
 7:      $u(i,j) = (0,-1)$ ▷ Modify the payoff matrix to get the dominating node
 8:    **end if**
 9: **end for**
10: **for** $i = 0; i < 2; i + +$ **do**
11:    **for** $j = 0; j < 2; j + +$ **do**
12:      **if** $u(i,j) == (0,1)$ **then**
13:        Add $Node_j$ to $B_l[k]$ ▷ Attacker node
14:      **end if**
15:    **end for**
16: **end for**

The Algo. 2 detects the attacker node by using the elegant game theory approach. The root node calculates the checksum based on the SRH and stores it in the reserved bits of RPL protocol. Each sensor node also maintains payoff matrix. When the attacker modifies the SRH as discussed in section II; the sensor node generates IPv6 packet to add new neighbour in DODAG. This fake node is actually not available in the network. Apart from this, the checksum calculated will be different from the original one. Figure 5 depicts our implementation approach for detecting attacker nodes. Our lightweight method does not rely on cryptographic techniques or MAC-based solutions [14].

A. Simulation Result

This section V-A outlines the outcomes of our implementation of the Hatchetman attack on various performance indicators.

1) Impact on Downward PDR: Fig. 7 shows impact on PDR as we increases number of nodes. For a normal scenario, PDR is 1, but when the attacker node is placed at 1-hop distance, it reduces the PDR.

2) Impact on AE2ED: Figure 8 demonstrates a comparable impact on the average end-to-end delay when the number of nodes is increased. As the quantity of generated and received packets decreases, the AE2ED also decreases.

3) Impact on Overhead Packets: The attacker node also increases the overhead packets as shown in Fig. 9. These packets are IPv6 control messages and some control messages are due to the hatchetman attack when the node cannot forward the packet.

VI. CONCLUSION AND FUTURE SCOPE

IoT has powered Industry 4.0, and it has significantly impacted our life. Each small to big device will be connected to the Internet shortly. This big resolution will have lots of pros and cons in the technology. The positive is that we can track all our belongings through web applications. It will help in the productivity of agriculture fields and, at the same time, will be helpful for the security of endangered species. IoT also helps in enhancing a country's security. The way IoT has emerged in recent times shows a new industrial revolution. But connecting every device, from small to big, to a network has led to a broader scope for attacks. This paper focuses on one of the IoT-specific, which is Hatchetman attack. The attack potentially decreases the PDR and increases the error messages in the network. This makes the network unstable and it tries to reconfigure itself by resetting the trickle timer.

In the future, we plan to implement different IoT-specific attacks and generate a multi-label attack dataset to detect the network's behavior. We also plan to showcase the attack's impact, where the attackers can coordinate with each other to make the attack more impactful. Coordinated attacks drastically impact the network, and the attacker nodes are not easily identifiable.

REFERENCES

[1] I. H. Khan and M. Javaid, "Role of internet of things (iot) in adoption of industry 4.0," *Journal of Industrial Integration and Management*, p. 2150006, 2021.

Fig. 5: **Hatchetman attack detection using matrix game approach**

Fig. 6: **How the Model Works**

Fig. 7: **Impact on Downward PDR**

Fig. 8: **Average End to End Delay with and without attack.**

Fig. 9: **Increase in Overhead Packets**

[2] M. Hasan, "State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally," https://iot-analytics.com/number-connected-iot-devices/, 2022, [Online; accessed 25-May-2022].

[3] M. Sain, Y. J. Kang, and H. J. Lee, "Survey on security in Internet of Things: State of the art and challenges," in *2017 19th International conference on advanced communication technology (ICACT)*. IEEE, 2017, pp. 699–704.

[4] 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.

[5] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2018, pp. 12–17.

[6] S. Oh, D. Hwang, K. Kim, and K.-H. Kim, "A hybrid mode to enhance the downward route performance in routing protocol for low power and lossy networks," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, p. 1550147718772533, 2018.

[7] R. Kumar, J. Grover, G. Sharma, and A. Verma, "Addressing dio suppression attack in rpl based iot networks," in *International Conference on Information Security, Privacy and Digital Forensics*. Springer, 2022, pp. 91–105.

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

[9] P. Thulasiraman and Y. Wang, "A lightweight trust-based security architecture for rpl in mobile iot networks," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–6.

[10] G. Sharma, J. Grover, A. Verma, R. Kumar, and R. Lahre, "Analysis of hatchetman attack in rpl based iot networks," in *International Conference on Emerging Technologies in Computer Engineering*. Springer, 2022, pp. 666–678.

[11] Z. Han, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[12] A.-u. Rehman, S. U. Rehman, and H. Raheem, "Sinkhole attacks in wireless sensor networks: A survey," *Wireless Personal Communications*,

vol. 106, pp. 2291–2313, 2019.

[13] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, "Coalitional game theory for communication networks," *Ieee signal processing magazine*, vol. 26, no. 5, pp. 77–97, 2009.

[14] G. Sharma, J. Grover, and A. Verma, "Qsec-rpl: Detection of version number attacks in rpl based mobile iot using q-learning," *Ad Hoc Networks*, vol. 142, p. 103118, 2023.