An Algorithm for Detection of Blackhole Attack on RPL

Meron Mekonnen
HiLCoE School of Computer Science and Technology
gatomeron@gmail.com

Taye Abdulkadir
Assistant Professor, HiLCoE School of Computer Science and Technology
tabdulkadir@gmail.com

Abstract

The Internet of things (IoT) is an emerging technology, which seems to have a promising future in linking the real and virtual world and hence changing the way we experience the world. As its implementation increases, attacks performed on it will also increase. Since it is connected with the physical world, the attacks can result in a serious damage. Consequently, it is required to provide protection mechanisms against the attacks.

Blackhole attack is one of the attacks, which is performed on RPL (Routing Protocol for Low- Power and Lossy Networks). In RPL network, the nodes connect to the network forming a tree topology. There are parents and children nodes, and packet is transferred only from children to parent or parent to children until it reaches its destination. In a blackhole attack, attacker node drops packets passing towards it, resulting in the packets from and towards its children not reaching their destination. This attack can result a serious consequence especially where real time data transfer is required.

In this work, an algorithm is designed for detecting the existence of blackhole attack. After an attack is detected, the attacker node (blackhole node) is identified and removed from the network. Blackhole attack is simulated using the Cooja simulator and the provided algorithm is tested by using *packet delivery rate* and *energy consumption* as evaluation metrics. Ensuring the improvement of packet delivery rate is vital because blackhole node drops packets. Additionally, IoT contains resource constrained devices and hence energy consumption has to be as minimal as possible.

The provided algorithm has successfully detected the existence of blackhole attack and the blackhole nodes are successfully identified and removed from the network. The simulation shows that the designed algorithm has improved the packet delivery rate without resulting higher energy consumption for a network of up to 22 nodes.

Keywords – *IoT, RPL, Blackhole attack, Detection*

1. Introduction

In the Internet of things (IoT), every day physical objects are connected to the Internet and are able to send and receive data. These objects generate data and communicate with each other as well as with their physical environment. Sensors play important role in IoT since they are the devices collecting heterogeneous data from the physical environment. They are resource- constrained devices with limited memory, processing capacity and battery power. Consequently, they use routing protocols designed for resource constrained devices.

RPL is a distance vector IPv6 routing protocol for low power and lossy networks [2] [3] [4]. Low power and lossy networks contains nodes which often operate in harsh and remote environment [2]. The term "lossy" refers to the unreliable nature of the communications in these networks [2]. RPL creates a routing topology in the form of a destination-oriented directed acyclic graph (DODAG) [2] [3] [4]. Directed acyclic graph is a graph that is not cyclic which means that it is not possible to start at one node and traverse the entire graph or come back to the same node by traversing the edges. The DODAG is a directed graph without cycles, oriented towards the root node which is the LBR (Low-power Wireless Personal Area Networks (LoWPAN) Border Router) [2]. Traffic flows from each node towards its immediate parent until it reaches the root and vice versa. There are various attacks performed on RPL and blackhole attack is one of these attacks [5].

In a blackhole attack, attacker node drops packets passing through it silently. Nodes oblivious of the malicious node, keep sending their packets while the malicious node keep receiving and dropping the packets like a hole which sucks in everything [5].

In the RPL protocol, since the sending node wouldn't expect confirmation message from the destination node, blackhole attack could remain in the network undetected. However, its impact can result in serious consequences and it needs to be identified and mitigated. This work focuses on providing an algorithm to identify and mitigate blackhole attack on the RPL routing.

2. Related Work

F. Ahmed, et al [6] have proposed a mitigation technique for blackhole attack in LLNs using local decision and global verification process. In the local decision process, nodes overhear data packets transmitted by their neighbors and they observe the communication behavior of their neighbors. If a node notices misbehaving activity in its neighbor, it counts the misbehaving activity and compares it to a given threshold value. If the count of misbehaving activity exceeds the threshold value, the neighbor is judged to be a suspicious node. When the threshold value of the suspected node is exceeded, the neighboring node which is the verification node initiates the global verification process. In the global verification process, the verification node sends a request to the root node to verify the suspicious node. The root node checks if the suspicious node is a blackhole node or not. If the suspicious node is found to be blackhole, the verification node broadcasts the blackhole node ID to its neighbors. The child nodes of the blackhole node will then avoid using it and will use another node to transfer data packet.

In order to identify blackhole nodes, each node in the network must actively listen to packets exchanged between their neighbors and observe the communication behavior of their neighbors. This results in high energy consumption for the resource constrained IoT devices which is a drawback on the provided solution [6].

J. Jiang, et al [7] have proposed a defense mechanism for blackhole attack on RPL. They proposed for the root node to actively track the packets instead of each node monitoring its neighbor's packet which removes the high energy consumption problem due to neighbor packet monitoring in [6]. They used a sequence number in each node's message which

increments as a message is sent and which is stored at the root. A counter variable is used at the root which records the number of packets sent out by a node and successfully received by the root. By using the stored sequence number and counted value, the packet loss rate is determined. Nodes with low packet delivery ratio are determined as blackhole nodes. Once the root node identifies a blackhole node, it broadcasts the message to the network. When a node receives the message, it broadcasts the message to its neighbors and if the blackhole node is its parent, it removes the blackhole node from its parents list. By this process the blackhole node is isolated by its children.

In [7], for determining the packet loss rate of a node, its packet is required to reach the root at some point so that packet loss rate of a node could be determined. This solution can be applicable for a blackhole node that drops some packets. However, if the blackhole node drops all the packets, then a node's packet is not going to reach the root, which means that it won't be possible to determine packet loss rate. Additionally, the node with low packet delivery ratio is identified as malicious node. However, having a low packet delivery ratio does not always indicate a blackhole node. A legitimate node could have a low packet delivery ratio if its children sends packet at a lower rate than the other nodes in the network. In this work, packet loss can be determined without requiring for a packet to reach the root. Furthermore, blackhole node identification is done not only by considering possible packet loss but also by considering that there might exist nodes that sends packets at lower frequency than the other nodes in the network, so that these nodes would not be misclassified as having packet loss.

3. The Proposed Solution

In order to detect the presence of blackhole attack on RPL and reduce its impact, an algorithm is provided in this work. The task of the algorithm is to detect blackhole attack on RPL and remove the attacker node from the network. When sending packets, nodes do not expect to receive confirmation for their packet reaching its destination. As a result, they remain oblivious if a blackhole node is dropping their packets which is why blackhole attack succeeds. In the provided algorithm, a sequence number is used in the sent messages to identify packet loss. The sequence number starts from one and increments as the node sends messages. Since IoT contains resource-constrained devices, it would not be possible to store the sequence number of each node on every other node. As a result, the root node is used for keeping the sequence numbers of all the nodes. The sequence number stored in the root is updated when a new packet is received from a node. Based on the stored sequence number on the root, each node's sequence number is checked relative to the other nodes, and it is determined whether the node is delayed or not.

"Sequence delay percentage (SDP)" is used to check sequence number of a node relative to the other nodes in the network. By comparing the sequence number of a node to the sequence number of all the other nodes in the network, the sequence delay percentage gives from what percent of the nodes in the network, is a node's sequence number low (delayed). Even if there is no current sequence number received from a node, the root can still compare the sequence number of the node relative to the other nodes using the previously stored sequence number of the node. This can be helpful when there is no packet to be received from a node

either due to the presence of blackhole attack or the node being a node which sends packets at lower frequency than the other nodes.

The steps used for computing the sequence delay percentage is discussed below

- *Perform the SD_{iff} of the node to be checked relative to node j , where node j represents all the other sender nodes excluding the node being checked*

$SD_{iff} = S_j - (I * S_c)$

Where SD_{iff} = sequence difference relative to node j S_j = sequence number of node j

S_c = sequence number of the node being checked I = interval

- *Count the number of nodes whose SD_{iff} is greater than zero*

- *Determine the sequence delay percentage for the node*

$$ctSDP = \frac{}{n-1} x100$$

Where SDP = sequence delay percentage

ct = count of number of nodes whose SD_{iff} is greater than zero n = total number of nodes excluding the root node The sequence difference of a node is its sequence number difference relative to the other node it is being checked to. It is used to determine whether a node has a smaller sequence number or not relative to the other node which is used for comparison. Positive SD_{iff} value indicates that the node being checked has a lower sequence number below the acceptable limit and hence the node is possibly delayed relative to the node it is being checked to. On the other hand, negative SD_{iff} indicates that the node being checked is not delayed relative to the node used for checking.

The count (ct) value determines the number of positive sequence difference values obtained. Higher count value for a node means a node's sequence number is less than its expected sequence value relative to large number of nodes.

3.1 Proposed algorithm in this work

Module1: *Sequence check of nodes*

```
Let nd = number of nodes excluding the root
for i=0 to nd-1 do
    ct=0
    for j=0 to nd-1 do
        SDiff = Sequence[j]-Interval*Sequence[i]
        if SDiff > 0
            ct=ct+1
        end if
    end for
    SDP[i] = (ct/(n-1))*100
```

```
11  end for
```

Module2: *Identify delayed nodes*

```
1  for i=0 to nd-1 do
2        if SDP[i]>50
3             delayed_node[i]=node_id(i)
4             delayed_time[i]=delayed_time[i]+1
5        else
6             delayed_node[i]=0
7             delayed_time[i]=0
8        end if
9  end for
```

Module3: *Send message to delayed nodes*

```
1  for i=0 to nd-1 do
2       if delayed_node[i]!=0
3            if delayed_time[i]>0 and delayed_time[i]<=3
4                 client_ipaddr=ip_return(delayed_node[i])
5                 uip_udp_packet_sendto((server_conn, msg,
                  sizeof(msg),client_ipaddr,
                  UIP_HTONS(UDP_CLIENT_PORT)) //This is is part of
                  Contiki OS
6            else if delayed_time[i]>3
7                 attacked_nodes_ct = attacked_nodes_ct + 1
8            end if
9       end if
10 end for
11 if attacked_nodes_ct!=0
12      blackhole_node_children() //Module4
13      blackhole_node_id() //Module5
14 end if
```

Module4: *Identify attacked nodes*

```
1  attacked_ct=0
2  for i=0 to nd-1 do
3       if delayed_node[i]!=0 and delayed_time>3
4            attacked_node[attacked_ct]=delayed_node[i]
5            attacked_ct=attacked_ct+1
6       end if
7  end for
```

Module5: *Identify blackhole node*

```
1  attacked_ct=get_attacked_ct()
2  for i=0 to attacked_ct-1 do
3        ancestor_get(attacked_node[i]) //gives ancestors of a node starting
       from the node's parent
4        ancestor=attacked_node[i]
5        for j=0 and ancestor!=1
6             ancestor=ancestor_table[i][j]
7             if SDP[ancestor]<50
8                  blackhole_node[blackhole_ct]=ancestor
9                  blackhole_ct = blackhole_ct + 1
10                 ancestor=1
11            end if
12       end for
13 end for
```

Module6: *Remove blackhole node*

```
1  blackhole_ct=get_blackhole_ct()
2  for i=0 to blackhole_ct-1 do
3        client_ipaddr=ip_return(bh_node_neighbour[blackhole_node[i]])
4        uip_udp_packet_sendto(server_conn, msg,
         sizeof(msg),client_ipaddr,    UIP_HTONS(UDP_CLIENT_PORT))
         //This is part of Contiki OS
```

3.2 Blackhole attack detection

The sequence delay percentage is determined for every node so that to identify whether there is packet loss or not. Higher SDP means the checked node's sequence delay percentage is higher compared to the sequence number of the other nodes in the network, which might indicate packet loss. However, having higher SDP does not always mean there is a packet loss. It might be because of the node being checked sending packets at lower frequency than the other nodes. Furthermore, packet loss necessary might not mean the presence of blackhole attack. The packet loss might be due to the lossy nature of the network. Which means that a packet might be lost since the network is unreliable. Consequently, 50% is used as a threshold value, and hence, for a node to be considered delayed, it needs to have a sequence delay percentage of greater than 50% which means that its sequence is delayed compared to more than half of the other nodes.

After a node is identified as delayed, a message is sent by the root to the delayed node to confirm whether the higher SDP value is due to the presence of blackhole attack or due to the required task of the node. The root then expects to receive a reply from the node. If the node gives back reply, it indicates that the node is not an attacked node but rather a node, which sends packet at a lower rate. However, if there is no reply from the node, the root then sends message again for the second and third time to the node because the first message might not have reached the node due to the unreliable nature of the network. However, if there is still no reply from the node, then it means than the node is attacked node and there is a blackhole node in the network.

3.3 Blackhole node identification

Once the existence of blackhole node is identified, the root then searches for the blackhole node. It starts searching from the parent of the attacked node and traverses through its ancestors towards the root. The first obtained non-delayed node or node with lower sequence delay percentage is the blackhole node. This is because in a blackhole attack, the blackhole node doesn't transfer packets of its children but its packets are transferred. percentage is the blackhole node. This is because in a blackhole attack, the blackhole node doesn't transfer packets of its children but its packets are transferred.

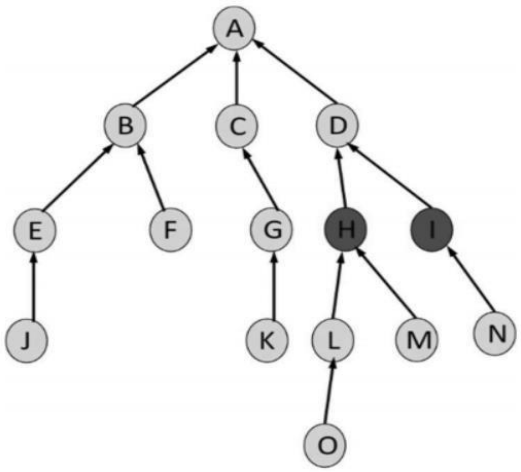

Figure 1: Multiple blackhole nodes

In figure 1, node A is the root node and node H & node I are blackhole nodes whereas node L, M, O & N are victim nodes. When one or all of node L, M or O are identified as attacked nodes, the root looks for a node with a non-delayed sequence delay percentage in their ancestor. When it reaches node H, it will identify it as a blackhole node and stops searching since Node H is not delayed. Similarly, after node N is identified as an attacked node, the root search for node N's ancestors and will find node I which is the parent of node N and a non-delayed node. Hence, node H and node I will correctly be identified as blackhole nodes.

3.4 Blackhole node removal

After the blackhole node is identified, the root sends message to the attacked nodes through their neighbours. When the blackhole node's children receives the message, they will drop the blackhole node from their parent list and will choose a node among their

neighbors, as their new parent. The blackhole node's neighbours will also remove it from their neighbours list. Consequently, the blackhole node will be isolated and removed from the network.

4. Experimentation

In order to evaluate the proposed algorithm, blackhole attack is simulated using Cooja simulator. Cooja is the network simulator for Contiki which is an operating systems used for wireless sensor networks. Two networks with different sizes are simulated and blackhole attack is applied in the networks. In the smaller network, a total of 12 nodes which contains 1 sink node, 2 blackhole nodes and 9 sender nodes is used. The larger network contains a total of 22 nodes with 1 sink node, 3 blackhole nodes and the remaining 18 sender nodes. The simulation is executed in both networks with and without the proposed algorithm which is a total of four simulations.

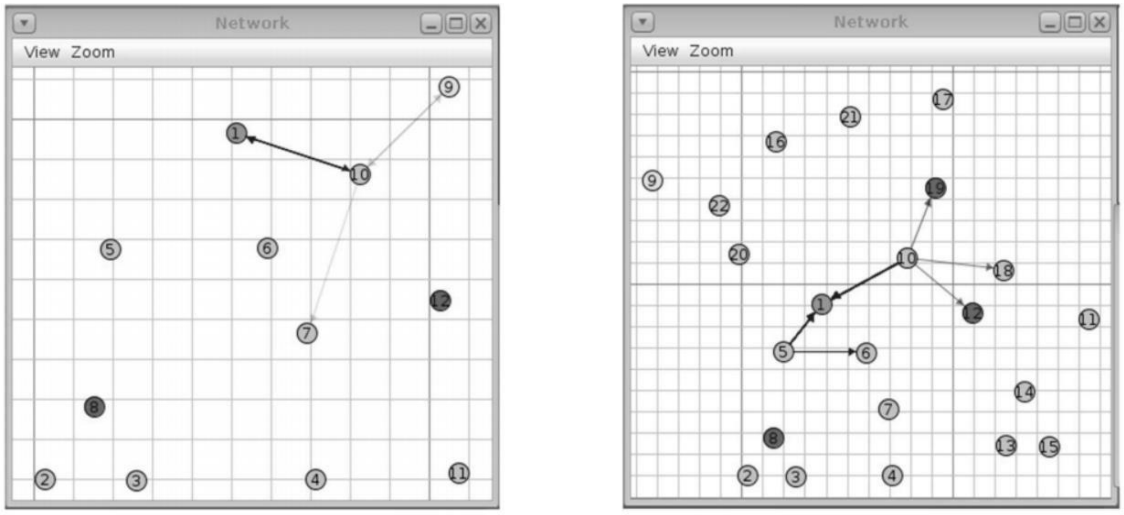

Figure 2: Simulation interface for smaller and larger networks

Figure 2, shows the simulation interface for the smaller and larger networks. In the smaller network, node 1 is root node, node 8 & 12 are blackhole nodes and the other nodes are legitimate nodes. In the larger network, node 1 is the root node, node 8, 12 & 19 are blackhole nodes and the other nodes are legitimate nodes. By applying the proposed algorithm, blackhole attack is detected and the blackhole nodes are identified. The attacked nodes have chosen a new parent and their packets were able to reach the root afterwards.Using the provided algorithm, packet delivery rate of the smaller network has improved by 42.1% and the packet delivery rate of the larger network has improved by 31.2%. This shows that packets which were previously being dropped by the blackhole nodes, are able to reach their destination when the provided algorithm is applied. There is an increase in energy consumption of the smaller and larger networks by 4.6% and 6.2% respectively using the proposed algorithm which is not high. This is important since IoT contains resource constrained devices.

5. Conclusions

In this work, an algorithm have been proposed to mitigate blackhole attack on the routing protocol RPL. The algorithm detects the presence of blackhole attack, identifies the attacker node and removes the attacker node from the network. The provided algorithm is tested by simulating blackhole attack. The result obtained in the simulation shows an improvement in packet delivery rate. This shows that the provided algorithm has succeed in detecting blackhole attack and removing attacker node. Furthermore, the energy overhead due to the provided algorithm is within acceptable limits. This shows that the proposed algorithm has provento be effective for detection of blackhole attack on RPL.

The proposed algorithm in this work has been tested and been proven to be viable for a network containing up to 22 nodes. For future work, providing a solution which could be viable for larger network size is recommended. Additionally, this work focuses only on blackhole attack. However, considering different attacks occurring at the same time could give the work more depth.

References

[1] Meron Mekonnen, "An Algorithm for Detection of Blackhole Attack on RPL," *A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science in Software Engineering, HiLCoE School of Computer Science and Technology,* December 2022.

[2] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand and C. Chauvenet, "RPL: The IP routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance,* April 2011.

[3] T. Tsvetkov and A. Klein, "RPL: IPv6 routing protocol for low power and lossy networks," *Network,* July 2011.

[4] J. V. V. Sobra, J. J. P. C. Rodrigues, R. A. L. Rabêlo, J. Al-Muhtadi and V. Korotaev, "Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications," *Sensors,* vol. 19, no. 9, January 2019.

[5] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," *In2015 International conference on pervasive computing (ICPC),* pp. 1-6, January 2015

[6] F. Ahmed and Y.-B. Ko, "Mitigation of black hole attacks in Routing Protocol for Low Power and Lossy Networks," *Security and Communication Networks,* vol. 9, pp. 5143-5154, October 2016

[7] J. Jiang, Y. Liu and B. Dezfouli, "A Root-based Defense Mechanism Against RPL Blackhole Attacks in Internet of Things Networks," *In 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE,* pp. 1194-1199, 2018