Annals of Emerging Technologies in Computing (AETiC)
Vol. 5, No. 5, 2021

Research Article

Sinkhole Attack Detection and Avoidance Mechanism for RPL in Wireless Sensor Networks

Ansar Jamil*, Mohammed Qassim Ali and Muhammed E. Abd Alkhalec

Universiti Tun Hussein Onn Malaysia, Malaysia
ansar@uthm.edu.my; mmk4580@ gmail.com; gi170023@siswa.uthm.edu.my
*Correspondence: ansar@uthm.edu.my

Received: 17th October 2020; Accepted: 15th November 2020; Published: 20th March 2021

Abstract: The security issue is one of the main problems in Wireless Sensor Network (WSN) and Internet of Things (IoTs). RPL (Routing protocol for low power and lossy networks) is a standard routing protocol for WSN, is not to be missed from being attacks. The performance of RPL is reduced significantly after being attacked. Sinkhole attack is one of the most common attacks to WSN and RPL, threatening the network capability by discarding packets and disrupting routing paths. Therefore, this paper proposes a new Secured-RPL routing protocol to detect and avoid sinkhole attacks in the network, which is called Cross Layers Secured RPL (CLS-RPL). This routing protocol is enhanced of the existing RPL routing protocol. CLS-RPL is a cross-layer routing protocol that uses information from the data link layer in its security mechanism. CLS-RPL uses a new technique and concept in detecting a sinkhole attack that is based on eave-listening (overhearing) that allows a child node to eave-listening its parent transmission. If the child node does not hear any transmission from its parent node after sending several packets, this means its parent node is a sinkhole attacker. Otherwise, if the node hears transmission from its parent node, this means that its parent node is legitimate and continues to send more packets. CLS-RPL implements a simple security mechanism that provides a high packet delivery ratio. The finding shows that CLS-RPL provides 52% improvement in terms of packet delivery ratio when compared to RPL protocol.

Keywords: ***Cross-layer; RPL; Security; Sinkhole attack; Wireless Sensor Network***

1. Introduction

The IETF ROLL working group (routing at a low value of power and loss networks) provides a new RPL routing protocol that supports IPv6 for WSN. The module protocols make smart routing decisions when performance data affect the exchange of information and carry data packets to other sensor nodes. If routing results are not smart enough, more recycling for each target data is required in the WSN network, which affects power consumption, bandwidth, and sensor node processing [1]. With a dozen sensors nodes, RPL can route to thousands of devices and support traffic flow from point-to-point, and point-to-multipoint traffic. Besides, RPL's performance at LLN (Low power and Lossy Network) has low data transfer rates and high loss rates due to restrictions such as limited sensor processing capacity, limited battery capacity, and limited memory. Finally, the RPL routing protocol enables the efficient use of energy from smart devices, calculates sources, provides flexible topology and data routing [2][3].

The usage of devices of low power wireless has become very familiar in our life. Security goal is becoming a requirement in the various WSN applications such as healthcare, automobile, military, the solutions of the environment where it supports several advantages. WSNs provide considerable opportunities in establishing security projects. Designs related to security concepts

Ansar Jamil, Mohammed Qassim Ali and Muhammed E. Abd Alkhalec, "Sinkhole Attack Detection and Avoidance Mechanism for RPL in Wireless Sensor Networks", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 94-101, Vol. 5, No. 5, 20th March 2021, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2021.05.011, Available: http://aetic.theiaer.org/archive/v5/v5n5/p11.html.

have an important part in transferring information, it has been one of the most significant areas in terms of routing. Approaching routing data and source position causes network security dangers [4].

There are many different types of attacks that can affect the network because of using wireless radio medium and cooperative nature in network protocols. Attackers can listen to radio transmissions, transmit into the channel, and response to the previously heard packets. The malicious nodes may be inserted by the attacker which has similar functionality and performance with legitimate nodes. Typically, this is done by purchasing the same node separately or by getting any legitimate node and then program it to be a malicious node. Furthermore, it's hard to define a technique that can be used in wired networks to be applied in a WSN with all the limitations of energy and capabilities. The attacker may disable a WSN using different types of attacks such as Sybil attacks, sinkhole attacks, wormhole attack, and jamming or packet injection attacks [5].

This work focuses on the detection of sinkhole attacks. In this kind of attack, the objective of an attacker is to attract almost all traffic in the network through a malicious node, which acts as a metaphorical sinkhole or a node with the lowest rank. To achieve it, the malicious node is purposely located near the base station. The malicious node just simply drops all received packets or corrupt it before sending it to the next node. The effect of sinkhole attacks becomes more severe when the malicious node attacking the main route to the base station [5].

The remainder part of this paper is organized as follows: In Section 2, we present the related work. Section 3 presents the sinkhole attack on RPL. In Section 4, we present our detection algorithm to prevent sinkhole attacks in WSN. In section 5, the performance of our proposed detection sinkhole attack mechanism is evaluated through simulations. Finally, the conclusion of our work in Section 6.

2. Related work

There have been many studies conducted to design security mechanisms to detect attacks on wireless sensor networks and measured its performance within different simulation frameworks, scenarios, applications, and types of routing protocol used. In this section, some previous studies are discussed related to the detection and avoidance of sinkhole attacks.

R. Prajapati and N. Manjhi [6] proposed a grid-based technique in which deploys a mobile agent in each grid based on acknowledgment and delay of data. The time is computed then the mobile agent transfers the information to the base-station and finds out the malicious behavior of nodes. A Grid Base Cluster approach was proposed to find out the malicious behavior in WSN. In each grid, nodes are within each other transmission range that allows communication between them. An active mobile node moves around in clusters and the sender sends information about ACK. The behavior of nodes and count drop packets is observed by the mobile agent. If the mobile agent detects missed ACK from a node more than a threshold, then the mobile agent declares the node as a malicious node and informs it to other nodes in the network.

Dvir *et al.* [7] proposed new security services to RPL routing protocol to counter a sinkhole attack. It is called The Version Number and Rank Authentication (VeRA) security scheme. VeRA prevents misbehaving (compromised) nodes from impersonating a DODAG root and sending a DIO message with an illegitimate increased Version Number. Besides, VeRA also prevent misbehaving (compromised) nodes from publishing an illegitimate decreased Rank.

Le *et al.* [8] proposed an intrusion detection system (IDS) to detect RPL-based network topology attacks that disrupting the optimal and stability of network operation. These attacks include the Neighbor, the Rank, the Local Repair, and DIS/DIO attack. To protect RPL-based network topology, the IDS design includes semi-auto building a specification-based IDS model. The aims are to learn the transitions, states, and relevant statistics based on analyzing the trace file. The generated model is integrated into the IDS server.

Sheela *et al.* [9] recommended the protection program against sinkhole attack failures. Mobile agents are a controlled piece of software. They are moved from one node to another, not only sending data but also performing calculations. It is an efficient model for distributed solutions and is specifically attractive in a dynamic network environment. A highly restrictive routing program according to mobile agents has been proposed. It applies mobile agents to gather data about all mobile sensor nodes to keep each node up to date with the entire network, thus an accessible node does not respond to false data on malicious or attacked nodes; An important characteristic of the recommended technique is that it does not need a decryption or encryption technique to recognize a good attack.

We can conclude that the existing intrusion detection detects sinkhole attacks by using agent distributed based intrusion detection system, DIO message-based mechanisms, agent cluster-based intrusion detection system, etc. Based on our knowledge, none of them uses the eave-listening (overhearing) concept in their security design to provide information about packet transmissions of parent nodes in detecting sinkhole attacks. Because of that, we proposed and developed a new Secured-RPL routing protocol to detect and avoid sinkhole attacks in the network using eave-listening (overhearing) concept, which is called Cross Layers Secured RPL (CLS-RPL).

3. Sinkhole Attack on RPL

A sinkhole attack is a kind of routing attack that changes the DODAG graph in RPL. Sinkhole attack on RPL happens in two stages. First, the malicious node advertises falsified information data for parent selection to attract considerable traffic in the network. Second, the malicious node may discard or modify it after receiving packets illegally. A sinkhole attack can be launched using at least one malicious node as shown in Fig. 1. The malicious node advertises false information to other nodes in the network that it's the best node to send data to the base station. False information can be acting as the base station, the lowest rank level, and a high-quality link. Because of this, most of the nodes change their routes by selecting the malicious node as its parent nodes. Traffic in the network starts concentrating on the malicious node. Then, the malicious node just simply drops the traffic to prevent it from reaching the base station that can disrupt the operation of the network. Malicious attack nodes are typically distributed near the base station for severe impact [10, 11]. This sinkhole attack could be so strong in combination with another attack [9].

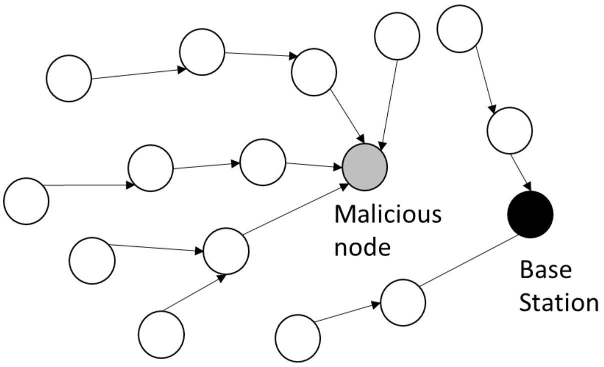

Figure 1. Sinkhole Attack

4. Cross Layers Secured-RPL (CLS-RPL) Design

CLS-RPL is a cross-layers security mechanism that involves the network layer and the data link layer. The data link layer responsible to overhear its parent transmission after sending a certain number of packets. Fig. 2 shows the overhearing concept used by CLS-RPL. Based on the figure, node B is located within the transmission of node A. Node B transmission covers node A and the base station. This mean, node B becomes the intermediate node for node A to deliver packets to the base station. Since node B is a legitimate node and the parent node of Node A, any transmitted packets by node A, node B forwards the packets to the base station. At the time node B is forwarding the packets to the base station, node A can overhear the transmissions of node B and receive it. Otherwise, all packets from node C will be dropped by the attacker node that caused node C does not overhear any transmission from the attacker node. CLS-RPL allows node A and node B to record the number of overhearing of its parent node respectively.

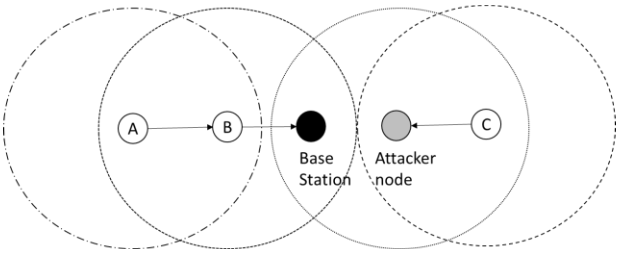

Figure 2. Overhearing concept in CLS-RPL

Based on this information, this layer determines the status of the parent node either attacker node or legitimate node. The updated status of each parent node is provided to RPL in the network layer. If an attacker node is detected, RPL changes its route avoiding the attacker node. The design implementation of CLS-RPL can be divided into three main parts: calculation of the number of overhearing transmission of a parent node, sinkhole attack detection, and sinkhole attack avoidance. Calculation of the number of overhearing transmission and detection of sinkhole attack are implemented in the data link layer. Sinkhole attack avoidance is implemented in RPL (which is in the network layer). Overall sinkhole detection and avoidance algorithm of CLS-RPL is shown in Table 1.

Table 1. CLS-RPL Sinkhole attack and avoidance mechanism

CLS-RPL security algorithm
a) Calculation number of overhearing
1. The node starts sending packets to its parent node.
2. The node starts overhearing for any packet transmission of its parent node.
3. When the node received a packet transmission, check the source of the packet, If the source address of the packet belongs to its parent node, add one to the number of overhearing
4. After 10 packet transmissions, determine the number of overhearing, *N*.
b) Detection of sinkhole Attacker
5. Based on the number of overhearing, N, the node determines the status of its parent node, p_{status} either legitimate node or attacker node.
6. Decision making: If $N > 0$; Legitimate node If $N = 0$; Attacker node
7. Update p_{status} to CLS-RPL routing protocol (network layer)
c) Avoiding Sinkhole Attack
8. If p_{status} is equal to a legitimate node, Remain the cost of its parent node. CLS-RPL keeps the parent node.
9. If p_{status} is equal to attacker node, The cost of its parent node is set to the maximum value, which removes the attacker nodes as the parent candidates. Then, CLS-RPL will select another node as a new parent that provides the least cost to the sink.

4.1. Calculation of Overhearing

Calculation number of overhearing, *N* for transmission of parent node requires modification on Contikimac and CC2420 radio driver. Contikimac is an asynchronous duty cycling mechanism, which enables by default in ContikiOS [12]. It does not require any signaling messages or additional packet headers to discover wake-up and sleep periods. The duty-cycling is based on static periodic wake-ups of nodes to assess the channel activity (clear or transmission). Nodes use unicast packet strobing and acknowledgment (ACK) packets to synchronize it wake-up cycles with their neighbor nodes. Received time of ACK packet is stored and used as a reference guide for neighbor's next wake-up cycles.

In the implementation of CLS-RPL, Contikimac determines the number of overhearing of its parent's transmission after every 10 packets transmission to its parent node. After the first packet is just transmitted, Contikimac starts recording the number of overhearing of its parent transmission. Each time Contikimac has detected a packet transmission of its parent node, the number of overhearing will be cumulated. Contikimac continues to overhear its parent transmission until completed sending 10 packets to its parent node. This cumulative value will be taken as the final number of overhearing and will be used to determine the status of its parent node either a legitimate node or an attacker node.

In order to allow Contikimac to overhear its parent transmission, radio address recognition must be disabled in the CC2420 driver. The CC2420 [13] can use hardware-based ACKS for received data frames that do not involve the upper layer, which enabled by default. These ACKs are generated and sent as the result of the integrated address recognition and the verification of the Frame Check Sequence (FCS) and CRC checksum in the last two bytes of IEEE 802.15.4 frames. The process is fully automatic at the physical layer without the involvement of the upper layer, which also excludes modification on the ACK frame. When the radio address recognition is enabled, the CC2420 only receives packets that belong to him and automatically rejects any other packets at the radio level (hardware layer). Thus, it is impossible to overhear any transmitted packet by its parent node because the packet is not destined for the node. By disabling the address recognition feature, CC2420 forwards all received packet transmission to Contikimac. Contikimac processes and examines each received packet. First, Contikimac checks the destination address of the packet, if it belongs to itself, just forward the packet to the upper layer, otherwise, it needs to check for the source address of the packet before drop it. If the source address belongs to its parent node, the number of overhearing is increased by one (which is $N = N + 1$). Otherwise, Contikimac just ignores it.

4.2. Sinkhole Attack Detection and Avoidance.

Based on the finalized N values, CLS-RPL determines the status of its parent node, p_{status} either legitimate node or attacker node. If the number of overhearing, N is more than 0, the p_{status} is set to a legitimate node. However, If the number of overhearing, N is equal to 0, p_{status} is set to an attacker node. Then, the parent status, p_{status} is updated to CLS-RPL routing protocol (which is network layer).

CLS-RPL routing protocol checks the parent status, p_{status}. If the p_{status} is a legitimate node, CLS-RPL does not change the path cost of the parent node and keeps the parent node as the best candidate to forward data to the base station. However, if the p_{status} is an attacker node, CLS-RPL will set the path cost of the parent node to the maximum value, which removes the attacker node as the parent candidates. This means CLS-RPL will not select the attacker as the parent node anymore and isolate it from the network. CLS-RPL avoids any node from sending packets through the attacker node. Then, CLS-RPL selects another node as a new parent that provides the least cost to deliver data to the sink from the best parent candidates.

5. Simulation Configuration

5.1. Contiki and Cooja Simulator

Contiki is a lightweight operating system (OS) for dynamic loading and replacement of individual systems and applications designed for WSN. In the simulation program, open-source is intended for resource-constrained embedded designs. Contiki OS requires at least 2 kB (RAM) and 30 kB (ROM). Contiki OS supports dynamic unloading and loading is functional in a resource-limited surrounding, whereas retains the main design lightweight and compact. Due to that, Contiki supports techniques that help in coding the intelligent object solutions. It supports libraries for the sake of communication abstractions and memory allocation. It is programmed in C language.

Instant Contiki is an Ubuntu Linux virtual machine that runs on VMware player. It acts as a virtual box consist of tools and compiler used for Contiki development, where different simulations on various scenarios can be run. This software can be executed on pre-installed Windows or Linux using VM ware player that runs the Linux virtual machine.

Cooja is the network simulator for the Contiki system. Each node runs on an actual compiled and executed code for a real hardware platform in the Contiki system, which is controlled and analyzed by the simulator. For example, COOJA informs the Contiki system to handle an event or fetches the entire Contiki system memory for analysis. This is performed by compiling Contiki for the native platform as a shared library and loading the library into Java using Java Native Interfaces (JNI). Several different Contiki libraries are compiled for different kinds of sensor nodes and loaded in the same COOJA simulation to create heterogeneous networks.

5.2. Simulation Configuration

In order to determine the proposed security mechanism, the simulation experiment was conducted in four different simulation scenarios as follows:

1. RPL without attacker node.
2. RPL with attacker node.
3. CLS-RPL without attacker node
4. CLS-RPL with attacker node

All the simulation scenarios are simulated using the same simulation configuration of network topology as shown in Fig. 4 and basic simulation configuration setting as shown in Table 1. The difference between each scenario is the routing protocol and network with an attacker node or not. The network consists of 12 sensor nodes, which is arranged in a tree topology in different scenarios. Over different Clear Channel Assessment (CCA) rates and packet rates. The selected CCA rate is CCA_2, CCA_4, CCA_8, CCA_16, and CCA 32 that allows Contikimac to keep checking the medium for any transmission at a rate $\frac{1}{2}$s, $\frac{1}{4}$s, $\frac{1}{8}$s, $\frac{1}{16}$s and $\frac{1}{32}$s respectively. CCA_2 is the lowest CCA rate and CCA_32 is the highest CCA rate. Packets are transmitted over different packet rates, which are 0.1, 0.2, 0.5 and 1 packet/s. Simulation time is set to 30 minutes for each scenario. As for the performance metric, the number of overhearing and packet delivery ratio are collected during the simulation for each scenario. Each simulation was repeated 10 times.

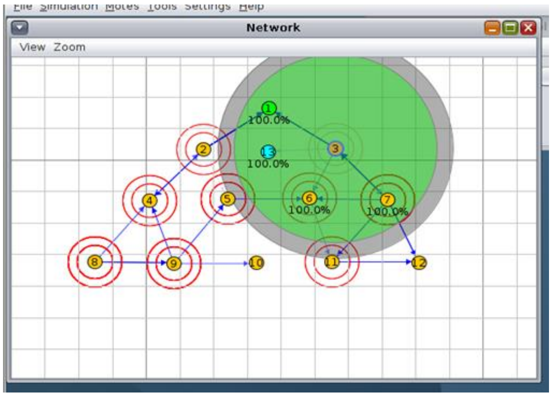

Figure 4. Network topology in COOJA

Table 1. Simulation Configuration

Simulation Parameters	
Simulation tool	Contiki 3.0 Cooja simulator
Mote type	Sky mote
Number of nodes	11
Number of sink node	1
Number of attacker node	1
Radio medium	UDGM: Distance Loss
Transmission range	30 m
Interference range	35 m

6. Results

Fig. 5 shows the average number of overhearing for different CCA rates. The graph indicates that the number of overhearing is increased when the rate of CCA is increased. It is expected because as the CCA rate is increased, Contikimac wake-ups to check the channel more frequently will increase the probability to overhear transmission from other nodes. Furthermore, if a packet transmission is detected during wake-ups, the receiver is kept on to be able to receive the packet and the subsequent packets. The CCA_2 recorded the lowest number of overhearing, which equal to 140. Otherwise, CCA 32 recorded the highest number of overhearing equal to 695.

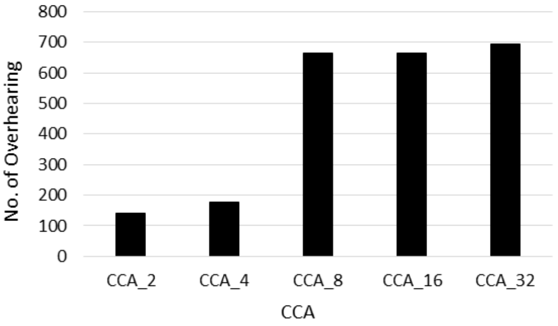

Figure 5. Number of overhearing with different CCA

Fig. 6 shows the average number of overhearing over different packet rates. The graph shows that the average number of overhearing is increased as the number of packets rates is increased in the network. The more packet transmitted in the network, the node detects more number of packet transmission that will increase the number of overhearing. The lowest number of overhearing is recorded by packet rate of 0.1 packet/s. The highest number of overhearing is recorded by packet rate of 1 packet/s.

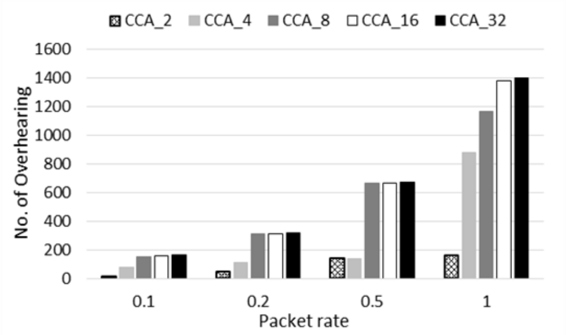

Figure 6. The number of overhearing for different packet rates.

Fig. 7 shows the average PDR for different simulation scenarios. The graph shows that CLS-RPL performance comparable performance when compared to RPL in the normal condition without any attack on the network. CLS-RPL and RPL recorded PDR equal to 0.9 approximately. During the sinkhole attack, CLS-RPL outperforms RPL by far. RPL suffers during the sinkhole attack just recorded a packet delivery ratio equal to 0.36. However, CLS-RPL able to deal with the attack by detecting and avoiding it provides a packet delivery ratio equal to 0.88. This means CLS-RPL gives a significant improvement in packet delivery ratio about 52% when compared to RPL. It is about just 2% lower when compared to its performance in a normal situation without attack.

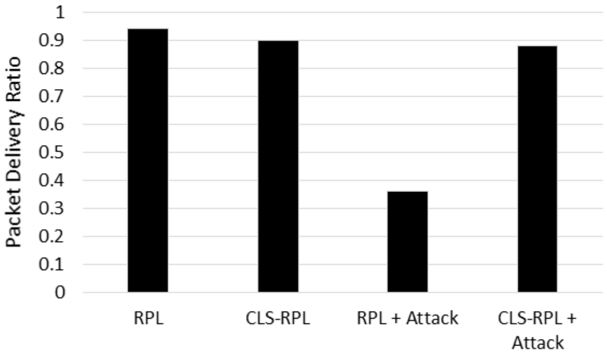

Figure 7. Average packet delivery ratio for different simulation scenarios

8. Conclusion

This paper proposed a security enhancement to RPL routing protocol which is called Cross Layers Secured RPL (CLS-RPL). The CLS-RPL routing protocol is a cross-layer routing protocol that uses information from the data link layer. CLS-RPL uses overhearing (eave-listening) to detect and avoid sinkhole attack. If a node does not hear any transmitted packets from its parent node after sending several packets, this means its parent node is a sinkhole attacker. The node removes the attacker node as its parent and finds an alternative parent node. Otherwise, if the node hears transmitted packets from its parent node, this means its parent node is legitimate and continues to send more packets. To determine the performance of CLS-RPL, Cooja was selected as the simulation tool. A WSN was deployed in Cooja environment consisting of 11 sensor nodes and one sink node. One sinkhole attacker node was introduced to attack the network. The finding shows that CLS-RPL provides 52% improvement in terms of packet delivery ratio when compared to RPL protocol.

Acknowledgement

The authors would like to express their thanks to Universiti Tun Hussein Onn Malaysia (UTHM) for providing financial support under H132-TIER 1 Grant.

References

[1] M. M. Khan, M. A. Lodhi, A. Rehman, A. Khan, and F. B. Hussain (2016). Sink-to-Sink Coordination Framework using RPL: Routing Protocol for Low Power and Lossy Networks, Journal of Sensors, vol. 2016, Article ID 2635429, 11 pages, https://doi.org/10.1155/2016/2635429.

[2] H. C. Chaudhari and L. U. Kadam (2011). Wireless Sensor Networks : Security, Attacks and Challenges, International Journal of Networking, vol. 1, no. 1, pp. 859–868.

[3] P. Janani, V. C. Diniesh, and M. J. A. Jude (2018). Impact of Path Metrics on RPL's Performance in Low Power and Lossy Networks, 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, pp. 0835-0839, DOI: 10.1109/ICCSP.2018.8524141.

[4] G. Ma, X. Li, Q. Pei, and Z. Li (2017). A Security Routing Protocol for Internet of Things Based on RPL, In 2017 International Conference on Networking and Network Applications (NaNA), Kathmandu, pp. 209-213, DOI: 10.1109/NaNA.2017.28.

[5] I. Abdullah, M. Muntasir Rahman, and M. Chandra Roy (2015). Detecting Sinkhole Attacks in Wireless Sensor Network using Hop Count, International Journal of Computer Network and Information Security, vol. 7, no. 3, pp. 50–56, DOI: 10.5815/ijcnis.2015.03.07.

[6] R. Prajapati and N. Manjhi (2016). Grid Base Cluster Approach for Detection of Sinkhole Attack in WSN, International Journal of Engineering and Computer Science, vol. 5, no. 17571, pp. 17571–17576, http://103.53.42.157/index.php/ijecs/article/view/2236.

[7] A. Dvir, T. Holczer, and L. Buttyan (2011). VeRA - Version Number and Rank Authentication in RPL, Proc. - 8th IEEE Int. Conf. Mob. Ad-hoc Sens. Syst. MASS 2011, pp. 709–714, doi: 10.1109/MASS.2011.76.

[8] A. Le, J. Loo, K. K. Chai, and M. Aiash (2016). A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology, Information 7, no. 2: 25, https://doi.org/10.3390/info7020025.

[9] D. Sheela, C. N. Kumar and G. Mahadevan (2011). A Non Cryptographic Method of Sink Hole Attack Detection in Wireless Sensor Networks, IEEE-International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, Tamil Nadu, 2011, pp. 527-532, DOI: 10.1109/ICRTIT.2011.5972397.

[10] Z. Zhang, S. Liu, Y. Bai, and Y. Zheng (2018). M Optimal Routes Hops Strategy: Detecting Sinkhole Attacks in Wireless Sensor Networks, Cluster Computing, vol. 6, pp. 1–9, https://doi.org/10.1007/s10586-018-2394-6.

[11] P. Pongle and G. Chava (2015). A survey: Attacks on RPL and 6LoWPAN in IoT, Int. Conf. Pervasive Computing, vol. 00, no. c, pp. 1–6, DOI: 10.1109/PERVASIVE.2015.7087034.

[12] Dunkels, Adam. (2011). The ContikiMAC Radio Duty Cycling Protocol, SICS Technical Report T2011:13 ISSN 1100-3154, December 2011.

[13] W. B. Pottner, S. Schildt, D. Meyer, and L. Wolf (2011). Piggy-Backing Link Quality Measurements to IEEE 802.15.4 Acknowledgements, Proc. - 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS) 2011, pp. 807–812, DOI: 10.1109/MASS.2011.92.