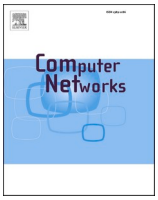
Computer Networks 218 (2022) 109391



Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet





A Lightweight Energy-Efficient Algorithm for mitigation and isolation of Internal Rank Attackers in RPL based Internet of Things

P.S. Nandhini [a,*], S. Kuppuswami [b], S. Malliga [a], R. DeviPriya [c]

[a] Department of Computer Science and Engineering, Kongu Engineering College, India
[b] Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, India
[c] Department of Information Technology, Kongu Engineering College, India

ARTICLE INFO

Keywords:
IoT
RPL
Rank
Rank attacks
Non-storing mode
Hash function

ABSTRACT

Internet of Things (IoT) has become a pivotal technology for connecting various physical things or devices to the internet. Despite the potential advantages, security in IoT network continues to be a major concern due to enormous volume of the data generated and transmitted. Since the devices in IoT are resource-constrained and heterogeneous, it is becoming harder to maintain security in routing protocols such as Routing Protocol for Low-power and Lossy networks (RPL). RPL uses control packets for the construction of tree topology and it is prone to various attacks such as flood, version, rank and so on. Rank attack leads to the formation of loop, increased latency in packet delivery, energy depletion in nodes, decreased delivery ratio of packets and so on. Since the impact of the rank attack is high on the network, a lightweight energy-efficient algorithm is needed for the detection and isolation of rank attackers. The proposed Rank Attack Detection (RAD) algorithm uses a non-cryptographic hash algorithm to preserve the integrity of control packets. It uses random sampling to reduce energy consumption. Once the rank attacker is detected, an alarm is generated against the attackers. The alarm is not a separate control packet; it is appended to the control message itself. The proposed algorithm is analysed using different topologies by comparing it with the already existing algorithms in terms of control packet overhead, delivery ratio, latency for delivery of data packets, accuracy and energy consumption. It achieves better accuracy in grid-centered topology and the average energy consumption is reduced in random topology.

1. Introduction

IoT is a network of disparate devices or gadgets that communicate with each other through internet without human intervention. The significance of an IoT system is that it improves the quality of life by delivering an appropriate solution or automated response to environmental changes. It entails the deployment of a large number of Low Power Lossy Networks (LLNs). The devices in LLN are low-powered with minimal processing speed, inbuilt memory and have a tiny operating system. The communication lines in LLN are sensitive to significant packet loss. So, traditional communication technologies and devices are restricted in IoT due to resource constraints. Therefore, communication, networking and connectivity protocols are chosen based on the type of IoT applications. The significant difference in LLN architecture is the incorporation of adaptation layer between data link and network layer. Both adaptation and network layers take care of routing. Traditional routing protocols are incompatible with LLNs [1,2]. To overcome this problem, a series of protocols has been formulated [3,4] for LLNs. Fig. 1 depicts the protocol stack of one of the LLNs, namely 6LoWPAN.

RPL is a network layer source routing and distance vector protocol that has been suggested and standardized by IETF [5]. It has grown in popularity across scientific, academic and industrial communities [6]. This protocol arranges the nodes in Destination Oriented Directed Acyclic Graphs (DODAGs) which are tree-like topologies and which use control packets like DODAG Information Object (DIO), Destination Advertisement Object (DAO) and DODAG Information Solicitation (DIS). The process of DODAG construction begins from the DODAG root. The root node broadcasts the DODAG information by transmitting DIO messages. The nodes that receive the DIO, calculates its rank and per hop routing metric in DODAG and replies with a DAO message (based on DAO Trigger Sequence Number) to inform its participation. After verification, an entry (DODAG Parent, Next hop Interface, Path metrics

* Corresponding author.
E-mail addresses: nandhinisv1@gmail.com (P.S. Nandhini), skuppu@gmail.com (S. Kuppuswami), mallisenthil.cse@kongu.edu (S. Malliga), scrpriya@gmail.com (R. DeviPriya).

https://doi.org/10.1016/j.comnet.2022.109391
Received 5 April 2022; Received in revised form 20 September 2022; Accepted 23 September 2022
Available online 29 September 2022
1389-1286/© 2022 Elsevier B.V. All rights reserved.

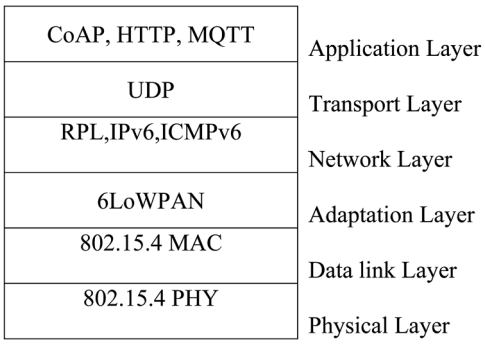


Fig. 1. Protocol Stack of 6LoWPAN.

value for each DODAG parent) for the confirmed node is created in the routing table of DODAG root. So, it is not possible to generate legitimate-looking consistent DAO messages by the attackers since the valid metrics of the routing table will not be known. Then, the confirmation is intimated through DAO-ACK packet from the root node. After that only a node can become a member of the DODAG and transmit the packets. If DAO messages are not generated, DAO-ACK messages will not be received. So, a node cannot become a member of a DODAG for transmitting the messages.

The rank value in DIO message indicates the position of a node in relation to the DODAG root as shown in Fig. 2

Rank value plays a vital role in all RPL operations. It is incremented in the downward direction. RPL protocol assumes that all the nodes in the DODAG are reliable (i.e. all the nodes are following the rules of RPL). It does not provide any security mechanism to check the behavior of the nodes [7]. Therefore, if a normal node is compromised, then it will turn into an attacker. Normal nodes turn greedy by altering the rank value (internal rank attackers) and influence the neighbor nodes to send the message through them. So, an internal rank attacker gains the control over the network [8]. This leads to the (i) formation of loops in routing [9], (ii) inefficient construction of routes, (iii) decreased Packet Delivery Ratio (PDR), (iv) increased generation of control packets and (v) increased latency. So, there prevails an urge to address these attacks.

The default security modes included in RPL standard have been investigated by a number of researchers. In [10], default security modes has been included in the implementation of RPL and it is found that it consumes more resources which are already a constraint in LLNs. So, the main motivation of the proposed research is to detect and isolate increased and decreased rank attackers in RPL based IoT by using lightweight energy-efficient algorithm. To do so, a Rank Attack Detection (RAD) algorithm is proposed. The primary objective of this algorithm is to guarantee reliable and secure data transfer across various nodes in the broad and dispersed IoT networks. Thus, RAD develops appropriate coordination among the normal nodes by isolating the rank attackers from DODAG.

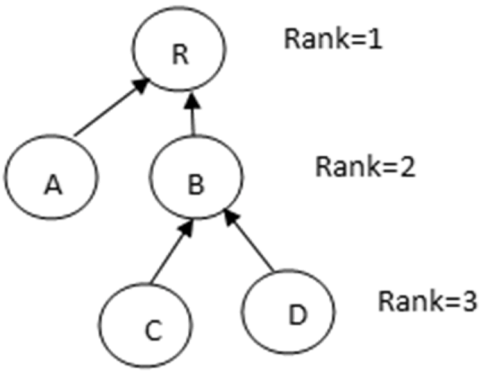


Fig. 2. DODAG with Rank.

Following research questions are addressed in the presented research.

1. How can rank attacks be detected and an internal rank attacker be identified in an energy-constraint network using non-cryptographic hash function?

2. Is it possible to detect internal attackers without using any extra control packet so that traffic overhead can be reduced in the network?

To address the first question, we propose an algorithm called RAD which uses integer hash function over certain fields of DAO message to detect the increased or decreased rank attack launched by internal rank attackers but in most of the existing systems cryptographic hash is used which consumes more energy.

To answer the next question, we have used the unused fields in the DIO and DAO messages for detection of internal rank attackers. So, this will not incur any additional control packets but most of the existing attempts have used extra control packets for detection of internal rank attackers. Our contributions in this paper are

- Reducing the number of control packets by appending extra fields in DAO and DIO messages
- Proposing a non-cryptographic lightweight hash algorithm to preserve the integrity of the control packets
- Proposing a local alarm to conserve energy in the case of minor impact caused by the attackers
- Using random sampling to conserve the energy instead of checking all the control packets

The remaining parts of the paper are organized as follow: Section 2 presents an overview of RPL and the recent literature reviews on the countermeasures for rank attack. The details of the proposed algorithm are given in Section 3. Simulation and its results are examined in Section 4. Section 5 focuses on the conclusion and future work.

2. Background

2.1. RPL

RPL optimizes the DODAG topology based on the specific goals of an application. A set of DODAGs form an RPL instance. Many RPL instances can exist in the network. Each instance in DODAG is based on a distinct Objective Function (OF). Minimum Rank with Hysteresis (**MRHOF**) is used in the current study to select the lowest-cost path to avoid the frequent changes of the network. [6,11] have provided a detailed study and comparative overview of standard RPL OF and research initiatives.

RPL operates in two modes: storing and non-storing [12]. In storing mode, the routing information is stored in all the nodes. According to RFC 6550, the intermediate nodes do not maintain the routing information. The root node is the only node that maintains the routing information of its DODAG. If a node wants to transmit a packet to the nodes not at one hop distance, then the packet is transmitted to the root and reaches the destination via source routing. Thus, the root knows the neighbours of nodes that are more than 1-hop away [13]. In the proposed algorithm, non-storing mode of RPL is considered.

Modification of values in the control packets would result in a wide range of security attacks. Based on the origin of attack, the attacks are classified as (i) Topology based attacks and (ii) DAG inherited attacks [14]. Rank attack is one of the DAG inherited attacks. The attackers broadcast the DIO message with increased or decreased rank attack. A small change in rank value alters the entire topology itself.

In **increased rank attack**, an attacker advertises an increased rank value to influence its neighbor nodes to choose a different parent. As a result, the child node in the sub-DODAG of an attacker searches for another preferred parent, thereby an attacker pretends as if it has invaded deeper in the DODAG. So, that the size of the parent set will be increased [15]. The increased rank attack is described in Fig 3.

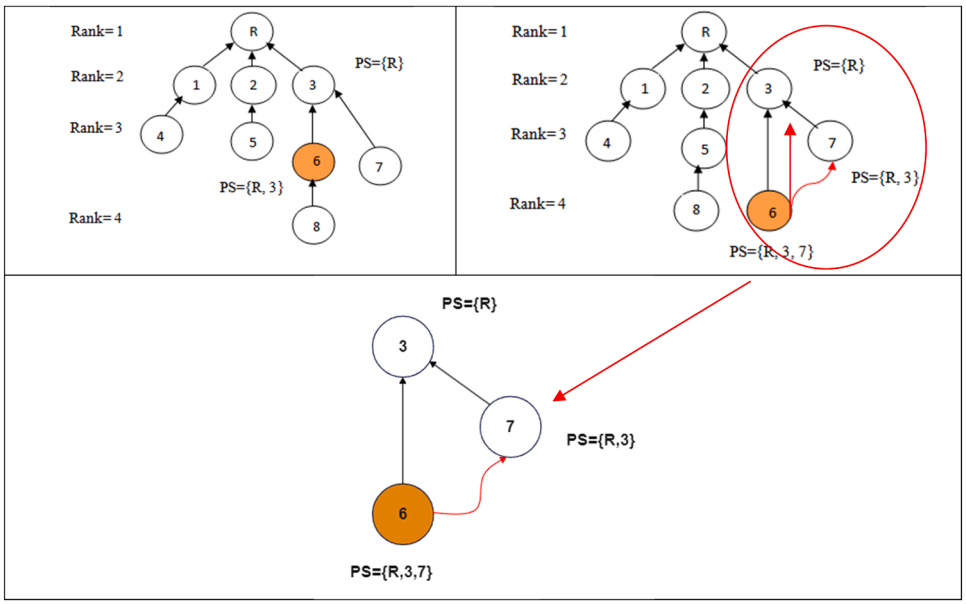


Fig. 3. Increased Rank attacker.

In Fig. 3, a simple **random** DODAG topology with 9 nodes is considered. Among the 9 nodes, R is the root node. The rank value of node 6 is 3. Since it is a greedy malicious node, it has increased its rank value from 3 to 4. Thus, node 6 has become an internal rank attacker (**Increased Rank attack**). Therefore,

Step 1: Node 8 (Child node of an attacker (6)) cannot have node 6 as its parent because the parent and the child cannot have same rank. So, node 8 chooses node 5 as its preferred parent (This is not a problem)
Step 2: The attacker node (6) pretends as if it has invaded deeper than node 7
Step 3: Rank of attacker 6 is increased from 3 to 4. So, the attacker node (6) receives the DIO message through nodes 3 and 7
Step 4: Attacker 6 responds with DAO messages for both DIOs received through nodes 3 and 7
Step 5: Thus, node 3 and 7 become the parents of attacker 6 and the Parent Set (PS) of the attacker (6) is increased from {R, 3} to {R, 3, 7}
Step 6: This leads to the formation of loop in DODAG topology which is against the rule of RPL

Similarly, the **decreased rank attackers** are those who advertise the decreased rank value in DIO messages. Since the rank value is decreased, the node seems to appear closer to the root and most of the nodes choose the attackers as their preferred parent to reach the root. The decrease rank attack has greater impact because it draws more traffic than really it should. It provides an opportunity to inspect, manipulate or drop the control packets. Therefore the integrity of the control packets would be compromised. Thus, this attack triggers various other attacks such as eavesdropping and sinkhole attacks [16], thereby the freshness of the routing information and the identity of the parents are lost.

In Fig. 4, a simple **random** topology with 8 nodes is considered. Among the 8 nodes, node 4 is an attacker and its original rank is 3.

Step 1: Node 4 decreases its rank value to be 2 in DIO message and transmits it to the nearby neighbours.
Step 2: The nearby neighbours assume that the node 4 is nearer to the root and chooses it to be the preferred parent.
Step 3: All the messages from the sub DODAG of an attacker reaches the root through it. Thus, this paves a way for various other attacks.

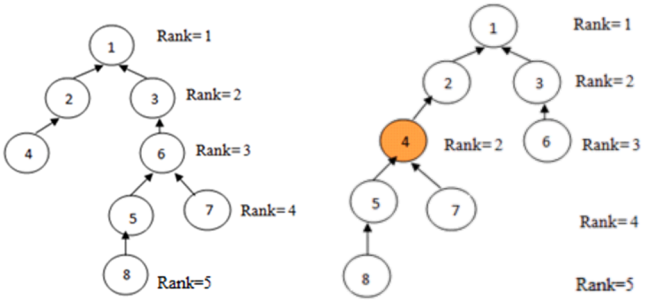


Fig. 4. Decreased Rank attackers.

2.2. Related works

With the emerging advancements in IoT, an extensive research has been carried out by the researchers to identify the solution for rank attacks in RPL. The aim of the rank attackers is to convince the neighbor nodes to become the preferred parent. After becoming a preferred parent, the rank attackers would alter, discard or drop the packets that flow through them. As a result, the entire throughput of the application is decreased. A summary of the existing researches including threshold based mechanism, trust based mechanism, cryptography and Intrusion Detection Systems (IDS) is provided.

In [17], the Sink Based IDS (SBIDS), a centralised approach for the detection of rank attackers in the non-storing mode of RPL, is proposed. By default, the root node is assumed to be a trusted one in this approach. Root node monitors the change in the rank values and defines the threshold value for parent switching accordingly. AES-128 is deployed for encryption. The downside of the proposed approach (SBIDS) is that it consumes more energy. So, a lightweight non-cryptographic hash function is used in the proposed algorithm to conserve energy.

In [18], a Secure RPL Routing Protocol (SRPL-RP) protocol is proposed for the detection, mitigation and isolation of rank and version attackers. This protocol deploys threshold value and attacks status table for the detection of rank attackers. Based on the comparison of rank with the threshold value, the attacker would be detected. The detected attackers are isolated and blacklisted. The blacklisted attackers are not included in the topology construction. Internal rank attackers are not considered for the evaluation of the protocol, but this is addressed in the

proposed algorithm. In [19], OF based on Echelon Metric is used. Rank Check Algorithm (RCA) is proposed to check the validity of the advertised rank. Echelon value is determined additively by the root node and its associated parents. RCA algorithm makes it easier to detect the attackers at the node level itself. The research gap identified is that no alarm or alert is generated against the attackers to the reachable normal nodes. If an alarm is generated, then the node that receives the DIO message from the attackers need not process the DIO message. Based on the consequences of the network, a local and/or global alarm is generated against the attackers in the proposed algorithm. Thereby the energy consumption of the network is reduced.

In [20], the effects of resource and topology attack in RPL is investigated. Increased and decreased rank attacks are categorised under resource based attacks. From the simulation result, it is inferred that all the considered RPL attacks decrease PDR, increase end-to-end Delay (E2ED) and power consumption. There is no mitigation or detection mechanism addressed in this paper. The authors [21] proposed an optimization algorithm for optimizing the routing paths and detection of rank attackers. This algorithm prevents the attacker from being selected as a preferred parent. Its limitation is that there is a problem of local optima. Its parameters can be adjusted by Fuzzy System to avoid this problem. The authors [22] proposed a design called SMTrust. The design calculates the trust value based on the mobility of sink and sensor nodes. The design is not implemented and so its effectiveness cannot be evaluated.

The authors [23] have included Message Authentication Code (MAC) in a DAO message to check the authentication of the nodes in the network. To detect the increased and decreased rank attacks, this mechanism incorporates the distributed module executed at all the nodes and the centralised module executed at the root node. These modules are executed to ensure that the rank values are not altered illegally by the nodes in the DODAG. The mechanism consumes more energy which is not appropriate for resource-constrained environments. So, a lightweight algorithm is proposed in the present study to conserve energy. S-MODEST is a security solution presented in [24] that more correctly detects the attacker while consuming less resources. The authors [9] have analysed the effects of decreased rank attack and proposed a detection scheme based on Multi-layer Perceptron model. Likewise, the authors [25] have used the Round Trip Time (RTT) of a request to the server to detect the decreased rank attack. In [26], the authors have deployed Support Vector Machine classifier for classifying the malicious behavior from the normal behavior. In [27], the authors have proposed a technique for the detection of rank and sinkhole attack in multi-sink DODAG topology. The paths that are not trusted are blocked instead of isolating the malicious nodes. Since multi-paths are considered, this leads to more energy consumption.

Already, an extensive review on the effects of the rank attacks and its countermeasures are done by us in [28]. In the review, the countermeasures are classified as Defense and IDS based mechanisms. Cryptography, trust and threshold based mechanisms are grouped under defense mechanism. Anomaly, Specification and Hybrid based solutions are grouped under IDS. The proposed RAD algorithm comes under cryptography based mechanism. To summarize, from the literature survey it is understood that most of the existing works require more computational power and do not use any light weight hash algorithm to detect the presence of internal rank attackers. The concept of local alarm and random sampling of inconsistent packets have not been discussed so far. Research gaps of the existing attempts are presented in Table 1. Table 1 presents a few of the most prominent available existing solutions for RPL based IoT networks.

From the literature survey, it is understood that none of the existing studies attempt to include all the parameters such as PDR, E2ED, control packet overhead, energy consumption and accuracy. These parameters play a vital role in evaluating the performance of the attack detection algorithm. So, in the proposed study, all the parameters are considered to evaluate the performance of RAD and it is the first of its kind. The research gaps identified in Table 1 are also taken into account when designing the proposed algorithm.

The proposed system uses a lightweight energy-efficient RAD algorithm for the detection of the rank attackers. This algorithm detects and isolates the rank attackers during the formation of DODAG. Only normal nodes would be participating in the forwarding operations and the rank attackers would be isolated from DODAG. In this way, security against rank attackers can be guaranteed in RPL based IoT networks.

3. Proposed methodology

3.1. Basic idea on working of RAD

RAD algorithm alters DAO message by appending fields such as rank,

Table 1
Summary of Related works.

References	Type of mechanism	Defense mechanism	Research Gaps	Performance Metrics	Validation Tools	Mobility
Bang et al., 2022 [19]	Threshold	Echelon Metric	Only less number of nodes is considered for evaluation.	PDR, E2ED and Power consumption	Cooja Simulator	No
Hkiri et al.,2022 [20]		–	Only the effects of the resources and topology attacks are analysed	E2ED, PDR, throughput and power consumption	Cooja Simulator	No
Seyfollahi et al., 2022 [21]	Trust based mechanism	MFO-RPL	Local optima problem due to MFO-RPL	Packet loss, Rank Switching Ratio and Convergence time	Real time analysis	No
Muzammal et al.,2022 [22]	Trust	SMTrust	Not implemented and tested	Attack detection and accuracy	–	Yes
Shafique et al.,2021 [17]	Cryptography	SBIDS	Increases power consumption and computational overhead	Accuracy and average power consumption	Cooja Simulator	Yes
Karmakar et al.,2021 [23]	Cryptography	LEADER	Energy-intensive since HMAC-LOCHA is used for MAC computation	PDR, latency, energy consumption and detection latency	Cooja Simulator	No
Almusaylim et al.,2020 [18]	Threshold	SRPL-RP	Internal rank attackers are not considered	Control message overhead, average energy consumption and accuracy	Cooja Simulator	Yes
Choukri et al., 2020 [9]	Graph	–	Only the effects of various rank attacks are analysed. No detection methods are proposed.	Average power consumption, ETX, packet loss and Beacon Interval	Cooja Simulator	No
Kiran et al., 2020 [24]	Trust	S-MODEST	Mobility of the nodes and the internal rank attackers are not considered	Detection accuracy, energy consumption and throughput	Cooja Simulator	No
Seth et al., 2020 [25]	Threshold	RTT	Internal rank attackers are not considered	Accuracy	Cooja Simulator	No
Said et al.,2020 [26]	Anomaly based IDS	SVM	IDS can be extended to detect anomalies in cloud based services also	Energy and Power consumption and Anomaly detection rate	Cooja Simulator	Yes
Iqbal et al.,2020 [27]	Anomaly based IDS	–	Multipath is used which increases the overhead in the network.	Detection rate, PDR and power consumption	Cooja Simulator	No

parent rank and hash value to detect the rank attackers. Upon receiving a DAO message, the root node compares the values in these fields with the values in the information table. If there is any mismatch between these values, then an alarm is generated. The proposed algorithm is elaborated in the following sections.

3.2. Modification of DAO message

In the proposed algorithm, the DAO message is used for detecting the rank attackers by appending extra fields to it. Target and Transit Information Options (TIO) in DAO message are also used for the detection of the rank attackers. The format of DAO message with options is shown in Fig. 5

Target option in DAO message is used for specifying the destination node that has to be reached. The final destination should be the IP address of the root node in non-storing mode. If the root node receives the DAO message with Target option not set to the IP address of it, then the root will discard the invalid DAO message. The above scenario occurs when an internal rank attacker fails to set the Target option, but reaches the root since the non-storing mode of RPL is considered. The target option is followed by TIO. This option is used for specifying the reachability of the destination and for extending the DAO message [29, 30]. Fig. 6 depicts the TIO option of the DAO message.

DAO messages are extended by adding extra fields such as rank (16-bits), parent rank (16 bits) and hash value (64 bits). Totally, 96 extra bits are added. Fig. 7 depicts the modified TIO option of the DAO message.

The rank and the parent rank values are obtained from DIO message and the hash value is calculated for rank, parent rank and IP address of the sender node by using the hash function.

3.3. Generation of hash value

RPL is a lightweight protocol used for energy-constrained networks [31]. So, integer hash function is used in the proposed work. The proposed integer hash function accepts 64-bits (Rank + Parent rank + IP address of the sender node of DAO message) as input and returns 64-bits as an output. Among the inputs to the hash function, IP address of sender node is unique. Since the IP address of the sender is unique for each node, the output produced by the hash function will also be unique. The pseudocode for the computation of hash value is given below.

x <- — $x \hat{} x >> 32$
x <- — x * 0xd6e8feb86659fd93U
x <- — $x \hat{} x >> 32$
x <- — x * 0xd6e8feb86659fd93U
x <- — $x \hat{} x >> 32$

If there is hash value collision, it will lead to same hash output. So, that the attackers can read or copy the hash value to perform more

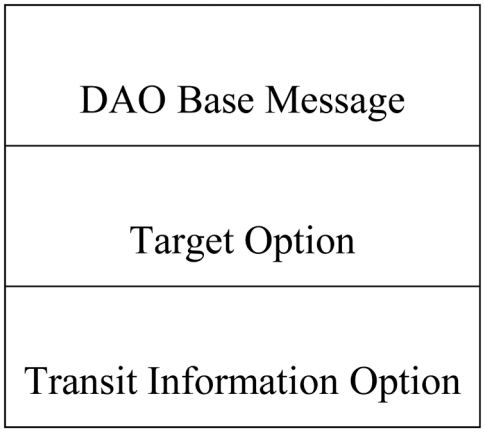


Fig. 5. DAO message with options.

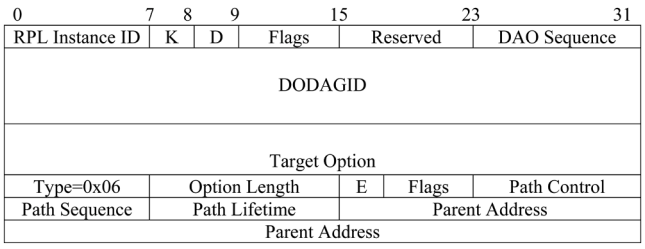


Fig. 6. TIO option of DAO message.

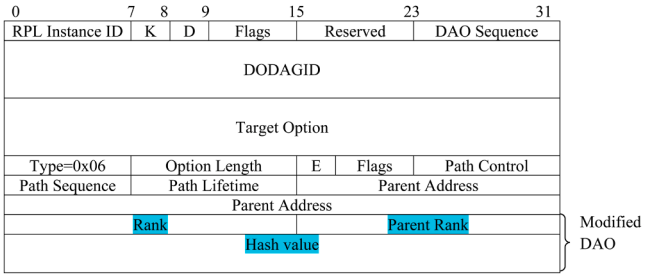


Fig. 7. Modified TIO of DAO message.

number of attacks and root node may wrongly identify a normal node as an attacker (False positive may occur) [32]. As the number of attacks increases, the generation of control packet also increases which depletes the energy of the nodes.

In the proposed hash function, the inputs and the output hash values are mapped in the ratio of 1:1. From this, it is inferred that there occurs no collision while using this hash function. Thus, the proposed integer hash function avoids collision. So, false positive would never occur.

The result of the hash function is included in the DAO message. When the root receives DAO message, it checks the received hash value with the values stored in the information table. If there is any mismatch in the hash value received, then the root confirms the presence of the rank attacker. To detect further whether it is an increased or decreased rank attacker, the detection procedure will be detailed below in Section 3.4.

3.4. Inconsistency check

Upon receiving DAO message, the root compares the received values with the values in the information table. The information table contains node ID (ID_N), Rank (Ra_N), Parent ID (ID_P), Parent Rank (Ra_P), Hash value and the ID of the blacklisted nodes (ID_M). In the proposed work, the behavior of the internal node is compromised to become an attacker. So, the initial data in the information table are valid and maintained by the root node. The topology considered in the proposed work is static and the OF is MRHOF, most of the data in the information table do not change often. If there is any change or updation in the DODAG, the root node alone is intimated by ICMP error message in non-storing mode. After intimation, based on the consequence of the topology change, the root node may carry out global or local repair mechanism. During repair mechanism, the root node will advertise the remaining nodes in the DODAG with DIO messages. DIO transmissions follow a Trickle timer to control the time of updation and construction of DODAG. According to RFC 6550, one inconsistency DAO message from a node is not considered a critical error. If consecutive packets are also detected as inconsistent from the same nodes, it may be considered as inconsistent. If there is any change in the parent set, repair counter value for the corresponding parent maintained in the root is incremented. The counter value is used to assess the impact of a topological change and assure the selection of siblings. If counter value is greater than the threshold (depends on the network size) in the Trickle interval, then only the values in the information table (maintained in the root) are updated. Otherwise it

is used for identification of attackers.

Inconsistent DAO messages can be received from parent or child node. If an inconsistent DAO message is received from the **parent node**, then the rank of the parent node is obtained from all its child nodes via information table. If a parent node has one or more child nodes, then the rank is verified among all the child nodes. The parent rank in its all child nodes should be the same (i.e., Parent rank = = claimed parent rank (child)). If the rank value of a node is less than the claimed parent rank value from the information table, then decreased rank attacker will be detected, but if the rank of a node is greater than the claimed parent rank value, then increased rank attacker will be detected. If inconsistent DAO message is received from the **child node**, then the verification of increased or decreased rank attack will be done based on RFC 6719 [33].

According to RFC 6719 standard, the rank of a child node can be a maximum of two values: **(i)** Rank of the parent set member who has the highest advertised rank, rounded up to the next higher rank value. **(ii)** Highest calculated rank in the parent set minus MaxRankIncrease. MaxRankIncrease is the maximum increase in the rank value that a node can advertise within the same version of DODAG. If the rank value in the received DAO message is greater than above two values, then the increased rank attacker will be detected.

Decreased rank attacker is detected if the rank of a node is less than the sum of parent rank and MinHopRankIncrease. According to RFC 6550 [34], MinHopRankIncrease is the minimum rise in the rank value between a node and any of its DODAG parents. This value is provisioned by DODAG root.

For example, consider a simple random topology with seven nodes as in Fig. 8. Among the seven nodes, root node 1 maintains the information table and node 3 is an internal rank attacker. For instance, if root node receives the DAO message from node 3 (Parent node of node 5 and 6), then the values in it are checked for inconsistency. If inconsistent DAO message is received, then the rank value is verified from the parent rank of node 5 and 6 in the information table. Assume that node 3 performs increased rank attack by increasing its rank value from 2 to 3. If the claimed parent node rank value (rank=3) is higher than the parent rank value of node 5 (rank=2) and 6 (rank=2) in the information table, then increased rank attack is detected. If the claimed parent node rank value is less than the value in the information table of node 5 and 6, then decreased rank attack is detected.

After the detection of internal increased or decreased rank attackers, local and/or global alarm is generated against the attackers (will be discussed in Section 3.5). The attacker node detected is isolated from the formation of DODAG and its ID is also inserted into information table. Then the information table is updated. If a root node receives the packet generated or forwarded from the blacklisted node, then the packets will be discarded. During repair mechanism, if there is any inconsistency detected then Rank- Error (R) flag is set. It is 1-bit flag to indicate whether a rank value is modified. After repair mechanism is completed, the inconsistency is verified and will be rectified.

ID_N	Ra_N	ID_P	Ra_P	Hash	ID_M
2	1	1	1	H_2	-
3	1	1	1	H_3	-
7	1	1	1	H_7	-
4	2	2	2	H_4	-
5	2	3	2	H_5	-
6	2	3	2	H_6	-

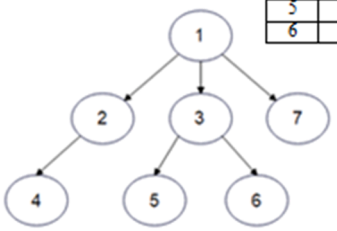


Fig. 8. Inconsistency Check.

3.5. Generation of alarm

After the detection of an increased or decreased rank attacker, the root node generates an alarm against the attacker in the network. To send the alarm, the root node uses the unused field in DIO message and inserts the IDs of the blacklisted nodes in these fields as shown in Fig. 9. The nodes that receive the alarm via modified DIO messages learn the IDs of the attackers from it. The alarm is included in DIO message itself to reduce the overhead of control packets in the network.

An alarm can be local or global depending on the repair mechanism of RPL. During global repair mechanism, root node generates the modified DIO message (global alarm) against the attacker to all the nodes in the network. During local repair mechanism of RPL, the modified DIO message (local alarm) is issued only to the neighbours. According to [35], the neighbours of a node are the subsets of nodes that are reachable during link local multicast in DODAG. The nodes that receive an alarm isolate the attacker from the formation of DODAG topology.

Let both local and global alarms be generated against the same malicious node at the same time, and only the global alarms be considered. If a normal node receives the local alarm first and subsequently the global alarm, the local alarm is ignored. If the global alarm is received first, then the global alarm is considered and the local alarm is discarded.

In non-storing mode, only the root node can generate local/global alarm to all the nodes directly. An attacker can try to generate an alarm but the values set in the options fields of the DIO messages are not known to the attackers. If the options set by the attackers were wrong, it is detected by the receiving node from the checksum field in the DIO messages. If the value of the checksum field is incorrect, then the receiving nodes will simply drop the DIO messages (alarm). Thus, a fake/false alarm generated by attackers has no consequence on the network.

In Fig. 10.a, a simple random topology with seven nodes is considered. Among the seven nodes, node 1 is a root node. Assume that there are no attackers in the DODAG initially. Later, greedy child node 6 turns into an attacker. Due to the presence of an attacker, there is a change in the topology. During local repair mechanism only certain nodes can receive the alarm based on local repair counter value in root node [36]. Other branches of the network are not impacted. If consequence of the topology change is less, it is not counterproductive to rebuild the entire DODAG for isolation of a single node. So in Fig. 10.b, during local repair mechanism, local alarm is sent in multicast only to certain nodes 3, 5 and 7 directly from the root. Thus, no extra control packet overhead is generated.

If node 3 in Fig. 10.a (which is close to the root) has turned into an attacker, it causes a massive change in the topology and so a large number of control packets are generated. Since the influence of an attacker (3) on the network is high, the global alarm is generated to all the nodes in the network as in Fig. 10.c. After detection of an attacker (3), the attacker is isolated from the DODAG topology. Eventhough node 3 is an attacker and isolated from DODAG, node 5 and 6 receives the alarm directly from the root node. Thus, the nodes under the coverage area of the root node can receive the alarm directly from it. The steps involved in the detection of rank attacker using RAD algorithm are summarised below.

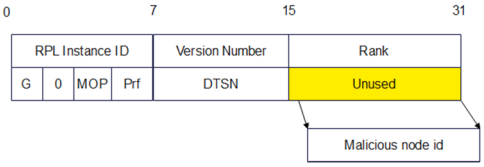


Fig. 9. Alarm in DIO message.

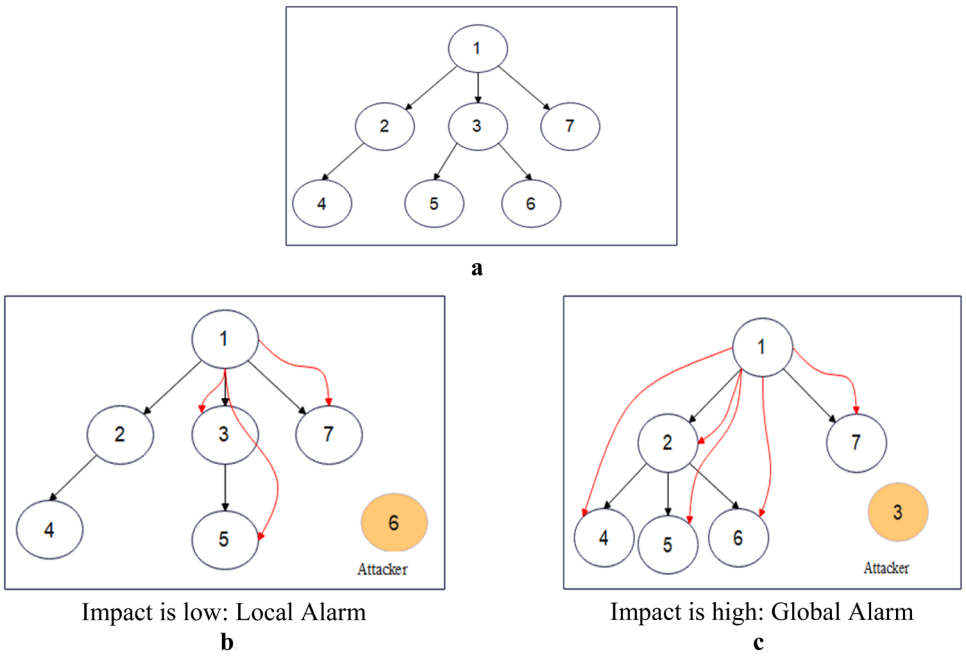


Fig. 10. a. DODAG topology without attackers b. Impact is low: Generation of Local Alarm c. Impact is high: .Generation of Global Alarm

3.6. RAD algorithm

Input: DODAG topology with Increased and Decreased Rank attackers
Algorithm:
Each node (N) appends Parent rank, Node rank and Hash value in DAO message
Root node after receiving the DAO message:
Checks the validity of the Target option in DAO message
if Target option = = IP address of Root Node in DAO message **then**
Retrieves ID (ID_N), Rank (Ra_N), Parent ID (ID_P), Parent Rank (Ra_P) and Hash value
if hash value = = hash value in the information table
Insert (ID_N, Ra_N, ID_P, Ra_P and Hash) into information table
else
// Check for Increased or Decreased Rank attack
Increase_Decrease( )
else
Discard invalid DAO message
end
Function: Increase_Decrease( ):
if ID_N = = Parent node **then**
Rank is claimed from the information table // Random Sampling
if (Ra_N < Claimed Rank from the information table)
Decreased Rank attacker is detected
alarm_generate( )
else
Increased Rank Attack is detected
alarm_generate( )
end
else // Child Node
Ra_{C1} = MinHopRankIncrease* (1+floor (Rank/MinHopRankIncrease)
Ra_{C2} = Largest Calculated Rank from the Parent set – MaxRankIncrease
$Ra_{C3} = Ra_P$ + MinHopRankIncrease
if ($Ra_N > Ra_{C1}$) || ($Ra_N > Ra_{C2}$) **then**
Increased Rank Attack is detected
alarm_generate( )
else if ($Ra_N < Ra_{C3}$)
Decreased Rank attacker is detected
alarm_generate ( )
end
end
Function : alarm_generate( ):
Invoke Local or global repair mechanism
Isolate attackers from DODAG
Insert the malicious node ID into Information Table
Update information Table
Output: Detection and isolation of Rank attackers in the topology

3.7. Random sampling of DAO messages

In this section, it is assumed that both increased and decreased rank attackers exist in the network. Both the rank attackers flood the DIO message with the increased and the decreased ranks respectively. In response to the flooded DIO messages, DAO messages will be generated by the nodes. In non-storing mode, all the generated DAO messages would be received by the root node. Instead of checking the inconsistency of all the DAO messages received from the nodes, random sampling is applied to check the inconsistency of the received DAO messages. If all the DAO messages are checked for inconsistency, energy consumption and detection delay by root node will be more. So, in this research, random sampling of DAO messages is adopted to check the inconsistency.

The communication between all the nodes and the DODAG root is assumed to be a Poisson processes. For analysing the probability of detecting the rank attackers, it is considered that all the attackers behave

Table 2
Parameters used in the rank attacker detection.

Parameters	Descriptions
N	Number of nodes in DODAG
y	Total number of attackers in the RPL network
R	Number of nodes through which the DAO message is passed on to the root node. R should always be less than N
P	Proportion of normal nodes in the network $p = \left(\frac{N-y}{N}\right)$
e_R	A Boolean event which means if the root node has sampled atleast one inconsistent DAO message
P_s	Probability that event e_R =TRUE has sampled atleast one inconsistent DAO message
P_d	Probability that an attacker is detected
e_F	A Boolean event which means an attacker has flooded more DIO messages to the neighbors
f	Number of DIO messages flooded by an attacker to neighbor nodes
n	Total number of messages received by each neighbor node
P_f	Percentage of flooded DIO packets sent to each of the neighbors i.e., P_f=f/n

independently. The parameters used in the rank attacker detection are presented in Table 2.

As in [37], if an attacker node floods the DIO message with the modified rank value, it is assumed to be received by 'R' nodes and is given by

$$P_d = P_s\left[1-\left(1-\frac{R}{N-1}\right)\cdot\left(1-p\left(\frac{R}{N-2}\right)\right)\left(1-p\left(\frac{R}{N-1}\right)PsPovp\right)\left(1-p2\left(\frac{R}{N-2}\right)PsPovp\right)\right. \quad (1)$$

Povp is negligible since it is assumed that Z is very small in reality while f is large. So, the equ (1) becomes

$$Pd = Ps\left[1-\left(1-\frac{R}{N-1}\right).\left(1-p\left(\frac{R}{N-2}\right)\right)\right] \sim Ps.\left(\frac{1+r}{N}\right) R$$

The number of DAO messages that the root node can sample is yZ. The confidence interval is 95%. So, the Z-value is 1.96 as in [38]. Table 3 presents the impact of number of samples being sampled with the detection delay and energy consumption.

From Table 3, it is inferred that the average delay and the energy consumption with random sampling decreases with increase in the number of attackers. As the number of inconsistent DAO message increases, there is a high probability for the sample to contain inconsistent DAO messages. The worst case in the detection of rank attacker is that the DAO sample does not contain the inconsistent DAO messages. It is noted that the average number of attackers detected increases with increase in number of attackers. Almost the number of attackers being detected is not affected by with and without random sampling. In [39], the authors have proved the effectiveness of random sampling. Fig. 11 represents the energy consumption at the root node with simulation parameters described in Table 4.

4. Results and discussion

To evaluate the efficiency of the proposed RAD algorithm, Cooja (Java based) Simulator is used, which works on open source Contiki OS. In this paper, Contiki 3.0 is used. It targets the IoT devices with low power, limited memory, less bandwidth and processing power.

4.1. Analysis of the proposed hash function

The proposed hash function is analysed using **avalanche effect** and **bias**. The avalanche effect is the number of bits that remains fixed on average when a single input bit is modified. For a good hash function, the avalanche effect should be greater than 50%. From [40], the avalanche effect of benchmark algorithms such as 3-DES,CR6, MARS and Blow fish is 30.8%, 47%, 50% and 48.4% respectively, but for the hash function used in the current study, the avalanche effect is **69.5**%.

Bias is the deviation of the calculated table from the ideal bias table as in [32]. For a good hash function, the bias should be as less as possible. The bias of the above used hash function is found to be **0.14252255**. Thus, the above used simple hash function is good with high avalanche effect and low bias value. This is the current need for RPL mechanism that

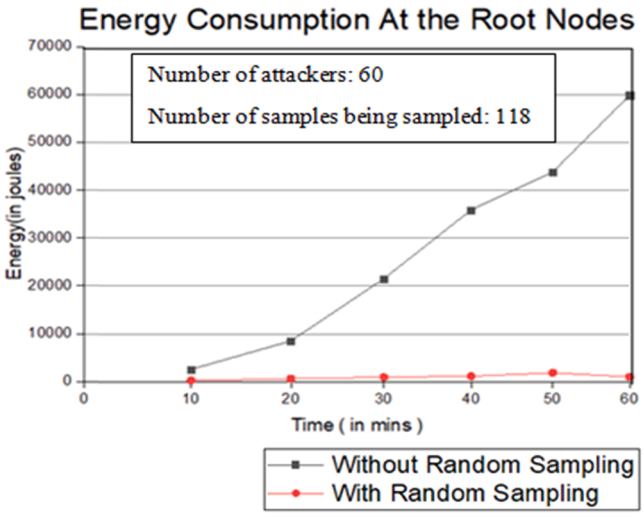


Fig. 11. Energy consumption with and without random sampling.

Table 4
Simulation Parameters.

Parameters	Values
Simulator	Cooja 3.0
Number of nodes	50 nodes (1 root node)
Node Type	Tmote Sky
Network Layer	RPL
MAC Layer	IEEE 802.15.4
Radio Medium	Unit Disk Graph Medium
Interference Range	100m
Transmission Range	50m
Topology dimension	200×200m
Simulation duration	60 min
Number of internal attackers observed	8 (4 increased rank attackers and 4 decreased rank attackers)
OF	MRHOF
Topology	Random/ Grid-center / Grid-random Topology

Table 3
Effectiveness of Random Sampling.

Number of attackers	Without Random Sampling			With Random Sampling			
	Average Detection delay (in *sec*)	Average Energy consumption(*in joules*)	Average Number of attackers detected	Average number of samples being sampled = yZ	Average Detection delay (in *sec*)	Average Energy consumption(in joules)	Average Number of attackers detected
10	3265	1587.69	8	20	2880	2896.51	6
20	4826	2769	17	40	2340	2679.45	14
30	5124	3184.75	26	60	2280	2543	23
40	6589	3698	36	78	2040	1623.31	37
50	6836	4156.95	44	98	1740	1358.69	44
60	7895	5996.6	56	118	1560	1023.41	55

operates in resource constrained based network.

Both avalanche and bias values are calculated at the same time. First, a 64×64 table is created and the values are initialised to zero. Rows indicate the input bits and the columns indicate the output bits. To track the association between the input bit (i) and the output bit (j), the row (i) and column (j) of the table are used. Each bit is flipped one at a time while iterating through entire 2^n inputs. If the output bit is flipped, then the corresponding element in the table will be incremented. Each element in the ideal table would be 2^{n-1}. It is found that for every input bit, an output bit is flipped atleast half the time. Finally, standard deviation is calculated to change the obtained output into a normalised value.

4.2. Simulation parameters

For simulation, static topology with Tmote Sky [41,42], a well known sensor motes are considered. Totally, 50 motes with one root are deployed in the region of 200×200 m.The transmission range of each mote is 50 m. The number of nodes that turns into the attackers is eight. Among the eight attackers, four perform increased rank attack and the other four perform decreased rank attack. The rank attackers are implemented based on the manipulation in Contiki RPL library. In simulation, the nodes send the temperature value to the root. The node delivers the sensed value via UDP packet periodically. Different topologies such as random, grid random and grid are considered for simulation. Table 4 lists the variables that are used in simulation.

Totally, 10 independent simulations are run and the average of all the values in each run is considered for plotting the graph. Different topology such as random (as in Fig 12.a), grid-center (as in Fig 12.b) and grid-random (as in Fig 12.c) are considered for the evaluation of RAD algorithm. In the figures below, the green color indicates root, yellow color indicates normal nodes and pink color indicates attackers.

4.3. Simulation results and analysis

The three types of performance metrics considered for the evaluation of RAD algorithm are (**1**) **Network Performance Metrics** (**2**) **Accuracy Metrics** and (**3**) **Resource Requirement Statistics**. The performance of RAD is compared with (i) SBIDS[17] (ii) SRPL-RP [43] and (iii) LEADER [23]. The RPL with and without attackers are also simulated for comparison.

4.3.1. Network performance metrics

The network performance metrics considered for the evaluation of RAD are:

(i) Average PDR of data
(ii) Overhead in the control message packets and
(iii) Average delay for the transmission of data packets

(i) PDR

The number of data packets received by a node divided by the number of packets transmitted is known as PDR. It is calculated at 15 min (during convergence stage), at 40 min (during stability stage) and at 60 min (during final stage). Among the topologies considered, the topology with less PDR is again used for the evaluation of RAD by increasing the number of attackers. The rationale behind choosing the topology with less PDR is to check whether PDR can be increased by implementing the proposed RAD algorithm. Fig. 13 depicts the simulation findings when different topologies are considered.

From the simulation result, it is clear that RAD algorithm is as good as RPL without attackers. This is true for all the topologies at different times. Even without attackers, RPL cannot achieve 100% of PDR. This is due to frequent retransmission of packets and the other MAC layer issues. The situation becomes even worse when RPL is targeted by attackers. In **grid-centered topology**, RAD algorithm achieves a PDR of 90.26% during convergence time, 91.52% during stability time and finally, 93.62% in the stabilized stage. In **random topology**, RAD algorithm achieves a PDR of 87.52% during convergence stage, 90.20% during stability time and 91.45% in the final stage. Likewise, in **grid-random topology**, RAD algorithm achieves a PDR of 92.14% during convergence stage, 93.18% in the stability time and 94.35% in the final stage. Thus, the PDR at the final stage in different topologies is summarised in Table 5.

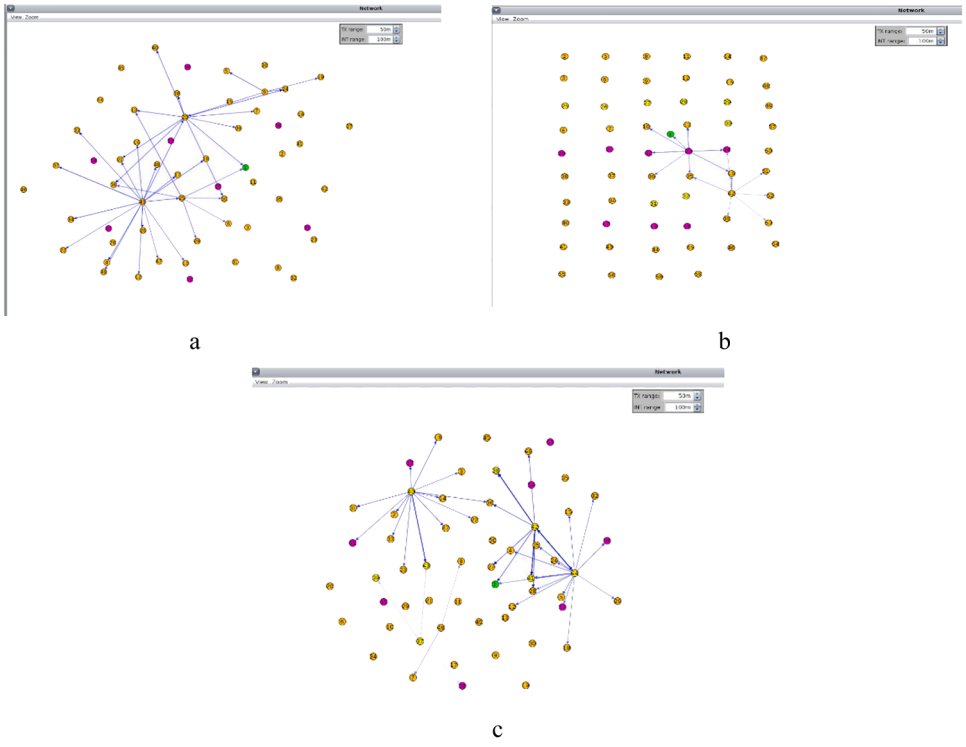


Fig. 12. a. Random Topologyb. Grid-center Topology c. Grid-random Topology.

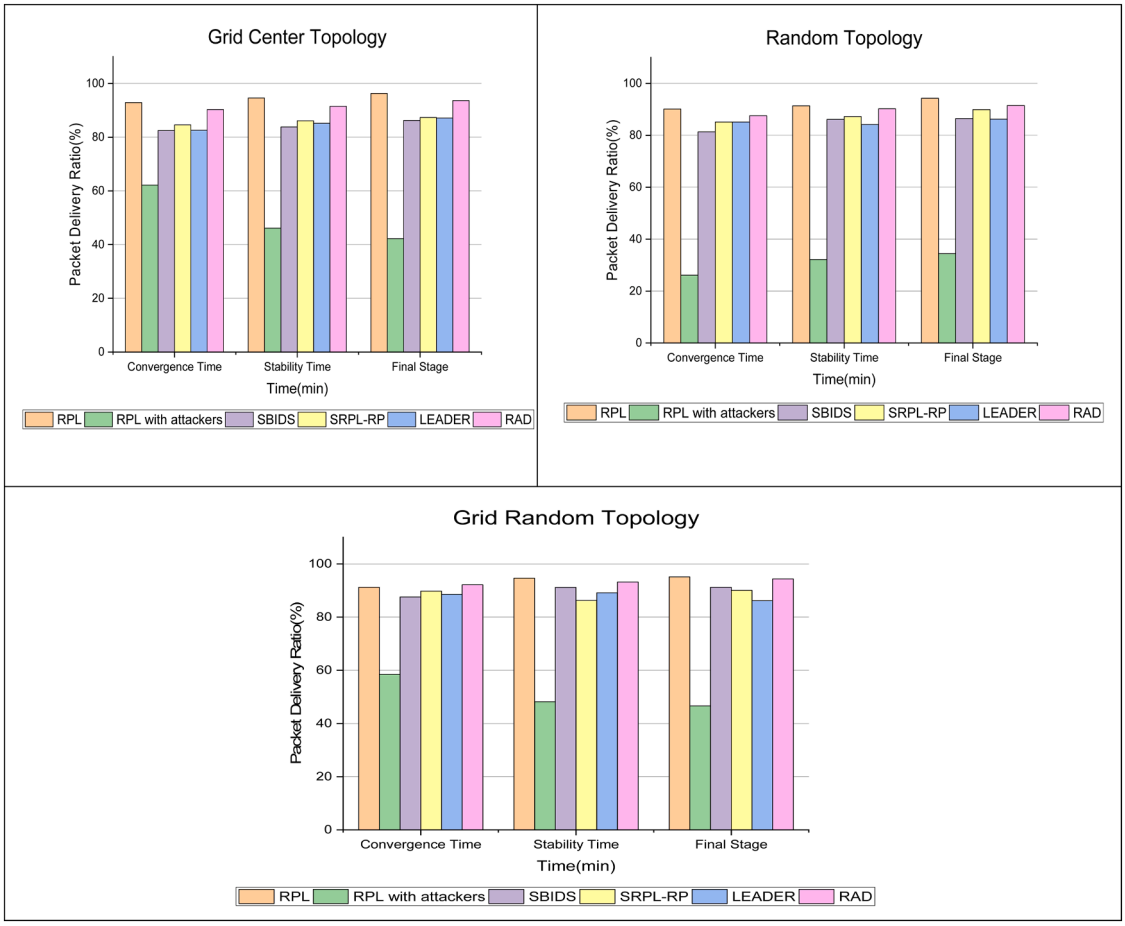


Fig. 13. Packet Delivery Ratio at different Topologies.

Table 5
PDR obtained at final stage for different topologies.

Mitigation Methods	Grid-center	Random	Grid-random
RPL with no attackers	96.23	94.26	95.12
RPL with attackers	42.21	34.41	46.58
SBIDS	86.21	86.45	91.25
SRPL-RP	87.34	89.89	90.06
LEADER	87.13	86.23	86.14
RAD (Proposed Algorithm)	**93.62**	**91.45**	**94.35**

Table 6
PDR versus number of malicious nodes.

No. of Malicious nodes	SBIDS	SRPL-RP	LEADER	RAD
8	91.05	90.06	86.14	**93.12**
16	88.45	89.53	84.26	**92.15**
24	89.36	84.52	82.89	**90.26**
32	79.23	80.32	80.18	**89.54**
40	78.89	79.56	78.95	**83.21**

As can be seen from Table 5, RAD algorithm achieves better PDR compared to RPL and the other algorithms. Among the topologies considered, random topology has a less PDR. This is due to the reason that most of the packets are dropped due to the routing errors caused by the increased and the decreased rank attackers and frequent reconstruction of DODAG. So, random topology is used for the evaluation of RAD algorithm by increasing the number of attackers. The same simulation parameters given in Table 4 are used, except the number of malicious nodes and the topology considered is random topology. The result obtained is presented in Table 5.

From Table 6, it is clear that in RAD and the other compared techniques, PDR decreases when the number of attackers increases. This is due to the existence of rank attackers, a portion of the sub-DODAG is isolated from the root node. So, the temperature values sent by the nodes do not reach the root node. From Table 5, it is inferred that RAD has a better PDR among the other compared techniques.

(ii) Overhead in control message packets

The number of control messages generated during the construction of DODAG topology is known as control packet overhead. Since the nodes are energy-constrained, reducing the amount of control packets required for the construction of DODAG is a tricky problem. The control packet overhead is calculated by

$$Control\ Packets\ overhead = \sum_{p=1}^{n} \mathrm{DIO(p)} + \sum_{r=1}^{s} \mathrm{DAO(r)} + \sum_{k=1}^{q} \mathrm{DIS(k)}$$

The number of control packets generated for different topologies is presented in Table 7.

Table 7 presents the control packet overhead for different topologies

Table 7
Control Packet Overhead in number of packets.

Mitigation Methods	Grid-center	Random	Grid-random
RPL with no attackers	1042	1363	1196
RPL with attackers	44,523	38,961	42,560
SBIDS	4062	3964	3304
SRPL-RP	2375	1998	2851
LEADER	3254	3652	3124
RAD (Proposed Algorithm)	**1060**	**1451**	**1212**

at the end of simulation (60 min). It is observed that the number of control packets generated by RPL without attackers is more in random topology than in the grid based topology. In random topology, if the nodes are placed in random fashion, the density of the placed node will not be uniform. The randomness with respect to the node placement decreases from the grid to the random topology. In grid-center and grid random topologies, the density of the node is almost uniform and so that a parent node can serve more child nodes. Thus, the child nodes can choose the optimized parent and form the DODAG with less number of control packets.

From Table 7, it is inferred that the RAD algorithm generates a less number of control packets compared to the other mitigation methods in all the topologies. The number of control packets generated by RAD is almost the same as the number of control packets generated by RPL with no attacker. This is because no extra control packets are used in RAD for the detection of rank attackers and the generation of alarm.

The control packet overhead is also calculated for RAD by increasing the number of malicious nodes in the network. The same simulation parameters presented in Table 4 are used, except the number of malicious nodes and the topology considered is grid-centered. Since the control packet overhead is high in Grid centered topology, it is considered here.

From Fig. 14, it is clear that in all the mitigation methods including RAD, the control packet overhead increases as the number of malicious node increases, but the rate of increase is less compared to the other methods. This is because the rank attackers detected would be isolated during the formation of DODAG topology.

(iii) Average delay for the transmission of data packets

The time taken for a packet to be generated until it is successfully received at the DODAG root is referred to as latency (delay). The total delay is calculated by

$$Total\ delay = \sum_{k=1}^{n} (\text{Received time(k)} - \text{Sent time(k)})$$

The average delay (in ms) for data packet transmission in different topologies is presented in Table 8.

From Table 8, it is inferred that the average delay is greater in random topology due to more control packet overhead and the longer path for the nodes. Due to the increase in the control packet overhead, the business of the channel also increases. Since the degree of densities is not uniform in random topology, there exists the longer path between the root and the nodes.

The average delay for the delivery of the data packets using RAD is 982 ms for Grid random topology, 854 ms for grid-center topology and 1054 ms for Random topology. Thus, RAD reduces the latency more efficiently. The results achieved from RAD are superior to those produced by SRPL-RP, which isolates the malicious node using a blacklist table. Since the malicious nodes are filtered, the generation of control packet is reduced which, in turn, reduces the average delay.

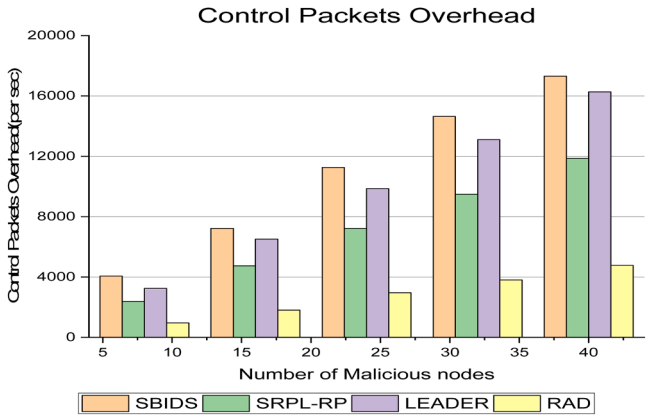


Fig. 14. Control Packet Overhead versus number of malicious nodes.

Table 8
Average delay for data packet transmission (in ms).

Mitigation Methods	Grid-center	Random	Grid-random
RPL	869	898	1024
RPL with attackers	2869	3985	2984
SBIDS	982	1231	1095
SRPL-RP	962	1153	996
LEADER	998	1224	1098
RAD (Proposed Algorithm)	**854**	**1054**	**982**

4.3.2. Accuracy metrics

The ratio of the number of nodes successfully identified as legitimate or malicious to the total number of nodes in the network is known as accuracy. The accuracy is calculated by

$$Accuracy = \frac{True\ Positive(TP) + True\ Negative(TN)}{TP + TN + False\ Positive(FP) + False\ Negative(FN)}$$

The accuracy of the mitigation methods for different topologies is presented in Fig. 15.a. From Fig. 15.a, it is clear that the accuracy of the Grid-centered topology is high compared to random topology. This is due to the high densities of the node in the grid-center topology. All the nodes including the attacker receive DIO message and reply back with the DAO message. From the DAO message received, the inconsistency in the rank can be easily detected. In random topology, the nodes are scattered randomly and due to the presence of the rank attackers, it has been noticed that a portion of the network becomes separated from the root. The malicious nodes do not receive the DIO message and so they do not reply back with the DAO message. Hence the detection accuracy is less in random topology compared to the other two topologies.

Since random topology has a lower accuracy rate, the efficiency of RAD is proved by increasing the number of attackers. The accuracy of the rank attacker detection with varying number of rank attackers in Random topology is presented in Fig. 15.b. If the number of rank attackers considered for simulation is 8, then the number of increased rank attackers will be 4 and the decreased rank attackers will be 4.

The accuracy of the RAD along with different mitigation methods are depicted in Table 9.

From Table 9, it is inferred that the detection rate of the rank attacker is reduced as the number of malicious nodes is increased. Due to the presence of the rank attackers, a portion of the nodes in the network is unable to contact their root node. So, the attackers presented in the disconnected part of the network are not detected. False positive is detected due to the decreased rank attackers in the network. The decreased rank attackers advertise the shortest path to their neighbor nodes to reach root nodes. From Table 8, it is clear that RAD shows the improved rate of accuracy than the other mitigation methods. This is due to the presence of more number of attackers, and the DIO packets generated by the attackers would also be more. Inconsistent DAO messages would also increase. So, when the root node samples the DAO message, the detection of inconsistency in the rank also increases.

4.3.3. Resource requirement statistics

The average energy consumption of the nodes is the resource requirement considered for analysis. The energy consumption is considered as it is a very pivotal resource in LLN.

Average Energy Consumption of nodes

Power Trace Plugin is used in Cooja Simulator to trace the energy consumption of the nodes. The energy consumed is categorized as CPU energy (Total energy consumption by CPU), LPM energy (Energy consumption of nodes in power saving mode), Radio Transmit and Listen energy (Energy used to transmit and receive the packets). In the above categories, the pairs Transmit and Listen, CPU and LPM are exclusive. Thus, the nodes can never be in all the modes at the same time. All these

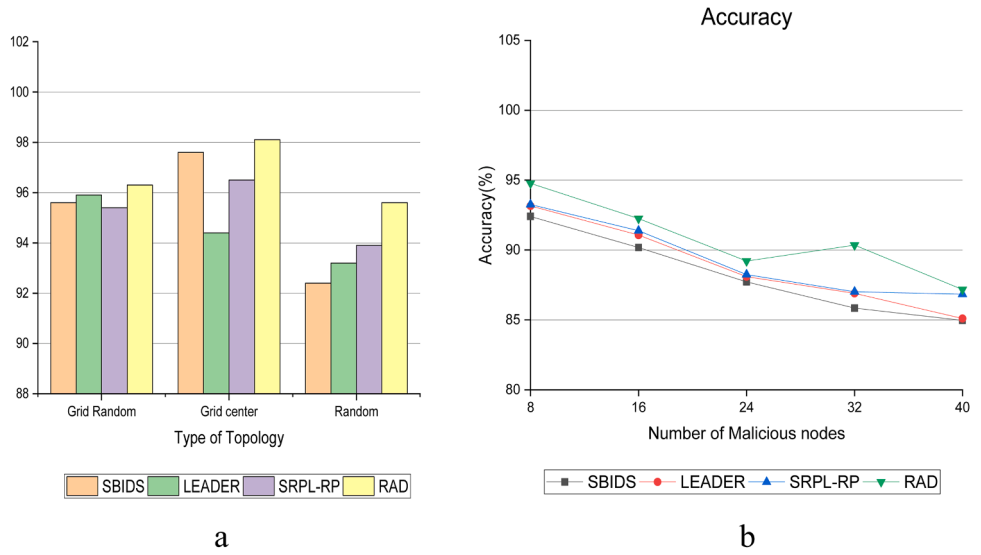


Fig. 15. a.Accuracy in different Topologies b.Accuracy vs. Malicious Nodes.

Table 9
Accuracy of the Mitigation Methods.

Mitigation Methods	Number of Normal Nodes	Number of attackers	Average number of (nodes) TP	TN	FP	FN	Accuracy (%)
SBIDS	**42**	**8**	40	4	2	4	88
	34	**16**	32	12	2	4	88
	26	**24**	23	19	3	5	84
	18	**32**	12	28	6	4	80
	10	**40**	8	31	2	9	78
LEADER	**42**	**8**	41	7	1	1	96
	34	**16**	33	14	2	2	94
	26	**24**	24	21	2	3	90
	18	**32**	12	30	6	2	84
	10	**40**	7	34	3	6	82
SRPL-RP	**42**	**8**	41	6	1	2	94
	34	**16**	33	12	1	4	90
	26	**24**	24	20	2	4	88
	18	**32**	11	30	7	2	82
	10	**40**	7	33	3	7	80
RAD	**42**	**8**	42	6	0	2	96
	34	**16**	33	14	1	2	94
	26	**24**	24	21	2	3	90
	18	**32**	15	29	3	3	88
	10	**40**	7	35	3	5	86

Table 10
Average energy usage (in joules) for various topologies.

Mitigation Methods	Grid-center	Random Topology	Grid-random
RPL without attackers	1433	1025	1169
RPL with attackers	3795	3128	3245
SBIDS	2859	2664.52	2762.43
SRPL-RP	2612	2412	2467.12
LEADER	2679.45	2442.1	2543
RAD (Proposed)	1623.921	1325.146	1486

modes of energy consumption are measured internally at run time by Contiki. Table 10 shows the average energy consumption of the nodes for data packet transmission in various topologies.

From Table 10, it is clear that the Grid-centered topology consumes more energy because in it, the nodes are placed at equal distance close to the root. Since many nodes are in transmission range, there is a frequent exchange of packets. This leads to more energy consumption. Moreover, due to the path diversity caused by attackers, the energy consumption is high. RAD also reduces the average energy consumption by approximately 40% and shows less energy consumption in random topology.

The average energy consumption of RAD is less because a light-weight non-cryptographic hash function is used in it. From Table 9, it is inferred that SBIDS consumes more energy. This is due to the deployment of AES which is computationally-intensive. LEADER employs HMAC using LOCHA which is also computationally-intensive, but less compared to SBIDS. It is also discovered that as the number of malicious nodes rises, the average energy usage of the node increases.

Compared to the results provided in the literatures (SBIDS, SRPL-RP and LEADER), the proposed RAD algorithm consumes less energy. The energy consumed by RAD algorithm is slightly higher than normal RPL (without attackers). This is due to hash value calculation, inconsistency detection and so on. The average energy consumption of RAD is reduced far better than the RPL with attackers. This is due to the isolation of detected rank attackers from the DODAG topology and an alarm is also generated against the attackers. So, that the internal rank attacker nodes cannot become a member of DODAG and cause disruptions. Table 11 presents the summary of the performance of RAD algorithm.

5. Conclusion and future work

The integration of IoT into various applications offers significant benefits. Along with the benefits, it is prone to various attacks. Therefore security is a prime concern to improve the performance of the IoT network. The proposed energy-efficient RAD algorithm is specially developed to improve the secure data communication between DODAG root and the remaining nodes. This algorithm is incorporated into the RPL protocol to improve the performance and to detect the rank attackers. The proposed RAD algorithm guarantees security against the internal increased and decreased rank attacks in RPL based IoT networks. Non-cryptographic hash function is used to preserve the integrity of DAO messages exchanged during the formation of DODAG. Instead of checking all the received DAO messages to detect the rank attackers, Random Sampling is used to conserve energy. After the detection of rank attackers, local or global alarm is generated against the attackers to the normal nodes. Alarm is appended in the unused fields of DIO message. Thus, it reduces the overhead of control packets and depletion of energy. The performance of RAD algorithm is evaluated using Cooja Simulator under various topologies. It gives 94.35% of packet delivery rate in grid random topology, reduces control packet overhead from 44,523 packets to 1060 packets in grid centered topology and reduces average delay from 2869 ms to 854 ms in grid-centered topology. It also achieves the

Table 11
Summary of the Performance of RAD algorithm.

Network Performance Metrics									
Mitigation Methods	**Average PDR**			**Average Control Packet Overhead** (in number of control packets)			**Average Delay** (in ms)		
	Grid-center	Random	Grid-random	Grid-center	Random	Grid-random	Grid-center	Random	Grid-random
RPL without attackers	96.23	94.26	95.12	1042	1363	1196	869	898	1024
RPL with attackers	42.21	34.41	46.58	44,523	38,961	42,560	2869	3985	2984
SBIDS	86.21	86.45	91.25	4062	3964	3304	982	1231	1095
SRPL-RP	87.34	89.89	90.06	2375	1998	2851	962	1153	996
LEADER	87.13	86.23	86.14	3254	3652	3124	998	1224	1098
RAD	**93.62**	**91.45**	**94.35**	**1060**	**1451**	**1212**	**854**	**1054**	**982**
		Increased			Decreased			Decreased	
Resource Requirement Statistics and Accuracy Metrics									
Mitigation Methods	**Average Energy Consumption** (in joules)			Accuracy (%)					
	Grid-center	Random	Grid-random	Number of Normal Nodes : 42					
RPL without attackers	1433	1025	1169	Number of attackers : 8					
RPL with attackers	3795	3128	3245	Grid-center	Random	Grid-random			
SBIDS	2859	2664.52	2762.43	97.6	92.4	95.6			
SRPL-RP	2612	2412	2467.12	94.4	93.2	95.9			
LEADER	2679.45	2442.1	2543	96.5	93.9	95.4			
RAD	**1623.921**	**1325.146**	**1486**	**98.1**	**95.6**	**96.3**			
		Decreased			Increased				

accuracy of 96% and the average energy consumption is reduced by 40% in random topology in comparison with SBIDS [17], SRPL-RP[18] and LEADER [23]. The simulation results show the effectiveness of RAD algorithm.

For the evaluation of the proposed algorithm, static topology is considered and it is assumed that a greedy or disgruntled node may turn into an attacker and inject an increased and decreased rank attack. This would not be the only chance for launching rank attacks. The attackers may also newly join and launch the attacks. In several IoT based applications, support for mobility is also demanded. Hence, as a future work, the proposed RAD algorithm can be extended to support the mobility of nodes and detect various other attacks like version, sinkhole and so in IoT networks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] H.-S. Kim, J. Ko, D.E. Culler, J. Paek, Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): a survey, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2502–2525.

[2] A. Verma, V. Ranga, Security of RPL based 6LoWPAN networks in the internet of things: a review, IEEE Sens J 20 (11) (2020) 5666–5690.

[3] A. Ghotbou, M. Khansari, Comparing application layer protocols for video transmission in IoT low power lossy networks: an analytic comparison, Wirel. Netw. 27 (1) (2021) 269–283.

[4] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: applications, challenges, and open problems, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2334–2360.

[5] H. Bouzebiba, M. Lehsaini, FreeBW-RPL: a new RPL protocol objective function for internet of multimedia things, Wirel. Pers. Commun. (2020) 1–21.

[6] H. Lamaazi, N. Benamar, A comprehensive survey on enhancements and limitations of the RPL protocol: a focus on the objective function, Ad Hoc Netw. 96 (2020), 102001.

[7] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, M. Chai, The impact of rank attack on network topology of routing protocol for low-power and lossy networks, IEEE Sens. J. 13 (10) (2013) 3685–3692.

[8] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, A. Durresi, Routing loops in dag-based low power and lossy networks, 2010, in: 24th IEEE International Conference on Advanced Information Networking and Applications, IEEE, 2010, pp. 888–895.

[9] W. Choukri, H. Lamaazi, N. Benamar, RPL rank attack detection using Deep Learning, in: 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), IEEE, 2020, pp. 1–6.

[10] A. Agiollo, M. Conti, P. Kaliyar, T. Lin, L. Pajola, DETONAR: detection of routing attacks in RPL-based IoT, IEEE Trans. Netw. Serv. Manag. (2021).

[11] S.R. Boualam, A. Ezzouhairi, New objective function for RPL protocol. Embedded Systems and Artificial Intelligence, Springer, 2020, pp. 681–690.

[12] O. Iova, P. Picco, T. Istomin, C. Kiraly, Rpl: the routing standard for the internet of things... or is it? IEEE Commun. Mag. 54 (12) (2016) 16–22.

[13] S.-W. Min, S.-H. Chung, H.-J. Lee, Y.-V. Ha, Downward traffic retransmission mechanism for improving reliability in RPL environment supporting mobility, Int. J. Distrib. Sens. Netw. 16 (1) (2020), 1550147720903605.

[14] P.S. Nandhini, B. Mehtre, Directed acyclic graph inherited attacks and mitigation methods in RPL: a review, in: International Conference on Sustainable Communication Networks and Application, Springer, 2019, pp. 242–252.

[15] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, A. Zanella, IoT: internet of threats? A survey of practical security vulnerabilities in real IoT devices, IEEE IoT J. 6 (5) (2019) 8182–8201.

[16] H.A. Abdul-Ghani, D. Konstantas, A comprehensive study of security and privacy guidelines, threats, and countermeasures: an IoT perspective, J. Sensor Actuator Netw. 8 (2) (2019) 22.

[17] U. Shafique, A. Khan, A. Rehman, F. Bashir, M. Alam, Detection of rank attack in routing protocol for Low Power and Lossy Networks, Ann. Telecommun. 73 (7) (2018) 429–438.

[18] Z.A. Almusaylim, A. Alhumam, N. Jhanjhi, Proposing a secure RPL based internet of things routing protocol: a review, Ad Hoc Netw. 101 (2020), 102096.

[19] A.O. Bang, U.P. Rao, EMBOF-RPL: improved RPL for early detection and isolation of rank attack in RPL-based internet of things, Peer-to-Peer Netw. Appl. 15 (1) (2022) 642–665.

[20] A. Hkiri, M. Karmani, M. Machhout, The routing protocol for low power and lossy networks (RPL) under Attack: simulation and Analysis, in: 2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET), IEEE, 2022, pp. 143–148.

[21] A. Seyfollahi, M. Moodi, A. Ghaffari, MFO-RPL: a secure RPL-based routing protocol utilizing moth-flame optimizer for the IoT applications, Comput. Standard. Interfaces 82 (2022), 103622.

[22] S.M. Muzammal, R.K. Murugesan, N.Z. Jhanjhi, L.T. Jung, SMTrust: proposing trust-based secure routing protocol for RPL attacks for IoT applications, in: 2020 International Conference on Computational Intelligence (ICCI), IEEE, 2020, pp. 305–310.

[23] S. Karmakar, J. Sengupta, SD LEADER Bit, Low overhead rank attack detection for securing RPL based IoT, in: 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2021, pp. 429–437.

[24] V. Kiran, S. Rani, P. Singh, Towards a light weight routing security in IoT using non-cooperative game models and dempster–shaffer theory, Wirel. Pers. Commun. 110 (4) (2020) 1729–1749.

[25] A.D. Seth, S. Biswas, A.K. Dhar, Detection and verification of decreased rank attack using round-trip times in RPL-based 6LoWPAN networks, in: 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), IEEE, 2020, pp. 1–6.

[26] A.M. Said, A. Yahyaoui, F. Yaakoubi, Abdellatif T Machine learning based rank attack detection for smart hospital infrastructure, in: International Conference on Smart Homes and Health Telematics, Springer, 2020, pp. 28–40.

[27] M.M. Iqbal, A. Ahmed, U. Khadam, Sinkhole attack in multi-sink paradigm: detection and performance evaluation in RPL based IoT, in: 2020 International

Conference on Computing and Information Technology (ICCIT-1441), IEEE, 2020, pp. 1–5.
[28] P.S. Nandhini, S. Kuppuswami, S. Malliga, Energy efficient thwarting rank attack from RPL based IoT networks: a review, in: Materials Today: Proceedings, 2021.
[29] M. Mahyoub, A.S.H. Mahmoud, M. Abu-Amara, T.R. Sheltami, An Efficient RPL-based Mechanism for Node-to-Node Communications in IoT, IEEE IoT J. 8 (9) (2020) 7152–7169.
[30] C. Pu, Sybil attack in RPL-based internet of things: analysis and defenses, IEEE IoT J. 7 (6) (2020) 4937–4949.
[31] K. Prathapchandran, T. Janani, A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST, Comput. Netw. 198 (2021), 108413.
[32] Skeeto (April 2020). https://github.com/skeeto/hash-prospector.
[33] Gnawali O., Levis P. (2012) The minimum rank with hysteresis objective function.
[34] Winter T., Thubert P., Brandt A., Hui J., Kelsey R., Levis P., Pister K., Struik R., Vasseur J.-P., Alexander R. (2012) RPL: iPv6 routing protocol for low-power and lossy networks.
[35] M. Pushpalatha, T. Anusha, T.R. Rao, R. Venkataraman, L-RPL: RPL powered by laplacian energy for stable path selection during link failures in an Internet of Things network, Comput. Netw. 184 (2021), 107697.
[36] K.D. Korte, A. Sehgal, J. Schönwälder, A study of the RPL repair process using ContikiRPL, in: IFIP International Conference on Autonomous Infrastructure, Management and Security, Springer, 2012, pp. 50–61.
[37] H. Lin, S. Sun, Estimation for networked random sampling systems with packet losses, IEEE Trans. Syst. Man Cybernetic. 51 (9) (2019) 5511–5521.
[38] S.K. Aityan, 2022) Confidence intervals, Bus. Res. Methodol. (2022) 233–277. Springer.
[39] H. Sun, J. Sun, J. Chen, Analysis and synthesis of networked control systems with random network-induced delays and sampling intervals, Automatica 125 (2021), 109385.
[40] S. Aljawarneh, M.B. Yassein, WaA Talafha, A multithreaded programming approach for multimedia big data: encryption system, Multimed. Tools Appl. 77 (9) (2018) 10997–11016.
[41] D.M. Kumar, A. Ghosh, Resource efficient routing in internet of things: concept, challenges, and future directions, Int. J. Comput. Digit. Syst. 8 (6) (2019).
[42] A.W. Abbas, S.N.K. Marwat, Scalable emulated framework for IoT devices in smart logistics based cyber-physical systems: bonded coverage and connectivity analysis, IEEE Access 8 (2020) 138350–138372.
[43] Z. Almusaylim, N. Jhanjhi, A. Alhumam, Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP, Sensors 20 (21) (2020) 5997.



P.S. Nandhini, currently pursuing PhD in Internet of Things. She has authored two books titled "Wireless Network" with ISBN: 978–1–63,485–178–7 and "Internet of Things" with ISBN: 978–93–88,982–47–4(HB). She has published many papers in International Conferences and Journals. She is a Gold Medalist in ME. She has undergone Summer Research Fellowship sponsored by IISC.