The Journal of Supercomputing
https://doi.org/10.1007/s11227-022-04921-6

Enhanced Rank Attack Detection Algorithm (E-RAD) for securing RPL-based IoT networks by early detection and isolation of rank attackers

P. S. Nandhini[1] · S. Kuppuswami[2] · S. Malliga[1] · R. DeviPriya[3]

Accepted: 30 October 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Internet of Things (IoT) is becoming a crucial requirement for linking numerous physical devices or gadgets to the web. Due to large volume of data exchanged, security in IoT still remains a key challenge. Additionally, maintaining security in routing protocols like Routing Protocol for Low-power and Lossy networks (RPL) is becoming more difficult because IoT devices are heterogeneous and resource-constrained. RPL builds a tree topology using control packets. Since more control packets are exchanged, modification in the content of control packets gives rise to a number of attacks including rank attacks, version attacks, flood attacks and so forth. Rank attack is a major attack that causes loop creation, energy depletion, decreased packet delivery, increased packet delivery latency and other problems. Since rank attacks have a significant adverse effect on the network, an effort has been made to enhance the default mechanism of RPL and so that the rank attackers can be quickly identified and isolated. The proposed E-RAD (Enhanced-Rank Attack Detection) algorithm deploys rate limiting technique to control the generation of DIO (DODAG Information Object) packet and also uses DIS (DODAG Information Solicitation) message solicitation to isolate the rank attackers. If an attacker is missed using above mechanism, it can be detected from the consistency check of hash value in DAO (Destination Advertisement Object) message. An alarm is set off against the rank attackers as soon as it is found. An alarm is an attachment to the control message itself and is not a standalone packet. E-RAD algorithm resulted in 1356 J of energy usage, 775 ms of packet delivery delay, 95.53% of delivery ratio, 97.23% of accuracy and 1023 packets of overhead in control packet generation.

Keywords IoT · RPL · Rank attack · Control packets · Hash · Alarm

✉ P. S. Nandhini
nandhinisv1@gmail.com

Extended author information available on the last page of the article

Published online: 16 November 2022

1 Introduction

IoT refers to the scenario in which the objects, things or sensors are provided with computing and network connectivity capabilities. Therefore, the objects can generate, transfer sensed data with no human intervention. The objects in LLN are constrained in terms of resources and should be able to be accessed from anywhere. There are large number of applications using IoT. It includes monitoring of environment, automation of home and industry, smart city and farming, tracking of shipment and so on [1]. Thus, IoT is essential to every aspect of human life [2]. But in the real world, the installation of IoT necessitates secure communication [3]. It is difficult to provide secured communication due to heterogeneity of devices. Some devices are powerful and others are resource constrained. It is also essential that the communication between the devices should be secured end-to-end. Researchers are working on a lightweight technique and protocols for securely connecting resource constrained nodes in LLN to the Internet.

RPL is one of the standardized protocols for Low Power Lossy Networks (LLNs) [4]. It builds a tree topology called Destination Oriented Directed Acyclic Graph (DODAG) for establishing communication between nodes. The root node uses DIO, DAO, DAO-ACK and DIS control packets to build the DODAG. Root initiates the DODAG construction by periodically broadcasting the DIO messages. The DAO is returned by the node that received the DIO message. The DAO-ACK packet is used by the root to send the acknowledgement. If DIO is not received by a node, then it solicits the DIS message. Nearby root nodes reply with the DIO message. After receiving DIO messages, the node selects its root based on the Objective Function (OF) [5].

To ensure loop free DODAG, each node has been given a rank value. Rank value is the number of hops to reach a node from its root. It increases in the downward direction as in Fig. 1. Each node in the network selects the parent node with the least rank value. RPL does not examine the nodes' behaviour since it presumes that all of the nodes are trustworthy. As a result, a newly joined attacker can join the DODAG and manipulate the rank value. Attackers can increase or decrease the value of rank in DIO to disrupt

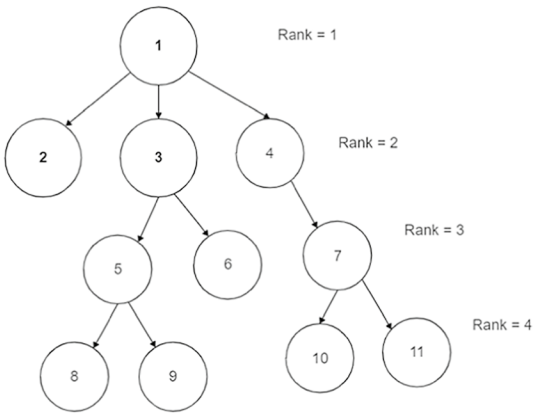


Fig. 1 Rank value

the DODAG. It leads to adverse effects such as (i) Formation of loops (ii) Unoptimized routing (iii) Decrease in Packet Delivery Ratio (iv) Increased DIO message generation (v) Unrecognised route and so on [6]. Therefore there is an urge to detect and isolate these attackers in DODAG topology.

Numerous researchers have looked into RPL's built-in security features and cryptographic defences against rank attacks (both increased and decreased) [7]. But, this resulted in computing overhead, which consumes more energy. This goes against RPL policy. Therefore, the primary goal of the proposed algorithm is to strengthen the default mechanism in RPL for identifying and isolating the rank attackers. For this, E-RAD algorithm is proposed, which finds and isolates the newly joined rank attackers in the DODAG topology. The main goal of the proposed algorithm is to ensure stable and secure data transfer between various nodes in the well-known IoT networks.

Proposed work answers the following questions.

1. Can the default mechanism in RPL be improved to identify and isolate the newly joined rank attackers from the DODAG?
2. Is it possible to detect the newly introduced attackers at the early stage (during transmission of control packets itself) before the transmission of data packets?
3. Can the existing tables maintained by the nodes be modified to include the black-listed nodes instead of separately maintaining the tables?

To answer the first question, in the proposed E-RAD algorithm, *rate limiting technique* of DIO generation and *solicitation of DIS messages* are used to detect the rank attackers. This is not a new but an improvised mechanism of RPL.

If newly introduced rank attackers are not detected by the above mechanisms, then in E-RAD algorithm the existing control packets and (DODAG) tables are modified to identify and isolate the attackers. This serves as an answer for second and third questions.

Our contributions of the paper are:

- Improving the default mechanism of RPL to identify and isolate the rank attackers
- Reducing the control packet overhead by identifying the attackers at the initial stage of construction of DODAG
- Reducing the computational overhead and energy usage at the root by using simple non-cryptographic hash function (32 bits)

The remaining of the paper is organised as follows: Sect. 2 briefs on background and existing systems. Proposed work is detailed in Sect. 3. In Sect. 4, the results are presented and analysed. Conclusion and future works are discussed in Sect. 5.

2 Background and existing systems

2.1 Background

RPL protocol has two operational modes: (i) storing & (ii) non-storing mode. When RPL is operating in storing mode, the state information is transmitted hop by hop. The routing table is kept up to date by each node in the DODAG. Therefore, it is not necessary to initiate DAO updates from every sub-DODAG. If RPL is operating in non-storing mode [8], the root node is responsible for maintaining the routing table. Each node uses a DAO message to inform its parents to the root. The communications are routed downward by utilizing IP source routing. Non-storing mode of RPL is taken into consideration in the proposed work.

An RPL Instance is comprised of a single DODAG with a root or several DODAGs with numerous root nodes. DODAGs in an instance share the same OF [9]. OF specifies the selection of the route by the nodes in RPL Instance. In the proposed work, *Minimum Rank with Hysteresis* (MRHOF) is chosen because it allows the selection of low-cost route which does not allow frequent changes of the paths.

Due to limited resources of LLN devices and the changes in the values of the control packets causes numerous routing attacks. In [10], the authors have classified the attacks based on the origin as (i) DAG and (ii) Topology inherited attacks. The DAG inherited attacks category includes rank attack. The rank attacker modifies the value of the rank in the DIO message to carry out decreased or increased rank attack in an effort to convince neighbouring nodes to relay the data packets through them. As a result, a loop forms and resources are wasted.

2.1.1 Decreased rank attack

Decreased rank attackers promote a lower rank value in DIO message to the other DODAG nodes. So that, majority of the nodes in the DODAG will choose the attackers as their preferred parents. In [11], the authors have found that due to decreased rank attack, more number of nodes have reached the root node via attackers. This paves the way for eavesdropping and manipulation of large part of the network traffic [12]. In Fig. 2a, consider a topology with one root node and 10 normal nodes (totally 11 nodes). A newly introduced rank attacker (A) causes the decreased rank. The rank value of the attacker is forged to be 2. So, that all the nearby nodes (8, 9, 10, 11) assume that attacker is close to the root and chooses node A as their preferred parent. Thereby, all the packets are routed to the root node via attacker. Thus, the attacker gains the information about the majority of nodes and has greater impact on the topology as in Fig. 2b.

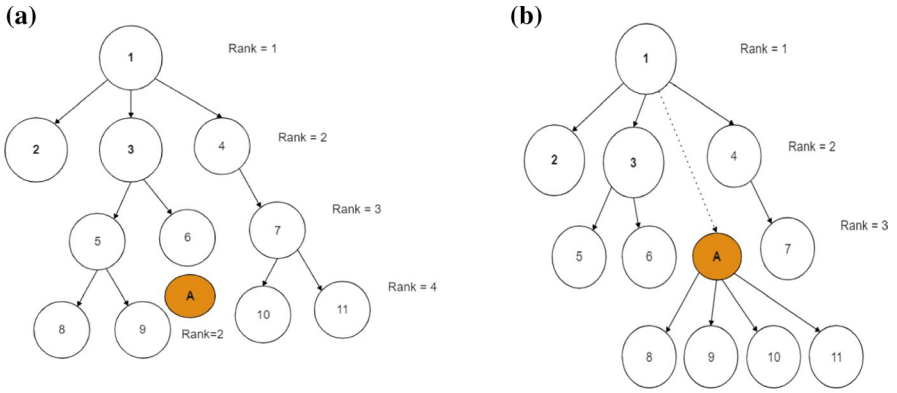


Fig. 2 **a** Decreased rank attacker. **b** Topology change

2.1.2 Increased rank attack

In an increased rank attack, the attacker promotes the higher rank value to sway the neighbours to select a new parent. The attacker descends deeper into DODAG which causes the child of the attacker in its sub-DODAG to search for a different preferred parent. This leads to the loop formation [13].

In Fig. 3a, consider a DODAG topology comprising of 11 nodes (1 root node and 10 normal nodes) and an attacker (A) deployed in random topology. Rank attacker (A) modifies the rank value as 5 in DIO and sends it to its neighbours. Since attacker A is close to node 10 (Rank = 4) and 6 (with Rank = 5), it chooses both the nodes (6 and 10) as its preferred parents. Thus, the attacker (A) has two parents (6 and 10) which lead to the formation of loop as in Fig. 3b.

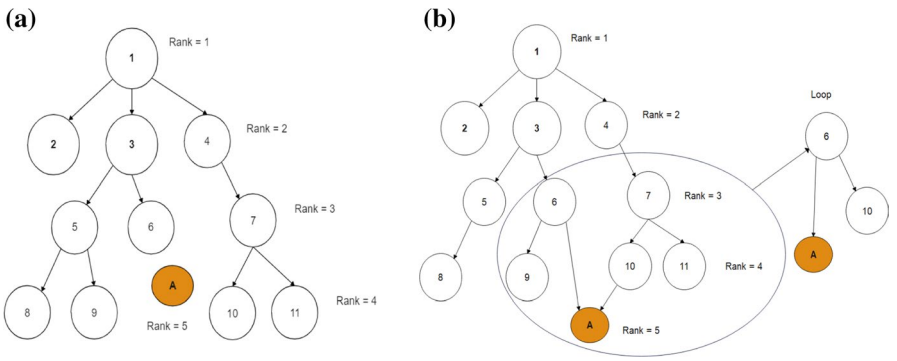


Fig. 3 **a** Increased rank attack. **b** Topology change-loop formation

2.2 Existing systems

In light of the advances in IoT, the experts have conducted a thorough examination to find a protection against RPL rank attacks. The objective of the decreased or increased rank attackers is to convince neighbouring DODAG nodes to act as the desired parent. The packets that pass through the rank attackers are altered or dropped [14]. Thereby, an IoT application's overall throughput is decreased. A summary of the recent research is provided.

Zahra et al. proposed Multiclass Classification (MC)—MLGBM (Machine Learning-based Gradient Boosting Machine) in [15] for detection of sensor network inherited and RPL specific attacks. The authors have generated a novel dataset to address the issue of the needed dataset not being available. The optimal feature selection method is applied to improve the effectiveness of the proposed work. The proposed model is evaluated using standard metrics such as accuracy, recall and precision.

Shafique et al. proposed Sink-based IDS (SBIDS) in [16] for identification of rank attackers. The authors have considered the root to be a reliable node. The root selects the appropriate threshold for changing parents by keeping track of changes to the rank value. AES-128 is used for encryption. SBIDS has the drawback of using more energy. As a result, the proposed approach improves an existing RPL method to preserve energy.

Osman et al. proposed Artificial Neural Network (ANN) in [17] for identification of decreased rank attack in RPL. It includes 3 phases such as pre-processing of the data, extraction of features using Random Forest classifier and ANN for detection of decreased rank attackers. The proposed model has been evaluated in scenarios of binary and multiple detections using IRAD dataset.

Karmakar et al. proposed a low overhead scheme called LEADER in [18] for detection of increased and decreased rank attack. The proposed scheme incurs low overhead since the control packet (DAO) is modified instead of new control packets. By employing a low-overhead MAC algorithm, it maintains fundamental privacy in terms of validity and integrity of the sent control message. The research gap identified in the proposed work is that there is no issue of alarm after identification of the rank attackers.

Boudouaia et al. [19] proposed the categories of all recently published papers on the rank attack, various mitigation methods and the impact on the network parameters. Finally, a review of some more recent research on security against rank attack using the Friedman test had been discussed. No mitigation method is proposed.

Boualam et al. [9] proposed a new OF called RAOF. Rank attack is implemented by altering the OF based on the rank value. It is assumed that the preferred parents or forwarding nodes are malicious. Depending on position of the attacker, this attack reduced the Packet Delivery Ratio (PDR) by 30–57% when compared to normal RPL. The disadvantages include increased network overhead and substantial delay. Raoof et al. [20] proposed a thorough investigation of RPL, its attacks and the countermeasures to these attacks. The classifications of RPL-based intrusion detection systems (IDSs), with an emphasis on the most recent IDSs are discussed. No mitigation method is proposed.

Glissa et al. [21] proposed a new secure RPL (SRPL). The primary goal of SRPL is to stop normal nodes from deliberately altering control message values, like node's rank, which could disrupt a network by fabricating a false topology. Rank thresholding is used in combination with the hash chain authentication approach to counter attacks like blackhole, sinkhole and others. According to the simulation results, SRPL is strong and resistant to attacks based on misleading modification of RPL metrics.

Almusaylim et al. proposed a Secure RPL (SRPL-RP) in [22] for identification and mitigation of version and rank attacks. A comparison of rank strategies is used to identify the attacker. For the identification of the version and rank attackers, the countermeasure uses threshold value and (attack) status tables. After detection, the attackers are isolated from the DODAG. The isolated nodes are included in the blacklist table. Alerts are generated against the attackers to skip them. Extra tables such as blacklisted and attack status tables are deployed in the proposed work which consumes more memory at the root nodes. In the proposed work, DODAG table is modified to include the blacklisted nodes which consume less memory at the root node.

Muzammal et al. [23] discussed a strategy based on trust called Security, Mobility (SM) Trust. The authors have addressed rank and blackhole attacks in RPL while taking into account both mobility and static nodes in IoT. The efficiency of the proposed work is evaluated using simulation. The findings suggest that SMTrust outperformed previous trust-based solutions for securing RPL. Despite only 2.3% rise in average consumption of power, the improvement in stability of topology is 46%, the rate of packet loss is lowered by 45% and there is a 35% improvement in throughput.

Bang et al. proposed a light-weight technique called Echelon Metric-based OF (EMBOF) in [24] to reduce and isolate the rank attack. The advertised rank's accuracy is determined and confirmed using the proposed OF. The root and preferred parent nodes in the RPL topology calculate the echelon value. The proposed approach not only detects but also isolates the attackers immediately. Nandhini et al. [25, 26] have reviewed the consequences of rank attacks and their defences. The authors have classified the countermeasures based on IDS and defense-based mechanisms. Specification, Anomaly and hybrid countermeasures are clustered under IDS mechanisms. Defense-based countermeasures include cryptography, trust and threshold-based mechanisms.

To summarise, it is understood from the existing methods that countermeasures for rank attack is still in its initial stages. Thus, more research and ideas can be sought to create more efficient methods to safeguard the DODAG topology from the rank attack. A very few attempts have been made to address the rank attack by improving the default RPL mechanism. To limit energy usage, the idea of local and global alert has not yet been considered. Most of the existing works consume more energy and incurs high computational overhead [27]. As a result, in the proposed improvement, the default RPL method has been enhanced in an effort to identify and isolate attackers with little cost and energy usage. Table 1 provides a summary of research gaps of the previous efforts.

Table 1 Summary of existing systems

References	Mechanism type	Proposed work	Investigation gaps	Metrics	Simulator	Mobility
Zahra et al. [15]	Trust	MC-MLGBM	More number of attacks are not considered	Recall, accuracy and precision	Simulator Cooja 3.0	Static or dynamic
Shafique et al. [16]	Based on cryptography	SBIDS	Increased overhead and power consumption	Power consumption and accuracy	Simulator Cooja 3.0	Dynamic
Osman et al. [17]	–	ANN	Accuracy can be improved	Accuracy, detection rate, F1-Score and precision	Simulator Cooja 3.0	Static
Karmakar et al. [18]	Threshold	LEADER	Energy intensive	PDR, latency, energy consumption and detection delay	Simulator Cooja 3.0	Static
Boudouaia et al. [19]	–	Friedman test	–	–	–	–
Boualam et al. [9]	–	RAOF	–	–	–	–
Glissa et al. [21]	Threshold	SRPL	Efficiency is not immediate for some attacks	Power consumption and Control messages overhead	Simulator Cooja 3.0	Dynamic
Almusaylim et al. [22]	Threshold	SRPL-RP	More memory consumption due to deployment of extra tables	Accuracy	Simulator Cooja 3.0	–
Muzammal et al. [23]	Trust	SM trust	Power consumption alone is evaluated	Throughput, Packet loss rate and power consumption	Simulator Cooja 3.0	Static and Mobile nodes
Bang et al. [24]	Trust	EMBOF	Alarm is not generated against the attacker	Accuracy and isolation time of attack	Simulator Cooja 3.0	Static and Mobile nodes

3 Proposed work

3.1 Setting the rate limit

The root node transmits DIO and calculates the path from the received DAO in non-storing mode. DIOs are not transmitted frequently [28]. Trickle algorithm, a gossiping algorithm [29], is used to schedule the transfer of DIO messages [30]. In our proposed scheme, the rate limit on DIO message is achieved based on the number of DODAG nodes and not on the threshold value. Each node itself is permitted to count the number of DIO packets because it is difficult for other nodes to count the amount of DIO created. The DODAG table provides the information on the number of nodes (as discussed in Sect. 3.4). As a result, dependent on number of node in the DODAG topology, the creation of DIO messages is inhibited.

Each node keeps track of its trickle time and DIO counter (C). DIO counter is used for monitoring the stability of the network. If a DIO is received with no differences from the preceding DIO message, then the value of C is incremented. If the value of C is greater than the redundancy threshold (K), the nodes reset the DIO counter and double the Trickle Time. K is a preset value which is configured by the operator. In the proposed work, the value of k is 10 which is the default value specified by the RPL protocol [31]. The nodes will minimise the trickle time and reset the value of C to zero if the incoming DIO changes. If 't' is the time within the current interval (I), $I_{\min}$ is the least interval size, '$I_{\max}$' is the highest interval size then the rate limited Trickle Algorithm is explained in Fig. 4

3.2 Pigeon hole principle

If the number of DIO messages generated (N_{m}) matches the number of nodes (N_{n}) in the DODAG, then only the normal nodes should get it, as shown in Fig. 5a. Therefore, all the normal nodes join the DODAG. If there is a newly introduced rank attacker node, then N_{n} should be greater than N_{m}. There are two possibilities under this scenario: Only normal nodes would have received the DIO message (as in Fig. 5b) or attackers (sitting close to the normal nodes) would have received the DIO message generated for the normal nodes (as in Fig. 5c). If normal nodes alone have received the DIO message, then it is not a problem. In second case, if (N_{n}) is greater than (N_{m}), then some of the normal nodes and the attackers would have received the DIO messages and joined the network based on Pigeon hole principle.

3.3 Solicitation of DIS message

Normal nodes that have not received the DIO message (for example node 7 in Fig. 6) solicit the DIS message to the root. There is a solicitation for a DIS message despite the fact that the root node has generated a number of DIO messages that equals the number of nodes in the DODAG. The timer value for solicitation of DIS message is kept low, so that the attackers can be detected earlier.

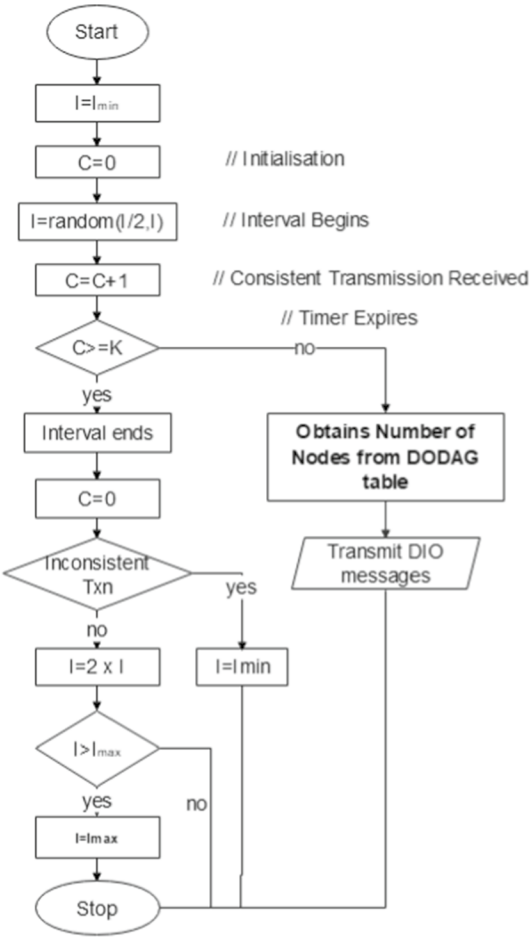


Fig. 4 Rate limited trickle algorithm

It triggers the root to check for inconsistency. The root node checks the set of DODAG parents, list of DAO parents and the status of the node in DODAG table for each node. From this check, the root node easily identifies the newly joined attacker nodes in the DODAG. Alarm is generated based on the effects of the attack when the attackers are found. Generation of alarm is discussed in Sect. 3.8. Based on the received alarm, the nodes isolate the attackers from DODAG topology.

Sometimes, normal nodes that have not received the DIO message do not solicit the DIS message. This happens when the normal node joins the nearby DODAG with the same OF. However, the DODAG continues to have a newly added attacker, which is handled in Sect. 3.7. The detection of the attackers in this scenario is done by introducing the hashed value in DAO message. Thus, the format of DAO message is modified to include the hash value.

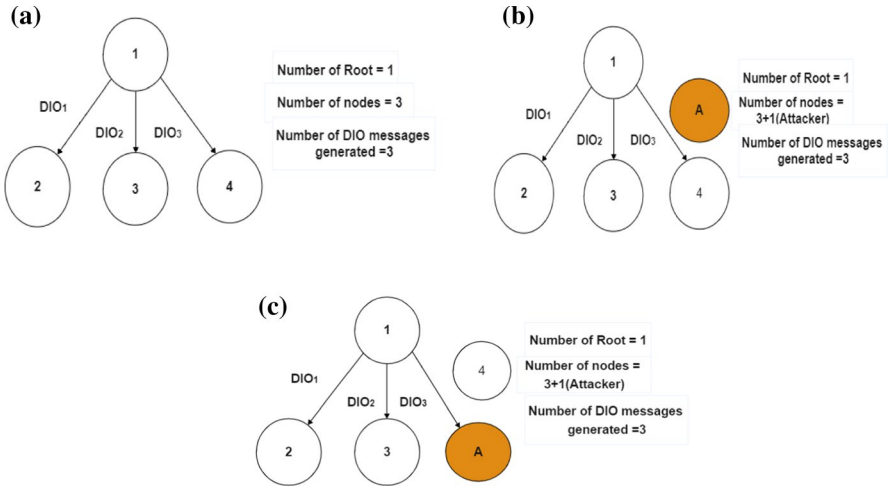


Fig. 5 **a** DIO messages received by normal nodes. **b** Topology with a newly introduced attacker. **c** DIO messages received by attackers

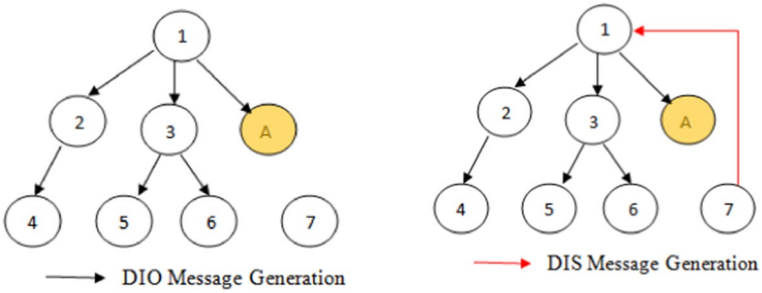


Fig. 6 Solicitation of DIS message

3.4 Modification of DODAG table

Each DODAG has a table that is kept up to date. The RPLInstanceID, DODAGID, Version number [32], Rank, list of DAO parents and other data can be found in the DODAG table. NodeID, hash value and malicious nodeID are added to the DODAG table. To save memory, the current DODAG database is expanded to include the necessary data rather than using a separate table for preserving nodeID and hash value.

3.5 Computing hash value

The RPL protocol is appropriate for networks with little resources. So, the hash value is calculated using a non-cryptographic hash function. The hash value takes 32-bits (nodeID and rank) as input and produces 32-bits output. NodeID is unique for each

node and is represented in 16-bits. Each node is given a unique nodeID while joining the DODAG. Since nodeID is unique, there will be no hash collision, i.e. input and output are mapped 1:1. The hash function's pseudocode is provided below

```
hash32( )
{
    x^=x>> 16;
    x*= 0x7feb352dU;
    x^= x >> 15;
    x* = 0x846ca68bU;
    x^=>> 16;
   return x;
}
```

DAO message is modified to include the computed hash value. The inverse of the hash function is given by

```
inversehash32( )
{
    x^=x>> 16;
    x*= 0x43021123U;
    x^=x >> 15^x >> 30;
    x* = 0x1d69e2a5U;
    x^=x>> 16;
   return x;
}
```

Root node can use above inverse hash code to check received values from each DAO message. But it drains more energy at the root node which is against the rule of resource constrained networks. Since the static topology of the node is considered in the proposed work, the hash value will not change more frequently. So, root node computes and stores the hash value corresponding to the nodeID in the DODAG table itself. The hash value can be verified and updated after each repair mechanism.

3.6 Modified DAO message

DAO message includes various options. It consists of one or more Transit Information Options (0×09) and Target Options (TIO: 0×05). The target IP address is specified using the target option. The target IP address for RPL's non-storing mode should be that of the root node. If the root node receives DAO messages, it determines whether the target IP address is the root node's IP address. If not, the received DAO message is discarded. A node specifies the DODAG parents via TIO. For the purpose of source routing, it is utilised to collect routing data. The ancestor in non-storing mode will be DODAG root. In Fig. 7, the amended DAO message is presented.

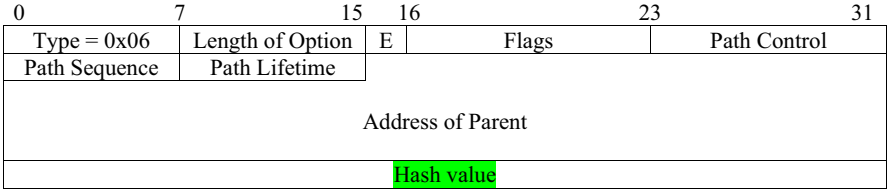


Fig. 7 Modified DAO message

3.7 Check for inconsistencies

Each node in the DODAG generates the modified DAO message in response to the DIO message. Root node checks the hash value in DAO message with the hash value in DODAG table. If inconsistency is detected, then DODAG parents, DAO parents, rank and node status of that node is checked from DODAG table to confirm that node to be an attacker. The value of rank in the altered DAO is compared with the value in the DODAG table to determine if it is a decreased or an increased rank attack. Increased rank attack occurs when the value of rank in the received DAO exceeds the value in the DODAG table. It is a decreased rank attack if the value of rank is lower than the DODAG table rank. Thus, thorough inconsistency checks, if an internal node that has turned into an attacker is also detected. Following validation, an attacker's nodeID is added to the DODAG table's blacklist column and an alarm is triggered as will be covered later.

3.8 Generation of alarm

The alarm is not issued as a distinct message in the proposed work. The network overhead increases if it is generated as a separate control message. This is against RPL policy. As can be seen in Fig. 8, the DIO message itself contains an alarm.

Global or local alarm will be triggered depending on how an attacker would influence DODAG. While global alert is issued including all nodes in the DODAG, local alarm will only be triggered to the nearest nodes. Following the alarm's production, the attackers are excluded from DODAG and their ID is added to the table's blacklist column. Through an ICMPv6 error message, the nodes notify the root node of their neighbour change. The root node starts the repair operation as soon as it receives the error notification. In a local repair mechanism, only adjacent nodes receive the DIO message with the alarm, whereas in a global repair mechanism, the DIO message is

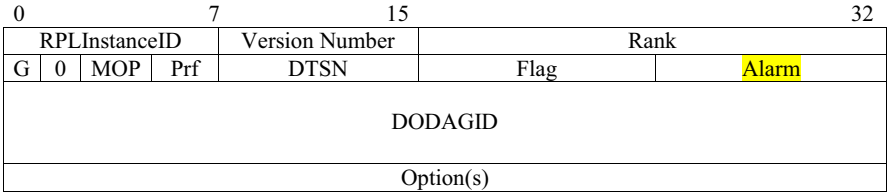


Fig. 8 Updated DIO

broadcast to every node in the DODAG. The DODAG table is consequently updated following each repair method. If an inconsistency is found during the repair method, the R (Rank error) bit is set and will be checked later.

3.9 E-RAD algorithm

E-RAD (Enhanced Rank Attack Detection):
Rank attacker detection in Non-Storing mode:
1. DODAG root generates DIO message based on the rate threshold (R_t)
2. **if** number of DIO messages generated (N_m) < Number of nodes (N_n) then
3. **Existence of newly introduced **rank attackers****
4. **if** normal nodes alone received DIO
5. Rank attackers would not have received DIO //**No problem**
6. break
7. **else**
8. Some of normal nodes and attackers would have received DIO // Pigeon hole principle
9. Normal nodes may or may not solicit DIS if they have not yet received a DIO message
10. **if** normal node solicit DIS // **Suspicious Activity detected**
11. attack_detect( )
12. **else**
13. Joined some other DODAG // No problem
14. presence_attack( ) // Still existence of newly introduced attacker
15. **end**
16. **end**
17. **end**
def attack_detect ( ):
for each node Root checks
DODAG parents, DAO parents, rank and node status from DODAG table
if inconsistency detected
Detects newly introduced and internal rank attacker
generates_alarm ( )
End
End
def presence_attack ( ):
Checks the hash value in DAO message with the hash value in the DODAG table
if inconsistency detected
Detects newly introduced rank attacker
generates_alarm ( )
End
def generates_alarm( ):
if rank value in modified DAO message > DODAG table rank value
Increased rank attack is identified
Else
Decreased rank attack is identified
Includes node in the blacklist of DODAG table
Solicit local or global alarm
Isolate the newly introduced attacker

4 Results and discussions

Cooja Simulator3.0 is used to assess the proposed E-RAD algorithm. It is a Java-based Simulator which runs on Contiki OS. Since it is highly portable and multi-tasking OS, it is widely used in Research.

4.1 Hash algorithm evaluation

Hash function used for the generation of hash value in E-RAD algorithm is assessed using (i) *bias* and (ii) *avalanche effect.* (i) *Bias:* Probability of an input bit influencing an output bit. If it is greater than or less than 50%, then the hash is 'biassed'. An effective hash function should have as little bias as feasible. This is because small amounts of bias normally have little impact on the quality of a hash function. If it is greater, then it leads to more hash collisions. The above-mentioned hash function's bias is 0.17353355999581582. (ii) *Avalanche effect:* Changing the input little causes the output to change significantly. An avalanche effect of at least 50% is required for a successful hash function. While benchmark algorithms like Blow fish, MARS, CR6 and 3-DES have avalanche effects of 48.4%, 50%, 47% and 30.8%, respectively, as in [33], the avalanche effect for the hash function utilised in the current work is 54.7%. As a result, the above used hash function performs well, having a large avalanche effect and a low bias value. The hash function is also simple and so it consumes very less energy at the root node for computation which is the need of the RPL.

4.2 Simulation parameters

The topology considered for evaluation of E-RAD algorithm is static topology. TmoteSky nodes are deployed. One sink node and 59 nodes were deployed. Number of newly introduced rank attackers was 10. Based on modification in the Contiki RPL library, rank attackers have been implemented. In the area of 300×300 m, a total of 60 nodes were deployed. Transmission range of each node is 50 m. During simulation, the nodes periodically transmit the temperature to the root through UDP packets. For simulation, many topologies are taken into account, including random topology (Fig. 9a), grid random (Fig. 9c) and grid-centered (Fig. 9b).

Green represents the root in Fig. 9a–c), mustard yellow represents normal nodes and pink represents malicious nodes.

4.3 Simulation results and analysis

Comparisons between the proposed E-RAD algorithm and SBIDS [16] and SRPL-RP [22] are made. The graph is plotted using the average of all the values from each run, which totals 15 different simulations. The duration of each simulation was 60 min. The parameters considered for evaluation of the proposed E-RAD algorithm are.

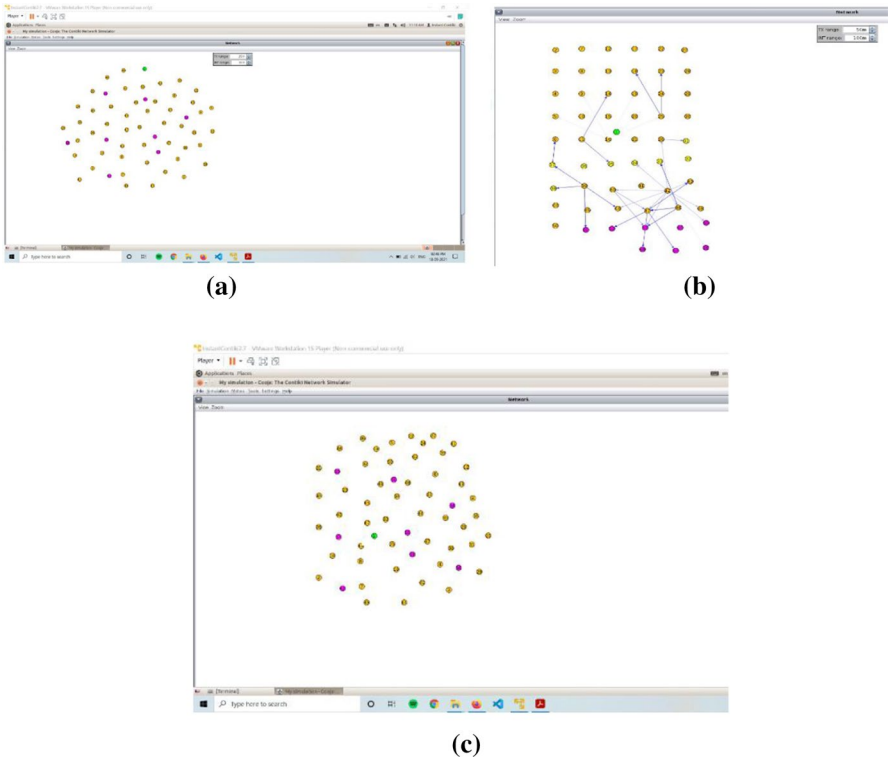


Fig. 9 **a** Random topology. **b** Grid centred topology. **c** Grid random topology

1. PDR
2. Accuracy
3. Control Packet Overhead
4. Average delay of packet delivery
5. Average energy consumption

4.3.1 PDR

PDR is defined as the ratio of total packet deliveries to total packet transmissions across the network from source nodes to destination nodes. PDR is calculated in different topologies and presented in Fig. 10

From Fig. 10, it is inferred that RPL could not attain 100% PDR in any of the topologies in the absence of attackers. This is due to MAC layer issues and the frequent retransmission of the packets. The proposed algorithm E-RAD obtained a PDR (95.53%) that was nearly equal to the RPL without attackers (97.26%) in all the topologies. This is a result of the attackers becoming isolated quickly. The attackers are detected and isolated during the transmission of control packets itself. In comparison to the other two topologies, the Grid-centered topology achieves better PDR.

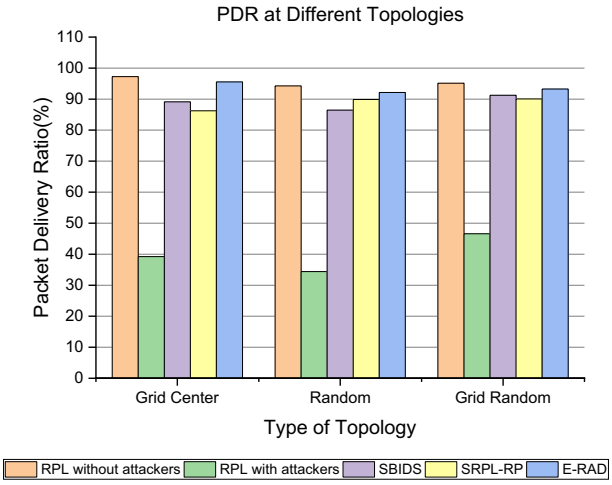


Fig. 10 PDR at different topologies

Table 2 Increase in number of attackers with PDR

No. of attackers	SBIDS	SRPL-RP	E-RAD
10	92.45	90.12	94.26
20	89.92	88.95	92.03
30	84.62	80.23	88.69
40	83.24	78.68	84.23
50	79.56	75.42	82.78

This is as a result of the high node placement density (i.e., nodes are placed closer). To determine the efficiency of the E-RAD algorithm while taking into account Grid-centered topology, the number of attackers is raised. The results are shown in Table 2.

The E-RAD method outperforms the other two examined mitigation techniques in terms of PDR despite an increase in the number of attackers, as shown in Table 2.

4.3.2 Accuracy

The percentage of nodes in the network that were correctly classified as benign or malicious is known as accuracy. The accuracy is determined using

$$\text{Accuracy} = \frac{\text{TP(TruePositive)} + \text{TrueNegative}(TN)}{\text{TP} + \text{TN} + \text{FalsePositive(FP)} + \text{FalseNegative(FN)}}$$

In Fig. 11, accuracy measured in various topologies is displayed. According to Fig. 11, the E-RAD algorithm in a Grid-centered topology delivers greater accuracy (97.23%) than previous countermeasures. Due to the existence of attackers among closely placed nodes, the attacker receives more DIO message than the normal

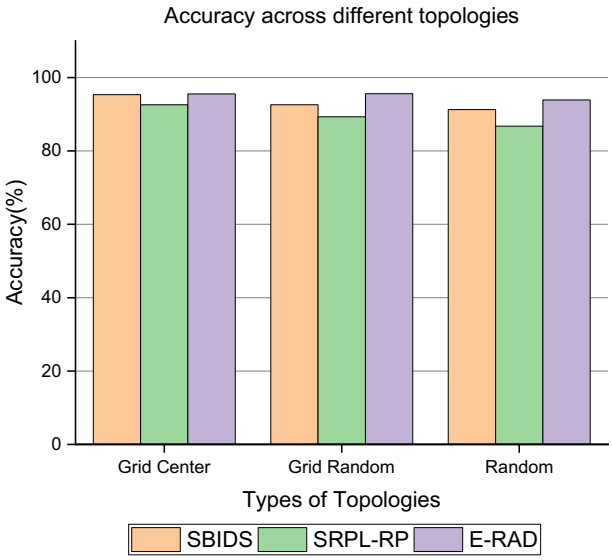


Fig. 11 Accuracy at different topologies

nodes. So, that the attackers join the DODAG by replying with the DAO message. Thus, the amount of inconsistent DAO generated will be more in this topology. As the quantity of inconsistent packets increased at the root node, more attackers were detected earlier. In random topology, accuracy is lower. The accuracy of SBIDS is 91.28%, SRPL-RP is 86.75% and E-RAD is 93.9%. The random placement of the nodes causes a portion of the DODAG to detach. Because they won't receive the

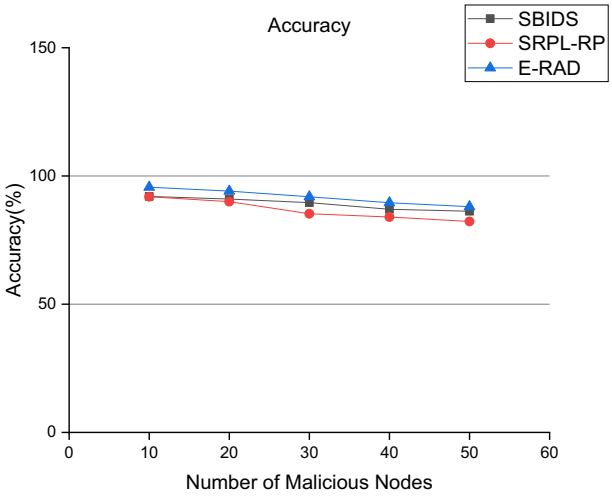


Fig. 12 Number of malicious nodes vs accuracy

DIO communication, the attackers in the detached section will be unable to join the DODAG. Accuracy of the E-RAD algorithm is verified by increasing the number of attackers in Grid-centered topology.

Figure 12 illustrates that accuracy declines with the increase in number of attackers. As the number of attackers increases, the attackers in the detached region cannot be reached. Despite the increase of attackers, the E-RAD algorithm still has a high level of accuracy (97.23%). This might be because rate limiting techniques were used to cut down on DIO. As the amount of DIO decreases, either the attacker or normal nodes may get it. There is no issue if the normal nodes receive it. If the attackers attempt to capture it, the demand for DIS messages rises. This allows for the early identification of more attackers.

4.3.3 Control packet overhead

Control packet overhead is the quantity of control packets generated during DODAG formation. Due to energy restriction of the nodes, it is difficult to reduce the quantity of control packets needed for the formation of DODAG. The control packet overhead is computed by

$$\text{ControlPacketOverhead} = \sum_{i=1}^{n} \text{DIO}(i) + \sum_{j=i}^{n} \text{DAO}(j) + \sum_{l=1}^{n} \text{DAO} - \text{ACK}(l) + \sum_{k=1}^{n} \text{DIS}(k)$$

Control packet overhead was calculated using different topologies and the result obtained is presented in Fig. 13

From Fig. 13, it is inferred that quantity of control packets created is less than RPL in all the topologies. This is due to the incorporation of rate limiting technique

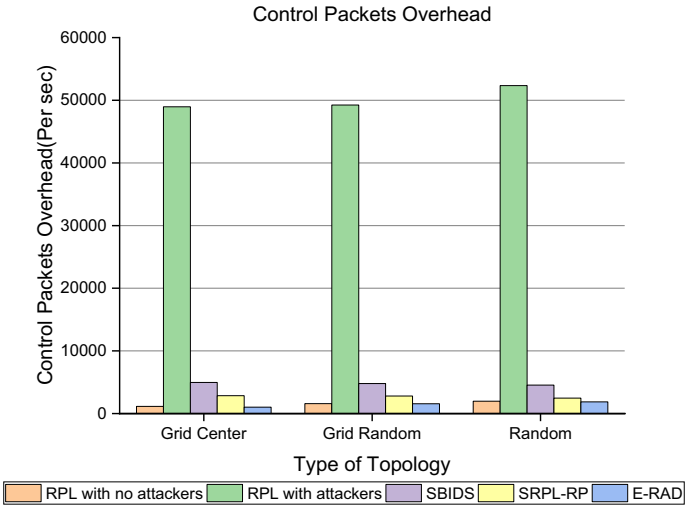


Fig. 13 Control Packet overhead

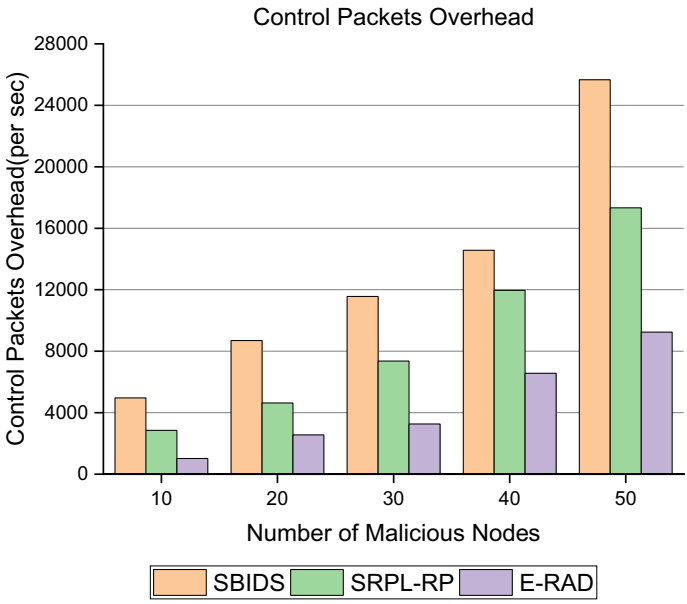


Fig. 14 Number of malicious node Vs control packet overhead

during the construction of DODAG. Nodes are wide spread in random topology which leads to frequent retransmission of control packets. As a result, compared to other topologies, random topology generates a greater amount of control packets. To check the effectiveness of incorporation of rate limiting in E-RAD algorithm, the performance is verified using increased number of attackers.

It can be concluded from Fig. 14 that E-RAD worked quite well in aspects of reducing the quantity of control packets created, which is crucial for RPL.

4.3.4 Average delay for delivery of packet

Delay of a packet is the interval of time between when it is created and when the DODAG root receives it.

$$\text{Averagedelay} = \sum_{n=0}^{k} \text{Receivedtime}(n) - \text{Senttime}(n)$$

By taking into account various topologies, the average data packet delivery delay (in ms) was computed and is displayed in Fig. 15.

From Fig. 15, it is evident that the Grid-centered topology has a lower average delay for data packet delivery. This is due to the closer placement of the nodes, which speeds up the delivery of data packets. It is higher in random topology (1205 ms) due to longer paths for the nodes. E-RAD offered a lower average delay

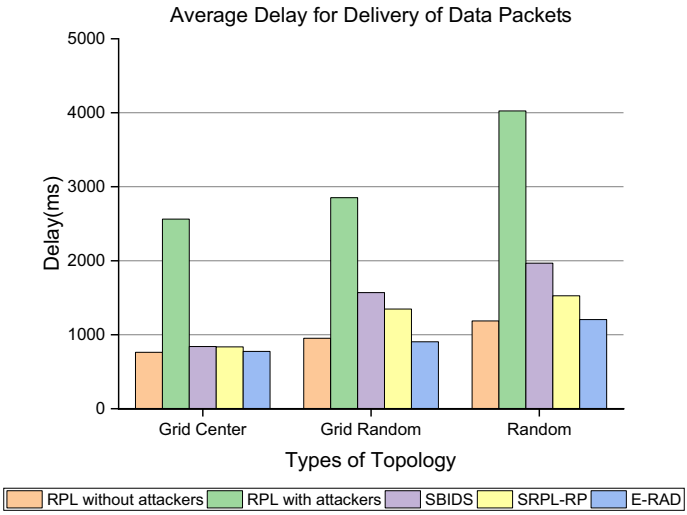


Fig. 15 Average delay for packet delivery (in ms)

for delivery of packets (775 ms) compared to all other countermeasures. This is because rate-limited control packets were generated which may lead to early detection of attackers. Since the attackers are isolated there will be no frequent formation of DODAG topology and there will be early delivery of data packets.

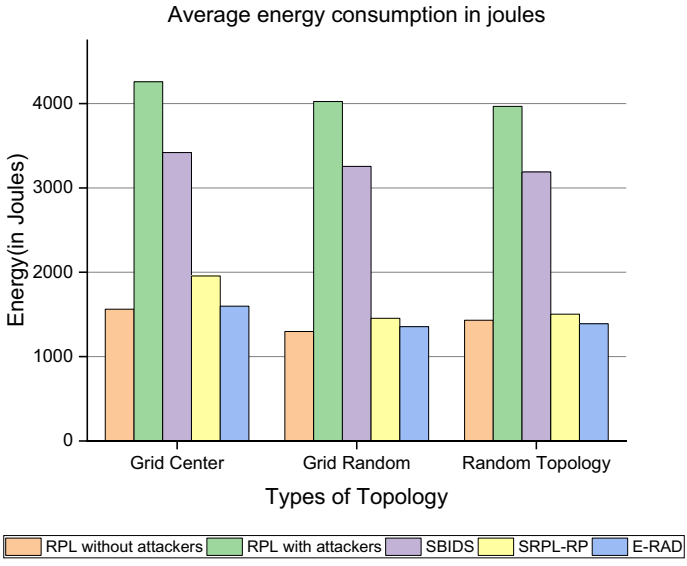


Fig. 16 Average energy consumption (in Joules)

4.3.5 Average energy consumption

The energy used by the nodes is their average energy consumption. The energy is categorised as LPM energy, CPU energy, Radio Transmit and Listen energy. A node cannot be active in all modes at once. Contiki measures each of these energy usage types internally as it is being used. Cooja Simulator uses the Power Trace Plugin to track the energy usage of the nodes. Figure 16 represents the average energy usage of the nodes (in joules).

The average energy consumption of the nodes in a grid-centered design is high (seen from Fig. 16). The nodes are evenly spaced apart and positioned close to the root, resulting in a high node placement density. Among the countermeasures compared, E-RAD consumes less energy (1356 J) because the generation of the control packets are rate limited. E-RAD consumes energy slightly higher than RPL without attackers. This could be as a result of the generation of hash values and consistency check. SBIDS uses the computationally demanding AES for security purposes, it consumes more energy.

5 Conclusion and future works

There are many benefits of incorporating IoT into different applications. In addition to the benefits, it is vulnerable to various threats. As a result, improving the security of IoT network is of the utmost importance. The proposed E-RAD algorithm incorporates rate limit on generation of DIO control packets and the solicitation of DIS control packets for identification of newly joined attackers. If attackers are missed in the above scenario, then it is detected from consistency check of hash value. Upon the identification of rank attackers, alarm is generated to isolate them from DODAG. Alarm is an attachment in the DIO message itself. Cooja Simulator is used to assess the potency of E-RAD algorithm. The algorithm achieves 95.53% of PDR, 97.23% of accuracy, average delay of 775msec, 1023 number of control packets in grid centered. The estimate energy usage is lowered to 1356 joules.

In most of the existing systems and the evaluation of E-RAD, static topology is considered. Support for mobility is crucial in many IoT-based applications. Thus, E-RAD algorithm can be expanded to incorporate node mobility and identification of other IoT attackers.

Data availability Data would be supplied upon request.

Declarations

Conflict of 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.

References

1. Maraveas C, Piromalis D, Arvanitis K, Bartzanas T, Loukatos D (2022) Applications of IoT for optimized greenhouse environment and resources management. Comput Electron Agric 198:106993
2. Pourghebleh B, Hekmati N, Davoudnia Z, Sadeghi M (2022) A roadmap towards energy-efficient data fusion methods in the Internet of Things. Concurr Comput: Pract Exp 34:e6959
3. Sharma S, Verma VK (2021) Security explorations for routing attacks in low power networks on internet of things. J Supercomput 77(5):4778–4812
4. Kaviani F, Soltanaghaei M (2022) CQARPL: congestion and QoS-aware RPL for IoT applications under heavy traffic. J Supercomput 78:16136–16166
5. Seyfollahi A, Moodi M, Ghaffari A (2022) MFO-RPL: A secure RPL-based routing protocol utilizing moth-flame optimizer for the IoT applications. Comput Stand Interfaces 82:103622
6. Almusaylim ZA, Alhumam A, Jhanjhi N (2020) Proposing a secure RPL based internet of things routing protocol: a review. Ad Hoc Netw 101:102096
7. Maidamwar PR, Bartere MM, Lokulwar PP (2022) Implementation of network intrusion detection system using artificial intelligence: survey. In: Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications. Springer, Berlin. pp 185–198
8. Hkiri A, Karmani M, Machhout M (2022) 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), 2022. IEEE, pp 143–148
9. Boualam SR, Ezzouhairi A (2020) New objective function for RPL protocol. In: Embedded Systems and Artificial Intelligence. Springer, Berlin, pp 681–690
10. Nandhini PS, Mehtre B (2019) Directed acyclic graph inherited attacks and mitigation methods in RPL: a review. International Conference on Sustainable Communication Networks and Application. Springer, pp 242–252
11. Sahraoui S, Henni N (2021) SAMP-RPL: secure and adaptive multipath RPL for enhanced security and reliability in heterogeneous IoT-connected low power and lossy networks. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-021-03303-9
12. Sahay R, Geethakumari G, Mitra B (2022) A holistic framework for prediction of routing attacks in IoT-LLNs. J Supercomput 78(1):1409–1433
13. Raoof AM (2021) Secure Routing and Forwarding in RPL-based Internet of Things: challenges and solutions. Carleton University, Ottawa
14. Seyfollahi A, Ghaffari A (2021) A review of intrusion detection systems in RPL routing protocol based on machine learning for internet of things applications. Wirel Commun Mobile Comput. https://doi.org/10.1155/2021/8414503
15. Zahra F, Jhanjhi N, Brohi SN, Khan NA, Masud M, AlZain MA (2022) Rank and wormhole attack detection model for RPL-based internet of things using machine learning. Sensors 22(18):6765
16. Shafique U, Khan A, Rehman A, Bashir F, Alam M (2018) Detection of rank attack in routing protocol for low power and Lossy networks. Ann Telecommun 73(7):429–438
17. Osman M, He J, Mokbal FMM, Zhu N (2021) Artificial neural network model for decreased rank attack detection in RPL based on IoT networks. Int J Netw Secur 23(3):496–503
18. Karmakar S, Sengupta J (2021) Bit SD LEADER: low overhead rank attack detection for securing RPL based IoT. In: 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), 2021. IEEE, pp 429–437
19. Boudouaia MA, Ali-Pacha A, Abouaissa A, Lorenz P (2020) Security against rank attack in RPL protocol. IEEE Netw 34(4):133–139
20. Raoof A, Matrawy A, Lung C-H (2018) Routing attacks and mitigation methods for RPL-based Internet of Things. IEEE Commun Surv Tutor 21(2):1582–1606
21. Glissa G, Rachedi A, Meddeb A (2016) A secure routing protocol based on RPL for Internet of Things. In: 2016 IEEE Global Communications Conference (GLOBECOM), 2016. IEEE, pp 1–7
22. Almusaylim ZA, Jhanjhi N, Alhumam A (2020) Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. Sensors 20(21):5997
23. Muzammal SM, Murugesan RK, Jhanjhi NZ, Humayun M, Ibrahim AO, Abdelmaboud A (2022) A trust-based model for secure routing against RPL attacks in internet of things. Sensors 22(18):7052
24. Bang AO, Rao UP (2022) EMBOF-RPL: improved RPL for early detection and isolation of rank attack in RPL-based internet of things. Peer-to-Peer Networking and Applications 15(1):642–665

25. Nandhini PS, Kuppuswami S, Malliga S (2021) Energy efficient thwarting rank attack from RPL based IoT networks: a review. Proc, Mater Today. https://doi.org/10.1016/j.matpr.2021.04.167
26. Nandhini PS, Kuppuswami S, Malliga S (2022) Classification of intrusions in RPL-based IoT networks: a comparison. In: Mobile Computing and Sustainable Informatics. Springer, Berlin, pp 849–862
27. Ragesh G, Kumar A (2022) Trust-based secure routing and message delivery protocol for signal processing attacks in IoT applications. J Supercomput. https://doi.org/10.1007/s11227-022-04766-z
28. Yassien MB, Aljawarneh SA, Eyadat M, Eaydat E (2021) Routing protocol for low power and lossy network–load balancing time-based. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-020-01261-w
29. Vallati C, Mingozzi E Trickle-F (2013) Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol. In: 2013 Sustainable Internet and ICT for Sustainability (SustainIT). IEEE, pp 1–9
30. Kalita A, Khatua M (2021) Opportunistic transmission of control packets for faster formation of 6TiSCH network. ACM Trans Internet Things 2(1):1–29
31. Vučinić M, Król M, Jonglez B, Coladon T, Tourancheau B (2017) Trickle-D: High fairness and low transmission load with dynamic redundancy. IEEE Internet Things J 4(5):1477–1488
32. Nandhini PS, Srinath P, Veeramanikandan P, Malliga S (2021) Version attack detection using claim algorithm in RPL based IoT networks: effects and performance parameters evaluation. In: 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC). IEEE, pp 209–215
33. Skeeto (2020). April 2020 https://github.com/skeeto/hash-prospector.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

P. S. Nandhini[1] · S. Kuppuswami[2] · S. Malliga[1] · R. DeviPriya[3]

S. Kuppuswami
skuppu@gmail.com

S. Malliga
mallisenthil.cse@kongu.edu

R. DeviPriya
scrpriya@gmail.com

[1] Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, India

[2] Department of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Namakkal, India

[3] Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, India