IJICTR
International Journal of Information & Communication Technology Research

Volume 13- Number 2 – 2021 (17 -28)

Intrusion Detection System to Detect Insider Attack on RPL Routing Protocol Based on Destination Advertisement Object

N. Jahantigh
Faculty of Electrical and Computer Engineering
University of Sistan and Baluchestan
Zahedan, Iran
nadiya_jahantigh@pgs.usb.ac.ir

A. Bakhtiyari Shahri*
Faculty of Electrical and Computer Engineering
University of Sistan and Baluchestan
Zahedan, Iran
bakhtiyari@ece.usb.ac.ir

Received: 15 April 2021 - Accepted: 20 May 2021

Abstract—**The increasing fascination with the Internet of Things has led to the extensive deployment of Low-power and Lossy Networks. IPv6 Routing Protocol over Low Power and Lossy Networks serves as the ideal routing protocol proposed by IETF for routing in IoT-LLNs. Routing attacks are one of the IoT challenges that can lead to network performance problems and often denial of service. The Destination Advertisement Object (DAO) insider attack is one of the most notable attacks in RPLs, and previous studies have not developed a complete method for its detection so as to separate the malicious node from the normal node. Using an anomaly-based intrusion detection system, this paper suggests three methods based on random, fixed, and dynamic threshold adjustment to prevent DAO insider attack and identify malicious nodes. The results showed that the proposed model has a detection rate of 100% and a very low rate of false alert.**

Keywords-DAO attack; Internet of Things; LLN; RPL; Security.

I. INTRODUCTION

The Internet of things (IoT) refers to any physical object that could be virtually detected, directed, and accessed. It is predicted that by 2025 around 35-75 billion devices will be connected to IoT networks [1, 2]. On the other hand, intrusion causes disruption in normal activities of computer networks. Hence, networks as wide as IoT are not an exception and they need to be equipped to manage imminent sudden attacks. For instance, when a user's signal is disconnected or stopped, his/her privacy could be exposed and certain information might be disclosed [3]. Even though various methods have been introduced to resolve security issues and challenges of IoTs, the inherent complexity and heterogeneous structure of these networks and their basic deployment difference from conventional networks have made it extremely difficult to ensure their security [4, 5]. Standardized by the IETF institute and the ROLL group for resource-limited nodes, the so-called RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol compatible with the IoT network layer. This protocol is designed specifically for the link layer (including IEEE 802.15.4 Mac and physical layers) to perform high data transfer operations in low-power networks.

* Corresponding Author

[DOI: 10.52547/itrc.13.2.17]

Although RPL is designed for secure interaction between nodes, but it involves a number of attacks and vulnerabilities which can disrupt network performance [6]. Thus, in the version number attack [7], a malicious insider node increases the version number by starting an unnecessary global network repair process. Similar to the version number attack, the DAO induction attack can cause a large number of network transfers. In this attack, the malicious node repeatedly increment its DTSN to cause unnecessary transmission of DAO control messages [8]. DAO insider attack is another instance in this context [9].

In this attack, the attacking node repeatedly sends fake DAO messages to the parent node. The parent node also sends the message to its child until it reaches the root node. This confronts the network with a large amount of DAO messages. In this attack, unlike other cases, the level of damage is not limited to the current attack range in the network. This is because DAO messages are transmitted end to end from the sensor node to the root node. This attack extremely affects the efficiency, power consumption, delay, and reliability of the entire network.

There are few studies on identifying and preventing DAO attacks, which highlights the need for developing a solution. In our proposed methods, machine learning algorithms are used offline in order to prevent DAO insider attack. In this approach, networks run with various simulation parameters and learn offline the number of different DAO packet transmissions. These learnings are then converted into a series of threshold values or conditions.

The main features of this article are as follows:

- First, a randomized algorithm is introduced, such that the higher the number of DAO messages sent, the more malicious the node could be and the less likely it is for its message to be sent to the parent node.
- We present a method which ensures that each node forwards the DAO messages received from child nodes. Based on the prepared dataset, it is predicted that this method will be applicable to all simulation scenarios without assuming specific network configurations or conditions.
- According to another algorithm, the threshold for sending a DAO message is dynamically set in each time period.
- We implement and evaluate the proposed algorithms using Cooja simulator in the Contiki operating system.

In the following, the concepts of RPL and intrusion detection system (IDS) are introduced.

RPL is primarily designed for networks such as 6LoWPAN, and it is a standard distance-vector routing protocol for low-power, lossy networks. The core of the RPL is a destination-oriented directed acyclic graph (DODAG) [6]. A directed acyclic graph (DAG) defines a tree structure to specify default routes between nodes. In this structure, a node may have several parents, while classic trees can only have one parent. In fact, RPL organizes nodes as a DODAG. A network can have one or more DODAGs that together form an RPL with a unit identifier. A network can also run multiple RPLs simultaneously, but these instances are logically independent of each other. On the other hand, a node can be connected to several RPL instances only if it belongs to a DODAG within each RPL instance.

A DAG is created according the configuration parameters defined in the node joining operation.

A node might have several parents. Choosing a more favorable parent depends on the rank that indicates the virtual distance of each node from the root and is employed to prevent and identify routing loops in RPL [8]. The rank rises as the distance widens from the root and falls as the root approaches. A rank is specified using an Objective Function (OF). The OF metrics can be the expected transmission count (ETX metric), hop count, energy, delay, or other metrics. Different OFs like Objective Function Zero (OF0) or the Minimum Rank with Hysteresis Objective Function (MRHOF) are presented in the RPL for rank calculation [10].

Figure 1 shows a DODAG that is a node with a border router given the lowest rank (i.e. 1). Other nodes also rank higher, depending on the distance they have in the network with the border router.

There are different kinds of control messages, including DAG Information Object (DIO), Destination Advertisement Object (DAO), and DAG Information Solicitation (DIS). They are defined in order to maintain and update routing information required to build a DODAG in the RPL. RPL control messages are sent based on three traffic patterns: multi-point to point (MP2P), point to multi-point (P2MP), and point to point (P2P). MP2P traffic is possible using upward routes from the child toward the root. P2MP traffic can be conducted through downward routes in the DODAG from the root or parent to the child. Finally, P2P traffic could take place using both upward and downward routes. RPL messages are defined as type 155 of ICMPV6 control messages [11]. The DIO message contains network information that is used to select the preferred parent, obtain DODAG information, as well as discover an RPL instance and learn its configuration parameters. When the root is the only DODAG node, it sends its location to all network levels using the DIO message. The DIS message is another type of RPL control message that is used by a node which is supposed to receive the DIO message from its neighboring node. When a node wants to join the graph, it multicasts the DIS message to its neighbors and requests a graph information message. RPL also uses the DAO control message to support the downward route.

A node seeking to be accessible by the root publicizes its address within a DAO message and transmits it to one of its DAO parents. DAO-ACK messages are sent by the receiver of the DAO message in response to that DAO message. The DAO receiver is the parent or root of the graph, meaning that the nodes transmit their routing information to their parent [11].

To create a DODAG graph, the root node first sends the required information to the nodes through a DIO message in order to detect an RPL instance and its configuration parameters. At each level, the receiver records its parent's route and adds a sender node to its

[DOI: 10.52547/itrc.13.2.17]

parent set. These nodes also determine their rank according to the OF mentioned in the DIO message.

Due to the acyclic nature of the graph, each node should not advertise a rank that is lower than any of the members of its parent set. They then broadcast the updated DIO message to their neighbor. Upon completion of this process, all nodes in the network have at least one routing entry with upward nodes which eventually reaches the root node based on an MP2P traffic pattern. This route includes all preferred parents. DIO messages are sent alternately based on the Trickle algorithm.

This algorithm adjusts the transmission frequency of control messages in accordance with the optimal status of the network. The node that wants to connect to the network sends a DIS message to receive a DIO message from its neighboring node. The DAO message is used to broadcast information about the presence of a downward route. Depending on the operation of the RPL in terms of storage in the DIO message, router nodes can preserve routing tables. To aggregate routing tables, each node, except the root, sends a DAO message with the prefix of both its parent and children.

In terms of downward routing, RPL operates in both storing (table-driven) and non-storing (source-routing) modes (Figure 2). However, the graph can function in just one mode. In the non-storing mode of routing when the related timer expires, the node collects the routing information and sends a new DAO message to its parent set. This process is repeated until DAO messages reach the root. Thus, the DAO message is unicast to the DODAG root. In this case, intermediate nodes do not record the prefix of their child nodes, and only the root is capable of storing and maintaining all downward routes. Therefore, intermediate parents send DAO messages to the preferred parent by storing their addresses in the stack of the reverse path of the received DAO package. In the storing mode, router nodes keep a routing table with all available destinations. To this end, each child unicasts a DAO message to the parent, and the latter records the content of these DAO messages [11].

This section proceeds with a security mechanism in the RPL protocol.

An intrusion detection system (IDS) can be used to detect attacks on a system. The role of an IDS is to monitor the performance of a computer or network system and to analyze it for signs of intrusion. In an IDS, three main modules of observation, identification, and analysis are employed. The first module observes network traffic and resources. Identification and analysis are the main components of an IDS which are used to detect intrusion based on a specific algorithm. Necessary alerts could be sent after identifying network intrusions. IDS can be used to monitor unknown traffic in a selected node, observe the performance of both network and nodes, and detect both external and internal attacks. In general, there are four types of IDS: signature-based, anomaly-based, specification-based, and hybrid [12].

Signature-based IDS focuses on identifying "known attacks," thus it is also called rule-based intrusion detection. This technique involves signatures or some predefined patterns that can be stored in a database. Depending on a predefined pattern or individual attack signature, each attack can be identified by exploring parameters that are known in a database for a specific attack. The known parameters form the attack signature and, thus, the type of an IDS. The limitation of this method is that unpredictable attacks on the system will not be detectable. Therefore, it is necessary for the database to be regularly updated with new attack or signature patterns [13].

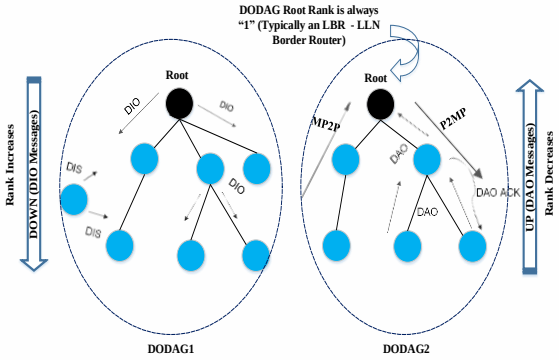

Figure 1. RPL architecture.

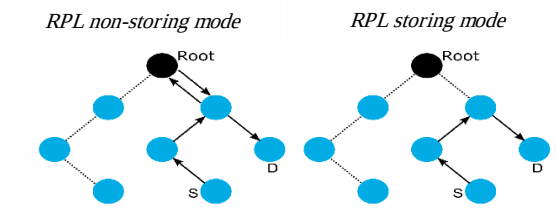

Figure 2. An RPL instance in both storing and non-storing modes.

Anomaly-based IDS (ADS) is used to identify malicious activities by analyzing network performance. This technique detects attacks that were previously unknown to the system by observing unusual system behaviors. Hence, this method identifies attacks more than does the signature-based method. In this technique, the malicious node is detected based on the deviation of the current protocol from the status of the protocol identified earlier. For example, an ADS might interpret the sudden forwarding of numerous packets by a node in a network as an attack [14].

Specification-based IDS (SIDS) is similar to ADS, but it provides a lower false rate than does ADS. In this method, the regular behavior of the network is manually defined for each profile, and intrusion is detected when the network behavior deviates from the defined rules. Finally, hybrid IDS uses a combination of the above methods.

Another type of security mechanism that is closely related to IDS is called Intrusion Prevention System (IPS) [15]. The purpose of IPS is primarily to intercept attacks and then to prevent further attacks. In the present study, the task of the IPS is to identify malicious nodes that disrupt the network. This system utilizes several approaches in preventing and detecting anomalies based on threshold values so as to detect DAO attacks.

The next section introduces a comprehensive review of the literature related works. The problem is

[DOI: 10.52547/itrc.13.2.17]

discussed and a description of the previous research objectives is presented in the third section. The fourth section includes the discussion of the research methodology. It explains the research procedure and research instruments utilized for this study. It also presents the result of simulation and performance of the proposed methods.

II. Review Literature

Numerous attacks and vulnerabilities have been introduced for RPL in the context of IoT which could disrupt network performance [16]. Consequently, the security of routing protocols on IoT has become a challenge for researchers [17]. Various studies have attempted to develop RPL and to identify and prevent attacks accordingly; also, several intrusion detection mechanisms and other methods have been proposed.

In the context of IDS in RPL-based networks, existing methods deal with attacks like sinkhole attack, selective-forwarding attack, and wormhole attack, and they deploy different techniques for their detection. For example, Chugh et al. [3] considered a normal network mode and two types of attacks: selective forwarding and sinkhole; in each case, the number of transmitted DIO messages, packet delays, waiting time, and packet loss rate were examined. However, this approach was not well-developed and no IDS was presented in that study.

Raza et al. [18] designed a real-time IDS for selective forwarding and sinkhole attacks called SVELTE. This method uses technologies such as IPsec and DTLS to provide security for end-to-end messages. The authors introduced a hybrid, centralized, and distributed approach and located IDS modules both in the 6BR and in constrained nodes. SVELTE features three major centralized components placed in the 6BR. The first component, called 6Mapper, collects information about the RPL and rebuilds the network in the 6BR. The second component, called the intrusion detection component, analyzes the data of the first component. The third component is a small firewall that prevents malicious traffic from entering the network. On the other hand, each node has two central components: the first one provides mapping information and the other operates with the firewall. The authors claimed that the proposed system is capable of detecting all attacks. However, this method involves some false alarms as well. Excessive use of resources and increased network overhead are other disadvantages of SVELTE.

In the Cooja simulator, Cervantes et al. [19] implemented an IDS called INTI to obstruct, identify, and isolate sinkhole attacks. INTI detects malicious nodes by clustering and monitoring the behavior of nodes when they send messages. It includes four components: cluster configuration, routing monitoring, attack detection, and attacker isolation. In the first component, clusters are created to increase the lifespan of the network and its scalability. The second component, having an “observer” node, is in charge of monitoring the routing process by counting the number of transmissions. In this method, it is assumed that if the number of incoming messages of a node is not equal to the number of its output messages, the node is not acting in accordance with normal nodes in the network. The third component detects malicious nodes. INTI uses the trust and reputation mechanism to determine whether a node is an attacker. The fourth component separates the malicious node from the cluster and alarms the neighboring nodes. This method is an improvement on SVELTE. While INTI reduces false positive alerts when the nodes are in motion, it consumes a high amount of resources.

Mayzaud et al. [20] explored two attacks related to topological inconsistency, including a direct scenario and a packet manipulation scenario. In these attacks, the node either sends inconsistent packets directly or changes the header of IPV6 packets. The nodes receive the packets and release incompatible ones, and also resume the timer by means of the Trickle algorithm. As a result, they heighten the overhead of control messages in the network, reduce channel access, and amplify energy consumption. To reduce the attacks, the authors considered fixed, adaptive, and dynamic thresholds. They found that the dynamic threshold performed better than the other two methods, but it did not yield a significant improvement in the case of aggressive attacks and packet delivery ratio.

Surendar and Umamakeswari [21] proposed the InDReS model for detecting sinkhole attacks. Their method included algorithms for leader node selection, calculation of the number of dropped packets, detection of malicious nodes based on evidence, and detection of malicious nodes based on ranking. In this method, the sum of the rank of each node and the minimum rise in rank from the current node to the neighboring node would be compared with the rank of the parent node. They assumed that if this value is less than the parent node, the node in question is considered malicious. Subsequently, it is isolated from the network, an alert message is transmitted to the network, and the network topology is reconstructed. Compared to INTI and SVELTE, this constraint-based method has a significantly higher packet delivery ratio and lower energy consumption as well as overhead, yet it does not show the detection rate.

Kiran et al. [22] proposed a method to detect distributed denial-of-service (DDoS) attacks based on trust and the frequency of packets. This method measures data frequency threshold with regard to data intervals. Each node involved in the data transfer process calculates the number of input packets and evaluates the trust of the resource according to the number of packets passing through the range in question. Neighboring nodes which calculate the amount of reduced trust send an alert message to calculate the ultimate trust based on the logic of data frequency, and they send information about the attacker node to all nodes in the network. Then, all nodes remove the information packet received from the attacker without sending it to the root. Consequently, unnecessary network traffic and energy consumption of nodes are minimized. However, in this method, all decisions must be made by the root instead of the neighboring nodes, because the root node has a higher energy than do other nodes.

Deshmukh-Bhosale and Sonavane [17] designed an IDS for wormhole attacks. They used the received

[DOI: 10.52547/itrc.13.2.17]

signal strength indicator (RSSI) to detect the attack and the attacker node. The system is based on a hybrid approach in which the centralized IDS is located at the root node and the distributed IDS is placed at other nodes. The distributed module includes four steps: neighbor validation, distance calculation, attack identification, and malicious node detection. In the centralized module, each node forwards rank and ID information of its parent as well as children to the root node to identify the attack. The authors obtained an attack detection rate of about 90% for 8 nodes in the network. Meanwhile, for networks with more nodes (e.g., 16 and 24), the detection rate was below 90%, which needs to be improved. Besides, this method does not report the false alarm rate.

Farzaneh et al. [23] provided a distributed anomaly-based IDS which uses threshold values to detect DIS and neighbor attacks. In this way, each node in the network collects its neighbor's information to detect intrusion. This method calculates the maximum number of DIS messages that neighbors receive in the normal mode. This model showed a true positive rate of 100% and a false positive rate of zero for DIS attack, and it is scalable and adaptable to a variety of networks. However, it does not take into account the mobility of nodes in detecting attacks. Also, the results of delay and packet delivery ratio are not reported.

Using three metrics of distance, residual energy, and expected transmission count, Farzaneh et al. [24] advanced a new fuzzy-logic-based approach to detect local repair attacks. The proposed method is independent of time and determines whether an attack has occurred or not based on the state of the nodes. The results of Cooja simulator in the Contiki OS revealed that the method was effective in identifying the desired attack; specifically, it yielded a detection rate of 94.41 and 95.75 after 8 and 12 minutes, respectively.

Khraisat et al. [25] focused on IoT IDS taxonomy and proposed a model based on IDS techniques which provides corresponding advanced IDS and detection capabilities for the detection of IoT attacks. They found that capture the context and purpose of each packet crossing the network is the essential to extract features in IDS. It seems that packets overlapping from different subnet networks represents a significant challenge in extracting the types of features from IoT network. Moreover, since the normal activities change frequently and may not be effective over time and most of the existing machine learning techniques are trained and evaluated on the knowledge provided by the old dataset, which do not include newer malware activities, it is necessary to newer and more comprehensive datasets that consist of a broad spectrum of malware activities.

Baghani et al. [8] introduced the DAO induction attack, in which the malicious node repeatedly amplifies its DTSN to cause unnecessary forwarding of DAO control messages. They also assessed the impact of this attack on network performance and provided a solution for its detection. To identify a malicious node, each child node has a preferred and a non-preferred parent. Therefore, when the malicious node increases its DTSN, any child hearing the DTSN-Incremented DIO message from its non-preferred parent will plan a DAO transmission through its preferred parent instead. As a result, the DAO message cannot be removed by the malicious node; the root then receives the DAO message and detects the attack. However, more than one malicious node should have been considered to ensure the adequacy of this method.

Finally, Ghaleb et al. [9] proposed the SecRPL model to prevent DAO insider attack. This model limits the number of DAO messages sent by parents. To this end, each parent node links a counter to each child node within its sub-DODAG. When the number of DAO messages sent for a child goes beyond a specified threshold (10), the parent throws away the DAO message that contains the respective child's address. To ascertain that no node is blocked because of the time factor, the counter is reset for all children of a parent that is going to forward a DIO message. The threshold used in this model is considered by the same type of network configuration. Therefore, this method cannot ensure network security in the case of topologies with different nodes. Moreover, this study has not provided an algorithm for detecting malicious nodes. In this regard, in order to complete previous models, we design an algorithm that, in addition to identifying malicious nodes, has a better performance in terms of reliability and delay. We propose several scenarios for detecting DAO attacks in the context of IoT. Comparing our results with those of Ghaleb et al. [9], fully discussed in the fourth section, demonstrates the superior performance of our model.

III. PROPOSED SOLUTION

In this section, a new model will be presented to prevent and detect the intrusion of DAO attacks. Since in this paper by building a huge database with different features, using classification algorithms, no relationship was found to detect malicious nodes, therefore, in this study, a method is presented that does not depend on the number of nodes in the network and the distance between each node and the server node. Using Cooja (a cross-level simulator for Contiki operating system), the proposed method collects the required data, which includes the number of DAO packets sent by each child node. By simulating the network, virtualizing network functions, and selecting thresholds for transmitted DAO packets, proposed model identifies DAO insider attacks and prevents them from being forwarded to high-level nodes. In order to design a mechanism for approving the transmission of DAO messages, we consider three threshold adjustment methods: random, fixed, and dynamic. In the random threshold method, the DAO message is responded using a random process. In the two other methods, a value is considered as the threshold number for DAO messages of child nodes to be sent to an upward route.The total number of DAO transmitted message includes the original DAO transmitted messages and forwarding DAO transmitted messages to its root.

The general outline of the proposed method is shown in Figure 3.

In the proposed method, in order to obstruct a DAO insider attack, the machine uses certain algorithms to learn offline the number of DAO packet transmissions in each case. This learning is then converted into a series of threshold values or conditions (fixed threshold

[DOI: 10.52547/itrc.13.2.17]

method); alternatively, the network traffic is checked online and the threshold value is set dynamically (dynamic threshold method). Finally, by selecting the threshold value, DAO insider attack is detected and damage to the entire network is prevented.

We use threshold values for each child node to detect DAO insider attacks. Receiving DAO messages more than the threshold value is an indicator for identifying attacker nodes in the network. Analyzing the rate of DAO packet transmission under different topologies with various numbers of nodes, our method collects the required data through a simulator. It then uses this data to determine the threshold value. Since each node has an overview of its own neighbors alone (rather than all neighbors in the network), the number of messages forwarded so far by child nodes is calculated via examining the number of DAO messages forwarded by neighbors. The general process for determining the threshold value is as follows:

1) Count the number of DAO messages received per all children in each node
2) Calculate the maximum number of control messages received by children in each node (MAX parameter).
3) Select a maximum from the MAX parameter of all nodes in the entire network
4) Select the threshold value from the maximum calculated in 100 different topologies

This dataset includes traffic data collected through Cooja simulator and Tmote Sky, which is similar to MSPSIM emulator. Traffic data were collected from networks with 100 different topologies that were spread with different numbers of nodes but uniformly with a distance of 20, 30, and 40 meters in 3,600 seconds. This dataset consists of the number of DAOs sent by each child node. The amount of forwarded DAO is measured over a specified period of time, i.e. 43 seconds. The dataset is used to determine the threshold in a model which is based on a proposed fixed threshold. The reason for choosing 43 seconds for a single period of time is related to the number of data sent per node. The results show that the maximum number of DAO messages is forwarded in less than 43 seconds. Because no accurate value could be adjusted for the number of DAO transmissions to check values less than 43 seconds, and since the malicious node is detected after a few periods of time, detection takes place at longer times. Therefore, we preferred 43 seconds as an ideal time period. Table I shows the maximum number of DAO messages received from children in different topologies with different numbers of nodes.

TABLE I. AN INSTANCE OF MAXIMUM DAO MESSAGES RECEIVED FROM CHILD NODES IN NORMAL NETWORK STATUS

		Number of Nodes		
Distance		20	40	80
	20m	3	3	3
	30m	4	4	3

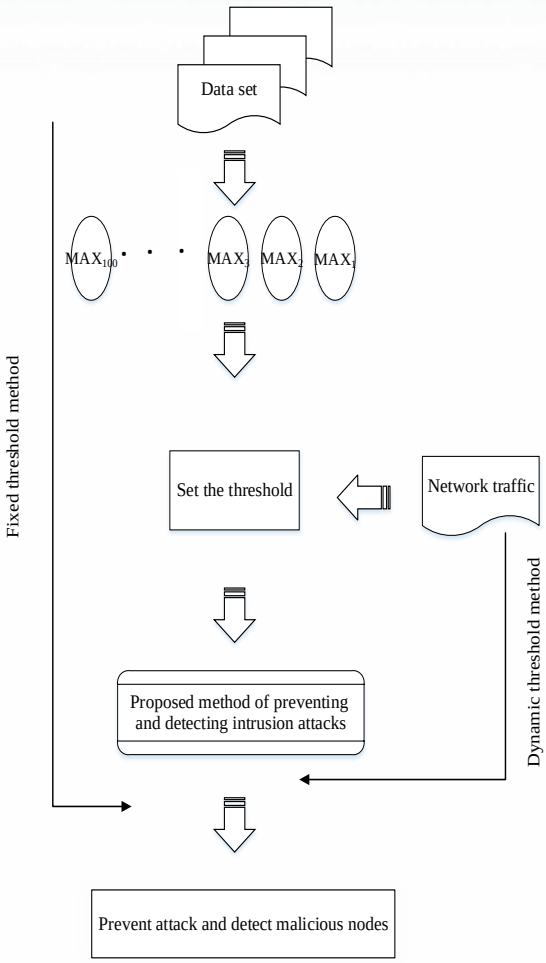

Figure 3. The general outline of the proposed method

To detect attacks that are based on sending a DAO message, we propose an IDS according to algorithm 1, in which child -> DAO_count indicates the number of DAOs received from each child; child -> block_count indicates the number of times it has exceeded the detection threshold; DAO_threshold is considered the threshold value for the transmission of messages by child nodes; and block_threshold indicates that the child node has exceeded the detection threshold twice. The detection algorithm identifies the attack by counting DAO messages. Thus, the number of messages received so far by child nodes is calculated and compared with the predetermined threshold value. If it exceeds the threshold specified for the number of DAO transmissions, it will be considered an attack and the child's DAO forwarding will be avoided. This node is then temporarily marked malicious and will be blacklisted if it exceeds the block_threshold.

[DOI: 10.52547/itrc.13.2.17]

Algorithm 1. Detection of DAO Insider Attack

```
1: For each child in Children list do
2: If ((child ->DAO_count) > DAO_threshold) then
3: If (child ->block_count < block_threshold) then
4: Temporarily block child
5: Else
6: Permanently block child
7: Print (child is an attacker)
8: End if
9: End if
10: End for
```

The method introduced for preventing attacks is based on limiting the number of DAO messages forwarded upward to the root node by other nodes. In each node, an independent counter is provided for each child node that will count the number of DAO transmissions so that if the threshold value is exceeded, the DAO messages of the respective child will not be forwarded to the DODAG root. In order to ensure that a node is not blocked in the "Permission to forward the DAO message" at all times, the counter of each child must be reset to zero after a time window (Algorithm 2).

Algorithm 2: Resetting the counter of DAO transmissions of each child to zero

```
1: Procedure Timer 43 second
2: For each child in Children_list do
3: Child_DAO_Counter = 0
4: End for
5: End procedure
```

The general procedure of the algorithm is as follows:

1) The threshold of the normal network status is determined.
2) In case of exceeding the detection threshold, the message is considered an attack or forgery.
3) The forwarding of DAO messages to the respective child is stopped.
4) A blacklist of nodes identified as malicious is added to each node.
5) In each parent node, if the child node at least twice exceeds the detection threshold defined for the number of DAO transmissions, the node is added to the blacklist and the forwarding of its DAO messages will be blocked.

In the following, we explain the three adjustment methods based on "random threshold", "fixed threshold", and "dynamic threshold" aimed at preventing and detecting DAO attacks.

In the random threshold method, a DAO message is answered using a random process and is directed upward. The proposed method does not depend on the number of nodes and their distance in the network, and the DAO message is answered only in a probabilistic way. As a result, any node attempting to send more DAO messages will be assigned a lower transmission probability.

In the proposed method, we use the rand(0.1) <=1/n condition, in which n represents the number of DAO messages transmission. This condition works based on probability concepts, such that the higher the n value, the lower its probability [26]. The proposed method is shown in Algorithm 3.

Algorithm 3. Random threshold method

```
1: Procedure CHILD'S DAO RECEIVED
2: If rand(0,1)<=1/n then
3: Forward the child DAO
4: Child_DAO_Counter ++
5: Else
6: Discard the child DAO
7: End if
8: End procedure
```

In the proposed fixed-threshold model, adjusting the threshold in each node to a default value in order to forward the DAO messages received from each child node upward to the root node helps mitigate the impact of the attack on the network. By analyzing different networks and according to the obtained data, we set the value of the threshold to 5 by default.

Algorithm 4. Random threshold method

```
1: Procedure INITIALIZATION
2: Set DAO_For_MAX With 5
3: End procedure
4: Procedure CHILD'S DAO RECEIVED
5: If child_DAO_Counter < DAO_For_MAX then
6: Forward the child DAO
7: Child_DAO_Counter ++
8: Else
9: Discard the child DAO
10: End if
11: End procedure
```

In the proposed dynamic-threshold model, the threshold for transmitting DAO messages received from each child node is dynamically adjusted at each time period. This change is based on the criteria applied to receive DAO messages from child nodes. In this method, the maximum data is suggested as the threshold value. This is achieved by defining the MAX parameter and updating it at each period. In this mechanism, the threshold is determined after removing the outliers, i.e. data sent by a malicious node. This disposal in the dynamic method relies on a fixed threshold, such that the data outside the fixed threshold are considered outlier.

Algorithm 5. Dynamic threshold method

```
1: Procedure Timer 43 seconds
2: For each child in Children list do
3: If ((child ->DAO_count) < DAO_threshold) then
4: If ((child ->DAO_count) > DAO_For_MAX)
then
5: DAO_For_MAX = child ->DAO_count
```


[DOI: 10.52547/itrc.13.2.17]

6: **End if**
7: **End if**
8: **End for**
9: **End procedure**
10: **Procedure** CHILD'S DAO RECEIVED
11: **If** child_DAO_Counter < DAO_For_MAX **then**
12: Forward the child DAO
13: Child_DAO_Counter ++
14: **Else**
15: Discard the child DAO
16: **End if**
17: **End procedure**

IV. RESULTS

In this paper, we implemented an IDS in the Contiki operating system and evaluated it using Cooja simulator, which facilitates an RPL implementation that functions on several platforms [27]. The base node for the simulation was Tmote Sky [28], which features 10kB of RAM and 48kB of ROM. It was chosen as the development platform due to the fact that its computational resources enable it to operate as an RPL router node with the Contiki RPL implementation.

We considered networks with a size of 20, 40, and 60 nodes for simulation. Figure 4 illustrates how the nodes are situated in the simulation. where the nodes are uniformly spread over an area of 100 x 100 meters, and the root node is located outside the square area at a distance of 10 meters.

Every node forwards one packet to the root every 60 seconds. Also, in order to obtain packet delivery ratio in downward traffic, the root node sends a response after receiving the packet. To assess the impact of a DAO attack on network capability and performance, malicious nodes send a fake DAO message in a pre-set time window (every 0.5 seconds). In each of these topologies, 10% of nodes in the network randomly act as malicious nodes. Simulation was performed at 1,800 seconds with 5 different seeds for each topology. Table II shows parameters needed for simulation.

In this paper, we evaluate the effect of DAO attack on true positive rate (TPR) and false positive rate (FPR). These metrics are calculated using true positive (TP), false negative (FN), true negative (TN), and false positive (FP).

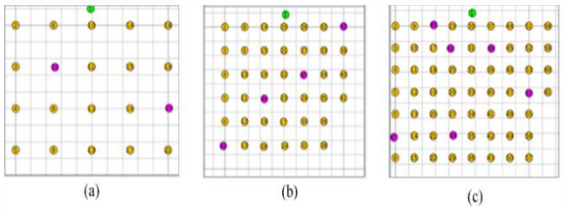

Figure 4. (a), (b) and (c) represent topologies with 20, 40 and 60 nodes, respectively. Green, yellow, and purple nodes respectively indicate server node, legitimate node,

$$TPR=\frac{TP}{TP+FN} \quad (1)$$

$$FPR=\frac{FP}{FP+TN} \quad (2)$$

The TP value indicates the number of correctly identified malicious nodes. The value FN number of malicious nodes classified as legitimate. The TN and FP values indicate the number of legitimate nodes properly classified and the number of legitimate nodes incorrectly classified as malicious, respectively.

- Packet Delivery Ratio (PDR): The ratio of the number of received packets to the total number of sent packets [29].

$$PDR = \frac{\text{Number of Packet received}}{\text{Number of Packet sent}} \quad (3)$$

Packet delivery ratio in the upward direction (Upward PDR) is the ratio of the average number of packets received by the root node to the number of packets sent by the network nodes. On the other hand, packet delivery ratio in the downward direction (Downward PDR) is the average ratio between the number of packets received by the nodes to the total number of responses sent by the root node.

TABLE II. NETWORK PARAMETERS USED FOR SIMULATION ANALYSIS

Simulation parameters	Value
Operating System	Contiki 3.0
Simulator	COOJA
Size of Deployment Area	100 × 100 m
Number of Nodes	20, 40, and 60
Mode of operation	Non-storing
Number of Roots	1
Distance from each Root Node to a Typology	10 m
Number of Simulations	5 per each topology
Transmission Rate of Control Packets of the Malicious Node	0.5 s
Onset of the Malicious Node's Attack	120 s After Starting Simulation
Radio Medium Model	Unit Disk Graph Medium (UDGM): Distance Loss
Range of Nodes	Rx and Tx: 20 m, 30 m
Mote Type	Tmote Sky
Duty Cycle	ContikiMAC
Physical Layer	IEEE 802.15.4
MAC Layer	ContikiMAC, IPV6
MAC Driver	CSMA/CA
Network Layer	ContikiRPL
Network Driver	Sicslowpan
Transport Layer	UDP
Traffic model	Constant Bit Rate
Simulation time	1800 s

[DOI: 10.52547/itrc.13.2.17]

• Average end-to-end delay: the average time interval between sending a packet by the sender and receiving it by the receiver [30]. It is calculated using Relation (4): In this relation, AVG Delay is average end-to-end delay, RecT is packet reception time, SendT is packet delivery time, and n is the total number of packets received.

$$\text{AVG Delay} = \frac{\sum_{i=1}^{n} \text{RecT} - \text{SendT}}{n} \qquad (4)$$

Upward latency (seconds) is the average end-to-end delay of all packets sent by nodes and received by the root node, and downward latency (seconds) is the average end-to-end delay of all response packets sent by the root node and received by the network nodes.

• Average energy consumption of network nodes: Due to the low power of the components of the Internet of Things, the amount of energy consumed is an essential consideration in these networks. Therefore, providing an algorithm with minimum energy consumption is one of the important evaluation criteria in this field. The average energy consumption of network nodes is calculated using Relation (5) [31]

$$Power(mW) = \frac{Energy(mJ)}{Time(s)} \qquad (5)$$

$$Energy(mJ) = \frac{\begin{pmatrix} Transmit * 19.5mA \\ +Listen * 21.8mA \\ +CPU * 1.8mA \\ +LPM * 0.0545mA \end{pmatrix} * 3\ V}{4096 * 8} \qquad (6)$$

In Relation (6), Time is the amount of time in seconds needed to perform the simulation, and Energy is the value derived from the PowerTrace tool. Other parameters are as follows:

Transmit indicates the number of clock ticks during which the node's radio chip was sending data.

Listen indicates the number of clock ticks during which the node's radio chip was listening.

CPU indicates the number of clock bits during which the node was performing computational processes.

LPM indicates the number of clock ticks spent in low power mode.

Figure 5 shows the detection rate of the proposed methods. In these mechanisms, TPR is 100% and FPR is zero, demonstrating the success of proposed methods.

TPR determines what percentage of network traffic detects by the designed mechanism, correctly. In the scenario of this study, different modes for the network are investigated but the only part of it is mentioned in the article due to repetition. One of the reasons for the accuracy of traffic separation is to check 100 different modes for the network (with different sizes and different nodes distance) to get the threshold for transmitting DAO messages. Therefore, it is not possible to send more than the mentioned threshold. When the malicious node starts its operation, the transmitting DAO messages exceeds this threshold and therefore, it is possible to detect the malicious node.

The error rate of zero suggests that the dataset tested for determining threshold is complete and comprehensive. On the other hand, the zero error is due to allowing a maximum of twice exceeding the threshold, so that normal nodes are not considered malicious in the event of a network error. It should be noted that in the method introduced by Ghaleb et al. [9] no mechanism was provided to identify the malicious node and separate it from normal network nodes so as to compare its TPR with the methods proposed in the present study.

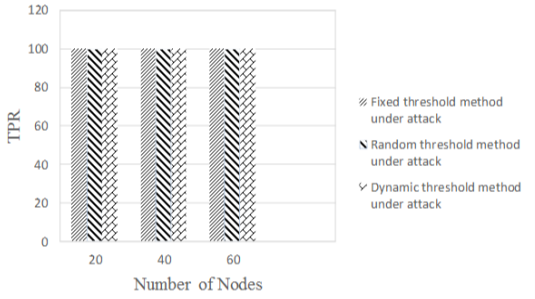

Figure 5. The Detection of DAO Insider Attack

Figure 6 and Figure 7, respectively, provide a comparison of packet delivery ratios in upward and downward routing approaches for five models in 1,800 seconds of simulation. The results confirm that our proposed methods feature a higher delivery ratio in both downward and upward routing compared to the method suggested by Ghaleb et al [9]. The results show that the PDRs of both traffic patterns are negative and their impact is proportional to the number of attackers on the network. In upward routing in the topology with 60 nodes, all three methods of fixed, random, and dynamic threshold adjustment succeeded in, respectively, improving PDR performance by 0.8353, 0.8168, and 0.8522 seconds, compared to the method of Ghaleb et al [9].

As for downward routing, comparing the results of packet delivery ratio in our study and that of Ghaleb et al. [9] demonstrates that the number selected in that study for topology with different numbers of nodes in the network is not appropriate. In topologies with 20, 40, and 60 nodes in the fixed threshold method, the results revealed that PDR performance improved by 0.9952, 0.9599, and 0.8792 seconds, respectively, which confirms the superiority of our method to that of Ghaleb et al [9]. Also, in the dynamic threshold method, PDR performance improved by 0.9939, 0.9522, and 0.8683 seconds in topologies with 20, 40 and 60 nodes, respectively. Finally, in topologies with 20, 40, and 60 nodes in the random threshold method, PDR performance improved by 0.9942, 0.9572, and 0.8739 seconds, respectively, implying the greater effectiveness of our method compared to that of Ghaleb et al [9].

[DOI: 10.52547/itrc.13.2.17]

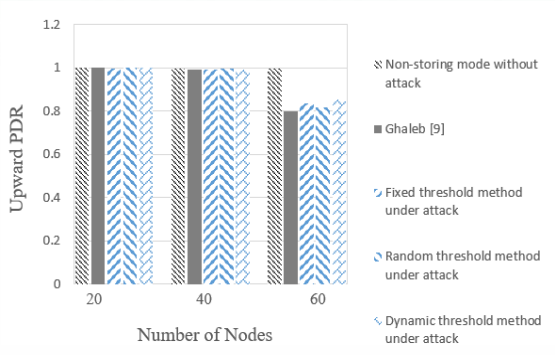

Figure 6. Comparison of the proposed methods with the Ghaleb et al's method in terms of package delivery ratio in upward routing

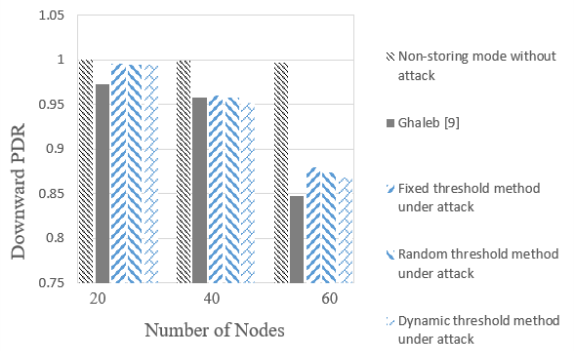

Figure 7. Comparison of the proposed methods with Ghaleb et al's method in terms of package delivery ratio in downward routing

Figures 9 and 10 show the average end-to-end delay in upward and downward routing associated with the models under comparison. This delay was affected by DAO insider attack, which is accompanied by the congestion of control messages. As can be seen, the proposed algorithms have a lower average end-to-end delay in upward and downward routing than does the method proposed by Ghaleb et al [9]. In the fixed threshold method, topologies with 40 and 60 nodes have an upward delay of 0.7907 and 2.3834 seconds, and a downward delay of 0.9595 and 1.2316 seconds, respectively. For topologies with 40 and 60 nodes, the dynamic threshold-based method reduced upward delay by 0.7818 and 2.3327 seconds and reduced downward delay by 1.001 and 1.2187 seconds, respectively. Similarly, for topologies with 40 and 60 nodes, the random threshold method reduced upward delay by 0.7191 and 2.3185 seconds, and reduced downward delay by 0.9127 and 1.1884 seconds, respectively.

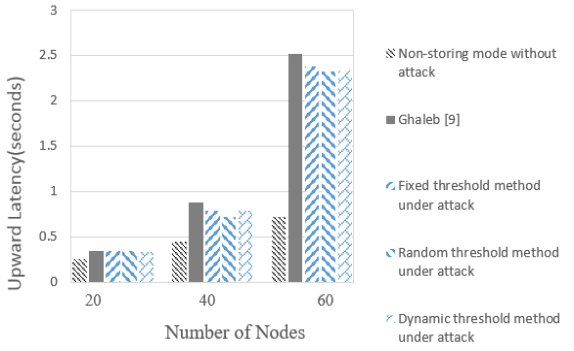

Figure 8. Comparison of the proposed methods with Ghaleb et al's method in terms of the average end-to-end delay in upward routing

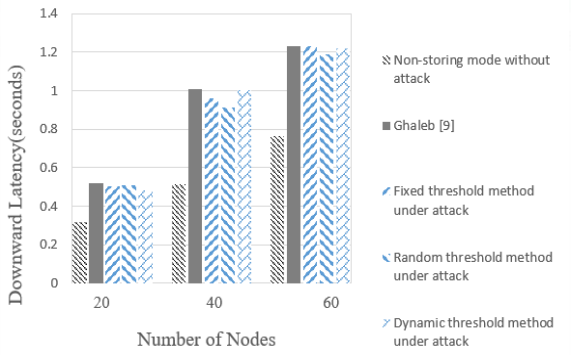

Figure 9. Comparison of the proposed methods with Ghaleb et al's method in terms of the average end-to-end delay in downward routing

Comparing the average energy consumption between different nodes in 1,800 seconds since the beginning of simulation (Figure 10) shows that the congestion of control packets has affected their energy consumption. As can be seen, the proposed methods have a lower average energy consumption compared to the method introduced by Ghaleb et al [9]. The energy consumption of the whole network will be influenced by the number of nodes involved in the transmission process; hence, it is proportional to the location of the malicious node. Specifically, the farther away the malicious node is from the network, the more intermediate nodes will be involved in the transmission process and the greater their impact will be on the energy consumption of the entire network.

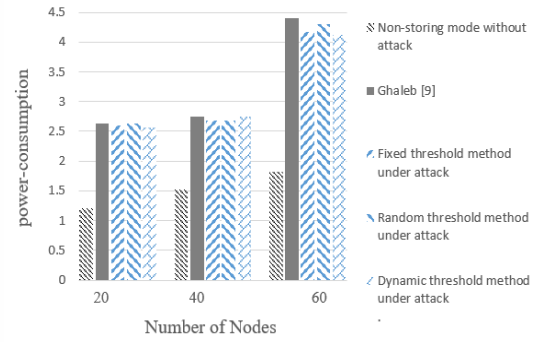

Figure 10. Comparison of different methods in terms of average energy consumption

V. Conclusion and Recommendations for Future Research

Providing an intrusion detection system is a solution to overcome common network attacks in the IoT environment and other attacks specific to RPL. One of these attacks is DAO, for which no adequate and comprehensive detection method has been introduced in previous studies to distinguish malicious nodes from normal nodes.

In summary, the attack can be described as follows. In terms of downward routing, RPL operates in both storage and non-storage modes. However, the graph can only function in one mode. In non-routing save mode, the table entries are refreshed based on a DAO message. When the timer associated with the trickle algorithm

[DOI: 10.52547/itrc.13.2.17]

expires, the node collects routing information and sends a new DAO message to its parent set. This process is repeated until the DAO messages reach the root node. In a DAO internal attack, however, the attack node on the network, known as the malicious node on the network, alternately sends DAO messages to its parent node. Thus, the network is exposed to a large volume of DAO messages and the efficiency, power consumption, latency and reliability of the network are severely affected. Unlike other attacks, the level of damage is very high because it is not limited to the current range of the destructive node and DAO messages are transmitted end-to-end from the sensor node to the root.

In this context, we developed a method to identify and prevent DAO attack with the following objectives:

1) Upgrade network performance through detecting and limiting DAO attacks from malicious nodes
2) Provide an adaptable method without assuming specific network configurations or conditions,
3) Provide a dynamic method to detect an attacking node based on online data.

In order to implement a method for detecting DAO attack, three methods are proposed. In the first proposed method, which is based on the random method, it probably responds to the DAO message. If a node wants to send more DAO, it will be less likely to be answered by the network nodes because according to the formula provided in this method, it will be less likely to send that number.

To use the next two methods, another database was needed. In this database, the maximum number of DAO messages sent by each child node is stored in different network modes. In these methods, the number of DAO messages sent by each node in 100 different networks with different characteristics such as network size and variable distance of nodes together in the network was examined and the maximum number of DAO messages sent by nodes was obtained. The method based on fixed and dynamic threshold was presented that the dynamic method is more complete than the fixed threshold and the variable threshold limit for nodes is considered. As a result, in the next methods, according to the different network conditions, the number of submissions is investigated and the threshold for sending nodes is considered.

The results of the evaluations exhibited that the proposed methods outperformed those of previous studies in obstructing DAO attacks. The proposed model is able to rapidly analyze DAO packets in real time sent in normal networks and detect the malicious node. The proposed methods have a detection rate of 100% and a false positive rate of zero, indicating the success of the proposed algorithms in detecting DAO attacks.

The results of the network attack mode showed how the DAO attack disrupted the Internet of Things, as well as the severity of the attack on the network. On the other hand, the results of experiments showed that the proposed model is able to analyze data in real time and in a limited time and detect malicious nodes. In this regard, the proposed methods showed a detection rate of 100%.

However, the result of the study by Farzaneh et al. [23] is close to the proposed method which is based on a fixed threshold but this approach does not report the results of delay and package delivery rate. Moreover in compare with [24], the proposed method do not depend on any conditions and are implemented without considering the network conditions while the study by Farzaneh et al. [24] depends on the network conditions. It should also be noted that the methods presented by them are DIS attack; and each attack is executed completely differently in the network The DAO attack level goes far beyond the network and wastes a lot of resources, while the DIS attack only affects the neighbor.

In line with the future works, threshold determination can be considered based on other statistical methods. Experiments have shown that considering the rank of the node is very effective in determining the threshold. So intervals can be considered for different ranks and a specific threshold can be set for each interval using the statistical methods mentioned.

Moreover, according to the experiments, network parameters such as the number of neighboring nodes and node rank have a great effect on determining the threshold.

In addition, to the attack for the DAO message, a malicious node can capture DAO messages sent from lower nodes and discard the packet instead of moving them to the root. So, the nodes in the downward routing fail to receive packets. Therefore providing a solution for this attack could be another case in future research.

REFERENCES

[1] C. Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses," *IEEE Internet of Things Journal*, 2020.

[2] H. Bypour, M. Farhadi, and R. Mortazavi, "An Efficient Secret Sharing-based Storage System for Cloud-based Internet of Things," *International Journal of Engineering*, vol. 32, no. 8, pp. 1117-1125, 2019.

[3] K. Chugh, L. Aboubaker, and J. Loo, "Case study of a black hole attack on LoWPAN-RPL," in *Proc. of the Sixth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Rome, Italy (August 2012)*, 2012, pp. 157-162.

[4] H. Garg and M. Dave, "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019: IEEE, pp. 1-6.

[5] H. Nasiraee and M. Ashouri-Talouki, "DoS-Resistant Attribute-Based Encryption in Mobile Cloud Computing with Revocation," *International Journal of Engineering*, vol. 32, no. 9, pp. 1290-1298, 2019.

[6] L. Dong and R. Li, "RPL based Named Data Routing Protocol for Low Power and Lossy Wide Area Networks," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019: IEEE, pp. 442-447.

[7] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A study of RPL DODAG version attacks," in

[DOI: 10.52547/itrc.13.2.17]

IFIP international conference on autonomous infrastructure, management and security, 2014: Springer, pp. 92-104.
[8] A. S. Baghani, S. Rahimpour, and M. Khabbazian, "The DAO Induction Attack Against the RPL-based Internet of Things," *arXiv preprint arXiv:2003.11061*, 2020.
[9] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, and L. Mackenzie, "Addressing the DAO Insider Attack in RPL's Internet of Things networks," *IEEE Communications Letters*, vol. 23, no. 1, pp. 68-71, 2018.
[10] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *rfc*, vol. 6550, pp. 1-157, 2012.
[11] J. Hui, "RFC6553 Option for Carrying RPL Information in Data-plane Diagrams," *RFC 6553*, vol. 33, pp. 3-8, 2012.
[12] S. Kalyani and D. Vydeki, "Survey of Rank Attack Detection Algorithms in Internet of Things," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018: IEEE, pp. 2136-2141.
[13] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25-37, 2017.
[14] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained IoT devices," *Mechanical Systems and Signal Processing*, vol. 136, p. 106436, 2020.
[15] A. Tabassum, A. Erbad, and M. Guizani, "A Survey on Recent Approaches in Intrusion Detection System in IoTs," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019: IEEE, pp. 1190-1197.
[16] M. Ahmadi and S. Jameii, "A Secure Routing Algorithm for Underwater Wireless Sensor Networks," *International Journal of Engineering*, vol. 31, no. 10, pp. 1659-1665, 2018.
[17] S. Deshmukh-Bhosale and S. S. Sonavane, "A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things," *Procedia Manufacturing*, vol. 32, pp. 840-847, 2019.
[18] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661-2674, 2013.
[19] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015: IEEE, pp. 606-611.
[20] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "Mitigation of topological inconsistency attacks in RPL-based low-power lossy networks," *International Journal of Network Management*, vol. 25, no. 5, pp. 320-339, 2015.
[21] M. Surendar and A. Umamakeswari, "InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016: IEEE, pp. 1903-1908.
[22] V. Kiran, S. Rani, and P. Singh, "Trust Based Defence System for DDoS Attack Detection in RPL over Internet of Things," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, vol. 18, no. 12, pp. 239-245, 2018.
[23] B. Farzaneh, M. A. Montazeri, and S. Jamali, "An Anomaly-Based IDS for Detecting Attacks in RPL-Based Internet of Things," in *2019 5th International Conference on Web Research (ICWR)*, 2019: IEEE, pp. 61-66.
[24] B. Farzaneh, M. Koosha, E. Boochanpour, and E. Alizadeh, "A New Method for Intrusion Detection on RPL Routing Protocol Using Fuzzy Logic," in *2020 6th International Conference on Web Research (ICWR)*, 2020: IEEE, pp. 245-250.
[25] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1-27, 2021.
[26] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 6, pp. 2193-2196, 2012.
[27] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th annual IEEE international conference on local computer networks*, 2004: IEEE, pp. 455-462.
[28] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 2005: IEEE, pp. 364-369.
[29] E. A. Shams and A. Rizaner, "A novel support vector machine based intrusion detection system for mobile ad hoc networks," *Wireless Networks*, vol. 24, no. 5, pp. 1821-1829, 2018.
[30] M. Conti, P. Kaliyar, M. M. Rabbani, and S. Ranise, "Attestation-enabled secure and scalable routing protocol for IoT networks," *Ad Hoc Networks*, vol. 98, p. 102054, 2020.
[31] B. Farzaneh, A. K. Ahmed, and E. Alizadeh, "MC-RPL: A New Routing Approach based on Multi-Criteria RPL for the Internet of Things," in *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2019: IEEE, pp. 420-425.

Nadia Jahantigh received the B.Sc. and M.Sc. degrees in Computer Engineering from the University of Sistan and Baluchestan, Zahedan, Iran. Her current research interests are Internet of Things, Security, Big Data Mining and Machine Learning.

Ahmad Bakhtiyari Shahri received his Ph.D. degree in Computer Science from "Universiti Teknologi Malaysia" (UTM), Malaysia; and M.Sc. degree in Electrical Engineering Telecommunications from Imam Hossein University, Tehran, Iran. His research interests include Security and Cyber Security, Information and Communication Technolog and ICT Policy.

[DOI: 10.52547/itrc.13.2.17]