Trust and Strainer Based Approach for Mitigating Blackhole Attack in 6LoWPAN: A Hybrid Approach

Himanshu B. Patel, Devesh C. Jinwala

***Abstract*—The conventional watchdog-based blackhole attack mitigation approaches require that all nodes in the network run in the promiscuous mode and result into the increased overhead. In this article, we improve the efficacy of mitigating the blackhole attacks in the RPL (Routing Protocol for Power and Lossy Networks). We analyze two approaches, viz. SIEWE (Strainer based Intrusion Detection of Blackhole in 6LoWPAN for the Internet of Things) and T-SIEWE (Trust and Strainer based Intrusion Detection of Blackhole in 6LoWPAN for the Internet of Things). SIEWE and T-SIEWE are based on statistically limiting the number of nodes in the network to be monitored – by identifying and filtering out the suspicious nodes. Our experimental evaluation and analysis show that as compared to other watchdog-based approaches and as compared to SIEWE, T-SIEWE optimally improves the packet delivery ratio of the system and accurately detects the Blackhole attack while entailing lesser memory and energy overhead.**

***Index Terms*—Blackhole Attack, 6LoWPAN, RPL, Intrusion Detection, Computational Trust, IoT.**

I. Introduction

The Cyber Physical Systems (CPS) touted as the Industrial Revolutions-4.0, exhibit three principle trends viz. high rate of device/data proliferation, integration at scale and autonomy in operation. As a result, the applications of the CPSs range across a wide horizon – from that in industrial automation, health care & medicine, electric power grid, agriculture, energy, smart cities automotive telematics, industrial process control, transportation, defense systems and tele-physical operations [1, 2, 3, 4]. However, due to the pervasive and ubiquitous nature of deployment and the resource constrained nature of the principle actors of the CPS viz. the Internet of Things (IoT) systems, ensuring the security and privacy of CPS is non-trivial [5].

The Low power and Lossy Networks (LLNs) form an important segment of the IoT. Typical IoT system protocol stack is based on the 6LoWPAN (IPv6 over Low Power Personal Area Networks) - the communication stack defined in RFC 6282 by the Internet Engineering Task Force (IETF). The 6LoWPAN stack ensures interoperability and IPv6 support for the resource constrained IoT devices. The 6LoWPAN adaptation layer allows IPv6 packets to be carried efficiently within small link layer frames similar to the one defined by IEEE 802.15.4 [6].

Manuscript received March 16, 2021; revised August 20, 2021.

Himanshu B. Patel is research scholar at the Department of Computer Science and Engineering, S. V. National Institute of Technology-Surat-395007, Gujarat, India. e-mail: hims247@gmail.com.

Devesh C. Jinwala is Professor at the Department of Computer Science and Engineering, S. V. National Institute of Technology-Surat-395007, Gujarat, India. e-mail: dcj@coed.svnit.ac.in

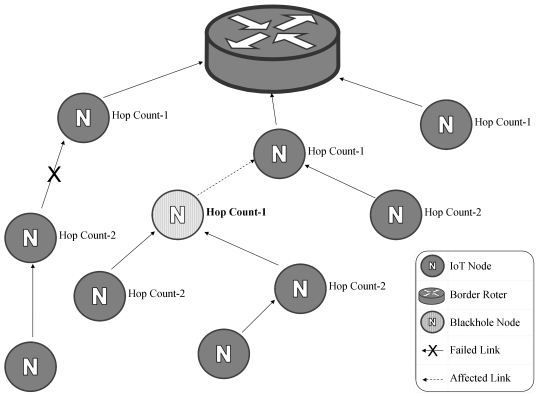


Fig. 1: Blackhole Attack on 6LoWPAN Network

One of the major issues in IoT deployment is routing. The IoT deploys a large number of resource-constrained constituent network nodes (typically, 50 billion [7]), with the constraint that the routing protocol therein is to be designed to operate with minimal overhead. The routing protocol used in the 6LoWPAN stack is RPL [8]. RPL supports optimized routing with respect to different routing metrics such as proximity with respect to the border router in terms of hop-count (Figure 1), reliable packet delivery, and energy overhead/residual energy at each node [9, 10]. However, RPL is susceptible to the Denial of Service attacks [11]. One of the prominent attacks therein is the blackhole attacks. In a blackhole attack, a malicious node in the network publicizes itself as a node having the shortest route to the destination node, i.e., to the border router. Figure 1 presents a typical scenario that depicts how a blackhole node can influence one of the routing metrics (i.e. hop-count) to gain more numbers of descendants in tree topology formed in the RPL protocol. We classify the approaches for mitigating the blackhole attacks in the following categories.

(a) Cryptographic approaches - that encrypt control messages [12, 13, 14].
(b) Acknowledgment based approach [15].
(c) Customized lightweight IDS based approach [16].
(d) Statistical approaches based on game theory [17] or machine learning [18].
(e) Watch-dog based approaches [19, 20, 21, 22, 23, 24].

The approaches proposed in the literature to mitigate the blackhole attacks either entail high computational overhead and latency [12, 13, 14, 15], high communication overhead [15, 16], or necessitate that each node in the network operates in the promiscuous mode resulting in higher communication

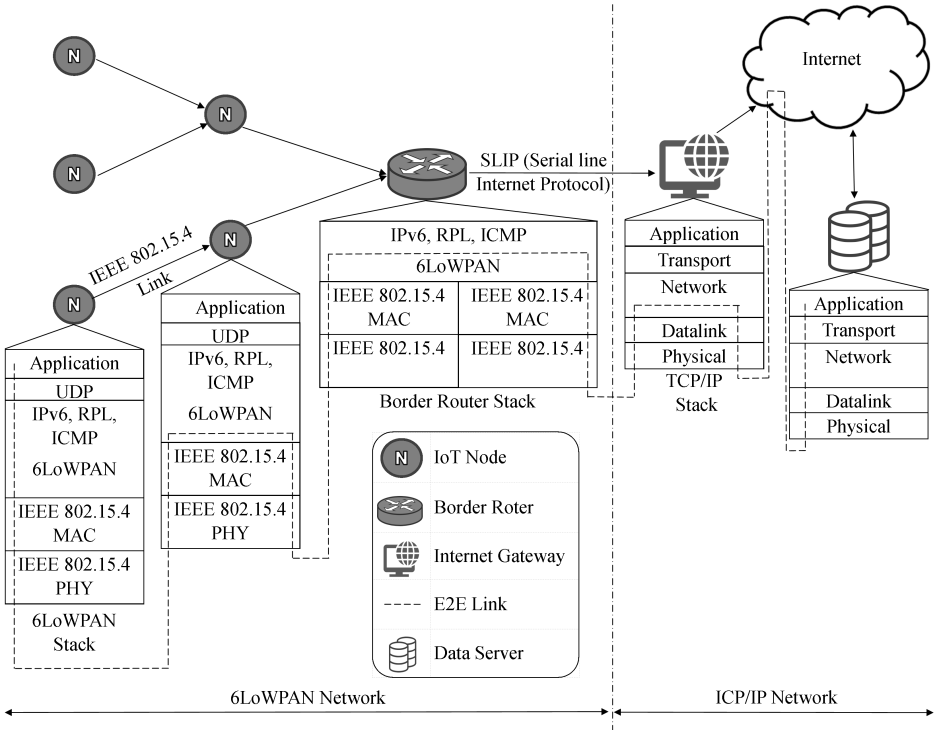


Fig. 2: IoT Communication Architecture

and energy overheads [19, 20, 21, 22, 23, 24].

Patel and Jinwala [25] proposed a proof-of-concept to demonstrate that a probabilistic watchdog based approach does not necessitate that each node in the network runs in the promiscuous mode and could be more efficient as compared to the watch-dog based approaches. Nevertheless, the approach [25] therein, tends to fall back to the watchdog-based approaches, with respect to the number of nodes required to operate in the promiscuous mode. The approach [25] employs a statistical mean value-based criterion, as a filter mechanism, to reduce the number of nodes required to be in the promiscuous mode. As per our evaluation in Section IV-E2, the same negatively impacts the true positive rate of filtering mechanism.

In this article, we critically evaluate and analyze the approach [25] with additional metrics of evaluation and discuss the issues therein. In addition, we propose, analyze and evaluate a T-SIEWE (Trust and Strainer based Intrusion Detection of Blackhole in 6LoWPAN for the Internet of Things). T-SIEWE is a trust-based approach that is more efficient and reliable and characterized by the following:

(a) In T-SIEWE, we use three different filtering criteria based on different combinations of mode value and standard deviation from the routing metric of a node [26], instead of using the statistical mean value-based criteria employed in [25].
(b) In T-SIEWE, we employ a trust value associated with the suspect node, computed collaboratively by the neighbor nodes. The trust values computed are used by the border router to generate reputation value (section IV-C3, IV-C5) of suspect node, and recognize a node as a blackhole node or otherwise.

We implement our approach in Contiki operating system [27] and simulate it with emulated Sky motes on Cooja [28] simulator with 16 and 32 nodes topology. We compare our findings with typical RPL execution scenarios [16, 25]. The evaluation and analysis using different metrics of evaluation (Section IV-D) lead us to conclude that T-SIEWE reduces the overall communication, improves energy efficiency, improves the packet delivery ratio, incurs negligible memory overhead, while improving the detection of the blackhole nodes.

II. Theoretical Background

The Low-power and Lossy Networks (LLNs) support point-to-point (Node-to-Node), point-to-multipoint (Border router to Nodes/Downwards) and multipoint-to-point (Nodes to Border router/Upwards) traffic patterns with multi-hop routing mechanism. The RPL protocol provides a mechanism to disseminate information over the LLNs, where topology formation is dynamic [29]. In this section, we discuss how the RPL protocol execution is vulnerable to the blackhole attacks.

While execution, RPL forms a tree based topology termed as Destination Oriented Directed Acyclic Graph (DODAG), that is rooted at the BR (Border Router) node. In the LLNs, depending upon the application requirements and overlapping domains, multiple DODAGs can be formed within a single network; each of them are rooted at different BR nodes [8].

In Figure 3.a, we show the very first step of DODAG formation. Here, R1 (BR) having IPv6 address aaaa::1 initiates DODAG formation process by flooding DIO (DODAG Information Object) packets in the network. A DIO packet contains basic information about current DODAG. Figure 4 shows DIO packet format.

After receiving DIO packets from the BR, nodes residing in the range of BR calculate their own ranks based on rank value specified in DIO and specific to the routing metric using objective function. Table I shows routing metric defined by RPL specification [9]. After the rank is calculated,

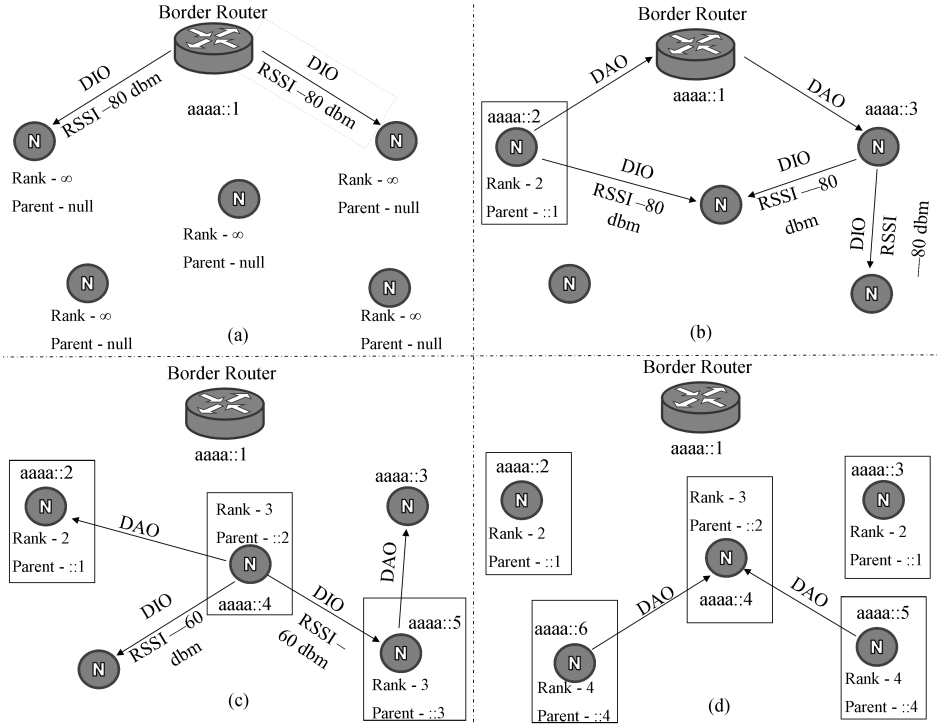


Fig. 3: RPL Execution

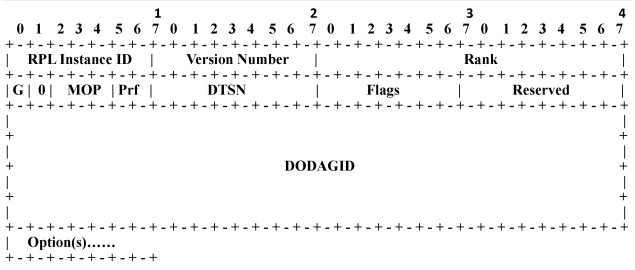


Fig. 4: DIO Packet Format

nodes position in DODAG is final, and it joins DODAG considering sender of DIO packet as parent and send DAO (Destination Advertisement Object) to parent node as an acknowledgment to DIO packet.

As shown in figure 3.B, every node that has successfully joined DODAG further disseminates DIO messages so that remaining nodes can join the DODAG, considering the present node as their parent.

In Figure 3.C and D, we show how routing metric value affects DODAG formation and parent selection procedure. Node having IPv6 address aaaa::5 receives a DIO packet from aaaa::3 having RSSI(Received Signal Strength Indicator) value -100 dbm and joins DODAG considering aaaa::3 as a parent and sets its own rank as 3. When a node aaaa::5 receives a DIO packet from aaaa::4 with better RSSI value, it switches its parent node and recalculates its rank value to 4. The mechanism used to optimize the DODAG also creates an opportunity for an attacker to attract nodes to join DODAG as a child node. In Figure 1, we show how an attacker can gain promising position in DODAG (it becomes a parent node over a large sub-tree of DODAG) by routing metric manipulation. Once a node has promising position, the node can affect accuracy and decision making capability of deployed system by dropping all the data packets of the children nodes included in its sub-tree.

III. Related Work: Blackhole Mitigation Techniques

The approaches to mitigate the blackhole attacks can be categorized as follows.

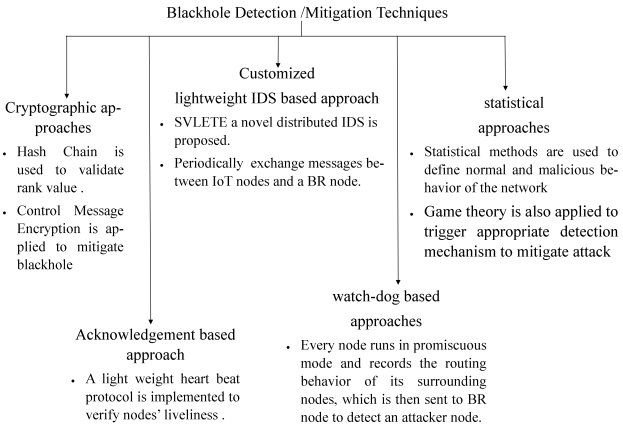


Fig. 5: Blackhole Detection/Mitigation Techniques

(a) Cryptographic approaches used to encrypt control messages [12, 13, 14].
(b) Acknowledgment based approach [15].
(c) Customized lightweight IDS based approach [16].
(d) Statistical approaches based on game theory [17] or on machine learning techniques [18, 21].
(e) Watch-dog based approaches [19, 20, 22, 23, 24].

The cryptography based approaches [12, 13, 14] for blackhole detection use either symmetric-key or asymmetric-key

TABLE I: Routing Metrics

Sr No	Routing Metric	Description
1	NSA (Node State and Attribute)	Represents node's characteristics such as CPU overload, Available memory.
2	Node Energy	Calculated based on energy related indicators like node's energy source, remaining battery life etc.
3	Hop Count	Specifies hop distance of a node from BR(Border Router) node.
4	Delay estimation	Sum of latency values of all link involved
5	ETX (Expected Transmission count)	Estimated numbers of transmission required for a packet to reach at destination node.
6	Physical layer metrics	They are calculated and provided by radio chip upon packet reception. RSSI (Received Signal Strength Indicator) and LQI (Link Quality Indicator) are currently used.

cryptography to hide the sensitive information in the RPL protocol execution. In the cryptography based approaches, either entire packet is encrypted (to prevent discerning between the data and the control packets of RPL) or the control packets are encrypted preventing further inference to the attacker [12, 13, 14]. However, the cryptography-based approaches incurs significant computation and/or communication overhead.

The acknowledgment-based approaches [15] necessitate that every node in the network must certify the proof of their liveness to the BR node at regular intervals using an special heartbeat packet. The BR node uses the received liveness information to periodically broadcast the status of those nodes that are not live; thereby allowing a network node to treat the parent node in its path to the BR node as a suspect and alter the path, in future communications. The acknowledgment-based approaches incurs significant communication overhead. In addition, only a few colluding nodes (e.g. in a Sybil attack) may render such approaches ineffective.

The customized IDS based approach [16] uses an intrusion detection system module executing at the BR node that relies on the signature and anomaly-based intrusion detection to flag a blackhole node. The IDS modules executing in the BR node rely on periodic information about the status of network nodes. Therefore, such approaches also entail higher communication overhead.

The statistical approaches use a hierarchical clustering (i.e. cluster-based approach) to collect information about the ongoing communication in the network. Instead of a single node (e.g. BR node) collecting all such information, a hierarchy of cluster-head nodes and observer-nodes (within a cluster) is used to gather the information, apply statistical functions (e.g. Dempster Shafer estimation) locally, to detect the compromised node and communicate the results to the BR node. The other alternative is to use a game theory based approach [17] with a game set up between legitimate nodes and an attacker nodes. However, the statistical approaches not only incur significant communication overhead, but also introduce significant computation overhead. Therefore, such approaches reduces the lifetime of the network. Authors [18] analyze the packet dropping attacks in cooja simulator and generate the PCAP file. A statistical learning technique is used to train the model and to detect attacks during the network operations.

The watchdog-based approaches [19, 20, 22, 23, 24] not only rely on collecting the communication information in the network from each node, but also expect a node to overhear the communication patterns of the neighbor nodes and forward the information to the BR node. The watchdog-based approaches use a trust value to identify the malicious nodes in the network. For example, the approach [20] detects of blackhole based on the calculation of trust values. In this scheme, authors consider a static topology. In addition, every node in the network is expected to run in the promiscuous mode, enabling them to overhear the transmissions of neighboring nodes. A black hole node typically drops all packets that are expected to be forwarded towards the BR node. The trust values of each neighboring node is calculated by a node using one or more combinations of the routing metrics (e.g. RSSI, LQI, energy, packet forwarding nature of the node etc.) along with the link quality metric (e.g. ETX - expected transmission time). Each node uses the calculated trust values to decide the parent node.

Khan et al. [22] show how trust can be embedded into the RPL routing decision. In this scheme, trust values like belief, disbelief, and uncertainty are calculated based on the positive and negative experiences of nodes. Here, each node in the network calculates positive and negative experiences of other nodes in proximity, depending upon their rank property, version number, and packet forwarding nature. Authors demonstrate three dissemination schemes, namely, Neighbor Based Trust Dissemination (NBTD), Clustered Neighbor Based Trust Dissemination (CNTD) and Tree Based Trust Dissemination (TTD). The TDD puts a constraint that a node can only supervise its parent node, thereby leaving the leaf nodes unsupervised. Authors argue that the effect of Blackhole node at the leaf level is not very significant. The approach reduces the network overhead.

Djedjing et al. [19] proposed a new routing metric named RPL Node Trustworthiness (RNT). RNT is used to reinforce the trustworthiness during RPL DODAG formation. Here, trust values are calculated based on some observation values that include honesty, energy, and unselfishness of the nodes. All the control messages exchanged during RPL DODAG formation use encryption for security. However, existing mechanisms appear to be costly on resource-constrained IoT nodes. Hence, authors have proposed the use of the Trusted Platform Module (TPM) chip to offload all complex computations. ERNT [23], an extended version of RNT,

calculates direct and indirect trust values using the TPM chip.

An Artificial Intelligence based Packet Drop Ratio (AIPDR)[24] uses neighboring information collected by nodes in promiscuous mode to generate a model at central node. The information is analyzed and used to detect selective forwarding attack. The approach improves PDR and end-to-end delay of the system.

In this article, we proposed an approach that reduces the number of nodes required to perform the computation of trust values.

IV. T-SIEWE: The Proposed Approach

In order to dissect the logic of the SIEWE and the T-SIEWE approaches for the blackhole mitigation, we first mention the scope of the 6LoWPAN model, the network, communication and the security model used in further discussions.

A. 6LoWPAN System Model

We define the 6LoWPAN system model used in our proposal for evaluation from physical, communication, and attack perspective. The physical Model establishes the distribution of nodes in the area and their capabilities. The communication model defines DODAG formation and interface patterns, and the attacker model sets the ability of an attacker and its capacity to disrupt network operation.

1) Physical Network Model: 6LoWPAN Network accommodates n similar devices(nodes). The entire set S of physical devices can be represented, as shown in the equation 1.

$$S = \{BR, N_1, N_2, N_3, \ldots, N_{n-1}\} \tag{1}$$

Each node N_i where $i \leq n-1$ and $N_i \in S$ has a unique IPv6 address. All the nodes in the present network are based on the IEEE 802.15.4 specification and are resource-constrained; except the Border router (BR) node. In addition, each one of them is assumed to have a limited range of 50 meters and has a bandwidth of 250 Kbit/s. We consider that the $BR \in S$, communicates through IEEE 802.15.4 defined communication protocol, and also capable of doing cost-effective computations. The Border router node is the one from where the DODAG formation process starts. It works as a gateway node. In addition, each node is expected to maintain the information about the neighbor nodes in a data structure viz. the neighbor table N_b, represented by the equation 2.

$$N_b = \{nb_1, nb_2, nb_3, \ldots, nb_c\} \tag{2}$$

Node N_b accommodates details of c number of nodes inclusive of their network attributes. RPL protocol specifications limit the size of the neighbor table to accommodate information about eight neighbors. As per IEEE 802.15.4 standard specification, an asynchronous lossy wireless channel is used for data transmission amongst the networked devices.

2) Communication Model: The wireless network is assumed to be an efficient communication medium for resource-constrained devices with a packet loss probability of 20%. RPL protocol manages optimized DODAG formation and maintenance procedures with its self-organizing and self-healing capabilities. We consider the following specifications. MRHOF (Minimum Rank with Hysteresis Objective Function) as an objective function and RSSI as a routing metric for parent selection. IEEE 802.15.4 standard has an inbuilt cryptography mechanism that provides confidentiality, integrity, and authentication services. Although an ad-hoc network, we assume that nodes, once deployed, do not change their position. In addition, RPL is assumed to be running in a non-storing mode, in which every node has to send a packet to Border Router first, followed by the same being forwarded by the Border router to the actual destination.

3) Attack Model: As discussed in the previous section, the network being in a non-storing mode for communication makes the BR node an intermediate entity for every communication within nodes or outside the network. Except leaf nodes, all other nodes in the network forward packets and implement routing operations. In addition, a DODAG, a tree topology network, is formed by RPL. In this tree topology, an attacker node can influence the value of routing metrics, to make itself a better candidate for being selected as a parent node of any other node in the network. Once the node gains a promising position in the network, the adversary node can drop all the data packets received from the descendant nodes and disrupt the network operation. Moreover, the node makes the attack more effective by actively participating in the DODAG formation and reformation process and by forwarding all control packets. The adversary node tries to replicate this behavior over a large section of the network and upon achieving a position over the sizable active span of the network. The attacker node starts dropping all the data packets directed towards and from the Border router; instead of forwarding them ahead. At the same time; the attacker node forwards all the control packets to participate actively in DODAG formation or reformation. As a result, the attack becomes more effective and gets extended over a larger period of time.

B. The SIEWE Approach for Blackhole Mitigation and Limitations

In general, in order to mitigate the blackhole attack in the watch-dog based approaches, the mechanism used is as follows: [19, 20, 22, 23, 24]:

(a) Check the value of the routing metric in the DIO packets received from each node and fill the neighbor table with relevant entries.
(b) Select a parent node based on the better value of routing metric published.
(c) Monitor the communication behavior of all the neighboring nodes as to whether anyone of those is dropping the packets or not.
(d) Locally conclude that a node is a blackhole node if it drops the inbound packets.
(e) Send all the observations to BR node to verify the blackhole node globally.

The consequences are the increased computational and communication overheads that in turn affect the energy efficiency and network lifetime. An interesting question that crops up is as follows: Is it possible to reduce the number of nodes required to be monitored (step(c)) and number of packets exchanged between nodes and BR node such that the overall overhead is reduced?

In general, an attacker node influences one of the routing metrics in such a way that there is an increased probability of the attacker node becoming the parent of its neighboring nodes. For example, as per the RPL execution mechanism, when a node receives a DIO control packet from x number of neighboring nodes, it chooses the node as a parent that communicates with the highest RSSI value amongst these x nodes. Therefore, an attacker node would always try to manipulate the routing metric in the DIO packet that it communicates (e.g., the RSSI value) to influence the routing decision of the neighboring node and attempt to become its parent. As shown in Figure 6, a malicious behavior of an attacker (in our case node 8) is observed and recorded by its neighboring nodes (4, 5 ,7 ,9 11, 12) only.

Figure 6 shows an active blackhole node in the network and the observer nodes. Only those nodes that are in the communication range of an attacker are capable of observing its malicious behavior; and thereby making their observations sufficient to detect the malicious node.

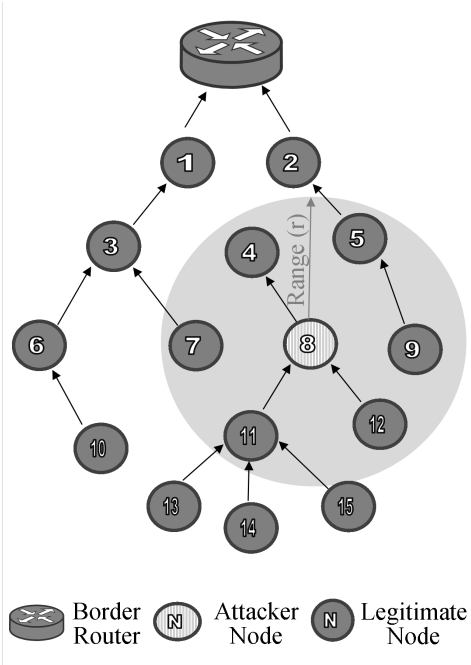


Fig. 6: Range of Node

In order to detect a blackhole attack, it is necessary execute a routine at a node that checks the routing metric (e.g. RSSI value) of the DIO packets received at that node. A node that has communicated with a higher routing metric value (e.g., RSSI value) could be categorized as a suspected node and required to be further watched out before being labeled as a blackhole node. However, a vital question that crop up here is as follows: How to consider a specific value to be higher i.e., how to decide the threshold value of the routing metric in the received DIO packets?

Authors [25] attempt to answer the following - albeit only proposing preliminary ideas.

(a) Check the value of the routing metric in the DIO packets received from each node.
(b) Deduce whether the value is disproportionate to influence the parent node selection and thereby routing.
(c) Prepare a list of suspicious nodes based on the step (b).
(d) Subsequently, monitor the communication of those nodes in the suspect list as to whether anyone of these nodes drops the packets or not.
(e) Locally conclude that a node, out of the suspicious node list, drops the inbound packets is a blackhole node.
(f) Send all the observations to the BR node to verify the blackhole node globally.

To arrive at a threshold value to be used for comparison, the approach in SIEWE uses a simple statistical average value of the RSSI values of all the DIO packets received by a node and maintained in a table. However, as shown in section IV-E2, our analysis shows that using a simple statistical average value of the routing metric, as a filtering criterion (as in SIEWE), may increase the false-positive ratio. In other words, instead of using the average value of a routing metric, some other more suitable statistical functions are required. Therefore, there is a scope for improvement in the strainer module of the SIEWE algorithm.

In this article, we improve the accuracy by applying various filtering criteria and practically comparing the accuracy of our improved approach with the one in SIEWE. In addition, we propose that instead of using a statistically derived threshold value as a criteria for filtering and reducing the list of suspicious nodes, each node should be assigned a trust value and the trust value of a node further be used to label a node as either a malicious node or a benign node. The analysis shows that the approach improves the accuracy of the blackhole detection as compared to the one presented in SIEWE.

C. Embedding Trust Into SIEWE

In Figure 7, we show the T-SIEWE architecture in which the detector and the collector modules from SIEWE are replaced with a trust calculator and a reputation generator, respectively. The new modules along with the improved strainer module are explained further in this section.

1) Strainer: In Figure 6, we show an active Blackhole node in the network and its observer nodes. Only those nodes that are in the communication range of a malicious node are capable of observing the malicious behavior; thereby making their observation sufficient to detect the attacker. Hence, filtering out such nodes is necessary to reduce the energy consumption of the network. The above functionality is implemented by a strainer module during RPL DODAG formation that eventually sets the nodes to run in the promiscuous mode.

To maintain the routing efficiency in 6LoWPAN based devices, every node in the network maintains a neighbor table that keeps relevant details of the neighboring nodes. As per the RPL specifications, a neighbor table accommodates information of eight nodes. Neighbor table stores information that can be used for routing optimization, such as node ID, routing metric value, time of last communication, reachable or not, and path cost of the node. In Table II, we show the neighbor table from the strainer module's perspective that consists of the node's IPv6 address and related RSSI metric value.

As RSSI value directly depends on the distance between nodes, in a dense adhoc deployment, a node may receive DIO packets from different sources with the same RSSI value.

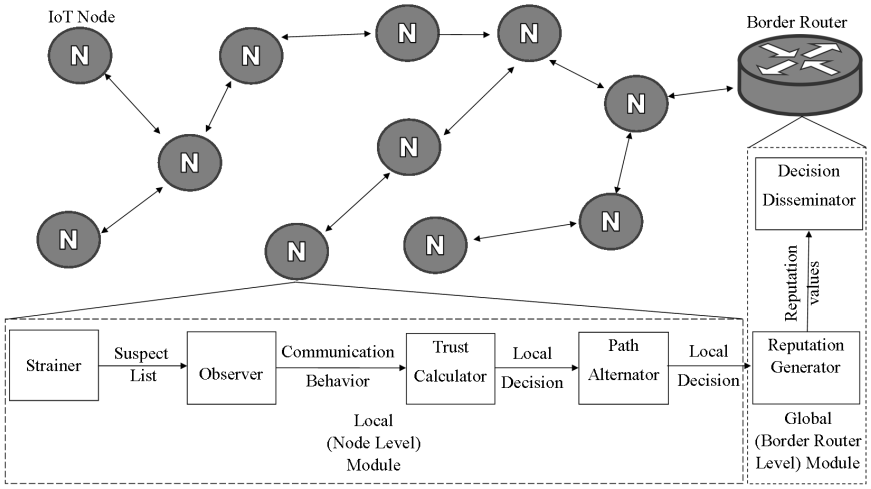


Fig. 7: Architecture of T-SIEWE

TABLE II: Neighbor Table

Node id	A::1	A::2	A::3	A::4	A::5	A::6	A::7	A::8
Metric	-60	-60	-40	-50	- 60	-90	-35	-50

Algorithm 1 Strainer_RSSI

procedure
  $nbr\_table \leftarrow empty$
  $suspect\_table \leftarrow empty$
  $sum \leftarrow 0$
  $count \leftarrow 0$
  $listcount \leftarrow 0$
  $max\_entries \leftarrow 8$
  $\Gamma \leftarrow 0$ ▷ mode value
  $\mu \leftarrow 0$ ▷ mean value
  $\sigma \leftarrow 0$ ▷ standard deviation
  $\delta \leftarrow 0$ ▷ rank threshold
  for each DIO message received **do**
    if $listcount > max\_entries$ **then**
      $nbr\_table \leftarrow ID, RSSI_{id}$
      $listcount = listcount + 1$
    end if
  end for
  if $listcount > max\_entries$ **then**
    $\Gamma = calculate\_mode(table, listcount)$ ▷ equation 3
    $\mu = calculate\_mean(table, listcount)$ ▷ equation 4
    $\sigma = calculate\_dev(table, \mu, listcount)$ ▷ equation 5
    $\delta = calculate\_threshold(\Gamma, \sigma)$ ▷ equation 6 / 7 / 8
    while $nbr\_table \neq empty$ **do**
      if $RSSI_{id} > \delta$ **then** ▷ equation 9
        $suspect\_table \leftarrow id$
      end if
    end while
  end if
end procedure

Hence, from the structure shown in Table II, we can easily calculate the mode value which occurs most often in the list as:

$$\Gamma = mode(rssi) \tag{3}$$

In case of multiple mode values, the largest value will be selected as Γ. For calculating the rank threshold value, the standard deviation for the RSSI value from the neighbor table is required. The RSSI value and the rank threshold can be defined based on the values of Γ, μ, and σ.

1) The mean value of routing metrics is calculated as follows.

$$\mu = \frac{\sum_{i=1}^{n} nb_i.rssi}{n} \tag{4}$$

2) The standard deviation σ of the list is calculated as follows:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \mid nb_i.rssi - \mu \mid^2}{n}} \tag{5}$$

3) The rank threshold is calculated as follows:

$$\delta = \Gamma + \sigma \tag{6}$$

The value of δ defines the range which consist of the value Γ which occurs frequently in the list and the expected value, and σ is a standard deviation for the RSSI value. For our analysis, apart from equation 6, we verify the accuracy of strainer's algorithm, with two other versions of δ as follows:

$$\delta_1 = \Gamma + (1.5 \times \sigma) \tag{7}$$

$$\delta_2 = \Gamma + (2 \times \sigma) \tag{8}$$

We calculate the rank threshold from the equations 6, 7 and 8. In addition, we decide whether a given node N_i is

suspicious or not using the following equation:

$$isSuspected(N_i) = \begin{cases} 1, & \text{if } N_{irssi} > \delta \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

From equation 9, we say that if the routing metric value of a particular node is greater than the rank threshold (δ), then the node is considered to be a suspicious node. The pseudocode for the strainer's function for identifying a suspicious node is shown in Algorithm 1. After the table gets populated with data of eight neighboring nodes, the Strainer algorithm calculates Γ, μ, and σ values and from these values, the rank threshold (δ) is derived. Nodes with a routing metric value exceeding that rank threshold will be added to the suspicious nodes' list for further verification. In Table III, we show an instance of the suspicious nodes' list. We implement and evaluate the strainer function for three different values of δ and verify the efficiency. In Section IV-E, we discuss the implementation of the strainer algorithm and the results.

TABLE III: Suspicious Nodes' List

Node ID	Packet-Received	Packet-Sent
A::5	8	6
A::6	8	0

For tableII, the calculated value of $\Gamma = -60, \mu = -53$, $\sigma = -16$ and $\delta = \Gamma + \sigma = -76$. Therefore, a node with ID A::6 is added to suspicious nodes' list as its RSSI value is higher than the δ.

2) Observer: As shown in Figure 6, after the strainer algorithm gets executed, each node in the range of an attacker node has an entry of an attacker node in its the suspicious nodes' list. Once a node has at least a single entry in the suspicious nodes' list, the observer/trust calculator module enables the promiscuous mode for the node. Due to RPL's tree-like topology structure, in which nodes are organized into a parent-child relationship, every node initially forwards packets to its parent node only. Hence, the observer nodes can be classified as children nodes, and observations from non-children nodes are used to decide the routing behavior. In contrast, the observations from the children nodes are used to find the packet forwarding behavior.

- **Routing Behavior**: Non-children nodes observe the routing behavior by overhearing the network traffic. Nodes check details regarding the sender and destination fields from packets, and if any of the two fields contain the ID of the suspicious node, it then sets packet-sent and packet-received values in the suspicious nodes' list. Procedure at line 1 in Algorithm 2 shows the observation mechanism for the routing behavior.
- **Packet Forwarding Behavior**: Children nodes observe the packet forwarding nature of the suspicious node as they directly forward the packets to them. A child node first forwards the packet to the suspicious node and increases the packet received value in the suspicious nodes' list, and then waits for a predefined time. If the parent node forwards the same packet, it increases the packet sent value in the suspicious nodes' list. Procedure at line 13 in Algorithm 2 shows the observation mechanism for the packet forwarding behavior.

Algorithm 2 Observer and Trust Calculator

```
1:  procedure OBSERVER(Non Child)
2:      for each communication received do
3:          if sender or receiver is in suspectlist then
4:              if packet is from node then
5:                  suspectlist.Outcount_id ++
6:              else
7:                  suspectlist.Incount_id ++
8:              end if
9:          end if
10:     end for
11: end procedure
12:
13: procedure OBSERVER(Child Node)
14:     for each Packet Sent do
15:         Incount_id ++
16:         Wait()
17:         if Overheard packet = Packet sent then
18:             Outcount_id ++
19:         end if
20:     end for
21: end procedure
22:
23: procedure TRUST CALCULATOR
24:     while True do
25:         for each entry in suspect list do
26:             tv ← Packet-sent/Packet-received
27:             if tv ≤ ρ then
28:                 if Parent is suspected then
29:                     Invoke local repair
30:                 end if
31:                 SendToBR(TrustPacket)
32:             end if
33:         end for
34:         Wait()
35:     end while
36: end procedure
```

As shown in table III, the observer module records the communication behavior of a suspicious node in the list which is then used by the trust calculator module for further verification.

3) Trust Calculator: The trust calculator module periodically checks values in the suspicious nodes' list and calculates routing and packet forwarding trust values for the suspicious node, as shown in equations 10 and 11. One of the column in the suspicious nodes' list stores trust values, as shown in Table IV. The module invokes RPL local repair mechanism by calling the path alternator module if the calculated trust value exceeds predefined threshold value (ρ).

- **Routing Trust**:

$$RB = \frac{DPS}{DPR} \quad (10)$$

Here, DPS stands for Packets-Sent and DPR stand for Packet-Received as shown in Table IV.

- **Packet Forwarding Trust**

$$FB = \frac{FP}{SP} \quad (11)$$

Here, FP stands for packet-sent and SP stands packet-received values as shown in Table IV.

In the proposed approach, we consider 20% packet loss due to the lossy nature of the communication channel. Therefore, we consider $\rho = 0.2$. The procedure for trust calculation starts at line 23 in Algorithm 2.

TABLE IV: Suspicious Nodes' List With Trust Values

Node ID	Packet-Received	Packet-Sent	Trust value
A::5	8	6	0.75
A::6	8	0	0

4) Path Alternator: Trust calculator locally verifies the malicious behavior of a suspicious node during the network operation. Once the suspicious node is identified, the path alternator module takes the following steps:

1) If the parent node is found to be suspicious and locally detected as malicious, then the path alternator initiates the RPL local repair mechanism to change the network path.
2) The path alternator creates trust dissemination packets and sends them to the Border router for the global verification process.

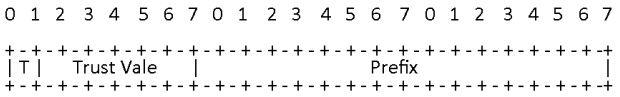


Fig. 8: Trust Dissemination Packet

In Figure 8, we show the packet format used for the communication of trust values to the BR. The first bit of the packet represents its type; value T=0 represents the Forwarding Behavior (FB) while T=1 represents the Routing Behaviour (RB). Instead of sending the real numbers, we use 0 to 9 numbers to represent trust values. The values are again represented as a real number by the BR node e.g., the value represented by 2 is changed to 0.2, and 0.9 is changed to 1. The prefix value here represents prefix of the IPv6 address of the suspicious node.

Strainer, Observer, Trust calculator and Path alternator are local modules or node level modules. Each node except the BR is pre-installed with local modules for filtering, observation, and attack detection at the local level. The global modules installed at the BR support the comprehensive verification process of the blackhole node. The trust calculator has its counterpart in the BR as a reputation generator. At the same time, the global module decision disseminator is responsible for sending a final decision about the blackhole node to all nodes in the network. In the remaining section, we describe the global modules.

5) Reputation Generator: Upon receiving trust dissemination packets from networked nodes, the reputation generator module calculates the reputation value for the attacker node represented by the prefix field. As the network runs in non-storing mode, the BR node holds a list of all nodes and their child nodes. Therefore, the reputation generator waits for readings from the child nodes, collects data from other observer nodes, and generates the cumulative routing behavior. Once the forwarding behavior from different child nodes are available, the overall reputation of a node is calculated.

TABLE V: Reputation Table

Node ID	CRB	CFB	Number of Child Nodes	Child Packets Received
A::5	0.6	0.2	2	1
A::6	0.3	0.5	3	1

In Table V, we show reputation values stored in the BR node. The BR node stores cumulative values upon receiving the trust dissemination packet from nodes. Upon receiving the packet reputation values, the generator first fetches the type bits and based on the type of value received, calculates the cumulative reputation values as follow.

$$CRB_{id} = \frac{\sum_{i=1}^{n_o} RB_i}{n_o} \tag{12}$$

CBR shows cumulative routing behavior that is calculated by computing the average of the received routing behavior collected by observer nodes.

$$CFB_{id} = \frac{\sum_{i=1}^{n_c} FB_i}{n_c} \tag{13}$$

CFB shows cumulative forwarding behavior that is calculated by computing the average of the trust values sent by child nodes.

Once trust values from all child nodes are received, the final reputation value for the node is calculated as follows:

$$Rep_{id} = \alpha \times CFB_{id} + (1 - \alpha) \times CRB_{id} \tag{14}$$

The α value here represents a weight within range of 0 to 1. For our approach, we consider α as 0.6, as we rely more on packet forwarding nature observed by the child nodes. Additionally, if the final reputation value calculated by the equation is less than or equal to 0.2, then the node is considered as a blackhole. The information is conveyed to all the nodes in the network by the disseminator module.

6) Decision Disseminator: If the reputation value for any particular node becomes less than or equal to the threshold value, i.e., 0.2, it can be confirmed that a blackhole exists within the network. In the proposed approach, a blacklist is maintained at every node to separate out such intruder nodes from the network. The decision disseminator module propagates node ID of the malicious node to the remaining nodes by embedding the ID within a DIO packet.

The DIO packet, as per the specifications defined in RFC6550, is shown in Figure 4. Two 8-bit fields in DIO packet viz. flags and reserved, are unused and ignored by the receiver [29]. Decision disseminator module uses the unused bits and sets them with the prefix ID of the malicious node and initiates the global repair procedure. After receiving the DIO packet, each node in the network fetches the ID of the malicious node and adds to its own local blacklist. In the proposed approach, we consider a blacklist to be persistent, and the suspect list is recreated for every instance of the DODAG.

D. Implementation and Analysis

We implement T-SIEWE in the Contiki-3.1 OS that is an efficient and very well known OS for IoT [27] devices. The OS also provides fully implemented RPL protocol as per 6LoWPAN specification. In Contiki OS, μIP, and IP stack modules provide IP communication, while the SICSLoWPAN module provides header compression. To implement the attacker model, we modify μIP and IP stack modules to drop data packets. In addition, we change the implementation of objective function to accommodate RSSI as a routing metric. We use the "Unit Disk Graph Medium (UDGM) with Constant Loss" model in the Cooja simulator as in the UDGM considers transmission range as a circular disk; all nodes within the range can receive all packets [28]. In addition, the UDGM allows users to set the Tx-Rx success ratio manually. For the proposed approach, we set the Tx-Rx success ratio to 80%. To simulate the T-SIEWE, we use the emulated Tmote Sky motes. Table VI shows the basic environment setup of the proposed simulation.

TABLE VI: Simulation Environment

Parameter	Value
Simulator	Cooja
Radio Model	UDGM
Node Radio Range	Rx and Tx 50m
Mote Type	sky
TX/RX Success Rate	80%
Size of deployment Region	$100 * 100$ m
Number of nodes	8 to 64
Type of nodes	3
Physical layer	IEEE 802.15.4
Routing protocol	RPL
Objective function	RSSI based
Additional Tools used	Collect View

Figure 9 A and B depict the basic topology that we used to verify the behavior of T-SIEWE. The topology has three types of nodes, a BR (Green), a blackhole node (Red), and regular nodes (Yellow). We consider four different network configurations consisting of 8. 16, 32 and nodes with 1,2,4 and 8 attacker nodes respectively. To normalize the observations, we perform each experiment 10 times and computed the average.

E. Evaluation

In this section, we discuss the empirical analysis of T-SIEWE. As shown in Figure 9a and Figure 9b, we set two different scenarios for the experiments; one with 16 nodes with two attackers and another with 32 nodes and four attackers. After performing simulation for the mentioned situations, we evaluate the true-positive rate and Packet Delivery Ratio (PDR), for different situations. In addition, we measure the energy overhead and memory requirements. We compare our results with the SVELTE [16] and the base RPL protocol in the lossy network. We used a collect view tool to monitor network topology during the simulation.

1) Effect of Blackhole attack on PDR: To implement the attack model, we change the MRHOF objective function in RPL to incorporate RSSI value as a routing metric. The parent selection mechanism of RPL is also modified to choose the parent based on the highest RSSI value. We increase the transmit signal power of the attacker node; hence all the nodes in the range of attacker node receive the packets with the highest RSSI value. The results suggest that by manipulating the RSSI routing metric value, the malicious node increases its probability of being selected as a parent node by 30% as compared to the regular (benign) nodes.

Collect view periodically collects the data from all the nodes in the network and plots the network graph from collected data. In Figure 9.C, we show a network graph formed by collect view tool 9 that has 32 nodes and 4 malicious nodes for 1:30 hours. Under the attacking scenario, the collect view application is not able to collect data from the whole network, and because of that generated network graph, doesn't have all the nodes. Figure 10 shows a comparison of PDR, between normal RPL protocol execution and the above defined scenarios. We have considered a lossy network, and, therefore, the packet loss is observed during the normal RPL; it is observed from Figure 10 that during the blackhole attack, PDR decreases drastically.

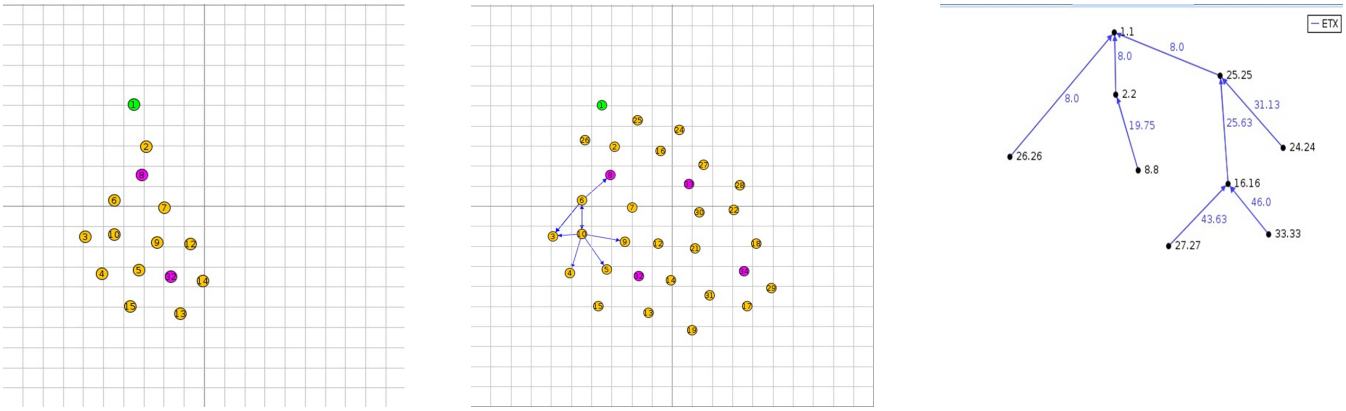
2) True Positive Rate For Strainer: We calculate the true-positive rate of T-SIEWE. The number of total successful alerts divided by the total number of alerts generated. In SVELTE, the root node waits for all [16] nodes' mapping request packets, and, then starts the detection process. In contrast, in the proposed approach, local detection modules start the detection process in the early phase of the route formation. We compare the Stainer module's true-positive rate with the one proposed in SVELETE [16] after 30 minutes of execution.

The results show that the proposed thresholds $\delta, \delta_1, \delta_2$ perform better as compared to the threshold considered in SIEWE but less than that of SVELTE [16] except for δ_2. The accuracy depends on the global verification phase that is performed at the BR. For the same, we consider the percentage of devices selected for running into promiscuous mode. In Figure 12, we show the percentage of devices selected to set into the promiscuous mode for monitoring of suspicious node's behavior.

Though δ_2 shows the promising result, as shown in Figure 12, it sets the least numbers of devices running into a promiscuous mode that eventually affects the overall efficiency of the global verification process. In addition, during the evaluation, we found that with trust-based global verification, it shows $60-80\%$ of True positive rate with δ_1 threshold. For further verification, we consider the proposed system with δ and δ_1 threshold.

3) Energy Overhead: To calculate the energy consumption of the network, we use Contiki power trace total time spent on different components during the execution. We use them with Tmote Sky's operation condition data sheet [16] to calculate the network's power consumption.

We consider the normal RPL protocol execution as a benchmark scenario to trace power consumption. We deploy all the nodes in the network by default as in promiscuous mode to monitor each other's activity and then compare the power consumption with T-SIEWE with δ and δ_1 as a



(a) 16 Nodes Simulation Setup With 2 Attackers (8 and 32)

(b) 32 Nodes Simulation Setup With 2 Attackers (8, 32, 33, 34)

(c) Network Graph Generated by Collectview For Setup Shown in Figure 9b

Fig. 9: Node Placements Within the Simulation Environment

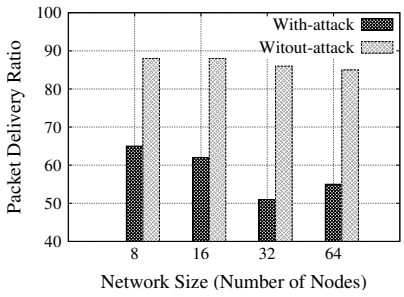


Fig. 10: PDR For Different Scenarios

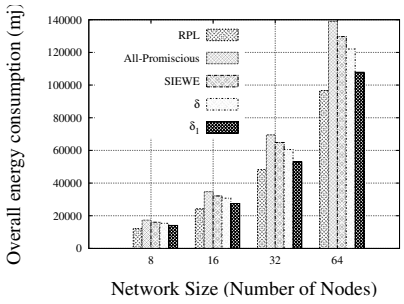


Fig. 13: Energy Usage in the Network by Nodes

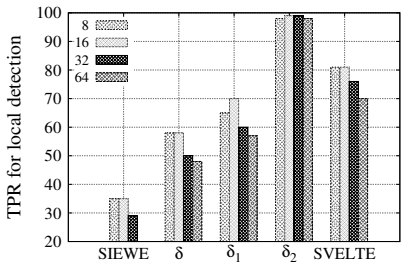


Fig. 11: True Positive Rate for Stainer in Local Detection Phase

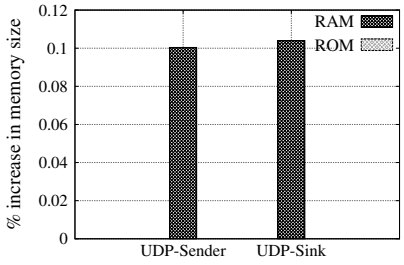


Fig. 14: Percentage Increase in Memory by the Proposed Approach

threshold. The figure shows that T-SIEWE consumes less energy as compared to running all nodes in the promiscuous mode.

4) Memory Overhead: As shown in Figure 14, the proposed approach requires minimum percentage of increment in size of RAM while size of ROM remains same.

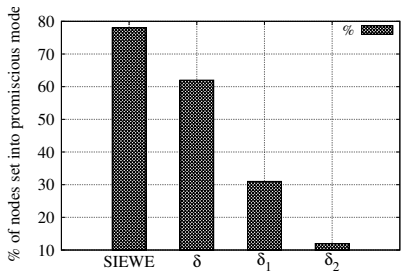


Fig. 12: Percentage of Nodes Set Into Promiscuous Mode

5) Packet Delivery Ratio: As shown in Figures 9.A and B, we run our simulation with T-SIEVE, verify PDR, and compare its PDR with our Benchmark RPL protocol.

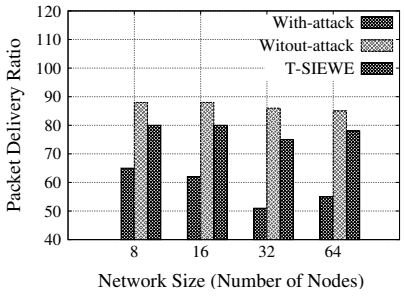


Fig. 15: PDR After Mitigation

After mitigation, PDR of system increases as the attacker is isolated from the network with the use of the blacklist.

We observe some differences between PDR after mitigation and benchmark RPL's scenario because during detention and mitigation, a blackhole node have dropped some packets.

V. Conclusion

We proposed a T-SIEWE, a trust and strainer based approach to detect the blackhole nodes in a network. The proposed approach is motivated by a novel idea that in order to improve the overall efficiency, the number of nodes that are to be watched for any anomalous communication patterns is to be reduced. The proposed approach T-SIEWE applies filtering criteria based on the statistical criteria and as our analysis and evaluation shows the improved efficiency by limiting the numbers of nodes set to run into the promiscuous mode. The experiments lead to reduce the count of these nodes by upto 50%. As compared to the state-of-the-art articles, in the proposed approach, the true positive rate for blackhole nodes detection is reduced. The trust-based global verification mechanism detects an attacker node with nearly 80% accuracy. T-SIEWE improves the network's PDR in the presence of a blackhole node with less energy consumption and negligible memory overhead.

References

[1] M. Tagashira and T. Nakagawa, "Biometric Authentication Based on Auscultated Heart Sounds in Healthcare." *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 343–349, 2020.

[2] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical Systems: The Next Computing Revolution," in *Design Automation Conference*, Anaheim, CA, USA, June 2010, pp. 731–736.

[3] Y. Cheng, P. Chao, H. Liang, and C. Kuo, "Smart Home Environment Management Using Programmable Logic Controller," *Engineering Letters*, vol. 28, no. 4, pp. 1174–1181, 2020.

[4] K. Lingjie, "Application of ZigBee-WSN Technology for Indoor Environmental Parameter Monitoring System," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 725–733, 2019.

[5] C. Greer, M. Burns, D. Wollman, and E. Griffor, *Cyber-physical systems and internet of things*. National Institute of Standards and Technology (NIST) Special Publication 1900-202, 2019.

[6] J. Hui, P. Thubert *et al.*, "Compression format for IPv6 datagrams over IEEE 802.15. 4-based networks," RFC, September 2011. [Online]. Available: https://www.hjp.at/doc/rfc/rfc6282.html

[7] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, and I. Godor, "Ericsson mobility report," Ericsson: Stockholm, Sweden, Tech. Rep., Feb 2016. [Online]. Available: https://www.ericsson.com/49db9f/assets/local/mobility-report/documents/2016/ericsson-mobility-report-feb-2016-interim.pdf

[8] E. Ancillotti, R. Bruno, and M. Conti, "The Role of The RPL Routing Protocol for Smart Grid Communications," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 75–83, 2013.

[9] O. Iova, F. Theoleyre, and T. Noel, "Stability and Efficiency of RPL Under Realistic Conditions in Wireless Sensor Networks," in *24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. London, UK: IEEE, 2013, pp. 2098–2102.

[10] W. Yin, B. Guo, H. Hu, Y. Hu, and Q. Li, "The Research on WSNs Scale-free Topology for Prolonging Network Lifetime," *Engineering Letters*, vol. 29, no. 1, pp. 238–243, 2021.

[11] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–6.

[12] K. Weekly and K. Pister, "Evaluating Sinkhole Defense Techniques in RPL Networks," in *20th IEEE International Conference on Network Protocols (ICNP)*. Austin, TX, USA: IEEE, 2012, pp. 1–6.

[13] G. Glissa, A. Rachedi, and A. Meddeb, "A Secure Routing Protocol Based on RPL for Internet of Things," in *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1–7.

[14] S. Luangoudom, D. Tran, T. Nguyen, H. A. Tran, G. Nguyen, and Q. T. Ha, "svBLOCK: Mitigating Black Hole Attack in Low-power and Lossy Networks," *International Journal of Sensor Networks*, vol. 32, no. 2, pp. 77–86, 2020.

[15] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the rpl-based internet of things," *International Journal of Distributed Sensor Networks*, vol. 9, no. 8, pp. 1–11, 2013.

[16] 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.

[17] H. Sedjelmaci, S. M. Senouci, and T. Taleb, "An Accurate Security Game for Low-Resource IoT Devices," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9381–9393, 2017.

[18] R. Sahay, G. Geethakumari, B. Mitra, and N. Goyal, "Investigating Packet Dropping Attacks in RPL-DODAG in IoT," in *IEEE 5th International Conference for Convergence in Technology (I2CT)*. Bombay, India: IEEE, 2019, pp. 1–5.

[19] N. Djedjig, D. Tandjaoui, and F. Medjek, "Trust-based RPL for the Internet of Things," in *IEEE Symposium on Computers and Communication (ISCC)*. Larnaca, Cyprus: IEEE, 2015, pp. 962–967.

[20] D. Airehrour, J. Gutierrez, and S. K. Ray, "Securing RPL Routing Protocol from Blackhole Attacks Using a Trust-based Mchanism," in *26th International Telecommunication Networks and Applications Conference (ITNAC)*. Dunedin, New Zealand: IEEE, 2016, pp. 115–120.

[21] M. Surendar and A. Umamakeswari, "InDReS: An Intrusion Detection and response system for Internet of Things with 6LoWPAN," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. Chennai, India: IEEE, Mar 2016, pp. 1903–1908.

[22] Z. Khan and P. Herrmann, "A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things," in *IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. Taipei, Taiwan: IEEE, 2017, pp. 1169–1176.

[23] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani,

"New Trust Metric for The RPL Routing Protocol," in *8th International Conference on Information and Communication Systems (ICICS)*. Irbid, Jordan: IEEE, 2017, pp. 328–335.

[24] V. Neerugatti and A. R. M. Reddy, "Artificial Intelligence-Based Technique for Detection of Selective Forwarding Attack in RPL-Based Internet of Things Networks," in *Emerging Research in Data Engineering Systems and Computer Communications*. Singapore: Springer Singapore, 2020, pp. 67–77.

[25] H. B. Patel and D. C. Jinwala, "Blackhole Detection in 6LoWPAN Based Internet of Things: An Anomaly Based Approach," in *TENCON 2019 - IEEE Region 10 Conference (TENCON)*. Kochi, India: IEEE, 2019, pp. 947–954.

[26] K. Gibert, M. Sànchez-Marrè, and J. Izquierdo, "A survey on pre-processing techniques: Relevant issues in the context of environmental data mining," *AI Communications*, vol. 29, no. 6, pp. 627–663, 2016.

[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*. Tampa, FL, USA: IEEE, Nov 2004, pp. 455–462.

[28] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level Sensor Network Simulation With Cooja," in *First IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2006)*. Tampa, FL, USA: IEEE, 2006.

[29] T. Winter, P. Thubert, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: Ipv6 routing protocol for low power and lossy networks, RFC6550," RFC, 2012. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6550