

Secure IoT Routing: Selective Forwarding Attacks and Trust-based Defenses in RPL Network

Jun Jiang, Yuhong Liu[†]

Department of Computer Science and Engineering, Santa Clara University, Santa Clara, CA, USA

{jjiang2, yhliu}@scu.edu

Abstract—IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) is an essential routing protocol to enable communications for IoT networks with low power devices. RPL uses an objective function and routing constraints to find an optimized routing path for each node in the network. However, recent research has shown that topological attacks, such as selective forwarding attacks, pose great challenges to the secure routing of IoT networks. Many conventional secure routing solutions, on the other hand, are computationally heavy to be directly applied in resource-constrained IoT networks. There is an urgent need to develop lightweight secure routing solutions for IoT networks. In this paper, we first design and implement a series of advanced selective forwarding attacks from the attack perspective, which can flexibly select the type and percentage of forwarding packets in an energy efficient way, and even bad-mouth other innocent nodes in the network. Experiment results show that the proposed attacks can maximize the attack consequences (i.e. number of dropped packets) while maintaining undetected. Moreover, we propose a lightweight trust-based defense solution to detect and eliminate malicious selective forwarding nodes from the network. The results show that the proposed defense solution can achieve high detection accuracy with very limited extra energy usage (i.e. 3.4%).

Index Terms—Selective forwarding attack, RPL, Routing security, IoT, Trust

I. INTRODUCTION

With the rapid adoption of Internet of Things (IoT) devices around the world [1], many of these devices are resource-constrained [2]. Since the existing Internet Protocols (IP) are too complex to be directly implemented on resource-constrained IoT devices [3], the Internet Engineering Task Force (IETF) designs a lightweight IPv6 protocol with a series of core protocols to ensure efficient and secure communications, such as IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) [4] and Routing Protocol for Low Power and Lossy Networks (RPL) [5]. In particular, RPL, as the core routing protocol for resource constrained IoT networks, has been adopted by a variety of applications, such as healthcare [6], smart grid [7], and smart city [8], etc.

Due to its popularity, RPL becomes an attractive attack target [9]. One attack that can cause massive damage to the RPL network is selective forwarding attack [10], where attackers interrupt network data flows by selectively dropping network packets. Compared to blackhole attacks which simply drop all packets, selective forwarding attacks are more deceptive

and can remain undetected for a longer time, causing long-term damage to the network. Despite their impact, existing selective forwarding attacks are still *lack of flexibility* in terms of dynamically identifying victim nodes and adjusting packet forward rates according to the state of the network.

On the other hand, there are three major types of defense mechanisms against selective forwarding attacks on RPL. The first type is to build a multi-path routing network to ensure the integrity of information transmission [11]. These mechanisms often *require excessive resources* to maintain the backup paths for nodes. The second type is distributed defense mechanisms, which deploy a monitor module on individual nodes [12]. However, these defense mechanisms often lead to *significant extra energy consumption* at the monitoring nodes. In addition, it is challenging to ensure that all the monitoring nodes are *long-term reliable and honest* in reporting their neighbors' behaviors. The third type is the centralized defense mechanisms [13], where a central node is employed to monitor and analyze malicious behaviors in the network. Nevertheless, the central node has to be deployed at a *core location in the network* to ensure coverage across the entire network, and may be easily *misled by complex bad-mouthing attacks*.

In this work, we aim to advance current studies from both the attack and defense perspectives. Specifically, from the attack aspect, we propose an advanced selective forwarding attack model, which can dynamically launch three types of malicious behaviors: (1) flexibly dropping packets from selected types of protocols, (2) adjusting the packet forward rate based on the average network packet forward rate to stay stealthy, and (3) dynamically selecting specific children nodes for bad-mouthing attacks. Furthermore, these attack behaviors can be combined to significantly increase the damages to the network and reduce the risk of being detected by state-of-the-art defense mechanisms.

Furthermore, from the defense aspect, we propose a novel centralized trust-based defense mechanism to combat selective forwarding attacks in RPL networks. Compared to distributed defense mechanisms, the proposed scheme can significantly save the energy consumption for anomaly detection by deploying the defense computation only at the root node. Unlike other centralized defense schemes, the proposed scheme takes advantage of RPL's tree-based network topology to evaluate the trustworthiness of nodes and avoids the introduction of a trusted third-party node. Furthermore, by carefully designing the detection/notification module, the proposed scheme can

[†]: Corresponding author

effectively balance the trade-off between detection delay and energy consumption.

The main contributions of this paper are as follows.

- This work proposes an advanced selective forwarding attack with three types of behaviors against RPL network. Malicious nodes in the proposed attack model cannot only flexibly choose the type of packets to drop, but also control the packet forward rates dynamically. As a result, these attackers are able to launch more stealthy attacks to avoid being detected. Furthermore, malicious nodes can also bad-mouth other normal nodes to cause false alarms in the system. The experiment results show that it can effectively evade RPL self-defense mechanism and state-of-the-art defense mechanisms.
- This work proposes a novel centralized trust-based defense mechanism. In particular, the proposed trust model integrates a self-trust value, which reflects a node's trustworthiness in its packet forwarding behavior, and a tree-based descendant trust value, which takes advantage of the RPL network topology to prevent bad-mouthing attacks. Furthermore, the proposed defense mechanism is deployed on the root node, which can effectively reduce the total energy consumption caused by distributed anomaly monitoring scheme, and eliminate the security risks caused by the introduction of third-party devices. Experiment results show that the proposed scheme achieves high detection accuracy and low energy consumption.
- This work proposes a novel anomaly report mechanism. Instead of using UDP packets, ICMPv6 control message is chosen to send information about malicious nodes. The reporting mechanism only starts when root node detects the malicious behaviors in the network. This reporting mechanism cannot only ensure that nodes in the network can be notified promptly, but also avoid causing information broadcast storms.

The rest of this paper is organized as follows. Section II discusses existing selective forwarding attacks and defense mechanisms in RPL networks. Section III introduces preliminaries of the RPL protocol. Section IV and V discuss the proposed selective forwarding attack and the defense mechanism in details. The results of the experiment are given in Section VI, followed by a conclusion in Section VII.

II. RELATED WORK

A. Selective Forwarding Attacks in RPL Networks

RPL network faces a variety of security threats, which are mainly divided into three categories [14], [15]. The first type of attack is resource attack, such as flooding attacks [16] and increased rank attacks [17]. In these attacks, the attacker aims to exhaust the victim node's energy and reduce its lifetime by misleading it to execute a large number of unnecessary instructions. The second type of attack is traffic attack, such as sniffing attacks [14] and identity attacks [10], where the attacker's main goal is to eavesdrop or manipulate the network's traffic. The third type of attack is network topology attack where the attacker undermines the security

and stability of the network by changing the topology of the network, such as sinkhole attacks [10], blackhole attacks [13] and selective forwarding attack [18].

Selective forwarding attack is one of most devastating type of attacks [19]–[21], which can cause severe damage to the network. However, in many of these attacks, the attackers set a fixed number of dropped packets or blindly attack all data packets, which increases the risk of being detected by defense mechanisms. In some advanced attacks [10], [12], the proposed attacks interrupt the victim node's communications by only forwarding RPL control messages while dropping all [10] or partial data packets [12]. Different from these existing attacks, in this study, the proposed selective forwarding attacks can perform more flexible malicious behaviors and even bad-mouth other innocent nodes to mislead state-of-the-art defense schemes.

B. RPL Network Defenses

The original design of RPL protocol has some basic security schemes, such as the local and global repair mechanisms [5], which can be triggered by changes of network topology, e.g. a link failure. However, these basic security repair mechanisms are far from adequate to resist the rapid evolving security attacks [22].

Beyond the basic repair mechanisms, there are mainly three categories of defenses. The first type is to establish multiple routing paths for each node to avoid selective forwarding attacks. In [11], the authors propose a secure routing protocol M-RPL, which establishes a hierarchical cluster network and backup paths for different clusters in the route discovery phase. In [23], the authors establish a temporary backup path for a node based on its packet delivery ratio. In [24], the authors propose to use the principle of Packet Replication and Elimination (PRE), through IEEE 802.15.4 Time-Slotted Channel Hopping (TSCH) as media access, to create parallel paths from nodes to the root node. Although the multi-path mechanism can effectively resist attacks, due to the introduction of redundant network routes and extra resources to maintain the backup paths, these defense schemes often cause significant increases in nodes' energy consumption.

The second type of defenses is distributed defense mechanisms that build monitor module on each network node. Due to its easy implementation, this type of defenses is the most popular one. In [12], [25], [26], the authors propose various distributed trust-based mechanisms, where each individual node monitors its neighbors' incoming and forwarding traffic and calculates their trust values. In [27], [28], a distributed, collaborative and layered trust-based IDS (T-IDS) and a Metric-based RPL Trustworthiness Scheme (MRTS) are proposed respectively, where each node monitors and cooperates with its neighbors to detect and report intrusions. These distributed mechanisms, however, have to be deployed on each IoT nodes, leading to significant extra energy expenditure. On the other hand, it also presents a new challenge, as how resource-constrained nodes can provide long-term reliable and honest reports on their neighbors' behaviors.

The third type is centralized defense mechanisms. In [13], the authors use a SVELTE intrusion detection system (IDS),

where a 6LoWPAN Mapper is placed on the IPv6 Border Router (6BR) to monitor and analyze the malicious behaviors in the network. In [10], the authors propose a lightweight heartbeat protocol, in which the root node detects the malicious node by periodically exchanging an echo signal with all its children nodes. In [29], authors introduce a control layer to achieve the hierarchical trust-based mechanism “CTrust-RPL” to monitor the nodes’ behaviors. These centralized detection mechanisms are often energy-efficient since only a limited number of nodes are involved for behavior monitoring. However, since the root node cannot directly monitor each node in the network, it can be easily misled by some malicious attacks, such as bad-mouthing attacks [30], where malicious nodes blame their parent/children nodes for packet dropping. As a result, it is challenging for such defense schemes to be robust against complex bad-mouthing strategies.

In our prior conference paper [31], we have studied black-hole attacks and proposed a centralized defense mechanism at the root node. On this basis, we further study the selective forwarding attacks and defense scheme in this work. Unlike the blackhole attack model, which blindly drops all packets, the attack model in this study launches attacks with three malicious behaviors, (1) flexibly select packet protocol, (2) dynamically adjust packet forward rate, (3) bad-mouth the victim children nodes. As a result, it can effectively hide the attack behaviors and cause long-term network damage. In addition, we propose a new lightweight centralized trust-based defense scheme to defend against selective forwarding attacks. Compared to our prior work, where root node simply uses the average packet forward rate as the trust value for each node, the defense scheme proposed in this work is more comprehensive by (1) integrating self-trust value and tree-based descendant trust value, (2) introducing a beta-based trust framework with a discounting factor to gradually reduce the impact of previous behaviors, and (3) assigning asymmetric discounting factors for good/bad behaviors, so that a node’s bad behavior is remembered for a longer time. Furthermore, more comprehensive experiments are performed to evaluate the performance of the proposed attacks and defenses from different aspects.

III. PRELIMINARIES: RPL PROTOCOL

As the attacks and defenses proposed in this work are based on RPL protocols, in this section, we briefly introduce some basis of RPL protocol. The Routing Protocol for LLN (RPL) is defined by the IETF’s Routing Over Low power and Lossy Networks (ROLL) Working Group. In particular, each RPL network may contain multiple RPL instances. Each RPL instance may contain multiple Destination Oriented Directed Acyclic Graph (DODAG). In a DODAG, the root node, which is usually the most powerful node, is responsible for storing and managing the routing paths. Non-root nodes can be added to one or more DODAGs.

RPL is a hierarchical-based routing protocol that relies on a DAG structure to exchange data among network nodes. Consequently, the parent and children relationship is essential for routing in a RPL network. Each DODAG has a specific objective function, which defines how each node selects its

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RPL InstanceID								Version Number								Rank															
G	O	MOP				Prf				DTSN								Flags				Reserved									
DODAG ID																															
Options(s)...																															

Fig. 1: Format of DIO message

parent node. Based on the objective function, an optimal path from any leaf node to the root node can be constructed.

A. Control Messages in RPL Network

There are four main types of control messages [5] to establish the DODAG, which are DODAG Information Object (DIO), DODAG Information Solicitation (DIS), Destination Advertisement Object (DAO) and Destination Advertisement Object Acknowledgement (DAO-ACK). Particularly, DIO is the most frequently used message, sent by the root node to all other nodes in the network, to advertise network structures for DODAG discovery, assembly and maintenance. Therefore, to rapidly report anomaly detection results while avoiding extra overhead introduced by anomaly reporting messages, we propose to insert detection results into the DIO message to distribute to all nodes in the network. The format of a DIO message is shown in Figure 1, which includes RPL InstanceID, Rank value, DODAG ID, Destination Advertisement Trigger Sequence Number (DTSN), etc. Non-root nodes must advertise and remain the values in DIO message, except for the update of the fields Rank and DTSN.

B. Default Security Mechanisms in RPL Networks

RPL networks can adopt standard mechanisms to ensure message integrity and confidentiality at different layers of the protocol stack. For example, the standardized IEEE 802.15.4 security, lightweight 6LoWPAN compression IPsec [32], and Datagram TLS (DTLS) are adopted to ensure security at the data link layer, IP layer and transport layer, respectively.

In addition, RPL adopts some simple repair mechanisms to recover from three network failures, including routing topology failure, link failure and node failure. When a small number of failures happen, the local repair mechanism starts. The local repair mechanism allows the nodes, which are impacted by failures, to detach from the original DODAG and change their rank values to infinite, then re-join the DODAG again. After multiple local repair mechanisms are initiated, the RPL protocol performs a global repair to rebuild the entire DODAG network by increasing the DODAG version number. Please note that the RPL uses a trickle timer to handle inconsistencies in the RPL DODAG. When the RPL network is stable, the interval of the trickle timer will increase exponentially. When a network inconsistency is detected, such as a loop generation, the trickle timer is reset.

As briefly discussed in Section II-B, these basic defense mechanisms are far from adequate when advanced attacks are launched.

IV. PROPOSED SELECTIVE FORWARDING ATTACK

Among diverse attacks against RPL networks, some attacks aim to cause as severe damage as possible to the network

Algorithm 1: Protocol-based Attack

Input: *Packet* (the Struct of IPv6 packet that need to be forwarded)

- 1 **if** *Packet.protocol* \neq *RPL* **then**
- 2 | Drop packet;
- 3 **end**

within a short time period, such as blackhole attacks [31]. These attacks are often easily detected and isolated by the defense mechanisms due to the aggressiveness of the malicious nodes' behaviors. Therefore, selective forwarding attacks, which can interrupt network communications in a flexible and stealthy way, are often launched to cause long term network damages. In this study, we propose an advanced selective forwarding attack model with three different types of selective behaviors.

1) *Protocol-based Attack*: We propose to selectively drop network packets according to their protocol types. Specifically, we propose to drop only data packets (i.e. non-ICMP packets) to achieve attack stealthiness. This is because in RPL networks, the loss of IPv6-based control messages (i.e. ICMP packets) will cause inconsistencies in network routing topology and trigger the RPL repair mechanisms. However, since data messages are transmitted based on UDP, the loss of such messages is difficult to be detected by RPL's self-defense mechanisms. In such cases, the RPL's self-recovery mechanisms will not be triggered [10].

As shown in Algorithm 1, before the malicious node forwards a packet, it first determines whether the type of the packet is a data packet. If so, this packet can be dropped. Please note that, to enable selective behaviors based on protocol types, the malicious node only needs to check the header part of the messages, which will not incur significant processing power.

2) *Packet Forward Rate-based Attack*: We assume that the data messages transmitted between nodes are encrypted, so that the attacker can control whether the malicious nodes discard the data packet but cannot change the content of the data packet. The malicious nodes can dynamically adjust their packet forward rates (PFR) according to the network conditions. It is not easy to determine an appropriate PFR, which can cause non-trivial damages to the network while avoiding being detected. In this study, we propose to achieve this goal by controlling the PFR to be slightly above the average PFR of the network. Specifically, the malicious node estimates the average PFR by monitoring all its neighbors' incoming and outgoing packets, and ensures that its PFR is slightly above the average PFR of the network (i.e. by a small value ε). Please note this PFR will be dynamically updated based on the changes of the average network PFR. Consequently, the malicious node can hide itself while still causing long term damage to the network. The equation to calculate a node N_i 's PFR is shown below

$$F_i^{\Delta t} = \frac{S_i^{\Delta t}}{R_i^{\Delta t}} \quad (1)$$

Algorithm 2: Packet Forward Rate-based Attack

Input: *Packet* (the Struct of IPv6 packet that need to be forwarded), $F_m^{\Delta t}$ (PFR of the malicious node within duration Δt), $\bar{F}^{\Delta t}$ (Estimated network average PFR within duration Δt)

- 1 **if** $F_m^{\Delta t} = (\bar{F}^{\Delta t} + \varepsilon)$ **then**
- 2 | Drop this packet;
- 3 **end**

where $R_i^{\Delta t}$ and $S_i^{\Delta t}$ represent the number of packets received and forwarded by node N_i within time duration Δt , respectively. In the proposed attack, a malicious node estimates the network average PFR (i.e. $\bar{F}^{\Delta t}$) based on its neighbors' PFR, as shown below

$$\bar{F}^{\Delta t} \approx \frac{\sum_{i=1}^p F_i^{\Delta t}}{p} \quad (2)$$

where p represents the number of neighbors of the malicious node.

With a larger value of p and a longer time duration Δt , the malicious node can achieve a more accurate estimation of $\bar{F}^{\Delta t}$, which, however, will also cause extra energy consumption and time delay. The proposed attack can flexibly adjust the trade-off according to specific attack scenarios. As shown in Algorithm 2, before forwarding a packet, the malicious node determines whether to drop the packet by comparing the its current PFR with the observed network average PFR plus ε . The value of ε can be adjusted according to the aggressiveness of the attack.

3) *Bad-mouthing Attack*: The proposed attack can arbitrarily choose one or multiple children nodes to achieve bad-mouthing attack. In bad-mouthing attack, the malicious node can frame the victim node (i.e. one of its children nodes) up by discarding data packets from the victim node.

For this attack, the most challenging part is to selectively choose the victim nodes and attack strategy. Since blindly selecting children nodes to attack or attacking all children nodes increase the risk of the attacker being exposed to the detection mechanism, we propose to only select specific children nodes as the victim nodes. More importantly, the attacker can flexibly choose victim nodes that are either located at critical network positions, or requiring minimum attack effort. For example, as shown in Algorithm 3, the malicious node identifies its children nodes with lower PFR as the victims as badmouthing these victims requires dropping less number of packets (i.e. less attack effort).

After identifying the ideal victim node, the malicious node can dynamically discard the victim's packets, misleading the root node to identify the victim as a malicious node that drops packets. If multiple malicious nodes coordinately launch attack at the same time, the false alarm rate will significantly increase for most trust-based defense solutions.

Please note that although we discuss these three attack behaviors independently for the sake of clarification, these attacks can be flexibly integrated to cause more damage.

Algorithm 3: Bad-mouthing

Input: *Packet* (the Struct of IPv6 packet that need to be forwarded), N_m (the Struct of malicious node), $\bar{F}^{\Delta t}$ (Estimated neighbor's average PFR), *ChList* (list of children nodes), *NumVict* (number of victim children nodes)

```

1 sort ChList in ascending order of PFR;
2 for  $n = 0; n \leq \text{NumVict}; n++$  do
3   if Packet.SrcMacAddr = ChList[ $n$ ].MacAddr
   and Packet.DesMacAddr =  $N_m$ .MacAddr
   and  $F_{ChList[n]} \geq \bar{F}^{\Delta t}$  then
4     Drop packet;
5     break;
6   end
7 end

```

V. LIGHTWEIGHT TRUST-BASED DEFENSE SCHEME

In this section, we propose a lightweight trust-based defense scheme, which is deployed on the root-node, against selective forwarding attacks, as shown in Figure 2. The input for the defense scheme is the data packets received from non-root nodes. The defense scheme includes three major modules. The *detection module* analyzes the trust value of each node based on the received data packets, whose propagation path is shown by the solid black line in Figure 2. After malicious nodes (e.g. node 4 in Figure 2) are identified, the *notification module* encapsulates such information in DIO packets and notifies all the nodes in the network, as indicated by the brown dashed line in Figure 2. In the *isolation module*, children nodes of the identified malicious nodes (e.g. node 7 in Figure 2) can isolate the malicious nodes and re-select their parent nodes based on received DIO messages. For example, the changed propagation path of data packets from node 7 is shown as the blue dashed line in Figure 2.

In the rest of this section, we first introduce the design of the trust model, which is the core of the proposed defense solution. Then, we discuss each module of the proposed scheme in details.

A. Trust Evaluation Model

An advanced trust model is designed to evaluate the anomaly of each node's behavior. In a RPL network, the root node tracks the behaviors of each individual node n_i and dynamically calculates the trust value, which is denoted as T_i . The trust value falls in the range from zero to one. When a node's trust value is below a trust threshold δ , it will be identified as a malicious node.

In the proposed trust model, the overall trust value T_i of a node n_i is composed of two parts: a self-trust value T_i^s , which is to capture failures of packet forwarding, and a tree-based descendant trust value T_i^d , which is to capture bad-mouthing attacks.

1) *Self-trust Value*: The proposed scheme adopts the Beta trust model [33] as its basis. Beta distribution is a family of continuous probability distributions that are often used to

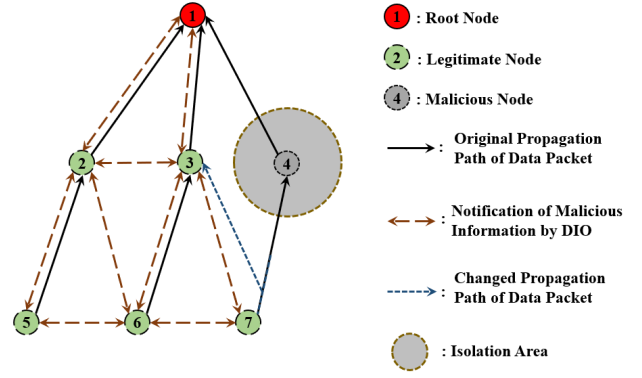


Fig. 2: Packet Propagation in the Proposed System

model binary events. In our case, we consider whether a node can successfully forward a packet or not as a random binary event. Then based on the prior observations on the number of successful and failed packet forwarding events, the probability of this node to successfully forward the next packet can be estimated as the expected value of the beta distribution, as shown in equation (3), where S and L represent the number of packets successfully sent or lost by node n_i , respectively.

$$T_i^s = \frac{S + 1}{(S + L + 2)} \quad (3)$$

From equation (3), we can observe that a node's self-trust value increases when more packets are successfully forwarded, or drops when more packets are lost.

However, the basic beta-based trust model cannot capture alternative behavior attacks [34], where malicious nodes alternatively perform good behaviors to accumulate high trust values and bad behaviors to interrupt network traffic. To prevent such attacks, we propose to introduce temporal information to discount a node's packet forwarding behaviors performed long time ago. Specifically, we introduce a discounting factor f to gradually forget a node's behavior over time, so that a behavior with a smaller discounting factor indicates a lower influence on the node's trust value. The calculation of the discounting factor for the k^{th} behavior of a node is shown below.

$$f_k = e^{-\lambda(q-k)} \quad (4)$$

where q represents the total number of behaviors performed by the node so far, including both successful and failed forwarding behaviors; and $(q - k)$ represents the total number of behaviors performed after behavior k . In other words, each time when a node performs a new behavior, all previous behaviors will have their $(q - k)$ value increased by 1, resulting in a smaller discounting factor. Please note that the latest behavior will always have its $\Delta_k = 0$, leading to its discounting factor value as 1. In addition, the forgetting speed λ is a constant value. A larger λ value will result in a smaller discounting factor and thus a higher forgetting speed.

More importantly, to punish bad behaviors further, we propose to design an asymmetric trust model so that past good behaviors can be quickly forgotten while past bad behaviors will be remembered for a longer time. To achieve this goal, two different λ values (i.e. λ_g and λ_b) are adopted to separately

discount past good and bad behaviors respectively, where $\lambda_g > \lambda_b$. Therefore, the self-trust value of node i with q behaviors (i.e. $T_{i,q}^s$) is calculated as follows.

$$T_{i,q}^s = \frac{\sum_{k=1}^q a_k f_{k,g} + 1}{\sum_{k=1}^q (1 - a_k) f_{k,b} + \sum_{k=1}^q a_k f_{k,g} + 2} \quad (5)$$

In equation (5), a_k value is 1 if behavior k is good, or 0 if behavior k is bad.

2) *Tree-based Descendant Trust Value*: Due to the restricted hierarchical structure of RPL networks, only parent nodes will forward data packets for their children nodes. It is very easy for a malicious parent node to control the packet transmissions of one of its children nodes to launch bad-mouthing attacks against this child. By only considering self-trust value, the victim child node's trust will drop while the malicious parent's trust value remains the same. To further defeat such attacks, we propose to also introduce a tree-based descendant trust value for each node, which considers the trust value of its direct descendants. The descendant trust value T_i^d is defined as follows.

$$T_i^d = \frac{\sum (w_j T_j^s)}{C_i} \quad (6)$$

where T_j^s is the self-trust value of node n_i 's child node n_j . Parameter w_j represents the weight of the child node n_j , which is determined by the number of data packet received by n_j per time period. The greater number of data packets received by n_j means larger weight assigned to node n_j . In addition, C_i denotes the total number of packets received by node n_i .

Please note that in the proposed scheme, a node's descendant trust value only depends on its children's self-trust value. Since RPL is a tree-like network, if a node's descendant trust value also considers its children's descendant trust values, it will lead to a recursive counting, where leaf nodes' trust values are over-emphasized. Since the descendant trust value is mainly designed to prevent bad-mouthing attacks, which can only effectively attack children nodes, we propose to not recursively count it.

When the malicious node bad mouth any of its children nodes, its own descendant trust value will be decreased. Moreover, the descendant trust value will significantly decrease if the number of the victim children nodes increases.

3) *Aggregated Trust Value*: Finally, the aggregated trust value T_i of the node n_i is the combination of the self-trust value T_i^s and the descendant trust value T_i^d . The calculation of the aggregated trust value is shown below.

$$T_i = w_s T_i^s + w_d T_i^d \quad (7)$$

where w_s and w_d are the weights for self-trust value and descendant trust value. This aggregated trust value will serve as the major criteria to identify suspicious nodes in the network.

B. Detection Module

In this section, we present the detection module, which involves the above proposed trust model as its core. Specifically, we make two assumptions. First, the data messages transmitted

between nodes are encrypted, meaning that malicious nodes on the routing path cannot change the content of the transmitted data information. Second, all data messages generated by non-root nodes are transmitted through the root node to the external network (e.g. the Internet). This is a reasonable assumption for most RPL networks [10], [35].

Current RPL protocol does not support the root node to record and track the packets sent by non-root nodes. To address this challenge, we introduce a sequence number, which is stored in the first byte of the data payload sent by each node. In particular, the sequence number is increased by one each time when the source node sends out a packet. The root node estimates the packet forward rate for each source node based on the number of received data packets and the corresponding sequence numbers.

In addition, since frequently calculating the trust value for each node greatly increases the workload of the root node, we introduce a sliding time window. The root node only calculates the trust value of each node once in each sliding time window. The length of the window can be determined according to the network status, such as the battery capacity of the root node and the sensitivity of the trust value.

By calculating the trust value of each node, the root node can identify the possible malicious nodes according to Algorithm 4. Confirmed malicious nodes will be added to the "blacklist". However, because the nodes may suffer from bad-mouthing attacks from their parent nodes, it may lead to high false alarm rate if we directly add all nodes with low trust values to the "blacklist".

To reduce the false alarm rate caused by bad-mouthing attacks, we propose to add a "watchlist" and a trust recovery time period. When the trust value of a node is lower than the threshold for the first time, it will be added to the "watchlist" as a suspicious node. If a suspicious node is required to change its parent for further investigation, the root node will reset a recovery timer and track if the suspicious node's trust value can recover after the parent change action. Please note that the length of the recovery timer can be determined according to the specific network status. A longer timer leads to longer detection delay but lower false alarm rate. Within the recovery time period, if the trust value of the suspicious node recovers back to the threshold, indicating that changing its parent node stops the anomaly, this node is considered as a normal node. Then its parent will be identified as the malicious node and added to the "blacklist". Otherwise, the suspicious node is identified as a malicious node and moved to the "blacklist".

C. Notification Module

After a malicious node is identified, the root node needs a reliable way to notify all the children nodes of the malicious node while avoiding information storms. This is challenging since the RPL network follows a strict tree-like topology for data packet forwarding, and a node can only receive data packets from its parent. It means that if data packets are used to disseminate the notifications, these packets will be simply dropped by malicious nodes and never reach their children nodes.

Algorithm 4: Malicious nodes detection

Input: trust threshold T_{th} , *ParentChangedList* (The nodes in the list have changed parent.)

Output: *blacklist* (containing malicious node ID), *watchlist* (containing suspicious node ID),

- 1 INITIAL: *watchlist*[] = empty, *blacklist*[] = empty, *timeperiod* = constant value, *TrustRecoveryTime*[] = constant value;
- 2 **while** *timeperiod* expires **do**
- 3 **foreach** $j \in \text{nodes}$ **do**
- 4 **if** $T_j < T_{th}$ **then**
- 5 **if** node n_j not in *watchlist* **then**
- 6 *watchlist.add*(n_j);
- 7 **else**
- 8 **if** *ParentChangedList*[j] = True and *TrustRecoveryTime*[j] expires **then**
- 9 *watchlist.remove*.(n_j);
- 10 *blacklist.add*(n_j);
- 11 *ParentChangedList*[j] = False;
- 12 **end**
- 13 **else**
- 14 **if** *ParentChangedList*[j] = True and node n_j in *watchlist* **then**
- 15 *blacklist.add*(n_j .old_parent);
- 16 *watchlist.remove*(n_j);
- 17 *ParentChangedList*[j] = False;
- 18 **end**
- 19 **end**
- 20 **end**
- 21 **end**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RPL InstanceID								Version Number								Rank															
G	O	MOP		Prf		DTSN								Flags				Reserved													
DODAG ID																															
Options(s)...																															
Malicious Node ID																															

Fig. 3: The proposed notification message

In this work, we propose to use the control messages (i.e. ICMPv6 messages) to disseminate these notifications, which can reach the children of a malicious node through other neighbor nodes. In particular, we recommend using the first byte in the payload of the ICMPv6 control messages to store the node ID, as shown in Figure 3. The first bit can be used to distinguish the suspicious node, which is set to 0, and identified malicious node, which is set to 1.

In order to reduce the energy consumption of the network, the root node executes the notification module right after the detection module. As shown in Algorithm 5, there are two lists: a “blacklist” and a “ParentChangingList”. The root node marks the node information in the two lists separately. The “blacklist” is used to notify all the children nodes, whose parents are identified malicious nodes in the “blacklist”, to re-select their parent node. The suspicious nodes, which are required to change parents, are added into the “ParentChangingList”. This distinction can avoid unnecessary parent selec-

Algorithm 5: Malicious nodes notification

Input: *blacklist*, *watchlist*

- 1 INITIAL: *ParentChangingList*[] = empty, *ParentChangedList*[] = False;
- 2 **while** (*watchlist* not empty or *blacklist* not empty) and *SendTime* **do**
- 3 **if** *watchlist* not empty **then**
- 4 *ParentChangingList.add*
- 5 (*FindMaxRankNode*(*watchlist*));
- 6 *broadcast*(*ParentChangingList*);
- 7 **foreach** $i \in \text{ParentChangingList}$ **do**
- 8 *ParentChangedList*[i] = True;
- 9 **end**
- 10 **end**
- 11 **if** *blacklist* not empty **then**
- 12 *broadcast*(*blacklist*);
- 13 **end**
- 14 **end**

Algorithm 6: Isolation malicious nodes

Input: *Packet* (received ICMP6 control message which contains malicious node ID M_{id} or suspicious node SUS_{id} , Parent ID P_{id} , Node ID N_{id})

- 1 **if** *Packet.M_{id}* = P_{id} or *Packet.SUS_{id}* = N_{id} **then**
- 2 *RplRemoveParent*(P_{id});
- 3 *PreferredParent* = *RplSelectParent*(*dag*);
- 4 *broadcast*(*Packet*);
- 5 **else**
- 6 *broadcast*(*Packet*);
- 7 **end**

tion process for non-relevant nodes in the network.

In addition, the root node finds sub-trees, which can cover minimum number of nodes in the “watchlist”, based on RPL tree-like topology. Then, the root node adds the nodes with the highest rank value in each sub-tree to “ParentChangingList”, which is the function “FindMaxRankNode” in Algorithm 5.

D. Isolation Module

As shown in Algorithm 6, when a non-root node in the network receives the notification message, it checks whether the node information in notification is its parent or itself. If any of two cases is true, this node removes the current parent from its parent list. Then, it re-selects its preferred parent and broadcasts this ICMPv6 control message to its neighbors. If not, the node broadcasts the ICMPv6 control message directly to all its neighbors.

After these three modules are completed, the malicious nodes launching selective forwarding attacks will be abandoned and isolated from the network.

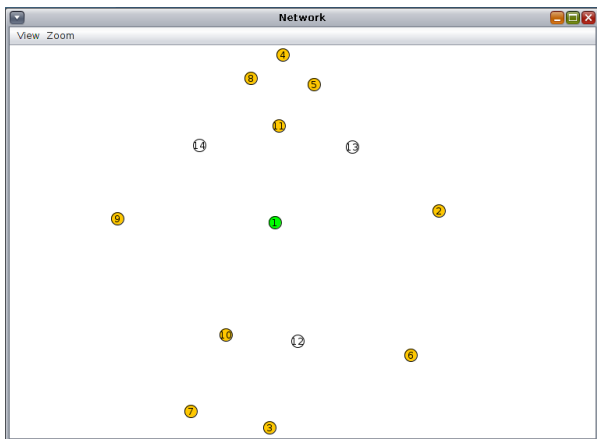


Fig. 4: Example of RPL Network Topology

VI. EXPERIMENT AND RESULT ANALYSIS

A. Experiment Set Up

This work adopts Cooja, which is a network simulator of Contiki OS [36], as our experimental platform. Specifically, fifteen nodes are randomly deployed in the experiments as shown in Figure 4, including a root node (i.e. node 1), eleven legitimate non-root nodes (i.e. node 2 to node 11) and three malicious nodes (i.e. node 12 to node 14). Minimum Rank with Hysteresis Objective Function (MRHOF) is selected as the objective function, where children nodes select their preferred parents according to ranks and Expected Transmission Count (ETX) values. The simulation parameters are summarized in TABLE I.

Based on the experiments, the performance of the proposed attack and defense scheme are tested and then compared with other state-of-the-art works. Specifically, this work adopts receiver operating characteristic (ROC) curve as the major performance metric because it can effectively reflect the trade-off between the detection rate and the false alarm rate when different thresholds are adopted. In each ROC curve, the x-axis and y-axis represent the detection rate and false alarm rate, respectively. The area under ROC curve (AUC) represents the accuracy. Larger area under the curve (i.e. the higher the AUC value) indicates better performance.

B. Performance of Proposed Attacks

In this experiment, malicious nodes may either selectively drop the victim node's data packets based on their perceived average network PFR, or launch bad-mouthing attacks against a specific child node. To illustrate the impact of different attacks, the ratio of bad-mouthing attacks to the total number of attacks in the network is divided into four cases, 25%, 50%, 75% and 100%.

The performance of the proposed attacks is evaluated against three defense schemes. In the first two schemes, the root node identifies the malicious nodes by comparing the PFR of each non-root node with a threshold value. The nodes with lower PFR are identified as malicious. The difference is that the first scheme (i.e. avg scheme) uses a node's average PFR, while the second scheme (i.e. rec scheme) uses only a node's

TABLE I: Simulation Parameters

Parameter	Value
Simulation platform	Contiki/Cooja 3.0
MAC	CSMA/CA
Transport	UDP/IPv6
Emulated nodes	Z1 mote
Simulation coverage area	130 m * 130 m
Total number of nodes	15
Malicious nodes	3
TX range	50 m
Interference	range 100 m
Packet size	46-byte
Data packet period	60 seconds
Routing protocol	RPL
Network protocol	IP based
Simulation time	150 minutes
Link failure model	UDGM with distance

most recent PFR. In the third scheme (i.e. def scheme), since the default RPL security scheme allows a node to re-select its parent node when network failures (e.g. link failure and node failure) are detected, the root node identifies malicious nodes by checking whether a node is discarded by its children.

The effectiveness of proposed attack model is illustrated in Figure 5. In all sub-figures of Figure 5, the def scheme shows the lowest performance with 0.44 average AUC. This is because the proposed attacks only drop data packets, which rarely cause failures in network routing. Although avg scheme and rec scheme show slightly better performances (with an average AUC as 0.70 and 0.65 respectively in Figure 5 (a)-(b)), their performances significantly drop (with an average AUC as 0.36) when the proportion of bad-mouthing attacks increases to above 75%, as shown in Figure 5 (c)-(d). This is because these two schemes consider a node with low PFR (i.e. either average PFR or the most recent PFR) as a malicious node, which can be taken advantage by bad-mouthing attacks to frame up the victim nodes.

C. Performance of Proposed Defense Modules

In this sub-section, we evaluate the effectiveness of each critical strategy proposed for the defense scheme. In particular, these strategies include (1) discounting factor, (2) asymmetric forgetting speed, and (3) integration of self-trust and descendant trust. Furthermore, the proposed attack models are launched with 50% selective forwarding behaviors and 50% bad-mouthing behaviors.

1) *Effectiveness of Discounting Factor*: In this subsection, λ_g and λ_b are set as the same value λ . By changing the values of λ , Figure 6 illustrates its impact on the performance of the proposed defense scheme. Specifically, four different λ values (i.e. 0, 0.3, 0.8, 100) are applied so that the discounting factor f_k ranges in the interval $[0, 1]$. From Figure 6, it can be observed that the defense scheme shows the best performance (e.g. AUC = 0.630) when $\lambda = 0.3$.

Specifically, when $\lambda = 100$, $f_k = e^{-\lambda(q-k)} = e^{-100(q-k)}$, which is approximately 0 for any $q - k > 0$. It indicates

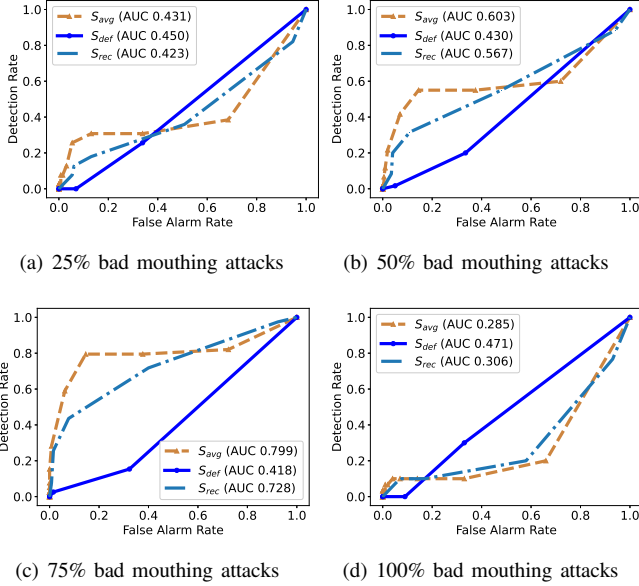


Fig. 5: Performance of the proposed attack model against three defense schemes. S_{avg} , S_{def} and S_{rec} represents avg scheme, def scheme and rec scheme, respectively.

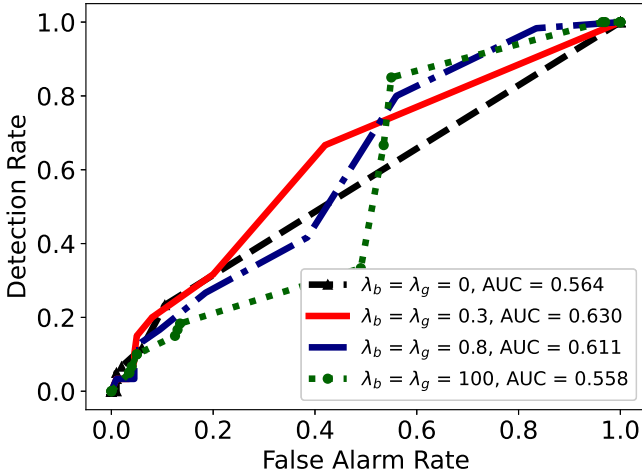


Fig. 6: Effectiveness of various forgetting speeds

that the defense scheme only remembers the most recent behavior for trust evaluation. In such settings, the defense scheme can mistakenly identify normal nodes with accidental packet losses as malicious nodes and therefore results in high false alarm rates, as shown by the green dotted curve in Figure 6. On the other hand, when the value of λ is 0, the defense scheme, which remembers all previous behaviors, also performs worse, as shown by the blue dash dot line in Figure 6. This is because malicious nodes can easily mislead the defense scheme by accumulating high trust values through good behaviors performed long time ago.

With an appropriate λ value, the proposed defense scheme can achieve high performance. Particularly, as shown in Figure 5 (b), when $\lambda = 0.3$, the AUC of the scheme with is 4% and 11% higher than that of the avg and rec schemes respectively,

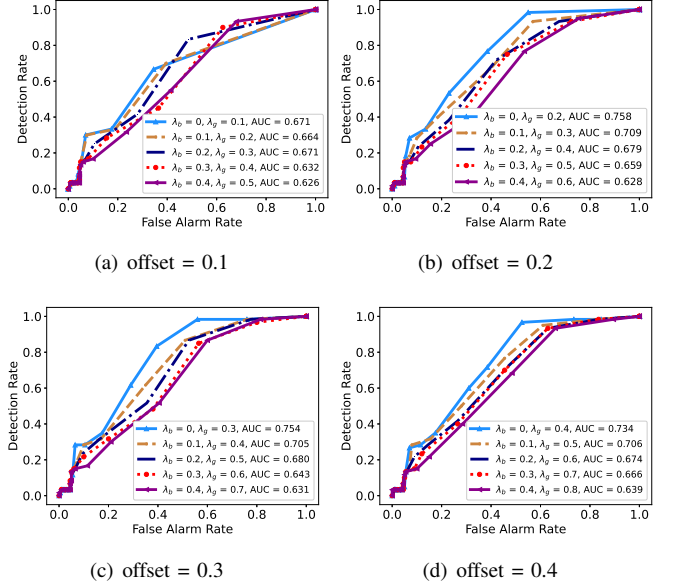


Fig. 7: Effectiveness of asymmetric forgetting speeds. Offset represents the difference between forgetting speeds λ_b and λ_g .

validating the effectiveness of the proposed defense scheme.

2) *Effectiveness of Asymmetric Forgetting Speeds:* Next, we evaluate the effectiveness of asymmetric forgetting speeds in Figure 7. Observed from Figure 6, the defense scheme with $\lambda = 0.3$ shows the best performance. Therefore, in this subsection, the value range of λ_b is set from 0 to 0.4. As we propose to forget bad behavior slower, the λ_g values are set to be 0.1, 0.2, 0.3 and 0.4 higher than λ_b . The results are shown in Figure 7.

Comparing the schemes in Figure 6 and Figure 7, it can be observed that 79% of the schemes with asymmetric forgetting speeds in Figure 7 perform better than the best scheme (i.e. $\lambda_b = \lambda_g = 0.3$) in Figure 6. In particular, the AUC of the best scheme (i.e. $\lambda_b = 0$ and $\lambda_g = 0.2$) in Figure 7 (b) is 20% higher than that of the best scheme in Figure 6. This observation validates the effectiveness of the asymmetric forgetting speed.

Furthermore, an appropriate offset (e.g. 0.2) between the two forgetting speeds makes the scheme perform better. Specifically, the scheme with $\lambda_b = 0$ and $\lambda_g = 0.2$ in Figure 7 (b) performs the best with the AUC as 0.758, which is 13%, 0.5%, and 3% higher than the highest AUCs in Figure 7 (a), (c), and (d). When the offset is too large, the scheme will be too sensitive to bad behaviors, which may increase the false alarm rate. When the offset is too small, the performance will be very similar to that of the scheme with identical λ_b and λ_g , which is less effective.

3) *Effectiveness of Integrating Self-Trust and Descendant Trust:* To detect bad-mouthing attacks, where the parent node frames its children nodes by discarding their data packets, we propose to integrate self-trust (T^s) and descendant trust (T^d). This subsection aims to evaluate the trade-off by adjusting the weights of these two aspects. In particular, w_s and w_d represent the weights assigned to T^s and T^d , respectively,

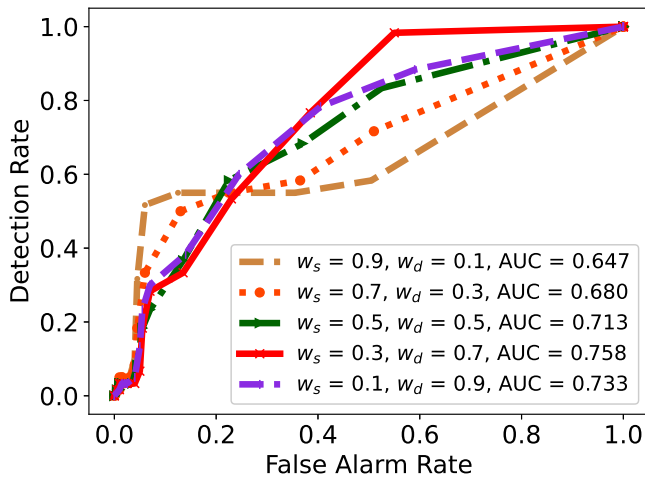


Fig. 8: Comparison of effectiveness of combination of two trust factors

where the sum of w_s and w_d equals one. The values of w_s and w_d are in the range of $[0.1, 0.9]$ with the step interval as 0.1. The values of λ_b and λ_g are fixed as 0 and 0.2 (i.e. the optimal values from Figure 7), respectively.

From Figure 8, the proposed defense scheme with $w_s = 0.3$ and $w_d = 0.7$ achieves the best performance. Furthermore, when $w_s \in [0.2, 0.4]$ and $w_d \in [0.6, 0.8]$, the AUC of the schemes are higher than 0.746, which validates the robustness of the proposed defense scheme.

More importantly, the AUC of the best scheme in Figure 8 is 26% and 34% higher than the avg scheme and rec scheme in Figure 5 respectively. This observation validates that the introduction of descendant trust value (T^d) enables the proposed defense scheme to effectively detect bad-mouthing attacks in the network.

Some of the schemes in Figure 8 have an inflection point in the detection rate range of $[0.55, 0.65]$, indicating that the increment of detection rate slows down. This is because this experiment places a limited number of nodes, including three malicious nodes, to prevent exceeding the capacity of the simulation platform (e.g. memory overflow).

D. Overall Performance Comparison

This section compares the proposed defense scheme with the state-of-the-art defense schemes, including: (1) the RPL default recovering scheme: MRHOF [5], (2) a centralized scheme: Heartbeat protocol (HP) [10], and (3) a distributed scheme: Trust-Aware RPL Routing Protocol (TPRP) [12]. All the schemes are applied on the same network topology and settings. In addition, the proposed defense scheme with $\lambda_b = 0$, $\lambda_g = 0.2$, $w_s = 0.3$ and $w_d = 0.7$ is launched.

1) *Detection Accuracy*: As shown in Figure 9, the proposed scheme yields the best overall performance among four defense schemes. Its AUC is 76%, 48% and 10%, higher than that of MRHOF, HP and TPRP, respectively.

In particular, the HP and MRHOF schemes fail to differentiate malicious nodes from normal ones, because these two schemes detect malicious nodes based on replies of the control

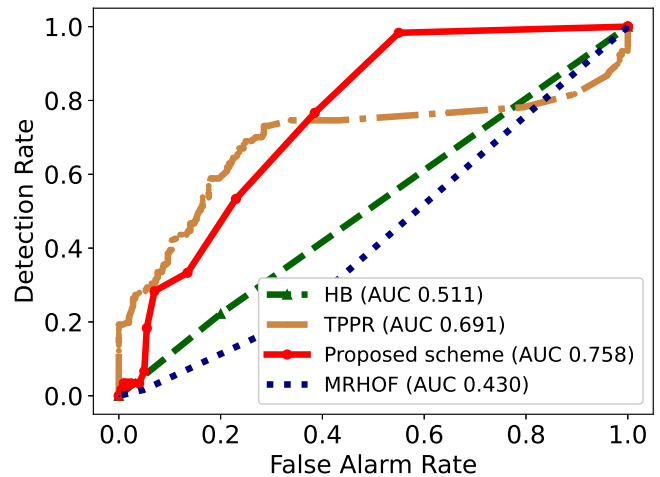


Fig. 9: Comparison of performance of different scheme based on proposed attack model

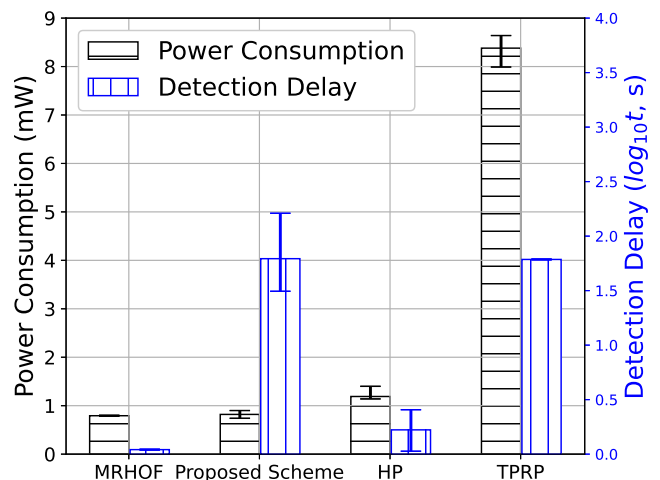


Fig. 10: Power consumption and detection delay comparison based on proposed attack mode

messages and thus cannot effectively capture the loss of data packets. Furthermore, when the false alarm rate is lower than 0.4, the detection rate of TPRP scheme stays high. This is because the TPRP scheme can effectively defend against bad-mouthing attacks by requiring each node in the network to monitor the sending and receiving packets of its neighbor nodes. However, the detection rate of TPRP scheme cannot be significantly improved. This is because it adopts average PFR as the trust value, which cannot effectively capture attacks alternatively perform good and bad behaviors.

2) *Energy Consumption and Detection Delay*: Figure 10 compares the detection delay and energy consumption of legitimate non-root nodes for different defense schemes. Specifically, the proposed attacks are launched against these defense schemes. Each defense scheme has two bars, representing its energy consumption and detection delay respectively.

From Figure 10, we can observe that the proposed defense scheme consumes very limited power, only 3.4% more than the default RPL recovery scheme (i.e. MRHOF). This is because the proposed scheme adopts the centralized design, where non-

root nodes do not need to monitor neighbor nodes' activities, but only forward notification packets on detected malicious nodes.

On the other hand, the HP and TPRP schemes are causing extra 50% and 957% power respectively when compared to the default MRHOF scheme. Although the HP scheme also adopts a centralized defense mechanism, it frequently launches a "request and reply" process between the root node and non-root nodes, which increases power consumption. Furthermore, the power consumption of TPRP scheme is the highest due to its distributed design, which requires each non-root node in the network to monitor, analyze, and share the activities of its neighbors. These requirements significantly increase extra work time and computational costs for each node, leading to much higher power consumption.

The detection delay is calculated based on only the successful detection of each scheme. In other words, if a detection scheme can only detect two malicious nodes out of three, the detection delay is the average delay of the two successful detection. As shown in Figure 10, the MRHOF and HP schemes show small detection delay, which, however, is calculated based on the very limited malicious nodes that can be detected. Furthermore, there is a relatively large variation in the detection delay of the HP scheme. This is because the HP scheme relies on the exchange of "request-reply" messages to detect anomaly. The detection delay may vary based on the frequency of the request messages. A higher frequency may lead to smaller detection delay, but higher power consumption.

In addition, the proposed scheme yields similar detection delay as the TPRP scheme, but a larger variation. This is because in the TPRP scheme, rather than relying on notifications from the root node, each node directly monitors its neighbors' behaviors, resulting in a relatively stable delay. The proposed scheme, however, can only detect anomaly when the data packets arriving at the root node show abnormal patterns, which may vary according to the network's data rate and the source node's network location.

In summary, compared to the state-of-the-art defense schemes, the proposed scheme yields much higher detection accuracy (i.e. 10% higher than the second highest one) and lower energy consumption (i.e. 31% lower than that of the second lowest one). Although its detection delay is higher than other schemes (i.e. the MRHOF and HP schemes), it is practical to be applied in a low power low data rate RPL network.

VII. CONCLUSION

In RPL networks, malicious nodes can damage routing paths by selectively dropping packets. In this paper, we propose an advanced selective forwarding attack with three flexible attack behaviors, including protocol selection, packet forward rate selection and bad-mouthing with children nodes selection. The flexibility enables the malicious nodes to hide their attack behaviors and maximize the long term attack impact. Furthermore, we propose a new centralized trust-based defense scheme, which consists of a self-trust based on the beta trust model with asymmetric forgetting speeds, and a

descendant trust value based on the RPL tree-like topology. Experimental results show that compared to the state-of-the-art defense solutions, the proposed defense scheme can effectively detect advanced proposed attacks with very limited energy consumption.

REFERENCES

- [1] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.
- [2] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, "Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment," *Sensors*, vol. 20, no. 22, p. 6420, 2020.
- [3] B. Marques and M. Ricardo, "Energy-efficient node selection in application-driven wsn," *Wireless Networks*, vol. 23, no. 3, pp. 889–918, 2017.
- [4] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6lowpan: a study on qos security threats and countermeasures using intrusion detection system approach," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1189–1212, 2012.
- [5] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, R. K. Alexander *et al.*, "Rpl: Ipv6 routing protocol for low-power and lossy networks." *rfc*, vol. 6550, pp. 1–157, 2012.
- [6] F. Gara, L. B. Saad, R. B. Ayed, and B. Tourancheau, "Rpl protocol adapted for healthcare and medical applications," in *2015 International wireless communications and mobile computing conference (IWCMC)*. IEEE, 2015, pp. 690–695.
- [7] J. Nassar, M. Berthomé, J. Dubrulle, N. Gouvy, N. Mitton, and B. Quoitin, "Multiple instances qos routing in rpl: Application to smart grids," *Sensors*, vol. 18, no. 8, p. 2472, 2018.
- [8] S. Junior, A. Riker, B. Silvestre, W. Moreira, A. Oliveira-Jr, and V. Borges, "Dynasti—dynamic multiple rpl instances for multiple iot applications in smart city," *Sensors*, vol. 20, no. 11, p. 3130, 2020.
- [9] P. Pongle and G. Chavan, "A survey: Attacks on rpl and 6lowpan in iot," in *2015 International conference on pervasive computing (ICPC)*. IEEE, 2015, pp. 1–6.
- [10] 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, p. 794326, 2013.
- [11] G. Ma, X. Li, Q. Pei, and Z. Li, "A security routing protocol for internet of things based on rpl," in *2017 International Conference on Networking and Network Applications (NaNA)*. IEEE, 2017, pp. 209–213.
- [12] D. Airehrour, J. Gutierrez, and S. K. Ray, "A trust-aware rpl routing protocol to detect blackhole and selective forwarding attacks," *Journal of Telecommunications and the Digital Economy*, vol. 5, no. 1, pp. 50–69, 2017.
- [13] 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.
- [14] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in rpl-based internet of things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.
- [15] Z. A. Almusaylim, A. Alhumam, and N. Jhanjhi, "Proposing a secure rpl based internet of things routing protocol: a review," *Ad Hoc Networks*, vol. 101, p. 102096, 2020.
- [16] A. Le, J. Loo, Y. Luo, and A. Lasebae, "The impacts of internal threats towards routing protocol for low power and lossy network performance," in *2013 IEEE Symposium on computers and communications (ISCC)*. IEEE, 2013, pp. 000 789–000 794.
- [17] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, and A. Durresi, "Routing loops in dag-based low power and lossy networks," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010, pp. 888–895.
- [18] F. Gara, L. B. Saad, and R. B. Ayed, "An intrusion detection system for selective forwarding attack in ipv6-based mobile wsn," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 276–281.
- [19] Y. Hu, Y. Wu, H. Wang *et al.*, "Detection of insider selective forwarding attack based on monitor node and trust mechanism in wsn," *Wireless Sensor Network*, vol. 6, no. 11, p. 237, 2014.
- [20] A. Mathur, T. Newe, and M. Rao, "Defence against black hole and selective forwarding attacks for medical wsn in the iot," *Sensors*, vol. 16, no. 1, p. 118, 2016.

- [21] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3718–3731, 2016.
- [22] D. Airehrour, J. Gutierrez, and S. K. Ray, "Securing rpl routing protocol from blackhole attacks using a trust-based mechanism," in *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2016, pp. 115–120.
- [23] M. A. Lodhi, A. Rehman, M. M. Khan, and F. B. Hussain, "Multiple path rpl for low power lossy networks," in *2015 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. IEEE, 2015, pp. 279–284.
- [24] T. L. Jenschke, R.-A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, "Multi-path selection in rpl based on replication and elimination," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2018, pp. 15–26.
- [25] C. Pu and S. Hajjar, "Mitigating forwarding misbehaviors in rpl-based low power and lossy networks," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–6.
- [26] Z. A. Khan, J. Ullrich, A. G. Voyiatzis, and P. Herrmann, "A trust-based resilient routing mechanism for the internet of things," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–6.
- [27] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, "A trust-based intrusion detection system for mobile rpl based networks," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 735–742.
- [28] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for iot security," *Journal of Information Security and Applications*, vol. 52, p. 102467, 2020.
- [29] T. ul Hassan, M. Asim, T. Baker, J. Hassan, and N. Tariq, "Ctrust-rpl: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based internet of things applications," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, p. e4224, 2021.
- [30] A. K. Gautam and R. Kumar, "An efficient packet reachability-based trust management scheme in wireless sensor networks," in *Safety, Security, and Reliability of Robotic Systems*. CRC Press, 2020, pp. 141–154.
- [31] J. Jiang, Y. Liu, and B. Dezfouli, "A root-based defense mechanism against rpl blackhole attacks in internet of things networks," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1194–1199.
- [32] S. Raza, S. Duquenooy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6lowpan with compressed ipsec," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. IEEE, 2011, pp. 1–8.
- [33] A. Josang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bled electronic commerce conference*, vol. 5, 2002, pp. 2502–2511.
- [34] N. Labraoui, M. Gueroui, and L. Sekhri, "On-off attacks mitigation against trust systems in wireless sensor networks," in *IFIP International Conference on Computer Science and its Applications*. Springer, 2015, pp. 406–415.
- [35] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the rpl-connected 6lowpan networks," in *Proceedings of the 3rd ACM international workshop on IoT privacy, trust, and security*, 2017, pp. 31–38.
- [36] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, "A survey on routing protocols supported by the contiki internet of things operating system," *Future Generation Computer Systems*, vol. 82, pp. 200–219, 2018.