EXAMPLE CHAPTER – Contributed chapter

# Chapter 9: Cyber Attacks and Countermeasures in RPL-based Industrial Internet of Things

Author(s):
*Cong Pu (Department of Computer Sciences and Electrical Engineering Marshall University, Huntington, WV 25755, USA, cong.pu@ieee.org, ORCID: https://orcid.org/0000-0002-7952-0038)
*(Include all authors' names, affiliations, and contact info. Indicate lead corresponding author with *)*

## 9.1 Abstract

In the emerging Internet of Things (IoT) era, billions of intelligent machines and devices are seamlessly interconnected with each other over the Internet to exchange information and support decision-making. The IoT is progressively becoming an important aspect of the fourth industrial revolution (widely known as Industrial Internet of Things (IIoT)) and provides an unprecedented opportunity to revolutionize traditional production and manufacturing processes. To achieve the goals and realize the vision of IIoT, a communication protocol specifically designed for networks with resource constraints and lossy communication links, a.k.a. RPL, has stood out from the crowd and quickly became a promising routing protocol for the IIoT. However, the security and privacy issues were not the major concerns when RPL was designed, thus, it fails to meet the security requirements of IIoT. In this chapter, RPL routing protocol and its major component, Trickle algorithm, are first introduced. Then, we identify and analyze various RPL-specific attacks in the IIoT, discuss their corresponding countermeasures, and present their performance impact via preliminary simulation results. Finally, we conclude the chapter with future research directions including interdisciplinary aspects and insights.

**Keywords**: Attacks, Countermeasures, Intrusion Detection, RPL, Industrial Internet of Things

## 9.2 Introduction

As we enter the third decade of the 21st century, we are heralding the dawn of a new era of Internet of Things (IoT). The conception of IoT, a large number of intelligent objects seamlessly collaborating to realize the objectives [1], has spread around the world and became an after-dinner conversation for people. For example, leveraging IoT and existing state-of-the-art technologies (i.e., 5G, blockchain, and artificial intelligence [2]), industrial ecosystem is moving towards the time of Industry 4.0, which is widely known as Industrial Internet of Things (IIoT) [3]. According to GlobeNewswire [4], the productivity benefits of IIoT are about to reach $265 billion in 2027. For example, according to the market survey report from Statista, the number of active IoT objects all over the world will reach 39 billion by 2025. As the usage of IIoT continues to grow across the world, there are high-demand jobs and employments in areas like data science, cybersecurity, and engineering & operations [5]. Data and information are at the core of IIoT. The true value of a ton of data and information generated by IIoT smart devices is to provide new insights for industry after being processed by various statistical and analytical techniques. As a substantial amount of data and information are being stored and processed, protecting them from escalating cyber attacks becomes a challenging issue. For example, ransomware can target IIoT device hardware as well as applications and data. From July to September 2020, Check Point Research has reported that the daily average number of ransomware attacks increased by 50% compared to the number obtained from January to June 2020. To protect various assets of IIoT, security experts need to be well-prepared with basic and advanced cybersecurity technologies such as intrusion detection and prevention systems, threat modeling and security analysis, and attack detection and defense. As industries keep investing in various IIoT technologies, it is no wonder that the IIoT makes significant contribution to the progress of mankind.

To achieve the goals and realize the vision of IIoT, efficient communication and reliable information distribution in the IIoT have an important role to play, deservedly, routing protocols come into the spotlight. The main goal of a routing protocol is to specify how IIoT devices communicate with each other to distribute information, which makes all IIoT devices be able to select routes between them in the IIoT network. Over the past few years, routing protocols of IIoT have received burgeoning attention in industry as well as academia. A telling example is the recent revelation that Cisco and IETF collaborate to propose a new communication scheme for networks with resource constraints and lossy communication links [6], called RPL [7], to address the devices with resource constrains such as those in the IIoT. In RPL routing protocol, there are a group of routing attributes which can be regarded as either routing constraints or routing metrics. If the routing attribute is being used as a constraint, the attribute can prune IIoT devices and communication links from candidate paths that do not respect the constraint. On the other side, the routing attribute is able to determine the least cost path if being used as a metrics. In academia, the authors in [8], [9], [10], [11], [12] proposed various communication protocols to improve the communication efficiency of IIoT network devices. For instance, in [13], the researchers design a tree-based communication protocol for IoT environment, where the consumption of energy and the delay of communication could be reduced by adopting a mobile sink. In order to maintain the routes in the IoT network, two approaches are introduced. The first approach is designed based on the traditional geographic routing protocol so that the consumption of energy can be properly balanced. The second approach tries to use a small number of control packets to maintain the tree-based routing structure. Assuredly, the above facts are enough to prove that routing protocols are the cornerstone of IIoT, and should be treated with respect.

When network engineers and academic researchers design or propose routing protocols,

however, functionality is their major focus, whereas security is an afterthought. Consequently, many well-performed routing protocols shine in a nurturing environment (i.e., network simulation), but fail in the realistic environment because they did not consider the security and privacy issues in their design. For instance, RPL routing protocol has several intrinsic and charming characteristics such as automated configuration, dynamic reaction to the change of network structure, routing loop correction, and the availability of different network instances [7]. To be specific, the feature of automatic configuration will make the network be able to discover routing paths dynamically. In terms of loop detection and avoidance, RPL routing protocol has an ability to identify routing loops whenever the topology of network changes, and is able to repair the routing loops. Basically, RPL is believed to be one of promising candidate communication algorithms for the IIoT because it has many attractive features that make it easier to satisfy various industrial applications' QoS requirements. Although RPL routing protocol has sufficient maturity, several challenging problems still remain unsolved. For example, according to RPL specification, the implementation of various security features is partially or fully optional because of concerns about system performance. As a result, RPL routing protocol becomes vulnerable to several well-known attacks inherited from wireless network and RPL specific attacks [14].

Although a large amount of effort has been dedicated to evaluating and enhancing RPL routing protocol's performance in various applications/systems, we will discuss the issues of security and privacy in RPL, mainly focusing on security attacks and countermeasures in the IIoT. This chapter is motivated in the matter of two facets. First, we concentrate on RPL routing protocol and Industrial Internet of Things, which are currently attracting a lot of attention because of their wide applicability. We carefully analyze the RPL routing protocol as well as the Trickle communication algorithm so that other researchers can obtain a better

understanding about the RPL-based IIoT. Second, we select state-of-the-art RPL specific attacks, and analyze their malicious operations as well as performance impact on IIoT, which highlights the necessity of advanced defense mechanisms and countermeasures to protect RPL-based IIoT. In this chapter, RPL routing protocol and its major component, Trickle algorithm, are first introduced. Then, we identify and analyze various RPL-specific attacks in the IIoT, and discuss their corresponding countermeasures. Finally, we conclude the chapter with future research directions including interdisciplinary aspects and insights.

## 9.3 Related Work

Since RPL routing protocol was released, many academic researchers and industrial engineers have investigated the security and privacy issues of RPL, and proposed state-of-the-art countermeasures to defend against various attacks.

In [15], the authors develop an intrusion detection system (also called DETONAR) to defend against RPL specific attacks in the IoT. First, the authors conduct extensive experiments and collect network traffic for several RPL specific attacks. The simulation results have been prepared and presented as a Routing Attacks Dataset for RPL (RADAR). Second, they develop an IDS, DETONAR, to detect some security attacks in RPL routing protocol. The main technique being utilized by DETONAR is the packet sniffing, where a group of security policies (signature/anomaly-based rules) are adopted to detect suspicious activities from incoming Internet traffic. As reported by their experimental results, DETONAR's positive detection rate exceeds 80% for ten attacks with a small amount of computation overhead. The authors in [16] first investigate the version number attack which try to exhaust network resources (e.g., energy power, memory storage, and computation capability) by targeting the global repair mechanisms of RPL routing protocol. And then, they propose a version number attack detection mechanism. In the feature extraction method,

a step forward feature selection scheme is used to choose the ideal features. Simulation and experiment results demonstrate that the detection mechanism is very competitive with good performance in detection accuracy and computation overhead. In [17], the authors conduct a literature review regarding RPL-related IDS in the IoT. [17]'s main contribution is that the authors identify several basic design requirements for intrusion detection systems based on various security attacks and their impacts. In addition, the authors discuss the best practices and research gaps in the research community of intrusion detection systems.

In [18], the authors put their efforts on a DODAG Information Solicitation (DIS) attack, investigate the attack characteristics, and then design a defense mechanism in IoT network running with RPL. In the DIS attack, the adversary multicasts DIS messages to frequently reset the timer of DODAG Information Object (DIO) messages, resulting in control messages congestion in the network. To identify the DIS attack, the researchers propose a countermeasure which can reduce the response rate of DIO messages to DIS messages. The experimental study has proven that the communication overhead as well as energy consumption can be reduced. The authors in [19] propose a deep learning based detection mechanism to detect/mitigate hello flooding attack in the IoT setting. The primary goal of hello flooding attack is to consume the limited resources of IoT devices. The authors conduct experiments with the comparison of existing benchmark schemes such as SVM and logistic regression. The authors in [20] investigate sybil attack and propose a defense mechanism. The basic idea is that all nodes are organized into a tree structure and each non-leaf node stores a detection table in the memory. When receiving a packet, the node examines the piggybacked source node ID and previous hop node ID with the entries in the detection table. If there is no matched entry, the piggybacked source node ID and previous hop node ID are added in the detection table. If there is a matching entry in the detection table, the receiving node broadcasts an alarm packet to announce the suspected sybil attack. After careful

analysis, it is found that the proposed scheme has a huge computation and storage overhead. E.g., if there are a huge number of entries in the detection table, the sybil attack detection overhead will significantly increase. In [21], the authors identify a new RPL specific attack, named non-spoofed copycat attack, and investigate its performance impact against IPv6 based wireless personal area networks. Through exploiting non-spoofed copycat attack, attackers implicitly monitor the DIO messages of neighbors, and then replay the captured DIO messages several times. The primary purpose of non-spoofed copycat attack is to affect RPL's performance (i.e., communication latency and packet delivery ratio) in IPv6 based wireless personal area networks.

In [22], the authors adopt the trust technique to detect sybil attack, where the behaviors of devices are evaluated based on the current and old trust values. If a sybil attack is detected, the adversary will be assigned with a lower trust value. As a result, the nodes with a lower trust value (possibly the adversary) cannot participate the regular routing operations. In [23], an intrusion detection system, named SVELTE, is proposed to detect network and routing layer attacks. In SVELTE, the intrusion detection scheme and firewall collaborate to examine network traffic and identify suspicious activities. In [24], the authors propose a camouflage-based approach to detect packet dropping attack in wireless networks with energy harvesting capability. In the proposed approach, each node intentionally disguises to be energy harvesting node (i.e., cannot perform implicit monitoring), and then stealthily monitors neighbor node's forwarding operation. If the adjacent neighbor chooses not to re-send the received packet, packet dropping attacks can be detected. In [25], the authors investigate a rank attack in the IoT network, where the adversary targets the rank value in RPL routing protocol and compromises the rank rule to affect the network performance. In [26], the authors propose a authentication scheme using secure hash functions to protect the communication from adversaries in the RPL-based IoT. The authors in [27] present an

analysis of security threat for RPL. In their work, the potential security challenges and basic defense mechanisms are presented and analyzed. In [28], the authors focus on IEEE 802.15.4 medium access control protocol and its usage and limitation in the IoT environment. A survey about denial-of-service attacks against IoT is provided in [29,30]. The authors in [31] review the history and development of RPL routing protocol and point out several directions for future research.

## 9.4 Background

### 9.4.1 RPL Routing Protocol

The Industrial Internet of Things (IIoT) helps once ‘‘dump’’ devices get more intelligent by empowering them to collect, process and send data traffic over the Internet, as well as communicate with other IIoT devices and information and communication systems seamlessly [32]. The acronym IIoT has a promise of improving the efficiency of regular operations with assistance of artificial intelligence, machine learning, and advanced wired/wireless communication technologies. But, the IIoT encompasses a broad range of industrial-grade applications, and its application range is very wide. Far from being restricted to just the concept model, the IIoT can be found in a variety of domains, from ABB Smart Robotics, to Airbus Future Factory, to Amazon Smart Warehousing [33]. The IIoT devices deployed for these applications typically operate under capacity constraints in terms of processing and storage capability, and battery energy [34]. The interconnection links between IIoT devices are featured with relatively low data rate and high packet loss ratio. In addition, the IIoT network can scale the number of devices from a few dozen to thousands, the communication traffic can be classified into one-to-one, one-to-many, and many-to-one modes. To achieve efficient and reliable communications in an IIoT environment, the routing protocol for networks with resource constraints and lossy communication links [7], widely

known as RPL, was proposed by Cisco together with IETF in 2012.

RPL-based IIoT example is shown in Fig. 1, where there are three DODAGs and two RPL instances. Specifically, RPL routing protocol organizes IIoT devices (later nodes) into a hierarchical tree structure which is termed the Destination-Oriented Directed Acyclic Graph (DODAG). Typically, a DODAG contains a gateway node and a set of regular nodes. Here,

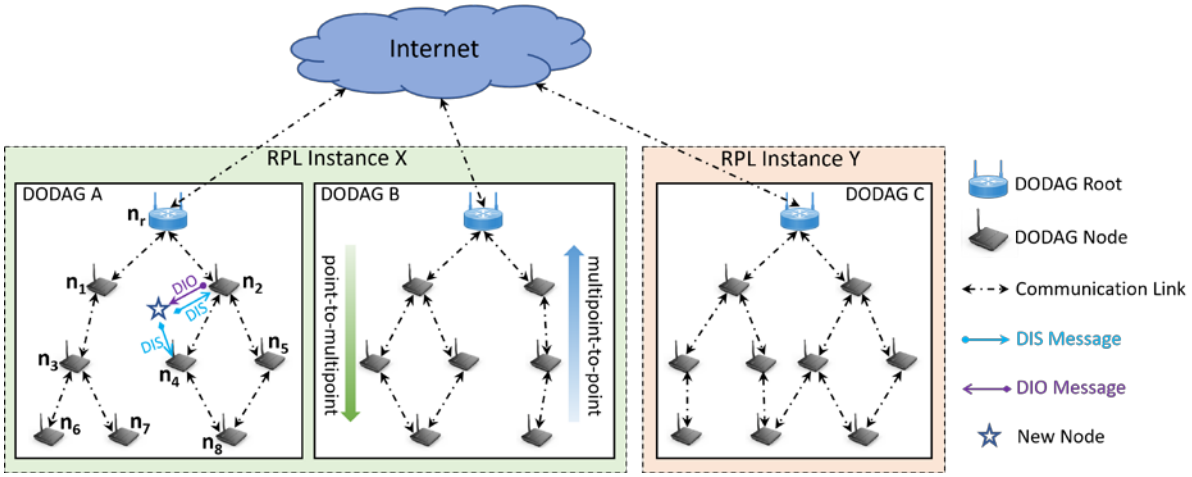


Figure 1: IIoT running with RPL, three DODAGs forming two instances of RPL [57].

the gateway node is termed the DODAG root, connecting with the Internet so that IIoT nodes can communicate across several networks. For large-scale IIoT networks (e.g., thousands of nodes are deployed in the industrial environment), nodes can be self-organized into numerous DODAGs, and DODAGs can be further grouped into different RPL instances. For multiple DODAGs from the identical RPL instance, they are given the same RPL instance ID. The rationale of creating and maintaining multiple RPL instances is to arrange different tasks in the IIoT network, where one RPL instance is responsible for one task and operates independently from other RPL instances. For example, two RPL instances can be established in the IIoT network, where one is liable for collecting and transferring temperature data, and the other is accountable for monitoring the movements of people. Every node has a rank

value. The more closer (or far) the node is located from the root of DODAG, the more smaller (or larger) the value of rank will be. In addition, the value of rank is being utilized to avoid DODAG loops as well as allow nodes to differentiate parent, child, and sibling nodes in the DODAG.

Speaking of traffic flows in RPL routing protocol, three different communication paradigms are supported: (i) one-to-one communication; (ii) one-to-many communication; and (iii) many-to-one communication. The one-to-one communication is adopted by any arbitrary pair of nodes in the DODAG to communicate. When the DODAG root has a command to be issued to other regular nodes in the DODAG, it employs the one-to-many communication mode. If the normal nodes have data for the DODAG root, they can use many-to-one communication mode.

In order to realize all the abovementioned functionalities, four control messages are defined in RPL routing protocol: DIO, DAO, DAO-ACK, and DIS. DIO message is carrying network-relevant information. Usually, the root node of DODAG initiates DIO message to form a new DODAG, establish downward routing paths, and assist new nodes to discover nearby DODAG to join. DAO message is transmitted by the leaf nodes of DODAG so that the upward routing information can be propagated from the leaf nodes to the root node of DODAG. DAO-ACK message is utilized to confirm that DAO messages have been successfully received. New nodes will adopt DIS messages to join the existing DODAG in the network.

To maintain DODAG downward routes, RPL routing protocol can be configured to operate with either of two modes: (i) caching mode; and (ii) non-caching mode. The basic idea of caching mode is that the route information to downwards nodes are stored by each node in the memory. For example, if a DAO message is received by a node, it first caches the piggybacked route, and then adds its node identifier in the aggregated route and passes on the

DAO message to the upward node. When the node receives a packet and the destination is one of its descendant nodes, it forwards the packet to the destination node via the cached

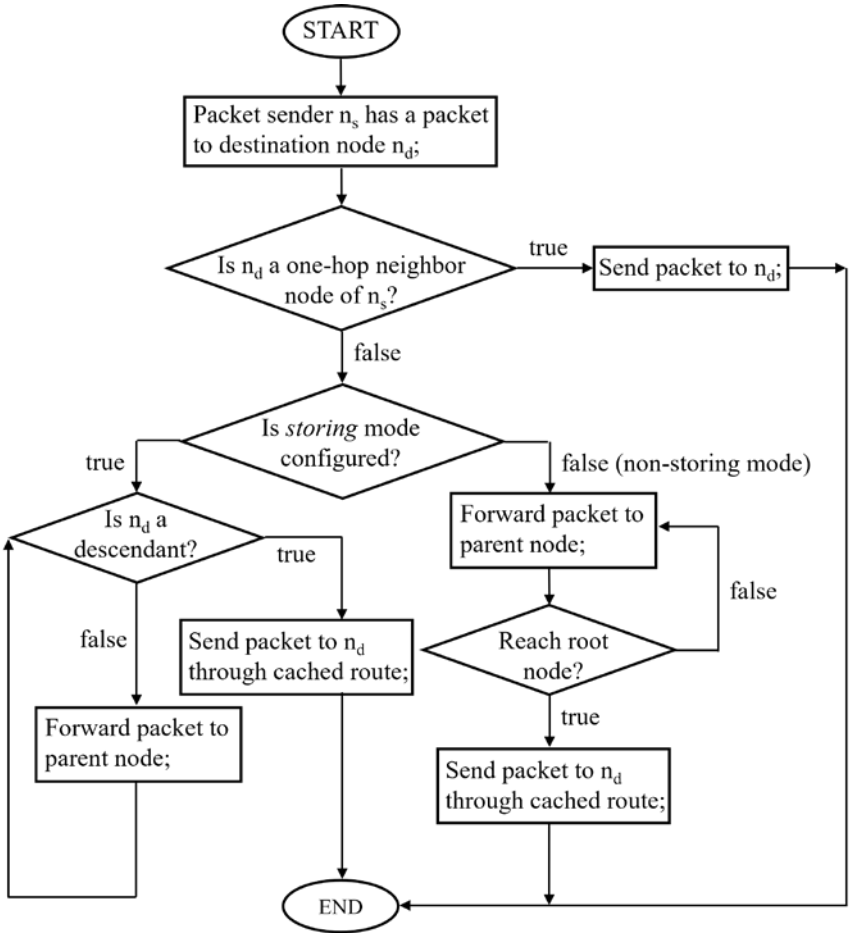


Figure 2: A flowchart of one-to-one communication with caching model and non-caching mode [57].

downward route. If its descendant node is not the destination of packet, the packet will be sent to the parent node. However, in the non-caching mode, the root node of DODAG is the only node who stores route information about downward nodes. Thus, when there is a packet to send, the packet has to be sent via the upward route to the root node of DODAG. And then, the packet will be forwarded to the destination node via the cached downward route by the root node of DODAG. A flowchart of one-to-one communication with caching mode and

non-caching mode is shown in Fig. 2.

### 9.4.2 Trickle Algorithm

In an unstable and hash environment, it is important to regulate the information exchange robustly and energy efficiently. The Trickle communication algorithm [35] is designed to achieve the goal of information consistency in the network. The logic of Trickle communication algorithm is that a node is able to dynamically adjust its packet transmission ratio based on the degree of information consistency. RPL's Trickle communication algorithm is adopted to control the transmission rate of DIO messages. Since DIO packets usually carry critical network information, as a result, the transmission of DIO packets deserves special attention. To be specific, if an inconsistency is detected by a node, the node will increase the transmission rate of its DIO packets. On the other side, if there are no inconsistent information detected, the node will decrease its DIO packet transmission rate. According to [35], six major system parameters are adopted to achieve the goals of Trickle communication algorithm.

- $I_{min}$: the lower bound of timer.
- $I_{max}$: the upper bound of timer.
- $k$: the redundant parameter.
- $I$: the length of current timer.
- $t$: a time within the current timer.
- $c$: a counting parameter.

The pseudocode of controlling DIO packet transmissions is shown in Algorithm 1 [57].

**Algorithm 1:** Trickle Algorithm in RPL Routing Protocol

**Input:** $I_{min}$, $I_{max}$, $k$
**Output:** $I$, $t$, $c$

```
1  Function Init(I_min):
       /* sets timer I to the first interval                    */
2      I ← I_min;
3      return I;
4  Function NewIntvl():
       /* doubles the interval length                           */
5      I ← I × 2;
       /* sets a counter variable c to 0                        */
6      c ← 0;
       /* sets the interval length to I_max                     */
7      if I_max ≤ I then
8          I ← I_max;
9      end
       /* sets t to a random point in interval                  */
10     t ← rand[I/2, I];
11     return t;
12 Function RecConsTrans():
       /* increases counter variable c                          */
13     c ← c + 1;
14     return c;
15 Function RecConsTrans():
16     if I_min < I then
           /* resets timer I                                    */
17         I ← I_min;
18     end
19     return I;
20 Function TimerExp():
21     if c < k then
           /* c less than redundancy constant k                 */
22         Transmit scheduled DIO;
23     else
24         Suppress scheduled DIO;
25     end
```

### 9.4.3 System and Adversary Models

We assume that the IIoT is composed of a set of DODAGs, where each node has constrained computing and storage capability, and battery energy. In the DODAG, the DODAG root and regular nodes communicate directly and indirectly via unreliable links. In addition, each node is pre-assigned a unique ID [36]. Since the IIoT nodes are usually deployed in a wide-open or unattended area, they can be easily captured and compromised by the adversary [37]. Through probing attacks, the adversary might be able to access the security-critical module of integrated circuit, retrieve sensitive data, and then reprogram it to turn it into the malicious node [38].

The IIoT nodes usually need to keep operating for a period of time in the areas of interest [39]. If the IIoT nodes are equipped with regular batteries and dedicated in the operations of monitoring and communication, their battery power only can last for 5.8 days [40]. Consequently, replacing the battery of IIoT nodes becomes inevitable so that the IIoT network can survive and continue to operate. However, the IIoT nodes are usually deployed in difficult-to-reach locations, thus replacing or refilling batteries becomes very challenging or even impossible. Thus, people usually choose to use drones to deploy new IIoT nodes in the interest of area to maintain the operation of network [41].

## 9.5 RPL Specific Attacks and Countermeasures

RPL is vulnerable to various cyber attacks due to the nature of wireless medium, the resource constraints of IIoT nodes, and the optional implementation of security mechanisms [42]. In addition, the security and privacy issues were not the first concerns when RPL was designed, thus it cannot satisfy the security requirements of current critical systems. It has been discovered that none of existing IIoT operating systems [43] [44], include the implementation of RPL‘s security mechanisms. There are several existing security

mechanisms to protect RPL-based IIoT from security attacks, such as intrusion detection systems [17] and hash chain authentication technique [45]. These security mechanisms can effectively detect outside attackers, however, they cannot defend against any inside attackers. An adversary might control a subset of IIoT nodes in the network, and also has access to the cryptographic keys so that the cryptographic protection mechanisms can be easily bypassed. In the following, several representative RPL specific attacks are presented and their corresponding countermeasure is briefly discussed.

### 9.5.1 Sybil Attack

For a new node, it can transmit a DIS message to request DODAG-related information from adjacent nodes to join the network. After receiving the DIS message, the neighbor node prepares and responds with a DIO message carrying RPL instance ID, version number, rank value, and other network configuration parameters. With those network-relevant information, the new node can choose a parent node, calculate the value of rank, and join the network. Unfortunately, the attackers can exploit the vulnerability of DIS messages to perform sybil attack against the IIoT network [46]. Specifically, the attackers first create many malicious DIS packets with fake node IDs, and then broadcast those packets. Here, the fake node identifiers could be media access control addresses that are randomly generated by the adversary. If a normal node receives the malicious packet, it thinks that fresh nodes are eager to become members of network. Based on the Trickle communication scheme, the normal node needs to restart its DIO Trickle timeout period, and then broadcast DIO packets. However, broadcasting DIO packets will require the normal node to consume energy resources [47].

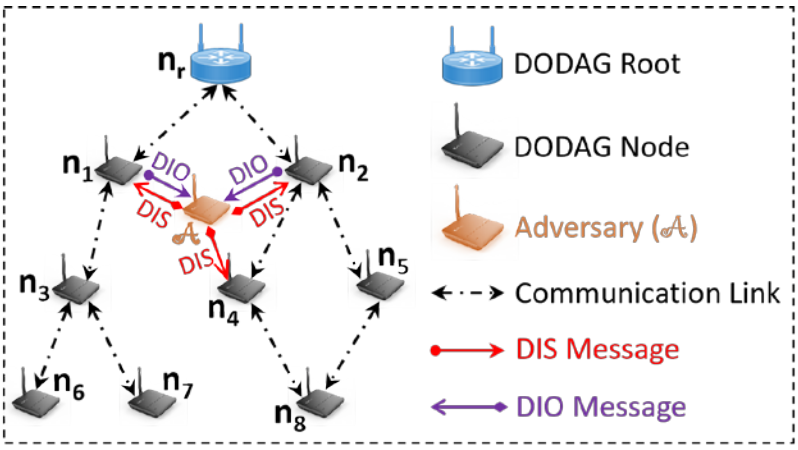


Figure 3: Sybil attack.

For example in Fig. 3, an adversary A is launching sybil attack against the legitimate node $n_1$, $n_2$, and $n_4$ by broadcasting an attack DIS message with the fake identifier. It is clearly shown that $n_1$ and $n_2$ are important bridge nodes between the root node of DODAG and other nodes in the DODAG. If the bridge nodes are not available (i.e., consuming all energy resource), the network will be divided into several parts. When $n_1$ and $n_2$ receive the DIS message, they assume that a non-member node might want to join the DODAG and is soliciting DODAG-related information from neighbor nodes. According to RPL specification, both $n_1$ and $n_2$ first restart their DIO Trickle timeout period to $I_{min}$, and then broadcast a DIO message when the timer expires. Here, since $n_1$ and $n_2$ are not adjacent nodes, the DIO message from one node will not be able to prevent the DIO message from the other node. When the attacker A transmits a large number of malicious DIS messages with fictitious ID, $n_1$ and $n_2$ have to respond by transmitting the corresponding amount of DIO messages. Due to the frequent receiving and sending messages, $n_1$ and $n_2$ will quickly consume their limited power of batteries. As a result, the lifetime of nodes will be shortened. When the battery energy is completely exhausted, the network partition will be formed in the DODAG, where other nodes (e.g., $n_3$) will not be able to communicate with the DODAG root $n_r$ anymore. To

demonstrate the severe consequences of sybil attack in the DODAG as shown in

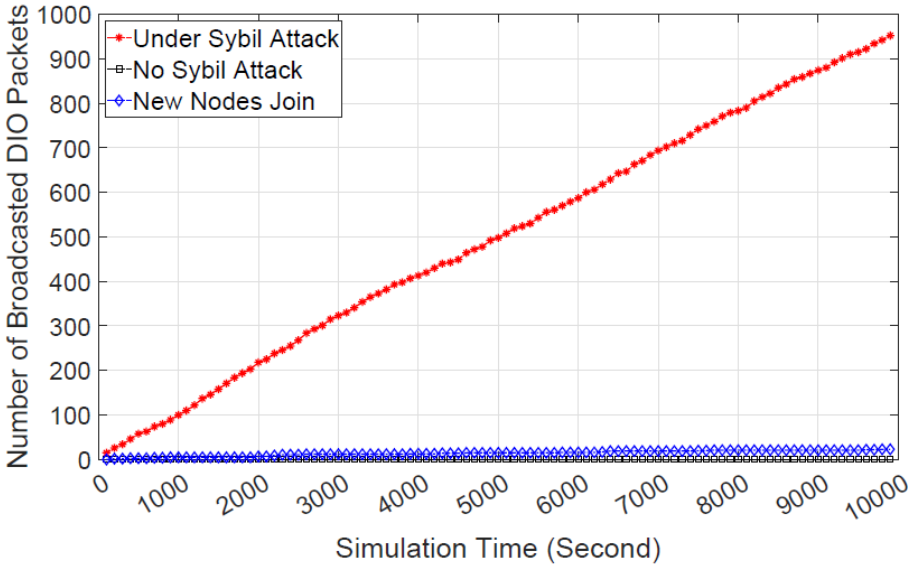


Figure 4: The impact of sybil attack [57].

Fig. 3, a preliminary experiment is conducted in OMNeT++ [48]. According to [35], we set $I_{min}$ = 0.1 sec, $I_{max}$ = 6,554 sec, and $k$ = 1 in the experiment. In Fig. 4, we obtain the amount of DIO packets against the experimental time. According to Fig. 4, it is clearly shown that the amount of transmitted DIO messages increase when the length of experimental time increases. On the other side, for the scenario without adversary, the number of transmitted DIO messages is maintained at the relatively low level.

In [49, 50], a Gini coefficient technique is being adopted to detect sybil attack by measuring the statistical dispersion. To be specific, during the observation window, every network node keeps track of the statistical dispersion of node IDs in the DIS messages, and then computes the Gini coefficient. After comparing the Gini coefficient with a pre-determined boundary value, the legitimate node can detect the existence of sybil attacks. However, the proposed Gini coefficient detection scheme has one potential drawback: the detection latency and accuracy are closely related to the length of observation window. If the

observation window is short, we can expect a low detection latency as well as a low detection accuracy. However, if the observation window is long, the detection latency will be increased, and we will have a high detection accuracy.

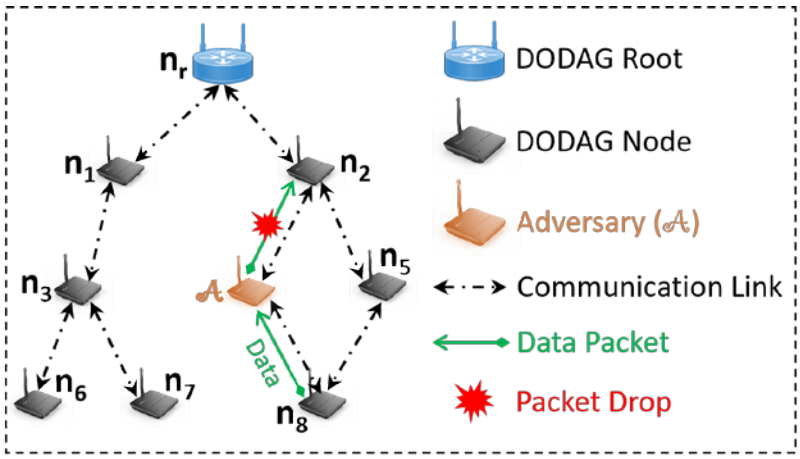


Figure 5: Packet dropping attack [57].

### 9.5.2 Packet Dropping Attack

In RPL routing protocol, the DODAG root needs to issue a DIO packet to form a DODAG after the network nodes are deployed in an area of interest. The DIO message is typically piggybacked with the ID of root node of DODAG, the DODAG root node's rank value, as well as an objective function. Here, the objective function specifies how each node calculates the rank value based on pre-defined metrics. After receiving the message of DIO, if the network node plans to join the existing DODAG, the sender ID of DIO message can be added to the list of parent nodes. In addition, it computes the value of rank for itself, and then passes on the DIO packet piggybacked with its own identifier and rank value to other nodes. If there are multiple nodes in a node‘s parent list, the member of parent list having the smallest value of rank becomes the preferred parent node. When the network node has data traffic to the root node of DODAG, the preferred parent node is automatically selected as the next-hop node to

send data traffic. However, an adversary can intentionally put a smaller value of rank in the DIO packet and transmit the packet to other network nodes, which make other nodes select the adversary as their preferred parent node. So,

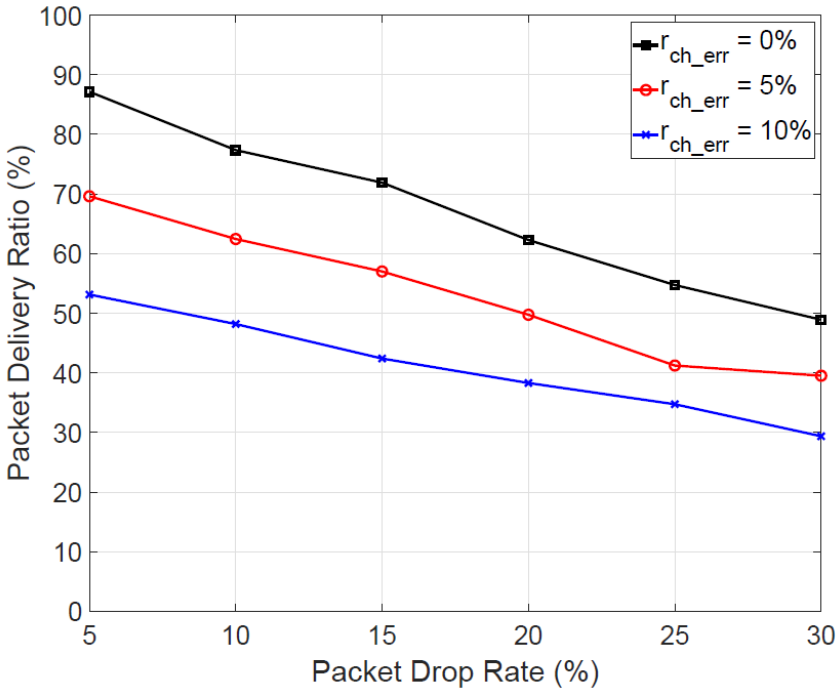


Figure 6: The impact of packet dropping attack [51].

when the adversary receives data traffic from other nodes, it can drop any data packets intentionally to deafen the root node of DODAG.

For example in Fig. 5, an adversary A is launching packet dropping attack in the DODAG. First, the adversary A broadcasts a DIO message with a smaller value of rank to make its child node (e.g., $n_8$) to choose it as the preferred parent node. Then, when $n_8$ sends a data packet to the adversary A, the adversary A randomly or strategically drops the data traffic without forwarding them to the next-relay node (e.g., $n_2$). As a result, the DODAG root $n_r$ cannot receive the data packet from $n_8$, and the network performance is significantly affected. We conduct a preliminary experiment to expose the severe consequences of packet dropping attack in Fig. 6, where the packet delivery ratio (PDR) is obtained with changing packet drop

rate and channel error rate ($r_{ch\_err}$). As shown in Fig. 6, when the packet drop rate increases, the PDR significantly decreases. Since more data packets are being dropped by the adversary A, less number of data packets can be transmitted to the root node of DODAG and a smaller PDR is obtained. In addition, the PDR is significantly affected by the channel error rate $r_{ch\_err}$. The overall PDR decreases as the $r_{ch\_err}$ increases. The reason is that data packets might get missing in the process of forwarding due to the bad wireless channel quality. Consequently, a lower PDR is measured.

In [51], a monitor based mechanism is designed to detect/mitigate malicious packet dropping behaviors and isolate the adversary from the RPL-based IIoT. After sending the data traffic to its preferred parent node, the packet sender continues to monitor the follow-up operation of packet receiver. If the packet receiver drops the packet without forwarding, the packet sender can compare the ratio of packet loss of preferred parent node with the average packet loss rate of adjacent nodes. If the ratio of packet loss of preferred parent node is lower or smaller than the average packet loss rate of adjacent nodes, the packet dropping activities of adversary can be detected. When the total number of caught packet dropping activities is larger than a pre-defined boundary value, the detection node will issue an Alarm packet so that all neighbor nodes will not send any packet to the adversary any more. The idea of monitor based mechanism is straightforward and easy to implement, however, its disadvantage is obvious, too. The pre-defined threshold value should be carefully selected. If a larger threshold value is adopted, a higher miss detection rate might be observed. On the other side, if the threshold value is too small, a false positive rate will be high.

### 9.5.3 DAO Divergence Attack

In RPL routing protocol, the caching mode enables network nodes to actively obtain the downward routing paths to its descendant nodes through buffering the route information

piggybacked in the DAO messages. Through caching the downward route information, every network node has the information about the next-relay node via which the data traffic can be delivered to the destination. However, the downward route in the buffer might become stale (e.g., one intermediate node along the route is not available), and the data traffic cannot be delivered to the destination if the stale route is selected to send data traffic. Thus, to get rid of stale routes from the buffer, the flag of Forwarding-Error in the header of packet is utilized to quickly identify the unreachable next-hop node and report the error route in the network. To be specific, when a node is unable to send the network traffic to the next-relay node according to the cached routing information, it sets the Forwarding-Error flag, creates an error message, and then sends the error message to the

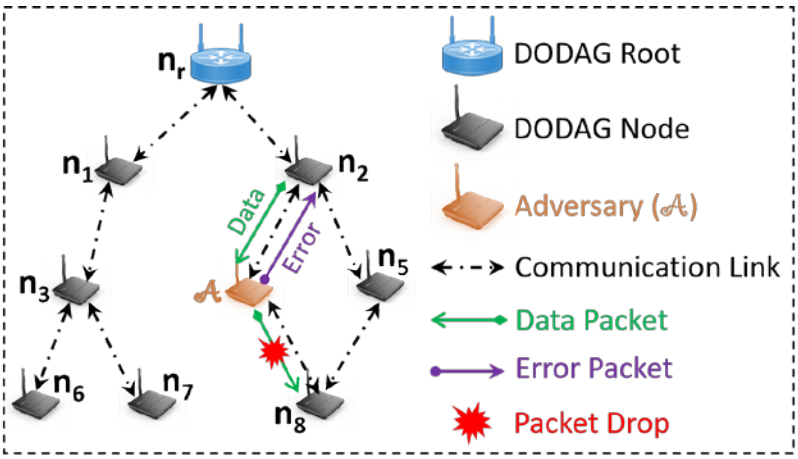


Figure 7: DAO inconsistency attack [57].

parent node. When the error message with the Forwarding-Error flag set reaches the parent node, the parent node has to remove the reported downward route from its buffer space according to RPL specification. Originally, the Forwarding-Error flag is designed to report and discard the error route in the network, and finally improve the performance of RPL routing protocol. However, this feature might be exploited by attackers to launch DAO

divergence attack [52] against the IIoT systems.

In Fig. 7, an adversary A receives a data packet from node $n_2$. Instead of forwarding the data traffic to the next-relay node $n_8$, the adversary A chooses to drop the data packet. After that, the adversary A creates an error packet with the flag of Forwarding-Error set, and responds $n_2$ with the error packet. When $n_2$ receives the error packet, it believes that the cached downward route to $n_8$ is no longer available, thus, it removes the cached downward route from the buffer. To show the severe consequences of DAO divergence attack in the IIoT network, we conduct a preliminary experiment. A downward route which consists

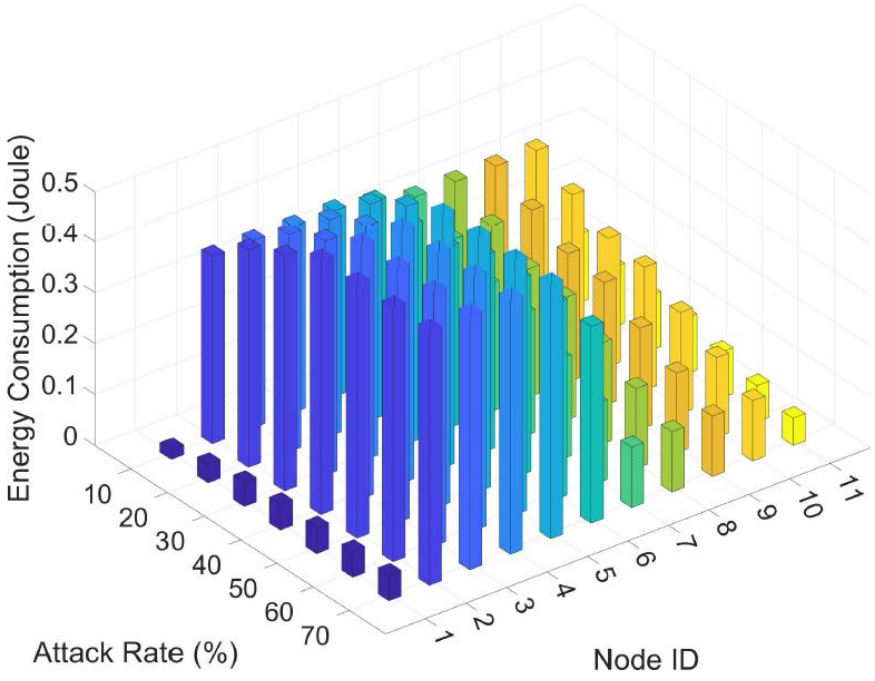


Figure 8: The consequences of DAO divergence attack [52].

of 11 nodes is set up in the network, where node $n_1$ and $n_6$ are the DODAG root and the adversary A, respectively. Here, the attack rate indicates how frequently the adversary $n_6$ replies the error packets with the flag of Forwarding-Error set. Each node's energy usage along the downward route is measured in Fig. 8, where the energy consumption of each node (i.e., $n_1$, $n_2$, $n_3$, $n_4$, and $n_5$) located prior to the adversary $n_6$ goes up when attackers perform

more attacks. As the adversary $n_6$ generates and replies more error packets, its parent node has to discard valid downward routing information cached within the buffer. If its parent or ancestral node has network traffic for the same destination later, the parent or ancestral node needs to generate and reply an error packet to their corresponding parent node. This is because the downward routing information which can be used to reach the destination nodes has already been discarded. Consequently, the adversary A's ancestors along the downward route need to respond with error packets, which makes the energy consumption of intermediate nodes increase. To defend against DAO inconsistency attack, the parent node can set up a dynamic

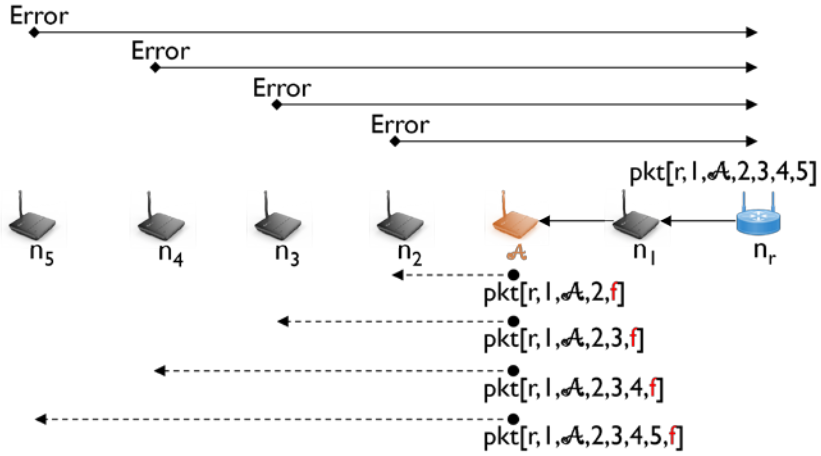


Figure 9: Hatchetman attack [53].

error packet acceptance rate which is adaptively fine-tuned according to the frequency of receiving error packets and the estimated channel error rate. With the error packet acceptance rate, each node can prevent the valid downward routes from being discarded.

### 9.5.4 Hatchetman Attack

In hatchetman attack, when the adversary A receives the data packets, it first creates a

huge amount of invalid packets by adding the error route information, and then sends these attack packets to normal nodes, which can make normal nodes discard the received attack packets and answer with a lot of error messages to the root node of DODAG [53]. In consequence, the legitimate nodes not only drop many data messages, but also reply many error messages, which waste the limited energy and communication resources. For example in Fig. 9, the DODAG root $n_r$ generates a packet piggybacked with the route ([r, 1, A, 2, 3, 4, 5]) and sends it to node $n_5$. When the data packet reaches the attacker A, the attacker A first replaces the post-hop nodes (i.e., 3, 4, and 5) of the target node (i.e., $n_2$) with a fake ID of destination node (i.e., $n_f$). After that, the attacking packet carrying the error routing information ([r, 1, A, 2, f]) is forwarded to the node $n_2$. In the error route, $n_f$ is the fictitious destination node address which is unreachable. When $n_2$ receives

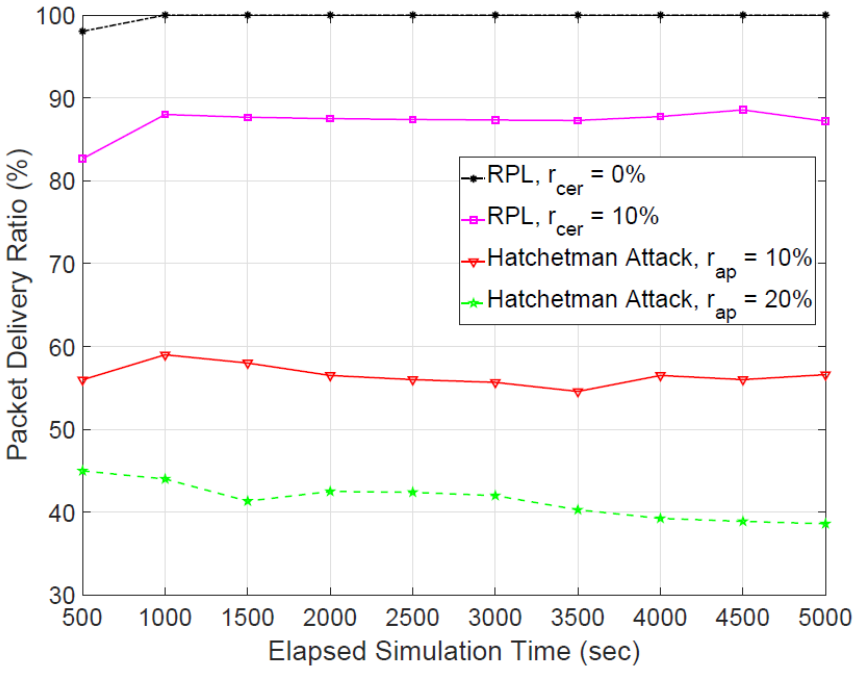


Figure 10: The impact of hatchetman attack [53].

the packet with the unreachable route, it attempts to send the packet to the node $n_f$. However,

$n_f$ does not exist in the network and the forwarding operation fails. After that, $n_2$ needs to drop the received packet and issue an error message to the root node of DODAG $n_r$. If the adversary A transmits the false packet piggybacked with the unreachable routing information to each downstream node, all received false packets will be dropped and an error packet will be replied by each downstream node. It is shown in Fig. 9 that the attacker A transmits the attack packets with the unreachable route to its downstream node, e.g., $n_2$, $n_3$, $n_4$, and $n_5$. After receiving the packet with the error route, all downstream nodes need to discard the packet. The reason is that the next-relay node is not reachable. Moreover, the downstream nodes have to generate and send error packets to the DODAG root $n_r$. In such case, every node between the attacker and the destination node has to receive and transmit many error packets, which consumes non-negligible amount of energy and communication resources in the network.

In Fig. 10, the ratio of packet delivery (PDR) is obtained with changing channel error rate ($r_{cer}$) and the percentage of attackers ($r_{ap}$). When $r_{cer}$ is 0%, the highest PDR is obtained. Since each node collaboratively forwards the received packets, the destination node will receive more packets and the highest PDR is shown. Without hatchetman attack, RPL's performance is vulnerable to the quality of wireless medium, where the PDR is oscillating around 76% with $r_{cer}$ = 10%. With $r_{ap}$ = 10% and 20%, the smallest PDR is observed by the hatchetman attack, comparing to the scenario running traditional RPL routing protocol without hatchetman attack. Since the attacker creates a large amount of attacking packets with the unreachable destination node and transmit them to the normal nodes, the normal nodes will drop more invalid data packets and a lower PDR is obtained. When the number of adversaries performing hatchetman attack increases, $r_{ap}$ = 20%, the PDR drops below 45%. This is because more attackers can create more attack packets with the unreachable route. So,

when the normal nodes receive those attack packets, they have to drop them.

### 9.5.5 Energy Abusing Attack

For RPL, the one-to-one communication is designed for arbitrary pair of DODAG nodes to communicate [54]. If the packet sender has a packet to its adjacent neighbor node, it just forwards the packet to the adjacent node directly, rather than sending the packet to its preferred parent node. In all other cases, the configuration of RPL, either caching mode or non-caching mode, will determine how the one-to-one communication is executed. To be specific, if the non-storing mode is configured in RPL routing protocol, the nodes, except for the DODAG root, do not cache any downward route towards descendant nodes. If there are some packets to send, the network node first has to transmit the packet to the root node of DODAG via the upward routing path. After receiving the packet, the DODAG root attaches the source routing information and transmits the packet to the destination node. However, if the RPL routing protocol is configured with storing mode, then each node will store the

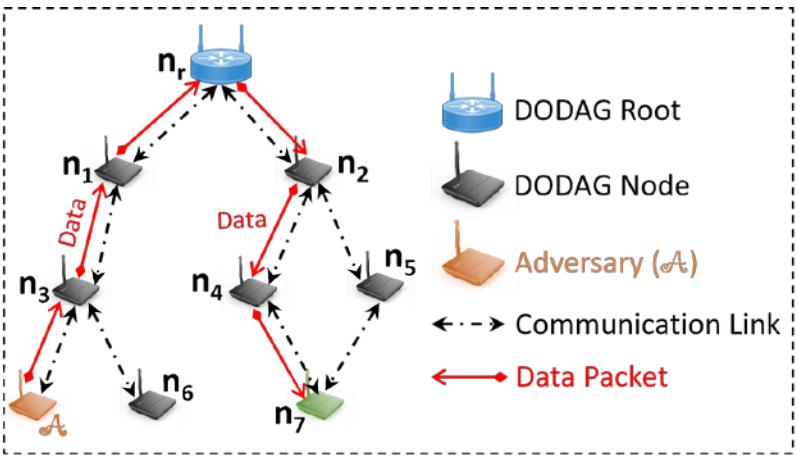


Figure 11: Energy abusing attack [55].

downward route towards descendant nodes. If there is a packet to its descendant, the packet can be forwarded via the cached downward route and finally reaches the descendant node.

Otherwise, the packet is supposed to be sent to the root node of DODAG via sender's preferred parent node. After the packet reaches the root node of DODAG, the root node of DODAG will then send the packet to the destination node.

In the RPL-based IIoT, the one-to-one communication is often used for sending data traffic as well as end-to-end acknowledgments between arbitrary pair of nodes in the DODAG. Unfortunately, the one-to-one communication could be exploited by the adversary in the malicious manner to affect the performance of network. In Fig. 11, the adversary A transmits many data packets to node $n_7$. If RPL routing protocol is configured with non-caching mode, the data packets have to be first transmitted to the root node of DODAG $n_r$ via the upward routing path. Then, the root node of DODAG $n_r$ attaches the source routing information in the data packets and sends them to $n_7$. If the caching mode is being configured in RPL routing protocol, the data packets need to be forwarded to the first common ancestor of adversary A and node $n_7$, which is the root node of DODAG $n_r$. After that, the DODAG root $n_r$ forwards all data packets to $n_7$. According to the above analysis, it does not matter which mode RPL routing

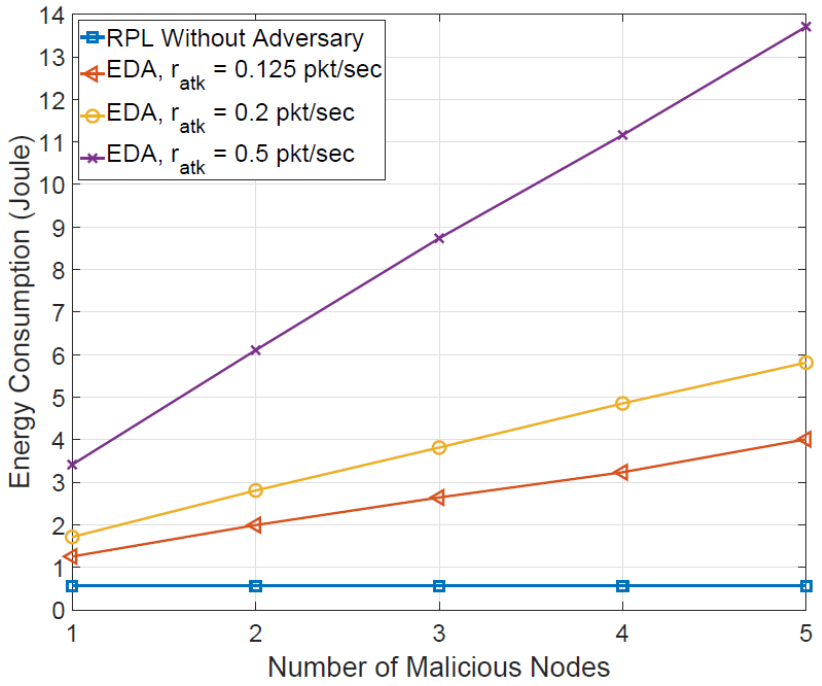


Figure 12: The impact of energy abusing attack [55].

protocol is currently using, a sequence of intermediate nodes, such as $n_3$, $n_1$, $n_r$, $n_2$, and $n_4$, need to send a large number of data packets. However, those many sending operations will cause the energy consumption increase. Since each IIoT node has very limited battery power, the energy depletion attack can quickly exhaust each node's battery power, and finally causes the network unable to work.

In Fig. 12, the energy usage is obtained with varying number of attackers and the attack rate ($r_{atk}$). Here, the attack rate $r_{atk}$ indicates the frequency of sending attack data packets by the adversary. It is shown in Fig. 12 that RPL routing protocol achieves the lowest energy consumption if there is no adversary in the network. As there are more attackers existing in the network, an increasing energy consumption can be observed under energy abusing attack (EDA). Since more attack data packets are being generated by the attackers, each intermediate node will need to transmit more attack packets. As a result, the energy consumption will increase. In addition, the energy consumption increases when attackers

perform more attacks. With a larger attack rate $r_{atk}$, attackers will send more attack data packets. As a result, more energy resource has to be consumed by intermediate nodes due to frequent receiving and forwarding operations.

In [55], the authors propose a detection mechanism to detect/mitigate the energy abusing attack based on the nodes' behaviour in RPL-based IIoT. The logic of the detection mechanism is that the number of sent packets during a pre-defined time period is being recorded by the adjacent node. Then, the counting number will be compared with a threshold value. When the number of sent packets is larger than the boundary value, the energy abusing attack can be detected.

## 9.6 Conclusion and Future Research Directions

The IIoT consisting of seamlessly interconnected smart devices has been seen in various industrial domains such as automated monitoring of inventory, quality control, supply chain optimization, plant safety improvement, etc. To improve the communication efficiency and revolutionize traditional industrial processes, a routing protocol named RPL has been proposed for the IIoT. Because the deployment of security mechanisms is missing in RPL routing protocol, however, the IIoT is vulnerable to several well-known attacks inherited from wireless network and RPL specific attacks. This chapter introduces several representative RPL-specific cyber attacks, such as sybil attack, packet dropping attack, DAO divergence attack, hatchetman attack, and energy abusing attack, and discuss the potential countermeasure against these attacks in the IIoT.

To further explore the potential of IIoT, we recommend two promising research domains with the trans-disciplinary opinions for future investigation. First, since IIoT devices are usually powered by traditional battery, as a result, battery replacement or energy replenishment is unavoidable or even impossible [40]. Thus, energy harvesting becomes an

ideal solution to extend the lifetime of IIoT devices [24]. Second, the traditional cryptographic schemes can provide fundamental protections, i.e., confidentiality, integrity and availability. Unfortunately, those cryptographic schemes cannot be directly applied to IIoT devices that operate with resource-limited microprocessors [56]. Thus, the lightweight security protocols become the only solution to secure IIoT networks.

Finally, we believe that this chapter has potential to help researchers discover novel research directions to pursue and contribute to the IIoT community through providing the detailed discussion of cyber attacks, countermeasures and future research directions.

**References**

[1] K. Tange, M. D. Donno, X. Fafoutis, and N. Dragoni, “A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities,” IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2489–2520, 2020.

[2] C. Pu, “A Novel Blockchain-Based Trust Management Scheme for Vehicular Networks,” in IEEE Proc. WTS, 2021, pp. 1–6.

[3] M. Bansal, A. Goyal, and A. Choudhary, “Industrial Internet of Things (IIoT): A Vivid Perspective,” Inventive Systems and Control, pp. 939–949, 2021.

[4] Industrial IoT (IIoT) Market Worth $263.4 billion by 2027, 2020, https://www.globenewswire.com/.

[5] The Opportunities of the Industrial Internet of Things, https://www.roevin.ca/blog/2019/august/opportunities-industrial-internet-of-things/.

[6] Securing the Internet of Things: A Proposed Framework, https://tools.cisco.com/security/center.

[7] T. Winter et al., “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” rfc, vol. 6550, pp. 1–157, 2012.

[8] F. Al-Turjman, “Cognitive routing protocol for disaster-inspired internet of things,” Future Generation Computer Systems, vol. 92, pp. 1103–1115, 2019.

[9] T. Behera, S. Mohapatra, U. Samal, M. Khan, M. Daneshmand, and A. Gandomi, “I-SEP: An Improved Routing Protocol for Heterogeneous WSN for IoT-Based Environmental Monitoring,” IEEE Internet of Things Journal, vol. 7, no. 1, pp. 710–717, 2019.

[10] R. Coutinho, A. Boukerche, and A. Loureiro, “A novel opportunistic power controlled routing protocol for internet of underwater things,” Computer Communications, vol. 150, pp. 72–82, 2020.

[11] B. Djamaa, M. Senouci, H. Bessas, B. Dahmane, and A. Mellouk, “Efficient and Stateless P2P Routing Mechanisms for the Internet of Things,” IEEE Internet of Things Journal, pp. 1–1, 2021.

[12] S. Jazebi and A. Ghaffari, “RISA: routing scheme for Internet of Things using shufed frog leaping optimization algorithm,” Journal of Ambient Intelligence and Humanized Computing, pp. 1–11, 2020.

[13] R. Yarinezhad and S. Azizi, “An energy-efficient routing protocol for the Internet of Things networks based on geographical location and link quality,” Computer Networks, vol. 193, p. 108116, 2021.

[14] A. Verma and V. Ranga, “Security of RPL based 6LoWPAN Networks in the Internet of Things: A Review,” IEEE Sensors Journal, vol. 20, no. 11, pp. 5666–5690, 2020.

[15] A. Agiollo, M. Conti, P. Kaliyar, T. Lin, and L. Pajola, “DETONAR: Detection of Routing Attacks in RPL-Based IoT,” IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1178–1190, 2021.

[16] M. Osman, J. He, F. Mokbal, N. Zhu, and S. Qureshi, “ML-LGBM: A Machine Learning Model based on Light Gradient Boosting Machine for the Detection of Version Number Attacks in RPL-Based Networks,” IEEE Access, vol. 9, pp. 83 654–83 665, 2021.

[17] G. Simoglou, G. Violettas, S. Petridou, and L. Mamatas, “Intrusion detection systems for RPL security: a comparative analysis,” Computers & Security, vol. 104, p. 102219, 2021.

[18] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, “Multicast DIS attack mitigation in RPL-based IoT-LLNs,” Journal of Information Security and Applications, vol. 61, p. 102939, 2021.

[19] S. Cakir, S. Toklu, and N. Yalcin, “RPL Attack Detection and Prevention in the Internet of Things Networks Using a GRU Based Deep Learning,” IEEE Access, vol. 8, pp. 183 678–183 689, 2020.

[20] P. Kaliyar, W. Jaballah, M. Conti, and C. Lal, “LiDL: Localization with early detection of sybil and wormhole attacks in IoT Networks,” Computers & Security, vol. 94, p. 101849, 2020.

[21] A. Verma and V. Ranga, “CoSec-RPL: detection of copycat attacks in RPL based 6LoW-PANs using outlier analysis,” Telecommunication Systems, vol. 75, pp. 43–61, 2020.

[22] D. Airehrour, J. Gutierrez, and S. Ray, “SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things,” Future Generation Computer Systems, vol. 93, pp. 860–876, 2019.

[23] 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.

[24] C. Pu and S. Lim, “Spy vs. Spy: Camouflage-based Active Detection in Energy Harvesting Motivated Networks,” in IEEE Proc. MILCOM, 2015, pp. 903–908.

[25] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, “The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks,” IEEE Sensors J., vol. 11, no. 10, pp. 3685–3692, 2013.

[26] A. Dvir, T. Holczer, and L. Buttyan, “VeRA-Version Number and Rank Authentication in RPL,” in Proc. IEEE MASS, 2011, pp. 709–714.

[27] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, “A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs),” RFC Standard 7416, January 2015.

[28] S. M. Sajjad and M. Yousaf, “Security analysis of IEEE 802.15. 4 MAC in the context of Internet of Things (IoT),” in Proc. IEEE CIACS, 2014, pp. 9–14.

[29] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, “Denial-of-Service detection in 6LoWPAN based Internet of Things,” in Proc. IEEE WiMob, 2013, pp. 600–607.

[30] A. Rghioui, A. Khannous, and M. Bouhorma, “Denial-of-Service attacks on 6LoWPAN-RPL networks: Threats and an intrusion detection system proposition,” Journal of Advanced Computer Science & Technology, vol. 3, no. 2, pp. 143–152, 2014.

[31] H. Kim, J. Ko, D. Culler, and J. Paek, “Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey,” IEEE Commun. Surveys Tuts., Sep 2017.

[32] C. Pu, J. Brown, and L. Carpenter, “A Theil Index-Based Countermeasure Against Advanced Vampire Attack in Internet of Things,” in IEEE Proc. HPSR, 2020, pp. 1–6.

[33] The Top 20 Industrial IoT Applications, https://www.iotworldtoday.com/2017/09/20/top-20-industrial-iot-applications/.

[34] K. Pister, P. Thubert, S. Dwars, and T. Phinney, “Industrial Routing Requirements in Low-Power and Lossy Networks,” rfc, vol. 5673, pp. 1–27, 2009.

[35] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The Trickle Algorithm,” Internet Engineering Task Force, RFC6206, pp. 1–13, 2011.

[36] C. Pu and X. Zhou, “Suppression Attack Against Multicast Protocol in Low Power and Lossy Networks: Analysis and Defenses,” Sensors, vol. 18, no. 10, p. 3236, 2018.

[37] C. Pu, X. Zhou, and S. Lim, “Mitigating Suppression Attack in Multicast Protocol for Low Power and Lossy Networks,” in IEEE Proc. LCN, 2018, pp. 251–254.

[38] C. Pu, S. Lim, B. Jung, and J. Chae, “EYES: Mitigating forwarding misbehavior in

energy harvesting motivated networks," Computer Communications, vol. 124, pp. 17–30, 2018.

[39] C. Pu, S. Lim, B. Jung, and M. Min, "Mitigating Stealthy Collision Attack in Energy Harvesting Motivated Networks," in IEEE Proc. MILCOM, 2017, pp. 539–544.

[40] C. Pu, T. Gade, S. Lim, M. Min, and W. Wang, "Lightweight Forwarding Protocols in Energy Harvesting Wireless Sensor Networks," in Proc. IEEE MILCOM, 2014, pp. 1053 1059.

[41] S. Mnasri, N. Nasri, and T. Val, "The Deployment in the Wireless Sensor Networks: Methodologies, Recent Works and Applications," in Proc. PEMWN, 2014.

[42] C. Pu and L. Carpenter, "Digital Signature Based Countermeasure Against Puppet Attack in the Internet of Things," in IEEE Proc. NCA, 2019, pp. 1–4.

[43] Contiki Operating System, http://www.contiki-os.org/.

[44] TinyOS, http://www.tinyos.net/.

[45] G. Glissa, A. Rachedi, and A. Meddeb, "A Secure Routing Protocol based on RPL for Internet of Things," in IEEE Proc. GLOBECOM, 2016, pp. 1–7.

[46] C. Pu, "Spam DIS Attack Against Routing Protocol in the Internet of Things," in IEEE Proc. ICNC, 2019, pp. 73–77.

[47] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wirless Sensor Networks: Design, Analysis, and Evaluation," IEEE Systems Journal, vol. 12, no. 1, pp. 834–842, 2018.

[48] A. Varga, OMNeT++, 2014, http://www.omnetpp.org/.

[49] B. Groves and C. Pu, "A Gini Index-Based Countermeasure Against Sybil Attack in the Internet of Things," in Proc. IEEE MILCOM, 2019, pp. 1–6.

[50] C. Pu, "Sybil Attack in RPL-Based Internet of Things: Analysis and Defenses," IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4937–4949, 2020.

[51] C. Pu and S. Hajjar, “Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks,” in IEEE Proc. CCNC, 2018, pp. 1–6.

[52] C. Pu, “Mitigating DAO inconsistency attack in RPL-based low power and lossy networks,” in IEEE Proc. CCWC, 2018, pp. 570–574.

[53] C. Pu and T. Song, “Hatchetman Attack: A Denial of Service Attack Against Routing in Low Power and Lossy Networks,” in IEEE Proc. CSCloud, 2018, pp. 12–17.

[54] C. Pu, “Energy Depletion Attack Against Routing Protocol in the Internet of Things,” in IEEE Proc. CCNC, 2019, pp. 1–4.

[55] C. Pu and B. Groves, “Energy Depletion Attack in Low Power and Lossy Networks: Analysis and Defenses,” in IEEE Proc. ICDIS, 2019, pp. 14–21.

[56] C. Pu and Y. Li, “Lightweight Authentication Protocol for Unmanned Aerial Vehicles Using Physical Unclonable Function and Chaotic System,” in IEEE Proc. LANMAN, 2020, pp. 1–6.

[57] C. Pu and K. Choo, “Lightweight Sybil Attack Detection in IoT Based on Bloom Filter and Physical Unclonable Function,” Elsevier Computers & Security, vol. 113, pp. 102541, 2022.