

Received 30 September 2023, accepted 28 October 2023, date of publication 13 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331030

RESEARCH ARTICLE

Mitigating Denial of Service Attacks in RPL-Based IoT Environments: Trust-Based Approach

FARAG AZZEDIN 

Information and Computer Science Department, Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

e-mail: fazzedin@kfupm.edu.sa

The author would like to acknowledge the support provided by KFUPM Interdisciplinary Research Center for Intelligent Secure System (IRC-ISS). This project is funded by IRC-ISS under project number INSS2202.

ABSTRACT In domains such as telehealth, intelligent transportation, and autonomous agriculture, ensuring secure routing of collected and exchanged data is paramount. Since its inception, there have been many research challenges for the RPL routing protocol that operates in resource-constrained environments and utilizes battery-powered IoT devices. Hence, researchers have focused on this crucial challenge by advising solutions to mitigate attacks that deplete nodes' energy and hence create energy gaps in the network. In this article, we study the impact of two energy exhaustion attacks (hello flooding and version number modification) on the RPL protocol and we present a novel mitigation solution based on behavioural trust. We present an in-depth study of the impact on radio energy consumption of the hello flooding and version number modification attacks in RPL as the number of network nodes increases. We showed that the impact of the former is localized to nodes in the vicinity of the attacker while the latter has a global impact that extends to the entire network. The obtained results from our simulations show that version number modification attack in particular has devastating impact on the network. We also propose a trust-based solution to mitigate these attacks and demonstrate its effectiveness. Accordingly, we conduct comparative study of these attacks and empirically investigate their impact on network performance by running extensive evaluation experiments. Our findings verify the effectiveness of our proposed trust system in mitigating both attacks.


INDEX TERMS Attacks, hello flooding, IoT, power drain, RPL, trust, version number modification.

I. INTRODUCTION

Internet of Things (IoT) has drawn much attention as it enables a wide range of exciting new applications such as health monitoring, home automation, industrial control, and smart cities [1], [2], [3], [4]. IoT is evolving as the concept of enabling intelligent machines' interactions using cutting-edge technologies. Therefore, the future is to transform real world objects into intelligent virtual environments. Furthermore, IoT aims to unify everything in our world under a common infrastructure, giving not only control of things, but also keeping us informed of the state of changes. Hence, IoT plays a vital role in network infrastructure of the Metaverse where IoT resource-constrained things provide

users with a completely real, lasting, and smooth interactive experience that bridges Metaverse and the real world.

IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is proposed to organize such network resource-constrained nodes in *Destination Oriented Directed Acyclic Graph* (DODAG) [5] topology with a single root. It is optimized for multi-hop and many-to-one communication, but also supports one-to-one messages. The root node manages all aspects of organizing other nodes. The root node regularly sends *DODAG Information Object* (DIO) messages to neighbouring nodes. These messages include information about the root node and other metrics such as the depth of the originating router (i.e., rank). Each node that receives one of these DIO messages advertises it to other neighbouring nodes. Eventually, all network nodes know the identity of the root and their rank. After the DODAG construction, nodes can communicate by sending messages to neighbour nodes

The associate editor coordinating the review of this manuscript and approving it for publication was Yufeng Wang .

which forward them until they reach their destination. This is possible because the algorithm that constructs and maintains DODAG guarantees that there are no cycles and no node is isolated [5].

It should be noted that the operating systems and communications protocols used by these gadgets are usually designed to have minimal footprint. ContikiOS [6] and TinyOS [7] are two popular IoT operating systems that have their respective open source RPL implementations [8] as lightweight communication protocol for IoT devices that became a standard by IETF [9].

However, there are still bottlenecks in RPL. IoT devices are gadgets that are designed to collect and exchange information with other nearby devices or through the Internet and perform lightweight computations. These devices usually have little to no security measures because most existing security solutions are designed to require heavy power and computational resources. As a result, IoT cybersecurity attacks have increased dramatically [10], [11], [12]. For example, Chalubo botnet and its predecessor Mirai [13] exploited vulnerabilities in more than 100,000 IoT devices. These high risks seriously impact IoT network topology, security, privacy, and energy levels [10], [13]. Since its standardization, many attacks on the RPL protocol have been devised [11], [12]. These attacks exploit properties of the protocol to inflict damage. Most potent of these attacks are *Denial of Service* (DoS) [11], [12], [14] and routing attacks [15], [16]. Specifically, RPL can be subject to several attacks when it transmits packets between nodes. One of the most common DoS attacks targeting RPL protocol is *DAG Information Solicitation* (DIS) *Hello flooding* and *version number modification* attacks, which has a detrimental effect on the node's limited processing power and energy level.

The rest of the article is organized as follows. The problem statement is discussed in Section II, which presents also the proposed trust-based mitigation solution. Section III gives an overview of the RPL protocol while Section IV outlines the related work. Section V explains the performance evaluation environment and the experiments conducted to evaluate the attacks impact. Section VI discusses the results of the statistical analysis performed on data collected from all evaluation experiments. Finally, Section VIII concludes the article and envisions future directions.

A. MOTIVATION AND CONTRIBUTIONS

Routing in resource-constrained environments is a challenge. As revealed by recent research [17], [18], [19], RPL should be lightweight to prevent these energy-constrained nodes from depleting their power. As such, developing RPL as an effective routing protocol is hampered by this constraint. That is, while energy is one of the scarcest resources for nodes [17], RPL should not be used to frequently exchange control messages. This RPL constraint can be exploited by cyber attackers to flood the network with control messages as in the case of *Hello flooding* that sends

many DIS messages. The attacker can also launch *version number modification* attacks that flood the RPL network with modified DIO messages. This highlights RPL vulnerabilities which motivates on-going and recent research towards more robust defense mechanisms [10], [20], [21]. Therefore, the contributions of this article are:

- We explore in-depth the impact of two RPL attacks namely, *hello flooding* and *version number modification*.
- We utilize the open source implementation of RPL protocol ContikiRPL to simulate the two attacks to find their impact on IoT devices and networks with respect to power consumption. These two attacks aim to drain the power of IoT by spamming control messages at the receiver or throughout the network and hence violate availability of RPL nodes and network.
- We propose novel trust-based solution to mitigate these two attacks.
- We conduct extensive evaluation experiments to evaluate the proposed trust-based mitigation solution.

As such, the above mentioned contributions' impact on the research community is to promote and enable RPL as an effective routing protocol. This need has been identified by the research community [17], [18], [19]. Our contributions also align with the requirements set by the research community [10], [20], [21].

II. PROBLEM STATEMENT AND PROPOSED SOLUTION

Information sources are the building blocks of any information system. Sensors in IoT are one example of these physical things belonging to various application domains, such as healthcare, and education. In spite of their benefits, the interconnected devices also pose some security challenges. Traditionally, *intrusion detection systems* (IDS) have been a vital tool for protecting information systems. However, IDS techniques developed for traditional information systems are not suitable for IoT due to its specific characteristics, such as resource-constrained devices, specific protocol stacks, and standards.

This article aims to explore attacks that violate the availability system requirement. These attacks exhaust victim's energy by sending many requests. There are two main attacks in this category namely, *hello flooding* and *version number modification*. The objective is to understand, evaluate, and examine the impact of each of these attacks. A trust-based solution to reduce the impact of these two attacks is proposed and evaluated.

One of the cornerstones of RPL is cooperation. Nodes in RPL network cooperate to construct and maintain a robust network, where messages flow optimally. In such an environment, trust-based solutions have proven very useful in mitigating security problems. Figure 1 shows the trust-based solution. Let us assume that node x is transmitting message m to node y . Once received, y checks if m is control or data message. If m is DIS message, then the *Information*

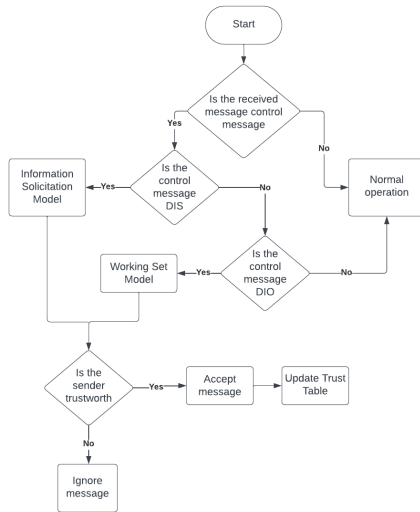


FIGURE 1. The proposed trust-based mitigation solution.

Solicitation Model is invoked. On the other hand, if m is DIO message, then the *Working Set Model* is invoked.

The working-set model is based on the assumption of temporal locality, which refers to not changing version number within a small time frame. This model uses a parameter Δ , to define the working-set window. The idea is to ensure the version number does not change during the working-set window. if the sequence of received version numbers from x within the working set at time t_1 is not changing, then x is trustworthy. By time t_2 , the working set can change and consequently the version number can change. Similarly, the information solicitation model controls the frequency of received DIS messages. When it is too high, we know that x is flooding y and hence x is untrustworthy.

We modified the part of the RPL code that deals with processing DIO and DIS messages, which are used in the *Version number modification* and *Hello flooding* attacks, respectively. The goal is to check if the current node is trustworthy or not before processing their messages. Then, a decision is made of whether to process their messages. In this initial sketch, we assume the existence of a trust model in the network that supplies binary trust level value to indicate the node's trustworthiness. If a node is trustworthy, its DIO and DIS messages are processed normally, otherwise they are discarded.

We consider a system model running LLN with RPL. In this model, resource-constrained nodes communicate directly or indirectly through lossy links. To simplify our system model, we assume that RPL maintains only one DODAG rooted at the DODAG root. In this model, an adversary can be:

- a malicious node or can capture and compromise a legitimate node and reprogram this legitimate node to behave maliciously. The malicious node then floods nearby nodes by broadcasting DIS messages to all nodes within its reach. Nodes that receive a DIS

message reply with a DIO message to advertise the existence of that neighbour. This floods the network with packets and keeps nodes continuously performing useless computations until their batteries are drained.

- a malicious nodes changes the DIO packet's version number to a higher number. Nodes that receive this DIO packet with a new version number, respond to the global repair to update the version to the new value and advertise the received version by sending DIO packets to their neighbor nodes. During a global repair, a new DAG is reconstructed for all the nodes to ensure an optimized and loop free tree. This reconstruction is expensive operation that requires computation from all nodes in the network. Therefore, a malicious node's goal is to trigger this operation continuously keep all nodes busy performing useless operations and consequently deplete nodes' energy.

III. RPL BACKGROUND

RPL is IPv6 routing protocol for low power and lossy networks designed by IETF routing over low power and lossy network (ROLL) group as a proposed standard. Given the significant overlapping between LLNs and IoT, and the fact that IPv6 is an essential feature in IoT environments, RPL has rapidly become the routing protocol for IoT. RPL design principle is to construct *Directed Acyclic Graph* (DAG). DAG is tree-like topology that has a single destination called *Destination Oriented Directed Acyclic Graph* (DODAG) [22]. This hierarchical design has the advantage of preventing network traffic loops [23]. Each node in the DODAG has a rank that indicates what it costs to get to the root; typically nodes closer to the root have a lower rank than nodes farther away [24]. RPL uses an objective function to calculate the rank of network nodes [24]. The objective function uses different metrics to determine the cost to reach the root node as energy consumption, hop count, or quality of the proposed paths [23]. For clarity and completeness purposes, we provide an overview RPL protocol. The overview covers RPL traffic flow and modes of operation, topology formation, RPL control messages, trickle timer, and finally, RPL security.

A. TOPOLOGY CONSTRUCTION

DODAG construction is shown in Figure 2. The root node broadcasts its information using *DODAG Information Object* (DIO) message. This DIO message will reach nodes that locate within the root's communication range. Typically, when a node receives a DIO message, it evaluates the routing information like RPL instance, the version number, the object function, and the mode of operation that represents the network information. DIO message also carries information about the sender, including node ID and node rank. Therefore, any node should add its information before sending the DIO message [24], [25], [26].

To join DODAG, new node sends a message called *DAG Information Solicitation* (DIS). Nodes within the communication range of the new node reply with DIO

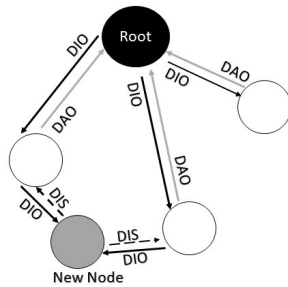


FIGURE 2. The DODAG construction mechanism.

message carrying node ID, objective function, and node rank [24]. Until the new node receives a DIO message from one of its neighbor nodes, it constantly broadcasts DIS messages at a set interval. This interval may vary with different RPL implementations. The new node stops sending DIS messages after receiving DIO messages from neighbor nodes and starts to view all senders as prospective parents. Alternatively, a new node can wait to receive DIO messages from its neighbors without sending a DIS message. The period between two consecutive DIO messages is dynamic, and the trickle timer determines this interval [27].

When a node receives a DIO message, it calculates its rank by considering a given objective function that aims to optimize the energy consumption, the hop count, or the quality of the proposed paths [24]. The main purpose of the objective function is to determine the rank of each node within the DODAG. Therefore, the root node is the sink with the minimum rank [25]. Additionally, it prioritizes the nearby nodes as prospective parents in an ordered list [28]. In DODAG, each node selects the preferred parent which is the node that offers the lowest cost or the minimum rank for this node [24], [25], [28]. The IETF has officially defined two objective functions: Objective Function Zero (OF0) which considers hop count as the routing metric [29]. The second objective function is Minimum Rank with *Hysteresis Objective Function* (MRHOF) which uses paths that minimize ETX (Expected Transmission Count) as a metric [30]. Based on the objective function and the rank of the sending node, nodes decide whether to join this DODAG [23].

When a new node selects its preferred parent, it registers itself by sending *Destination Advertisement Object* (DAO) message to its preferred parent [24]. In storing mode, each node maintains a routing table that maps all reachable destinations in its sub-DODAG to their corresponding next-hop nodes, as discovered when receiving DAOs. While in non-storing mode, the DAO is delivered directly to the root. When the root receives the DAO, it adds the node to its routing table and stores the parent-child relationship, which is later utilized for source routing. The DAO message may optionally, upon explicit request acknowledged by their destination. The *Destination Advertisement Object-Acknowledgement* (DAO-ACK) message is sent back to

the DAO sender. RPL relies on the trickle algorithm to define the intervals between checking the consistency in the transmission based on DIO messages. Trickle algorithm is also designed to control the redundant messages and to save the power by exchanging routing information within dynamic periods if an inconsistency is detected. Specifically, trickle algorithm transmits DIO messages when DODADG shows inconsistency and reduce the period between two consecutive evaluation to the minimum. But, when nodes exchange consistent information about DODAG, trickle algorithm increases the interval of checking transmission until a predetermined maximum interval. To implement this mechanism, the trickle algorithm defines three parameters [31], [32]. The first and the second parameters are the minimum and the maximum interval times. The last parameter is the consistency threshold which indicates invoking the suppression mechanism after receiving K consistent DIO messages. It is worthwhile to mention that all these values are determined by the DODAG root.

IV. RELATED WORKS

In this section, we discuss behaviour trust and trust models. This section also study various RPL DoS attacks and their countermeasures. We conclude this section by comparison and analysis to highlight the novelty of our proposed trust-based approach.

A. BEHAVIOR TRUST

Trust models [33], [34], [35], [36] are proposed to tackle behaviour-related issues. Recently, behavior issues have started to get attention from the research community [33], [37]. Trust is conceptualized in diverse ways and there are many trust models discussed in the literature [33], [34], [35], [36], [37]. These trust models argue and raise awareness for trust-based processes and also outline malicious behavior's effect on information systems.

Behavior trust is identified as a vital component in any Internet-based transaction and lack of behavior trust is a major obstacle for the potential growth of Internet communities [33], [34], [35], [36], [37]. Operating in open and dynamic environments, a device encounters unfamiliar and possibly hostile devices. There is a lack of consensus in the literature on the definition of behavior trust and on what constitutes behavior trust management [33], [34], [35], [36], [37]. Trust is a multi-dimensional notion that is suitable for a wide range of relationships [34], [35], [36]. Researchers have defined trust in different ways, which often reflects the researcher's background. The definition of behavior trust used in this article is adopted from [35] and [38]:

A device is trustworthy if there is a firm belief in the competence of the device to act as expected such that this firm belief is not a fixed value associated with the device but rather it is subject to the device's behavior and applies only within a specific context at a given time.

B. TRUST MODELS

Khan and Herrmann in [39] proposed trust-based IDS mechanism to overcome some attacks in IoT network. In this mechanism, a node can listen to its neighbors to rate them based on behaviour. The collected trust level is forwarded to the root node to evaluate and calculate the reputation value. A node is detected as malicious if the combined reputation value shows a high distrust value. This mechanism can detect forward attacks, rank attacks, and version number modification attacks. The distrust value can vary between attacks, as this mechanism gives nodes that generate a version number modification attack a higher distrust value.

Collective trust technique is used by Tandon and Srivastava [40] to provide protection against rank and sybil attacks in IoT. The overall trust value for each node is used to determine the node trustworthiness. Overall trust value is computed based on direct and indirect trust. Direct trust is calculated based on node behavior. The indirect trust considers the trust value received from neighbor nodes and the direct trust value calculated for the node. The attack detection is done based on the overall trust value.

Muzammal et al. [41] introduce a trust-based secure routing protocol in RPL. The overall trustworthy value of nodes is calculated based on direct and recommended trust values. Each node calculates the trustworthy value for its neighbor nodes. The calculation of the trust value takes in consideration, node mobility and trust metrics specify in RPL topology to detect attacks [41]. Authors in [42] propose time-based trust-aware RPL routing protocol. The proposed protocol provide protects against rank attacks and sybil attacks in IoT networks. The node's total trust value is calculated by its direct neighbors and recommended value. The trust value is evaluated as a time-based successful packet exchange between nodes to determine the node reliability. Malicious nodes with a lower trust value are isolated from the network to improve performance.

Hashemi and Aliee in [43] propose a *Dynamic and Comprehensive Trust Model for IoT* (DCTM-IoT). Trust is calculated based on three dimension. The first dimension is the quality of communication between two devices. The trust level for this dimension is computed using history information, direct observation, direct information, indirect observation, and indirect information. The second dimension calculates the trust based on processing the quality of service provided by the device. The last dimension calculates the trust based on processing contextual information. With these dimensions, the overall trust measure is computed. The attack detection is based on the node trust value. DCTM-IoT countermeasures rank, blackhole, and sybil attacks.

In [44], two lightweight techniques to mitigation version number modification attack was introduced. The first technique accepts direct veriosn number updates only from the root towards leaf nodes. This technique eliminates any malicious version number updates. The second technique uses a trust mechanism known as Shield. In Shield, a node changes the version number if the majority of nodes near

the root claim that there is a change. A single update from root is not trustworthy to change the version number. Shield maintains a list of neighbor nodes close to the root. Updates are based on information provided by the majority of nodes on this list.

In [45], a cooperative trustworthiness scheme is used for securing the routing topology. Trustworthiness of nodes is evaluated using node energy, honesty, selfishness, and expected number of retransmissions. Selfish nodes try to conserve their resources and consume other nodes resources. Decisions are made via IDS running on each node. Nodes with the highest overall trust value are chosen as parents. This implies more reliable routing.

C. DOS ATTACKS

DIS flooding attack: The malicious node floods nearby nodes by broadcasting DIS messages during this attack, which leads to resetting their trickle timer. Another strategy this attack could involve is targeting a specific neighbor by sending DIS messages and initiating its response in the form of a DIO message. The effect of this attack is to increase the exchange of control packets that leads to network congestion. This attack has more impact when the attacker broadcasts DIS messages. As per RPL standard [25], broadcasting DIS messages leads to resetting the receiver's trickle timer, which also leads to flooding the network with DIO messages and to increases the power consumption of nodes.

Hello flooding attack [46] aims to drain the battery of targeted nodes by keeping them awake and doing useless computations. A malicious node sends many DIS messages to all nodes within its reach. Nodes that receive a DIS message reply with a DIO message to advertise the existence of that neighbour. This floods the network with packets and keeps nodes continuously performing useless computations until their batteries are drained.

DIO Suppression Attack: This is another DIO replay attack where an attacker receives DIO packet from legitimate node and replays it repeatedly at fixed frequency. This flood of consistent DIO packets causes victim nodes to suppress their own DIO transmission due to the trickle algorithm specification. As a result, victim nodes stop sending their DIO packets after receiving enough consistent DIO packets [47], [48]. The DIO suppression attack uses the same mechanism as neighbor attack. However, this attack replays the received DIO packet at fixed frequency [47], [49].

Sybil Attack: Here, the attacker generates multiple redundant DIS packets with fake identities. This illegitimate packet generation causes surrounding nodes to reset their trickle algorithm [50], [51]. Authors in [48] and [51] modeled sybil attackers as nodes that multicast several DIS packets with false IDs to all their nearby nodes. Additionally, [52] stated that any moving node that sends DIS packets using new IPv6 address could be viewed as suspicious by its neighbors.

DAO Insider attack: To launch this attack, the attacker node frequently generates unaltered DAO packets to its parent

nodes to force ancestor nodes to flood the network with DAO packets [53], [54], [55]. While RPL is operating in storing mode and as the transmission of DAO packets moves upward towards the sink node, the attack's damage expands beyond its immediate vicinity [53], [55].

Version number modification attack [56] operates in a similar way to the *hello flooding* attack by flooding the network with control messages. However, in *version number modification* attack, the malicious node forwards all the DIO messages it receives with one of its parameters changed. This parameter is called version number and the malicious node usually increments this number, which causes the root to start global repair of the network. During a global repair, a new DAG is reconstructed for all the nodes to ensure an optimized and loop free tree based on some objective function. This reconstruction is expensive operation that requires computation from all nodes in the network. Therefore, a malicious node's goal is to trigger this operation continuously keep all nodes busy performing useless operations. Verma et al. [57] presented RPL attack taxonomy categorized as resources, topology, and traffic attacks. Resources attacks exhaust resources such as power and memory, whereas topology attacks disturb normal network topology by isolating nodes and connecting other to reduce network throughput. Finally, traffic attacks target network malicious traffic such as eavesdropping.

D. DOS ATTACKS COUNTERMEASURES

Countermeasures can be categorized as prevention, detection, and mitigation. Detection solutions provide systems with the ability to identify any suspicious behavior and trigger the system when an attack is present. Typically, detection solutions complement prevention solutions when it becomes impossible to fully prevent attacks either by the failure of the prevention solution or, in some cases, applying a protection solution becomes a burden on network resources [58].

On the other hand, mitigation solutions allow the attack, but react to minimize its impact on the network [59]. Unfortunately, current studies [60], [61] show that detection solutions are given more attention than mitigation solutions, especially in RPL environment.

Many approaches were proposed in the literature for detecting attacks on resources. These approaches can be categorized as three approaches namely, centralization, authentication, cryptographic puzzles. Centralization is basically the use of a trusted backend server to detect and throttle attackers. The basic idea in all solutions that belong to this category is to detect anomalies in behavior such as, sending too many RPL control messages in a given time period. Examples of approaches that belong to this category are [62] and [63]. Authentication approaches utilize lightweight authentication schemes using either symmetric or asymmetric keys to identify requests from legitimate nodes. Examples of such approaches are [64] and [65]. Finally, cryptographic puzzles throttle attackers by requiring them to

solve cryptographic puzzles before processing their requests. Furthermore, the difficulty of these puzzles increases for a particular node as more requests are generated. This increases the time between their requests. Examples of approaches in this category are [66] and [67]. The impact of these attacks on edge IoT nodes was demonstrated by Ioulianou et al. [62] and their simulations showed that node activity has increased from 35% during normal operation to 50% during an attack for affected nodes. However, one of the limitations of most of these studies is that their simulations, if any, are small scale simulations with simplistic scenarios to show the significant impact on battery drain [46], [62], [63].

Authors in [68] utilized RPL attack framework to simulate three different RPL attacks: hello-flood, decreased-rank, and increased-version. The attacks were simulated separately and simultaneously, with a primary focus on detecting them using an artificial neural network (ANN)-based supervised machine learning approach. The proposed ANN-based model detected the attacks in each scenario. In [54], authors collect network traffic traces from simulated environment in normal and attacker settings. Then, an AdaBoost ensemble model termed as Ada-IDS is developed to detect attacks in RPL-based IoT.

The proposed approach in [69] detects resource-based attacks in RPL networks. This technique involves deploying allied parent nodes throughout the network to monitor all nodes and identify any malicious activity. These nodes then report back to the root node. By implementing this technique, resource-based attacks can be detected and eliminated more efficiently, resulting in reduced control overhead and increased topology lifetime. The allied node approach protects the RPL topology, improves packet delivery, and prolongs the network's lifespan by reducing node latency and energy consumption.

Federated-transfer-learning assisted customized distributed (FT-CID) IDS model is proposed in [70] to identify RPL intrusion in IoT environments. The design process of FT-CID includes three steps: dataset collection, FTL-assisted edge IDS learning, and intrusion detection. According to the authors, the FT-CID model accomplishes high RPL security by implicitly utilizing the local and global parameters of different IoTs with the assistance of FTL.

In [71], a RPL defense scheme called DIS-mitigation is proposed. This system effectively reduces the impact of DIS flooding attacks on network output and identifies the nodes responsible for these attacks. The experimental results show that the proposed model can quickly and effectively mitigate DIS flooding attacks in both static and dynamic network conditions, without creating significant overhead for the nodes.

Authors in [72] present a solution to combat RPL version attacks that is both lightweight and effective. By making some simple modifications to the RPL functionality, a collaborative and distributed security scheme has been added to the protocol design. Experimental results demonstrate that the proposed solution is scalable and improves the protocol's resilience against simple and composite version attacks in

different experimental setups. The solution enables fast and accurate attack detection, quick topology convergence, and efficient management of network stability, control overhead, and energy consumption during different scenarios of version attacks.

E. COMPARISON OF COUNTERMEASURES

Table 1 summarized the RPL attacks countermeasure solutions. As stated in [60] and [61], most countermeasure solutions fall under the detection category. A mitigation solution is proposed in [72]. The drawback of this solutions is the assumption that the sink node cannot be exposed to any kind of attack. The sink node is specified to be the node that legitimately initiates a global repair and modifies the advertisement in the DIO messages. This is not a realistic assumption and hence makes this solution not implementable in real RPL environment setting. This solution also relies on centralized mitigation solution as it depends on sink node. On the other hand, Our proposed trust-based mitigation solution is distributed in nature and assumes that any node in the RPL network can be exposed to attacks. Furthermore, our proposed solution monitors RPL nodes to mitigate their malicious behavior.

TABLE 1. Comparison of countermeasure solutions.

Ref.	Countermeasure				Mitigation
	Detection				
	Centralized Based	Authentication Based	Cryptography Based	Others	
[62]	✓				
[68]				✓	
[66]			✓		
[63]	✓				
[65]		✓			
[72]					✓
[70]				✓	
[67]			✓		
[64]		✓			
[71]				✓	
[69]	✓				

V. EXPERIMENT SETUP

This section describes the performance evaluation environment used to investigate the impact of the *hello flooding* and *version number modification* attacks on energy consumption. There are 3 components that make up the environment of all simulation experiments: Contiki, Cooja and the RPL Attacks framework. Sections V-A, V-B and V-C discuss each of these, respectively.

A. CONTIKI

Contiki [73] is an operating system for networked, memory-constrained systems with focus on low-power wireless IoT devices. Contiki provides multitasking and built-in Internet Protocol Suite (TCP/IP stack), yet needs only about 10 kilobytes of RAM and 30 kilobytes of ROM. Contiki also provides its own implementation of the RPL protocol (ContikiRPL [74]) which we use in our simulations. Every node in our simulations runs Contiki.

TABLE 2. Configuration of all simulations.

Group	# of Nodes	Malicious Type	Area
1	10	None Hello Flooding Version Number Modification	100m
2	25	None Hello Flooding Version Number Modification	250m
3	50	None Hello Flooding Version Number Modification	500m
4	100	None Hello Flooding Version Number Modification	1000m

B. COOJA NETWORK SIMULATOR

Cooja is network simulator for Contiki that allows simulating IoT connected devices. Cooja has a graphical user interface that allows users to configure all the simulation environment parameters and gives access to statistics about all network nodes, including but not limited to power consumption and raw traffic. We use Cooja to simulate all the scenarios we present in section V-D. To create a realistic simulation environment for low-power IoT devices, we rely on Contiki-NG because of its lightweight design, along with flexible resource management modules. Contiki-NG already includes the RPL protocol, making it an ideal choice for our needs. To simulate the compiled IoT networks, we use Cooja within the Contiki-NG environment.

C. RPL ATTACKS FRAMEWORK

The RPL attacks framework is a user-friendly framework that abstracts most aspects of the running simulations. The framework allows users to configure campaigns of simulations as JSON files and take care of running all those campaigns and collecting and saving all results. For example, to run 100 different simulations the user defines 100 simulation configurations and asks the framework to run all these simulations and output all the results to a certain directory.

D. ATTACKS IMPACT

We conducted statistical analysis to gain better understanding of the simulation results. A total of 12 simulations are run as shown in Table 2. These simulations are divided into 4 groups based on the number of nodes. For each group, a baseline simulation without a malicious node is performed along with two other simulations for each malicious node type. Then, the radio energy consumption values are collected for each node from each simulation as shown in Table 3. Finally, descriptive statistics are generated from the data of each simulation and different comparisons are performed, as follows:

- Compare baselines of each group to determine the normal energy consumption values as the number of network nodes increase.

TABLE 3. Energy values description.

Value	Description
ON	Energy consumed while node radio is on
RX	Energy consumed while node radio is receiving data
TX	Energy consumed while node radio is sending data

- Compare the simulation results with malicious nodes to the baselines of their respective groups to determine the impact of each attack type on energy consumption.
- Compare the simulations results with malicious nodes of the same type across different groups.

We use the following 3 descriptive statistics:

- The average, maximum and minimum of ON values in the network.
- The average, maximum and minimum of TX values in the network.
- The average, maximum and minimum of RX values in the network.

VI. RESULTS AND DISCUSSION

Here, we analyze the results of the statistical analysis performed on data collected from all simulations under different network sizes. It is organized as follows. Section VI-A discusses the normal (baseline) RPL network operation while section VI-B discusses the RPL network operation under *Hello flooding* attack. Section VI-C discusses the RPL network operation under *Version number modification* attack. Finally, section VI-D evaluates our proposed trust-based solution in mitigating *Hello flooding* and *Version number modification* attacks.

A. NORMAL (BASELINE) OPERATION

This section presents the results of baseline simulations of the network under normal conditions without any malicious nodes. We present radio energy consumption statistics for networks of different sizes to establish a baseline for the network to measure the impact of each attack.

Figure 3, shows the average values for radio ON, RX and TX as the number of network nodes increases from 10 to 100. The results demonstrate the ability of the RPL protocol to minimize power consumption as the network grows in size. As the network size increased ten-fold from 10 to 100, the average values of ON, RX and TX increased by 2.5, 3.3 and 6, respectively. This sub-linear increase in radio energy consumption levels is owed to the efficient routing mechanism used in RPL.

Figure 4, shows the radio energy consumption levels for most stressed node in the network. These values are highly sensitive to the topology of the network and node distribution in the network, i.e a dense network will result in higher maximum values and vice versa.

Figure 5, shows the radio energy consumption levels for least stressed node in the network, which usually exist at

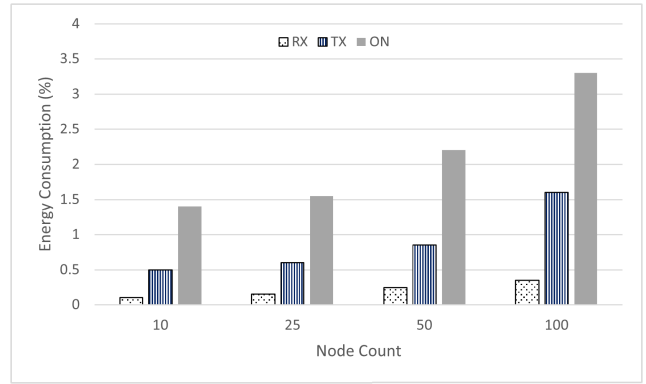


FIGURE 3. Average power consumption with different node counts.

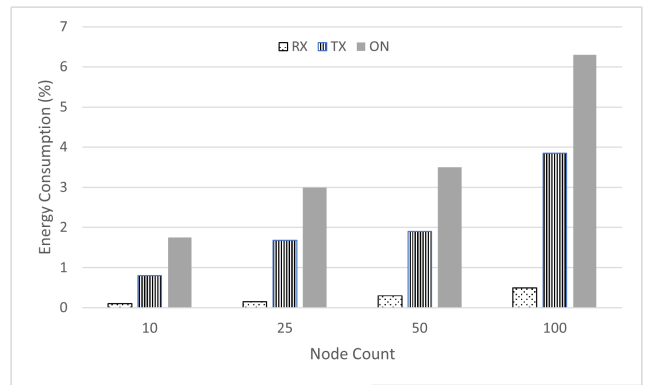


FIGURE 4. Maximum power consumption with different node counts.

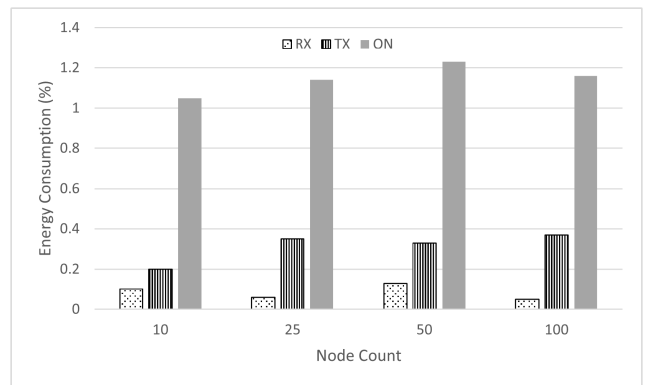


FIGURE 5. Minimum power consumption with different node counts.

the network edge. The differences in energy consumption are almost negligible as network size increases.

B. HELLO FLOODING OPERATION

We present radio energy consumption to demonstrate the locality of the *Hello flooding* attack and quantify its impact. Figure 6 shows a comparison of the average radio energy consumption levels between networks of different sizes and their respective baselines. The results indicate that the impact of the attack is localized (i.e limited to the nodes

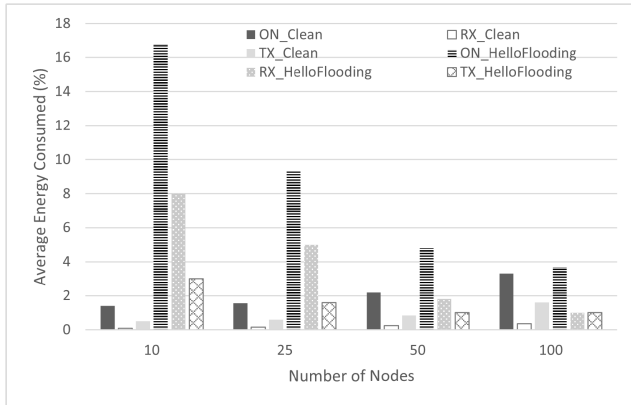


FIGURE 6. Average power consumption influenced by Hello flooding attack.

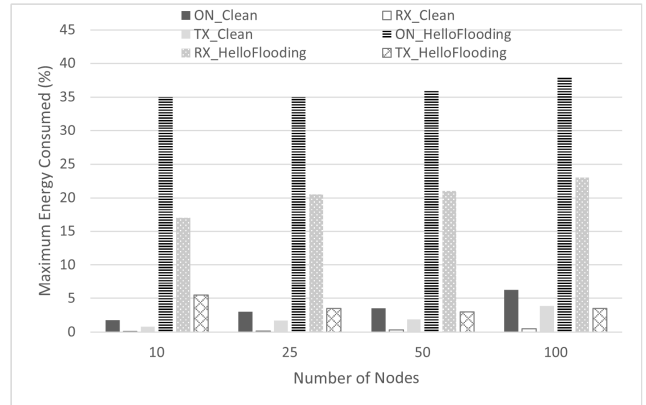


FIGURE 8. Maximum power consumption influenced by Hello flooding attack.

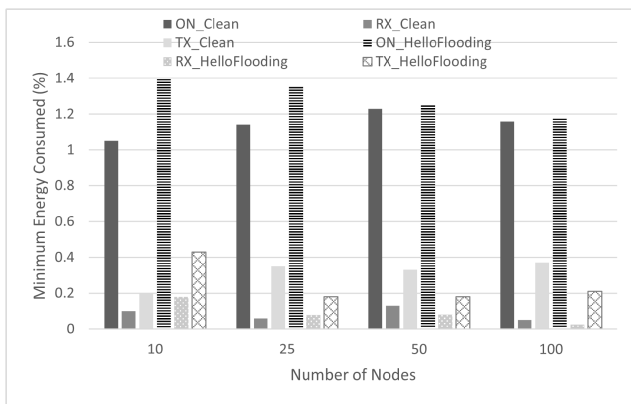


FIGURE 7. Minimum power consumption influenced by Hello flooding attack.

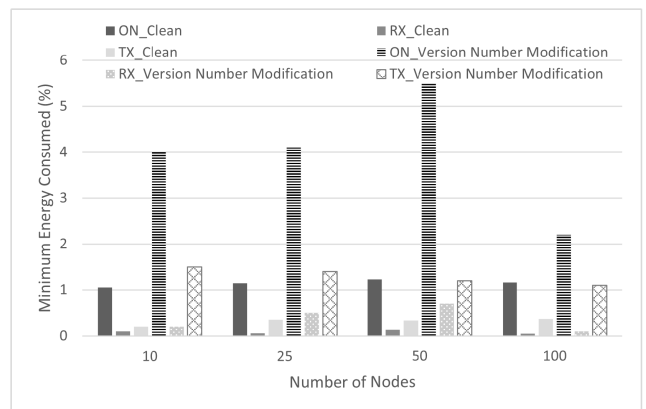


FIGURE 9. Minimum power consumption influenced by version number modification attack.

in the vicinity of the attacker), because as the number of nodes increases, the average of the attacked network nodes decreases to become comparable the baseline. This fact is also verified by the radio energy consumption levels for least stressed node in the network, which are comparable to the baseline, as shown in Figure 7.

Since the impact of the *Hello flooding* attack is localized to the nodes within range of the attacker, we can quantify the impact of the *Hello flooding* attack by comparing the energy consumption of the most stressed nodes in the network under attack and a comparable network free of the attack, as shown in Figure 8. It is worth noting that the malicious node consumes the same amount of energy as the most stressed node in the network.

C. VERSION NUMBER MODIFICATION OPERATION

We present radio energy consumption statistics to demonstrate the global impact of the *Version number modification* attack. Figure 10 shows a comparison of the average radio energy consumption levels between networks of different sizes and their respective baselines. The results indicate that the impact of the attack is global (i.e extends to all nodes in the network). As the number of nodes increases, the average of

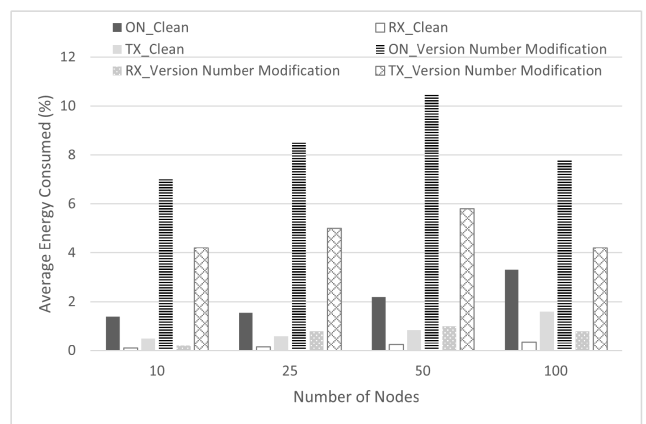


FIGURE 10. Average power consumption influenced by version number modification attack.

the attacked network nodes increases. This fact is also verified by the radio energy consumption levels for least stressed node in the network, which have doubled in the best case scenario and more than quintupled in the worst case scenario, as shown in Figure 9.

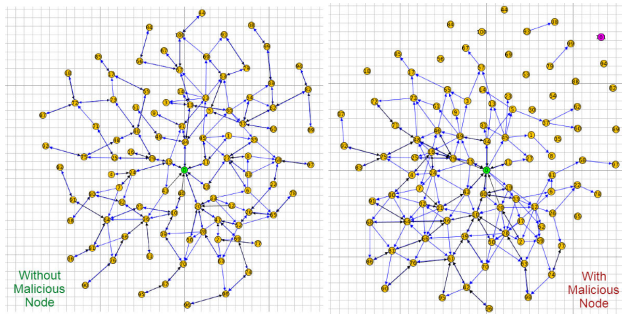


FIGURE 11. Average power consumption influenced by version number modification attack.

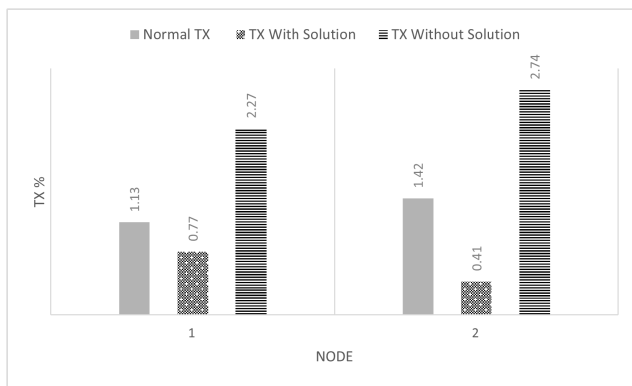


FIGURE 12. Energy consumed (TX%): Top 2 stressed nodes under Hello flooding attack.

In addition to increasing the energy consumption of all nodes in the network, the *Version number modification* attack could isolate portions of the network for sometime. This attack occurs because rebuilding the DODAG is time consuming for large number of nodes (≥ 100). By the time the root node finishes constructing the DODAG, the attack is launched again. Figure 11, shows a snapshot of the network at the end of the simulation (5 minute mark), for the baseline network and the network with the attack. The arrows indicate that there has been communication in the past minute between connected nodes. Notice how several nodes are completely disconnected, meaning they do not have any route to the root node. Furthermore, the energy consumed by the malicious node is less than the average of entire network, meaning that the malicious node did minimal work to cause all this disruption. This is in contrast to the *Hello flooding* attack, where the malicious node sends a lot of messages to overwhelm its neighbours.

D. PROPOSED SOLUTION EVALUATION

The first set of experiments is designated for the *Hello flooding* attack. The power drained from the victim nodes was reduced because the victims were no longer sending replies to the attacker. These victim nodes just discarded the message. This result is not surprising because a node cannot assess the legitimacy of a message before receiving and processing it

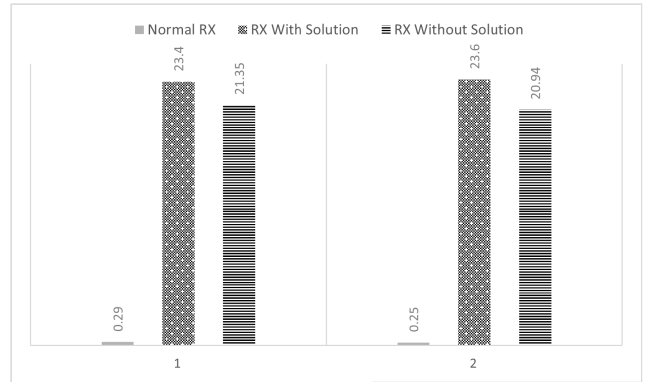


FIGURE 13. Energy Consumed (RX%): Top 2 stressed nodes under Hello flooding attack.

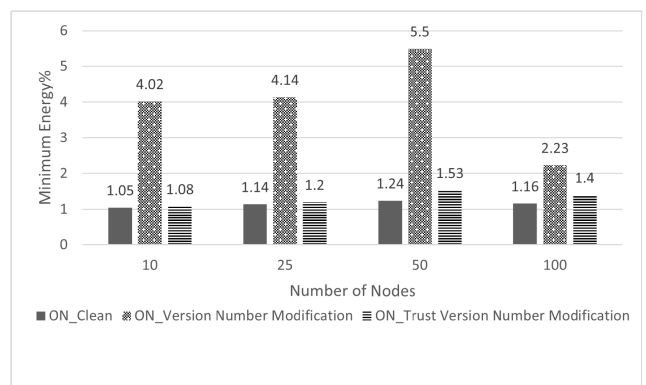


FIGURE 14. Least stressed node: ON Energy consumption under Version number modification attack.

(i.e RX radio energy consumption cannot be controlled). Figures 12 and 13 illustrate this fact through the RX and TX energy consumption of the 2 most stressed nodes in a network of 50 nodes in different states.

The results of the second set of simulations for the *version number modification* attack showed more promising results. Our solution was able to completely mitigate the impact of this attack as illustrated in Figures 14, 15, and 16. The global impact of the *Version number modification* attack was fully mitigated by our solution. This can also be validated by looking at the average ON value for each network size when the solution is deployed, which drops to match normal network average.

E. OBSERVATIONS AND REMARKS

Experiments conducted in Section VI-A show the energy consumption when the RPL network is under no attack. Section VI-B exposes the RPL network to *Hello flooding* attack. This attack is localized and hence it affects only nodes within range of the attacker. As such, as the number of nodes increases, the average of the attacked network nodes decreases to become comparable the baseline. This fact is also verified by the radio energy consumption levels for least stressed node in the network, which are comparable to the

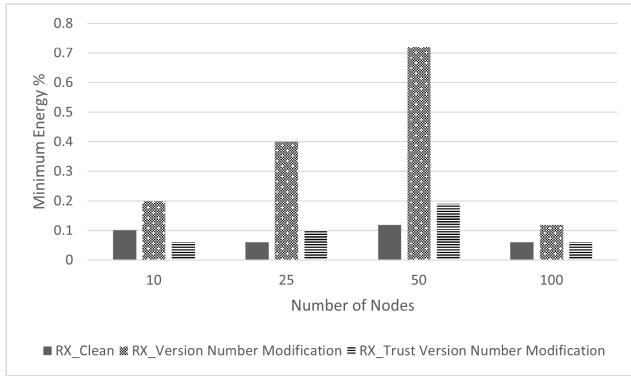


FIGURE 15. Least stressed node: RX Energy consumption under Version number modification attack.

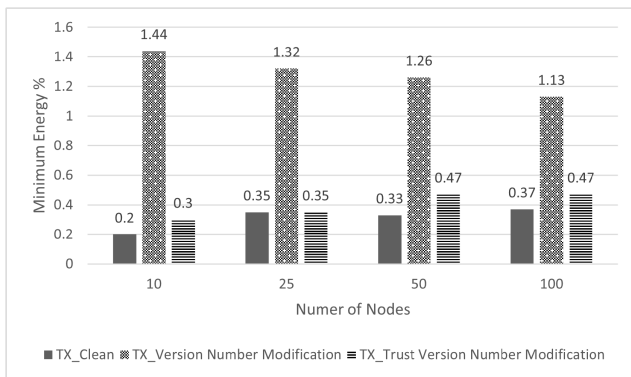


FIGURE 16. Least stressed node: TX Energy consumption under Version number modification attack.

baseline. *Version number modification* attack is evaluated in section VI-C. This section shows that the energy consumed by the malicious node is less than the average of entire network, meaning that the malicious node did minimal work to cause all this disruption. This is in contrast to the *Hello flooding* attack, where the malicious node sends many messages to overwhelm its neighbours. Evaluation of our proposed trust-based solution is shown in section VI-D. Our proposed solution is effective in mitigating both DoS attacks and hence reducing their impact on the RPL network.

VII. LIMITATIONS AND POTENTIAL IMPROVEMENT

One of the issues not considered in the proposed solution is whitewashing. This is a challenge for any trust-based system. If a node changes its identity, it can basically hide any previous malicious history. Another issue is bootstrapping the system and the trade-off using pessimistic versus optimistic trust-based system. Furthermore, when computing trust, trust may decay with time. Therefore, a decay function needs to be applied when obtaining trust. There are some issues that need to be sorted out before the decay function can be evaluated. First, how does the decay function apply to the trust levels. We need to explore the issue of quantity versus time. That is, by how much a trust level should be decayed and what is a reasonable time interval to decide applying the decay.

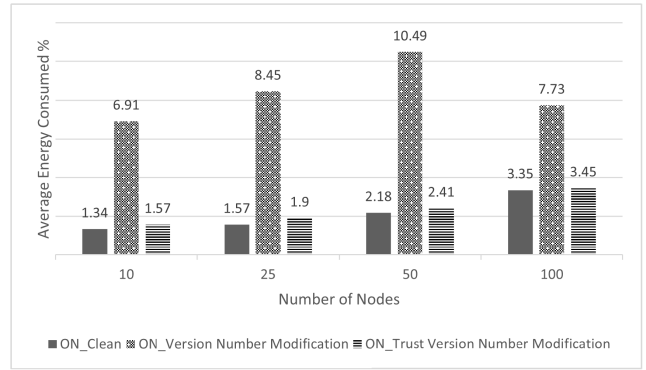


FIGURE 17. Average ON energy consumption under Version number modification.

Trust involves specifying and reasoning about beliefs. In the trust model proposed in this paper, trust can be represented as fuzzy values. The trust notion is a subjective and vague point of view about how the future behavior of other entities would fit in the expectations of others. Therefore, fuzzy sets can be used to combine trust levels and formally define the notion of trust. It will be interesting to incorporate a simple rule-based IF X AND Y THEN Z approach to solving the trust level estimation. In addition, since the trust model is a learning process, the learning operation can be formalized using such logic.

Incentive mechanisms to discourage or even prevent untrustworthy behavior. Since honest recommenders are important and vital component in any trust model, incentive mechanisms can be used to encourage honesty and hence reward honest recommenders.

VIII. CONCLUSION AND FUTURE WORK

This work studies the effects of RPL-specific DoS attacks namely, *Hello flooding* and *Version number modification* and evaluates a proposed trust-based solution to mitigate these two attacks.

Firstly, we simulated sensor nodes in Cooja which supports application development for ContikiOS. Simulations were performed for studying the characteristics of normal as well as malicious IoT environments. In the two DoS attack scenarios, compromised nodes perform specific DoS attack utilizing RPL control messages. The attacks negatively impact nodes' power consumption. Experimental results showed that the impact of *Hello flooding* attack is localized to nodes in the vicinity of the attacker while *Version number modification* attack has a global impact that extends to the entire network. Furthermore, we implemented and evaluated a trust based system in the environment of each attack type.

Based on these results, we developed trust-based mitigation solution. The results show that our proposed system was able to partially mitigate the impact of *Hello flooding* attack while it fully mitigated the impact of *Version number modification* attack. Since the impact of *Hello flooding* attack is localized to nodes within range of the attacker, we can

quantify the impact of *Hello flooding* attack by comparing the energy consumption of the most stressed nodes in the network under attack and a comparable network free of the attack.

Looking ahead, we are planning to study other RPL attacks and mitigation techniques and propose taxonomies to classify RPL attacks as well as countermeasures. We are also working on developing subjective-logic-based trust model to identify and mitigate other RPL malicious behaviour including replay and neighbour attacks.

REFERENCES

- [1] L. Alkharji, S. De, O. Rana, and C. Perera, "Semantics-based privacy by design for Internet of Things applications," *Future Gener. Comput. Syst.*, vol. 138, pp. 280–295, Jan. 2023.
- [2] C. Lira, E. Batista, F. C. Delicato, and C. Prazeres, "Architecture for IoT applications based on reactive microservices: A performance evaluation," *Future Gener. Comput. Syst.*, vol. 145, pp. 223–238, Aug. 2023.
- [3] S. Kumar, J. L. Buckley, J. Barton, M. Pigeon, R. Newberry, M. Rodencal, A. Hajzeraj, T. Hannon, K. Rogers, D. Casey, D. O'Sullivan, and B. O'Flynn, "A wristwatch-based wireless sensor platform for IoT health monitoring applications," *Sensors*, vol. 20, no. 6, p. 1675, Mar. 2020.
- [4] B. A. Alohal, V. G. Vassilakis, I. D. Moscholios, and M. D. Logothetis, "A secure scheme for group communication of wireless IoT devices," in *Proc. 11th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2018, pp. 1–6.
- [5] E. Baccelli, M. Philipp, and M. Goyal, "The P2P-RPL routing protocol for IPv6 sensor networks: Testbed experiments," in *Proc. 19th Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2011, pp. 1–6.
- [6] A. M. A. Ferreira, L. J. D. M. D. Azevedo, J. C. Estrella, and A. C. B. Delbem, "Case studies with the contiki-NG simulator to design strategies for sensors' communication optimization in an IoT-fog ecosystem," *Sensors*, vol. 23, no. 4, p. 2300, Feb. 2023.
- [7] *TinyOS: An OS for Embedded, Wireless Devices*, TinyOS, Jun. 2019. [Online]. Available: <https://github.com/tinyos/tinyos-main>
- [8] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. K. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, 2012, pp. 1–157.
- [9] A. Maheshwari, R. K. Yadav, and P. Nath, "Enhanced RPL to control congestion in IoT: A review," in *Proc. Int. Conf. Internet Things*. Cham, Switzerland: Springer, 2023, pp. 1–13.
- [10] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.
- [11] H. F. Lautert, D. D. J. D. Macedo, and L. Pioli, "Micro IDS: On-line recognition of denial-of-service attacks on IoT networks," in *Advanced Information Networking and Applications*, vol. 1. New York, NY, USA: Springer, 2023, pp. 446–459.
- [12] S. Mishra, V. M. Phuc, and N. Van Tanh, "Lightweight authentication encryption to improve DTLS, vank combined with overhearing to prevent DoS and MITM on low-resource IoT devices," in *Internet of Things—ICIOT 2022*. Honolulu, HI, USA: Springer, 2023, pp. 108–122.
- [13] C. Wei, G. Xie, and Z. Diao, "A lightweight deep learning framework for botnet detecting at the IoT edge," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103195.
- [14] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *Proc. 5th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/4th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Jun. 2018, pp. 12–17.
- [15] S. Rabhi, T. Abbes, and F. Zarai, "IoT routing attacks detection using machine learning algorithms," *Wireless Pers. Commun.*, vol. 128, no. 3, pp. 1839–1857, Feb. 2023.
- [16] G. Bansal, V. Chamola, A. Hussain, and M. K. Khan, "Cracking the anonymous IoT routing networks: A deep learning approach," *IEEE Internet Things Mag.*, vol. 6, no. 1, pp. 120–126, Mar. 2023.
- [17] P. Shahbakhsh, S. H. Ghafouri, and A. K. Bardsiri, "RAARPL: End-to-end reliability-aware adaptive RPL routing protocol for Internet of Things," *Int. J. Commun. Syst.*, vol. 36, no. 6, Apr. 2023, Art. no. e5445.
- [18] L. B. Furstenau, Y. P. R. Rodrigues, M. K. Sott, P. Leivas, M. S. Dohan, J. R. López-Robles, M. J. Cobo, N. L. Bragazzi, and K.-K.-R. Choo, "Internet of Things: Conceptual network structure, main challenges and future directions," *Digit. Commun. Netw.*, vol. 9, no. 3, pp. 677–687, Jun. 2023.
- [19] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function," *Ad Hoc Netw.*, vol. 96, Jan. 2020, Art. no. 102001.
- [20] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. A. Bahashwan, I. H. Hasbullah, M. A. Aladailah, and G. A. Mukhaini, "A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100741.
- [21] H. Xia, N. Huang, X. Feng, R. Zhang, and C. Liu, "Starlet: Network defense resource allocation with multi-armed bandits for cloud-edge crowd sensing in IoT," *Digit. Commun. Netw.*, Mar. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352864823000652>, doi: 10.1016/j.dcan.2023.03.009.
- [22] K. Avila, D. Jabba, and J. Gomez, "Security aspects for RPL-based protocols: A systematic review in IoT," *Appl. Sci.*, vol. 10, no. 18, p. 6472, Sep. 2020.
- [23] B. Mohamed and F. Mohamed, "QoS routing RPL for low power and lossy networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, Nov. 2015, Art. no. 971545.
- [24] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Comput. Netw.*, vol. 56, no. 14, pp. 3163–3178, Sep. 2012.
- [25] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks Abstract Low-Power*, document RFC 6550, Internet Engineering Task Force (IETF), 2012, pp. 1–157.
- [26] O. Iova, P. Picco, T. Istomin, and C. Kiraly, "RPL: The routing standard for the Internet of Things... or is it?" *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 16–22, Dec. 2016.
- [27] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," Internet Eng. Task Force (IETF), Fremont, CA, USA, Tech. Rep. RFC 6206, 2011.
- [28] A. Parasuram, D. Culler, and R. Katz, "An analysis of the RPL routing standard for low power and lossy networks," Dept. Elect. Eng. Comput. Sci., Univ. California Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2016-106, 2016, p. 98.
- [29] P. Thubert, *Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document RFC 6552, RFC Editor, Mar. 2012.
- [30] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document RFC 6719, Internet Engineering Task Force (IETF), Request for Comments, 2012.
- [31] E. Arvan, M. K. H. Dehkordi, and S. Jalili, "Secured location-aware mobility-enabled RPL," *J. Netw. Comput. Appl.*, vol. 209, Jan. 2023, Art. no. 103516.
- [32] S. P. Senthilkumar and B. Subramani, "RPL protocol load balancing schemes in low-power and lossy networks," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 7–13, Feb. 2023.
- [33] G. K. Ragesh and A. Kumar, "Trust-based secure routing and message delivery protocol for signal processing attacks in IoT applications," *J. Supercomput.*, vol. 79, no. 3, pp. 2882–2909, Feb. 2023.
- [34] F. Azzedin, H. Suwad, and Z. Alyafeai, "Countermeasuring zero day attacks: Asset-based approach," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2017, pp. 854–857.
- [35] F. Azzedin, "Taxonomy of reputation assessment in peer-to-peer systems and analysis of their data retrieval," *Knowl. Eng. Rev.*, vol. 29, no. 4, pp. 463–483, Sep. 2014.
- [36] C. Liang, F. Wen, and Z. Wang, "Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks," *Inf. Fusion*, vol. 46, pp. 44–50, Mar. 2019.
- [37] M. Aaqib, A. Ali, L. Chen, and O. Nibouche, "IoT trust and reputation: A survey and taxonomy," *J. Cloud Comput.*, vol. 12, no. 1, pp. 1–20, Mar. 2023.
- [38] F. Azzedin, "Trust-based taxonomy for free riders in distributed multimedia systems," in *Proc. Int. Conf. High Perform. Comput. Simulation*. IEEE, Jun. 2010, pp. 362–369.
- [39] Z. A. Khan and P. Herrmann, "A trust based distributed intrusion detection mechanism for Internet of Things," in *Proc. IEEE 31st Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2017, pp. 1169–1176.

- [40] A. Tandon and P. Srivastava, "Trust-based enhanced secure routing against rank and Sybil attacks in IoT," in *Proc. 12th Int. Conf. Contemp. Comput. (IC3)*, Aug. 2019, pp. 1–7.
- [41] S. M. Muzammal, R. K. Murugesan, N. Z. Jhanjhi, and L. T. Jung, "SMTrust: Proposing trust-based secure routing protocol for RPL attacks for IoT applications," in *Proc. Int. Conf. Comput. Intell. (ICCI)*, Oct. 2020, pp. 305–310.
- [42] D. Airehrour, J. A. Gutierrez, and S. K. Ray, "SecTrust-RPL: A secure trust-aware RPL routing protocol for Internet of Things," *Future Gener. Comput. Syst.*, vol. 93, pp. 860–876, Apr. 2019.
- [43] S. Y. Hashemi and F. S. Aliee, "Dynamic and comprehensive trust model for IoT and its integration into RPL," *J. Supercomput.*, vol. 75, no. 7, pp. 3555–3584, Jul. 2019.
- [44] A. Anış, S. B. Ö. Yalçın, and S. F. Oktuğ, "New lightweight mitigation techniques for RPL version number attacks," *Ad Hoc Netw.*, vol. 85, pp. 81–91, Mar. 2019.
- [45] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani, "Trust-aware and cooperative routing protocol for IoT security," *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102467.
- [46] P. Maurya and V. Kushwaha, "Impact analysis of hello flood attack on RPL," in *Advanced Network Technologies and Intelligent Computing*. Varanasi, India: Springer, 2023, pp. 554–568.
- [47] R. Sahay, G. Geethakumari, and B. Mitra, "A novel network partitioning attack against routing protocol in Internet of Things," *Ad Hoc Netw.*, vol. 121, Oct. 2021, Art. no. 102583.
- [48] B. Groves and C. Pu, "A Gini index-based countermeasure against Sybil attack in the Internet of Things," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2019, pp. 1–6.
- [49] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO suppression attack against routing in the Internet of Things," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2524–2527, Nov. 2017.
- [50] C. Pu and K.-K.-R. Choo, "Lightweight Sybil attack detection in IoT based on Bloom filter and physical unclonable function," *Comput. Secur.*, vol. 113, Feb. 2022, Art. no. 102541.
- [51] C. Pu, "Sybil attack in RPL-based Internet of Things: Analysis and defenses," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4937–4949, Jun. 2020.
- [52] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, "A trust-based intrusion detection system for mobile RPL based networks," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber. Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Exeter, U.K., Jun. 2017, pp. 735–742.
- [53] I. Wadhaj, B. Ghaleb, C. Thomson, A. Al-Dubai, and W. J. Buchanan, "Mitigation mechanisms against the DAO attack on the routing protocol for low power and lossy networks (RPL)," *IEEE Access*, vol. 8, pp. 43665–43675, 2020.
- [54] A. A. Anitha and L. Arockiam, "Ada-IDS: AdaBoost intrusion detection system for ICMPv6 based attacks in Internet of Things," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, pp. 1–8, 2021.
- [55] B. Ghaleb, A. Al-Dubai, E. Ekonomou, M. Qasem, I. Romdhani, and L. Mackenzie, "Addressing the DAO insider attack in RPL's Internet of Things networks," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 68–71, Jan. 2019.
- [56] A. D. Seth, S. Biswas, and A. K. Dhar, "LDES: Detector design for version number attack detection using linear temporal logic based on discrete event system," *Int. J. Inf. Secur.*, vol. 22, pp. 961–985, Mar. 2023.
- [57] A. Verma and V. Ranga, "Security of RPL based 6LoWPAN networks in the Internet of Things: A review," *IEEE Sensors J.*, vol. 20, no. 11, pp. 5666–5690, Jun. 2020.
- [58] M. E. Whitman and H. J. Mattord, *Principles of Information Security*. Boston, MA, USA: Cengage Learning, 2011.
- [59] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks and countermeasures," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 616–644, 1st Quart., 2020.
- [60] A. M. Pasikhani, J. A. Clark, P. Gope, and A. Alshahrani, "Intrusion detection systems in RPL-based 6LoWPAN: A systematic literature review," *IEEE Sensors J.*, vol. 21, no. 11, pp. 12940–12968, Jun. 2021.
- [61] G. Simoglou, G. Violettas, S. Petridou, and L. Mamas, "Intrusion detection systems for RPL security: A comparative analysis," *Comput. Secur.*, vol. 104, pp. 1–56, May 2021.
- [62] P. P. Ioulianou, V. G. Vassilakis, and M. D. Logothetis, "Battery drain denial-of-service attacks and defenses in the Internet of Things," *J. Telecommun. Inf. Technol.*, vol. 2, no. 2019, pp. 37–45, Jul. 2019.
- [63] S. Hristozov, M. Huber, and G. Sigl, "Protecting RESTful IoT devices from battery exhaustion DoS attacks," 2019, *arXiv:1911.08134*.
- [64] N. Ruan and Y. Hori, "DoS attack-tolerant TESLA-based broadcast authentication protocol in Internet of Things," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw.*, Jul. 2012, pp. 60–65.
- [65] P. Porabage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2728–2733.
- [66] R. Hummen, H. Wirtz, J. H. Ziegeldorf, J. Hiller, and K. Wehrle, "Tailoring end-to-end IP security protocols to the Internet of Things," in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2013, pp. 1–10.
- [67] T. Aura, P. Nikander, and J. Leiwo, "DOS-resistant authentication with client puzzles," in *Proc. Int. Workshop Secur. Protocols*. Berlin, Germany: Springer, 2000, pp. 170–177.
- [68] S. Sharma and V. K. Verma, "AIEMLA: Artificial intelligence enabled machine learning approach for routing attacks on Internet of Things," *J. Supercomput.*, vol. 77, pp. 13757–13787, May 2021.
- [69] R. Challa and K. S. Rao, "Resource based attacks security using RPL protocol in Internet of Things," *Ingénierie des Systèmes d'Inf.*, vol. 27, no. 1, pp. 165–170, Feb. 2022.
- [70] N. Abosata, S. Al-Rubaye, and G. Inalhan, "Customised intrusion detection for an industrial IoT heterogeneous network based on machine learning algorithms called FTL-CID," *Sensors*, vol. 23, no. 1, p. 321, Dec. 2022.
- [71] S. Ankam and D. N. S. Reddy, "A mechanism to detecting flooding attacks in quantum enabled cloud-based lowpower and lossy networks," *Theor. Comput. Sci.*, vol. 941, pp. 29–38, Jan. 2023.
- [72] I. S. Alsukayit and A. Singh, "A lightweight scheme for mitigating RPL version number attacks in IoT networks," *IEEE Access*, vol. 10, pp. 111115–111133, 2022.
- [73] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.
- [74] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2010, pp. 406–407.



FARAG AZZEDIN received the B.S. degree in computer science from the University of Victoria, Victoria, BC, Canada, and the M.Sc. and Ph.D. degrees in computer science from the University of Manitoba, Manitoba, Canada. He is currently an Associate Professor with the Information and Computer Science Department and a Researcher with the Interdisciplinary Research Center Affiliate for Intelligent Secure Systems (IRC-ISS), King Fahd University of Petroleum and Minerals (KFUPM). His research interests include trust modeling, cybersecurity, digital twins, network security, and the IoT. He published conference papers and journal articles in these areas.

...