IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 March 2024, accepted 17 April 2024, date of publication 22 April 2024, date of current version 1 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3391918

RESEARCH ARTICLE

Enhancing Security in LLNs Using a Hybrid Trust-Based Intrusion Detection System for RPL

S. REMYA[1], MANU J. PILLAI[2], C. ARJUN[2], SOMULA RAMASUBBAREDDY[3], AND YONGYUN CHO[4]

[1]Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri, Kollam, Kerala 690525, India
[2]Department of Computer Science and Engineering, TKMCE, Kollam, Kerala 691005, India
[3]Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana 500090, India
[4]Department of Information and Communication Engineering, Sunchon National University, Suncheon, Jeollanam-do 57922, South Korea

Corresponding author: Yongyun Cho (yycho@scnu.ac.kr)

This work was supported in part by the Innovative Human Resource Development for Local Intellectualization Program through the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government, Ministry of Science and ICT (MSIT), South Korea, under Grant IITP-2024-2020-0-01489, 50%; and in part by MSIT under the Information Technology Research Center (ITRC) support program supervised by the IITP under Grant RS-2024-00259703, 50%.

ABSTRACT An extensive worldwide network known as the Internet of Things (IoT) links different electronic devices and facilitates easy communication and group work. This interdependency is especially apparent in Low Power and Lossy Networks (LLNs), where resource-constrained devices adhere to specified protocols for effective communication. Such systems frequently use Routing Protocol for LLNs (RPL). Nevertheless, due to its basic simplicity, there are numerous ways to exploit it, thereby compromising network security. It is also difficult to carry out complex computational operations on LLNs due to their resource constraints. A highly developed system called the Trust-Based Intrusion Detection System for RPL (TIDSRPL) is presented in this research study. Complex trust computations are offloaded to the root node by TIDSRPL, which assesses node trust based on network behavior. Reduce the possibility of resource depletion with this strategic transfer that preserves energy, storage, and computational resources at the node level. Comparative analysis with the default RPL Objective Function (OF), MRHOF-RPL, demonstrates TIDSRPL's superior efficacy in detecting and isolating malicious nodes engaged in Sinkhole, Selective forwarding, and Sybil attacks. Notably, TIDSRPL exhibits a 20-35% reduction in average packet loss ratio and attains 33-45% greater energy efficiency compared to MRHOF-RPL, reinforcing its robustness in securing LLN operations.

INDEX TERMS Low power lossy networks, routing protocol, trust-based intrusion detection system, IoT, malicious node detection, energy efficiency.

I. INTRODUCTION

The rapid advancements in mobile computing and wireless communications have made the Internet of Things (IoT) a critical paradigm, driving research and the industrial revolution [1]. IoT is distinguished by an all-encompassing, worldwide network that enables the supervision and manipulation of the physical world by gathering, examining, and handling data acquired by sensors in IoT devices [2]. Remote control and management are made possible by the internet connection of these devices, which are furnished with a variety of sensing and communication interfaces like RFID, GPS, infrared sensors, and wireless networks. Applications such as telehealth, autonomous cars, cyber-physical systems,

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

and surveillance of animals and the environment are made possible by IoT which enables communication between machines and humans [3], [4].

Nevertheless, the incorporation of multiple networks in IoT presents unique security obstacles, such as safeguarding network privacy, authenticating across many networks, controlling access, and ensuring security in routing across different machines. The fundamental objective of IoT is to provide a seamless connection and facilitate communication, identification, management, and control among devices. Although IoT offers substantial advantages in several areas, its widespread adoption requires careful consideration of social and technological concerns. The implementation of IoT relies on a strong system architecture, data processing, and widespread computing and communication technologies. These technologies must handle both physical and cyber interconnectivity [5], [7]. However, security remains a crucial concern, especially in safeguarding private data from unauthorized access and compromise in the vast network of billions of interconnected devices. This emphasizes the necessity for considerable research in IoT security.

Furthermore, the IoT holds the promise of delivering a wide variety of sophisticated services and applications that enhance daily lives and benefit individuals and organizations. However, successful implementation requires addressing key aspects such as system architecture, network security, and social-technological challenges. As the network forms the backbone of IoT, robust routing mechanisms are crucial, and the focus on security is paramount to instill user confidence in the face of increasing interconnectivity among billions of devices. The ongoing and future research in IoT security will ensure the integrity and privacy of networks in this transformative paradigm.

In IoT, devices exhibit heterogeneity and utilize various communication architecture standards, as illustrated in Fig.1. One such standard is 6LoWPAN [8], illustrated in Fig.2, a low-cost communication protocol designed for applications with constrained resources, particularly suitable for Low Power and Lossy Networks (LLN) [9]. LLN encompasses networks where routers and interconnects face limitations in processing power, memory, energy (referring to node battery power), instability, and a low rate of data. LoWPAN, exemplified by wireless sensors, connects the physical environment to real-time applications with characteristics like low bandwidth, small-sized packets, and star and mesh topology. Notably, 6LoWPAN introduces an Adaptation Layer between the Link layer and the Network Layer, responsible for compression, decompression, packet fragmentation, reassembly, and routing. Routing decisions in 6LoWPAN can be classified into Route-over and Mesh-under [10], [11], with the latter handled by the Adaptation Layer. Among various IoT standards, the Internet Engineering Task Force (IETF) introduced the IPv6 Routing Protocol for LLNs (RPL) specifically for resource constrained networks.

Ever since its inception, RPL has been a favorite topic among researchers. Numerous works concentrating on enhancing the efficiency and security of RPL underline its relevance as the de facto IoT routing protocol. However, the existing works aimed at improving the security of RPL leave a lot of questions regarding the compatibility with resource-constrained LLN nodes. As per the definition by IETF, LLNs fall under the category of resource-constrained networks. The LLNs consist of LLN Border Routers (LBRs) and other LLN nodes (i.e., the "things" in IoT). Being resource-constrained in nature, LLN nodes cannot be expected to perform complex computations as it can result in severe performance degradation. This is extremely important as an under-performing LLN would naturally mean that the Quality of Service (QoS) requirements of the higher layer applications are not met.

In this paper, we propose a Trust-Based Intrusion Detection System for RPL (TIDSRPL) that considers the resource constrained nature of LLN nodes. The proposed protocol goes beyond traditional approaches by incorporating a trust-based evaluation of node behavior, providing a more dynamic and adaptive security mechanism. Thus, the proposed approach enhances the security of RPL, and provides a layer of defense against several attacks (Refer Section II for the details of possible attacks in IoT networks). Most importantly, TIDSRPL strategically offloads complex trust computations to the root node, resulting in a notable reduction in computational and repository levels at the individual node level. This resource conservation is a critical aspect that contributes to the sustainability of LLNs, particularly when dealing with resource-constrained IoT devices. The salient contributions of this paper are as follows:

- A novel approach named Trust-Based Intrusion Detection System for RPL (TIDSRPL) that incorporates an adaptive security mechanism to RPL.
- Offloading the complex trust computation tasks to the root node. This ensures better resource utilization at LLN nodes.
- Experimental results demonstrating the better performance of TIDSRPL compared to the existing protocols using Contiki-COOJA simulator.

The effectiveness of TIDSRPL in detecting and mitigating various types of attacks, combined with its resource conservation and efficiency improvements, positions it as a holistic security solution for LLNs. This comprehensive approach contributes to the academic advancement of knowledge in securing IoT networks, especially those characterized by low-power and lossy communication environments. The subsequent sections are organized as follows. Section II provides the necessary background. Section III reviews the existing literature. Section IV meticulously sketches the proposed methodology, elucidating the strategies devised for enhancing RPL's security framework. Section V systematically presents the results derived from the experimentation, furnishing a

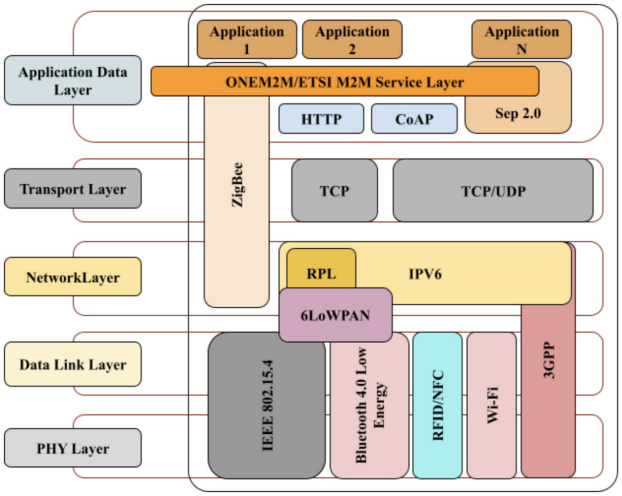

FIGURE 1. Proposed network architecture.

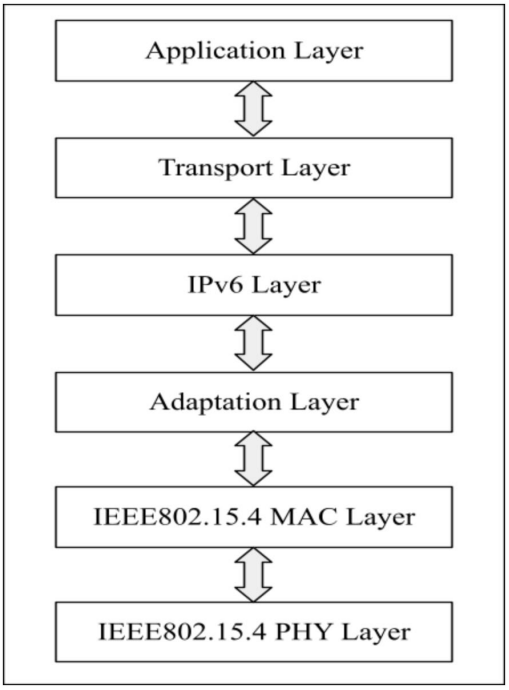

FIGURE 2. 6LoWPAN protocol architecture.

detailed analysis of the outcomes. The concluding sections VI and VII encapsulate final remarks, summarizing the key findings and insights obtained throughout the study, while also delineating potential avenues for future research endeavors.

II. IPV6 ROUTING PROTOCOL FOR LOW POWER AND LOSSY NETWORKS (RPL)

Routing Protocol for LLNs(RPL) [12], intended for resource-confined networks within IoT, operates as a Distance Vector routing protocol determining both distance and direction for network links employed in LLN, such as Radio Networks. If predefined topology is absent then RPL efficiently discovers links and selects nodes. It serves as a source routing protocol, allowing transmitters to specify complete or partial routes for packet transmission. RPL constructs Directed Acyclic Graphs (DAGs), organizing nodes hierarchically into Destination Oriented DAGs (DODAGs) [13]. Objective functions guide RPL in optimizing topology on the basis of predefined goals like energy consumption and hop count. RPL employs control messages such as DODAG Information Solicitation (DIS), DODAG Information Object (DIO), and Destination Advertisement Object (DAO) [14], [15]. Each DODAG has a unique identifier and utilizes values like RPLInstanceID, DODAGVersionNumber, DODAGID, and Rank for topology maintenance. The protocol supports multiple instances with distinct objective functions, and the routing decisions can be Route-over or Mesh-under. The MRHOF [16] minimizes metrics, utilizing latency to mitigate network-supporting metrics like hop count, latency, or ETX. RPL holds significance in establishing efficient and secure communication within IoT networks.

The diagram designated as Fig3 illustrates the hierarchical structure of a DODAG using a ranking mechanism. The border router connected to the network is assigned a rank value of 1. Afterwards, the RPL network is assigned a rank according to the particular network. The rank assignment can be used to identify the loops within the network. Every node in the DODAG is also assigned with candidate neighbors. The candidate nodes are subsets of nodes that can be reached by link-local multicast [17], [18]. The candidate neighbor set is restricted to the parent group, and the desired parent must be a member of the parent set. The assignment of a parent to a node is chosen by evaluating both the rank and energy transmission criteria. A node is not allowed to assign a rank that is lower than any of the members in its parent set. Nevertheless, there are specific instances in which the immediate predecessor of a node does not hold the lowest position among the nodes in the parent set. However, there exists a commonly employed method for determining the parent of a node. Commence by choosing the node with the lowest rating that is nearest to the border router. In addition, throughout each update, the node continuously calculates its rank and then performs actions based on the updated rank. Each node has a rank that increases gradually as it gets further away from the border router.

A. DODAG CONSTRUCTION, MAINTENANCE, AND IOT ATTACKS

DODAG construction involves the broadcast of DIO control messages, establishing routes from the root to the clients, and the unicast of DAO messages in the upward direction, constructing routes from clients to the root. It contains essential information such as DODAGID, rank information for node positioning, and Objective Function (OF) [19]. OF optimizes routes in the RPL instant, during the selection process, determines the translation of nodes, and also defines how a parent node is selected for a particular node. Objective

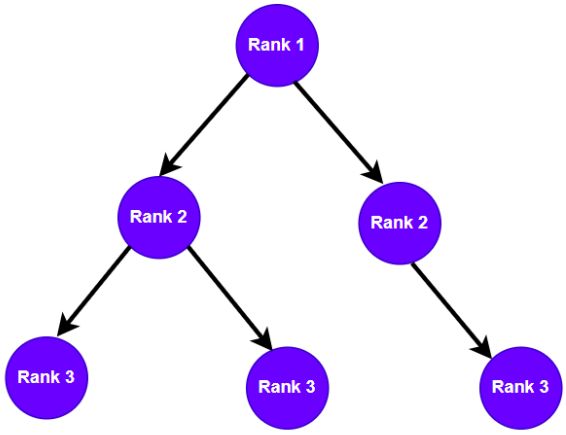

FIGURE 3. DODAG.

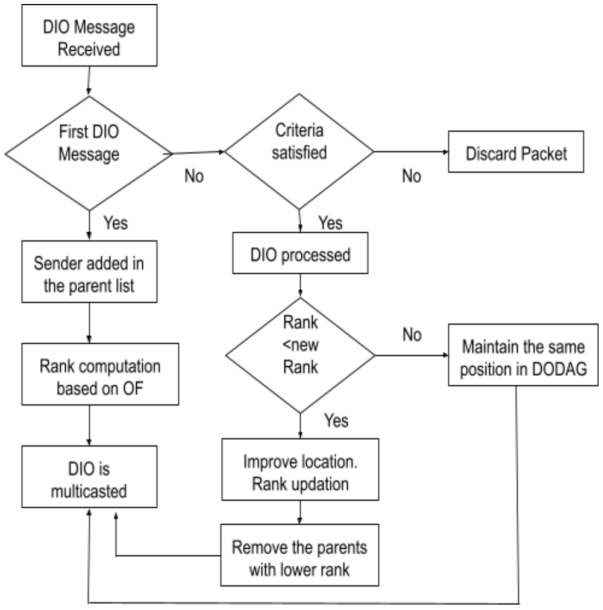

FIGURE 4. DODAG construction and maintenance.

function is classified into two types such as OF_0(Objective function zero) and MRHOF [20].

Nodes receiving DIO messages decide to join or not based on their willingness and membership status. If a node wishes to join, it is added to the address of the parent list of the sender, then the rank is computed according to the OF and forwards the updated rank in the DIO. If already a member, processing of additional DIO messages can involve dropping them, continuing in the DODAG, or changing location based on rank comparisons. DODAG maintenance ensures the avoidance of loops, with rank changes leading to node removal from the parent list. Grounded DODAGs offer connectivity for application goals, while Floated DODAGs provide routes but may not satisfy the goal, useful for maintenance scenarios. The workflow of the DODAG construction is shown in Fig.4.

Network security attacks pose a threat to IoT networks, with unauthorized actions aimed at destroying, modifying, or stealing sensitive data. As enterprises increasingly embrace mobile device access, networks become vulnerable to data theft or destruction. In wireless mobile networks, route information transmission during route establishment is susceptible to attacks by malicious nodes introducing false information, leading to various attacks such as routing table overflow, fabricated route advertisements, and false route error messages. These attacks can compromise network security and disrupt the routing process. Attacks are classified into Topological attacks, affecting the RPL network topology, and Performance attacks, where malicious nodes deceptively deplete resources, including Blackhole, Selective Forward, Greyhole, and Wormhole attacks. Understanding and addressing these attacks is crucial for ensuring the security and performance of IoT networks.

III. RELATED WORKS

In IoT networks, the vulnerability of compromised sensor nodes poses a significant threat to data routing integrity, leading to potential security attacks such as transmitting incorrect control information, dropping packets, injecting false routing data during aggregation, and hindering composite data forwarding. Recent years have seen a notable increase in interest in the research community due to the resilience of RPL to several types of attacks. Numerous research projects have been undertaken to provide efficient ways to reduce RPL attacks and improve network security. This section provides an overview of several significant techniques that have been offered in the literature, highlighting their salient characteristics, benefits, and drawbacks.

RPL is commonly used for packet transmission in IoT devices, but its susceptibility to security attacks necessitates effective defense mechanisms [21], [22], [23], [24]. Various secure protocol-based defense mechanisms have been proposed, including cryptographic solutions like Version Number and Rank Authentication (VeRA) [25], Secure-RPL (SRPL) [26], Trust Anchor Interconnection Loop (TRAIL) [27], and others. These mechanisms address illegitimate rank changes, and topological inconsistencies, and provide defenses against attacks like Sinkhole, Blackhole, Selective forwarding, and Rank attacks [28], [29]. Some of these have been reviewed in this section based on their classifications.

Trust-based mechanisms, such as Trusted Computing Architecture (TCA) [30], Secure Parent Selection, Lightweight Trust-Aware RPL, SecTrust-RPL [31], and Trust-based Security System (TIDS) [32], focus on evaluating trustworthiness and detecting malicious nodes through factors like rank, trust values, and geographical parameters. These mechanisms aim to enhance security by considering trust metrics, though challenges like energy consumption and vulnerability to specific attacks persist. SVELTE, a real-time IDS, employs anomaly-based detection for attacks in 6LoWPAN, while RIDES combines signature and anomaly-based

methods for robust intrusion detection [33]. Furthermore, heartbeat detection systems, exemplified by the Lightweight Heartbeat Protocol (LHP) [34], utilize periodic requests to identify blackhole attacks, with potential improvements involving the use of cryptographic methods or alternative protocols like User Datagram Protocol (UDP). Overall, the literature underscores the need for comprehensive security solutions in RPL-based IoT networks, addressing challenges in energy consumption, attack detection delays, and scalability [35], [36], [37].

To combat RPL attacks, the designers of [48] unveiled the Metric-based RPL Trustworthiness Scheme (MRTS). To assess the behavior of nearby nodes, MRTS makes use of both direct observations and indirect suggestions. Nodes can choose preferred parents based on link quality, energy availability, and trust value by computing the Extended RPL Node Trustworthiness (ERNT). Results showed that despite retaining low energy usage and high packet delivery ratios, MRTS effectively eliminated hostile nodes. But one important MRTS requirement is that nodes must run in promiscuous mode to watch neighbor behavior.

As described in [39], the Secure-RPL (SRPL) protocol aims to prevent malicious nodes by constructing false topologies and manipulating rank values. SRPL uses hash chain authentication for node authentication during topology changes and introduces a threshold technique to limit rank changes. The simulations showed that rank attacks may be successfully defended against, despite sending more RPL control signals.

A noteworthy addition to previous research is the Secure RPL Routing Protocol (SRPL-RP), which was proposed in [40]. It incorporates mitigation techniques for both rank and version assaults. In comparison to other methods, SRPL-RP achieves better detection and mitigation accuracy by identifying and isolating malicious nodes using threshold-based algorithms. However, similar to other approaches, it requires nodes to monitor network traffic, potentially leading to increased energy consumption and storage requirements.

By utilizing a trust-based method to identify attackers, the SecTrust-RPL protocol, which was presented by Airehrour et al. [41], improves RPL security. SecTrust-RPL uses less energy than conventional RPL and provides a strong defense against rank and Sybil attacks by operating nodes in promiscuous mode and analyzing direct and recommended trust values. This method, while effective, has the usual flaw of needing constant monitoring and storage space.

Iuchi et al. presented a Secure Parent Node Selection Scheme in [23] to let nodes select safe parents and ward off attackers by looking at neighboring node ranks. Simulation studies showed that, in comparison to normal RPL, the technique may both outwit attackers and enhance network performance. It does, however, require nodes to actively monitor network traffic, just as other trust-based techniques.

This review reveals diverse defense mechanisms for securing RPL-based IoT networks against various security threats. While cryptographic solutions offer robust protection, their resource-intensive nature may limit their applicability in resource-constrained IoT devices. Trust-based mechanisms introduce lightweight alternatives, assessing trust values and geographical parameters. IDS demonstrates effectiveness in detecting attacks, though challenges related to energy consumption and delayed detection persist. Heartbeat detection systems provide a means of identifying blackhole attacks, with potential enhancements through cryptographic methods or protocol modifications [42], [43]. A comprehensive security approach may involve integrating multiple mechanisms to address the evolving landscape of security challenges in IoT networks.

Thus several research have improved RPL security and reduced assaults by using several strategies like authentication protocols, threshold approaches, and trust-based procedures. Although these approaches show promise in preventing RPL assaults, they frequently necessitate that nodes remain in promiscuous mode for ongoing observation, which leads to inefficient use of energy and storage issues. Furthermore, a lot of suggested approaches don't have thorough implementations. To maximize detection performance and minimize resource overheads, future research areas can examine hybrid techniques that integrate intrusion detection systems (IDS) with protocol-based mechanisms. More effective and reliable ways to protect RPL-enabled networks from changing threats may be made possible by such integration. The summary of the state of the art is shown in Table 1.

IV. PROPOSED METHODOLOGY

A TIDS is introduced in this section, incorporating Jøsang et al. Subjective Logic [44] and a Heartbeat Monitoring System [45]for trust propagation and information gathering. Noteworthy is the focus on mitigating the resource constraints typically associated with IoT devices, as several other IDS systems are identified to impose excessive energy, storage, or memory demands. The other trust-based approaches in the existing literature consider a set of attacks that is different from the set considered in this paper. In this research work, we have considered three attack types together - Sinkhole attack, Sybil attack, and selective forwarding attack. Since there are no existing works that consider these three attack types, we have compared them with standard RPL. Also, such a comparison helps to get insights on the effects of these attacks on an RPL topology, if left unaddressed. The proposed methodology employs both a distributed and centralized approach for detection. The distributed facet involves each network node observing its neighboring nodes and relaying this information to the central or root node, which serves as the centralized agent. The root node, endowed with comparatively greater resources, analyzes the observations, computes trust values, and broadcasts these values in the network. The root node corresponds to the LLN Border Router (LBR). It acts as a Gateway/Edge Router for the

TABLE 1. Literature review of RPL-based IoT security mechanisms.

Paper Title	Methodology	Key Findings	Strengths	Weaknesses
VeRA	Cryptographic solutions	Provides defense against illegitimate rank changes and topological inconsistencies mitigates attacks like Sinkhole,Blackhole, Selective forwarding, and Rank attacks	Utilizes cryptographic methods for enhanced security	Requires computational overhead and potential vulnerabilities in key management
Secure-RPL (SRPL)	Hash chain authentication; threshold technique	Successfully defends against rank attacks despite increased RPL control signals; limits rank changes	Implements authentication techniques to prevent malicious nodes; achieves better detection and mitigation accuracy	Requires nodes to monitor network traffic, potentially increasing energy consumption and storage requirements
Secure RPL Routing Protocol (SRPL-RP)	Mitigation techniques for rank and version assaults; threshold-based algorithms	Achieves better detection and mitigation accuracy; identifies and isolates malicious nodes using threshold-based algorithms	Provides robust defense mechanisms against rank and version attacks	Requires nodes to monitor network traffic, potentially leading to increased energy consumption and storage requirements
SecTrust-RPL	Trust-based method; operates nodes in promiscuous mode; analyzes direct and recommended trust values	Provides strong defense against rank and Sybil attacks; uses less energy than conventional RPL	Offers improved security by utilizing trust-based approach	Requires constant monitoring and storage space
Metric-based RPL Trustworthiness Scheme (MRTS)	Direct observations and indirect suggestions; Extended RPL Node Trustworthiness (ERNT) computation	Effectively eliminates hostile nodes while maintaining low energy usage and high packet delivery ratios	Offers effective defense against hostile nodes through comprehensive trust evaluation	Requires nodes to run in promiscuous mode to watch neighbor behavior
Secure Parent Node Selection Scheme	Trust-based method; nodes select safe parents based on neighboring node ranks	Enhances network performance and defends against attackers; outperforms normal RPL	Improves network resilience by selecting safe parent nodes	Requires nodes to actively monitor network traffic
SVELTE	Real-time IDS; anomaly-based detection	Anomaly-based detection for attacks in 6LoWPAN	Provides real-time detection of intrusions	May require significant computational resources for real-time detection
RIDES	Signature and anomaly-based methods	Combines signature and anomaly-based methods for robust intrusion detection	Offers robust intrusion detection capabilities	May require significant computational resources for signature and anomaly-based detection
Lightweight Heartbeat Protocol (LHP)	Heartbeat detection; periodic requests	Utilizes periodic requests to identify blackhole attacks; potential improvements with cryptographic methods or alternative protocols like UDP	Offers lightweight detection of blackhole attacks	May have limitations in scalability for large-scale networks

LLNs. Hence, the computational resources available for an LBR will be much greater than that of the normal LLN nodes. This design minimizes computational complexities at resource-constrained network nodes. The trust matrix, derived from these trust status values, plays a crucial role in routing decisions. Malicious nodes identified at the centralized root node are broadcast to other nodes, prompting the nodes to segregate the identified malignant entities from the network and exclude them from routing considerations.

The trust-based system as shown in Fig.5 incorporates a reputation system where each node's actions are observed and evaluated by proximate nodes, determining adherence to the RPL protocol. In a single trust scenario, the trustworthiness value of the nodes willing to provide a particular service is calculated based on the policy of the system under consideration (Eg. Weighted Sum). Once information has been gathered from the system about a set of selected parameters that need to be aggregated to arrive at a single trust value. In the multi-trust scenario, multiple attributes are identified and defined for trust computation. Challenges, such as potential deception by malicious nodes within the reputation system and the need to filter out deceitful messages,

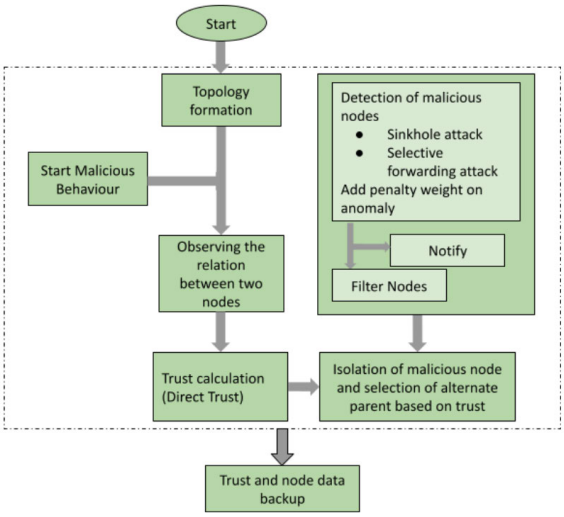

FIGURE 5. Trust based system.

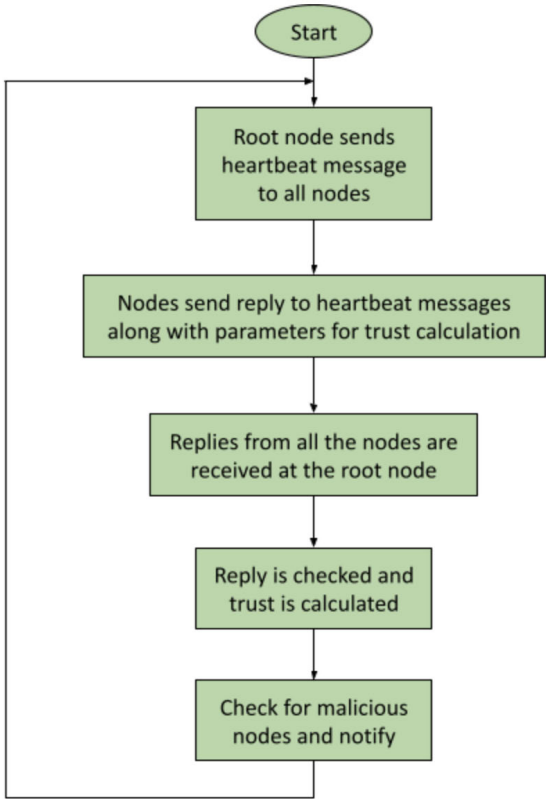

FIGURE 6. Heart beat based system.

are acknowledged. The trust computation process involves dimensions like trust composition, propagation, aggregation, update, and formation. Trust composition encompasses QoS trust [46] and Social trust, reflecting the components considered in trust computation. Trust propagation involves either distributed or centralized approaches for distributing trust observations among peers, while trust aggregation focuses on collecting and combining these observations. Trust update methods include event-trust update and time-trust update, determining how and when trust is updated. Trust formation addresses the amalgamation of various trust properties, distinguishing between single-trust and multi-trust scenarios. Additionally, a Heartbeat Monitoring System as shown in Fig.6is introduced, utilizing periodic hello requests from the root to all other nodes to maintain an up-to-date network picture and identify node reachability. This combined approach aims to enhance the RPL security in IoT networks while mitigating resource overhead [47].

A. JØSANG'S SUBJECTIVE LOGIC

The trust-based strategy utilizes Subjective Logic [48], an advanced methodology that admits not only trust and distrust but also integrates uncertainty. Three variables, namely belief (b), disbelief (d), and uncertainty (u), are introduced to represent uncertainty based on positive(pos) and negative(neg) trust valuations. The equations for these variables are given as follows as shown in Equations 1to 3, where c is a constant (typically 1 or 2).

$$belief = \frac{pos}{pos + neg + c} \tag{1}$$

$$disbelief = \frac{neg}{pos + neg + c} \tag{2}$$

$$uncertainity = \frac{c}{pos + neg + c} \tag{3}$$

This technique relies on transceivers supporting idle listening mode for 1-hop neighbors' data traffic. The root node calculates these values based on observations from network nodes, rating neighbors positively for adherence to the RPL protocol and negatively for non-compliance. Subjective Logic proves effective as it provides variables for both trust and uncertainty and allows one to make decisions regarding the node's trustworthiness. It also offers flexibility for accommodating various attacks, making it a suitable choice for IDS.

To propagate trust computation information to the root node and distribute decisions to network nodes, the proposed solution uses the heartbeat mechanism. The neighborhood data required for trust computation is gathered by the root node, which sends out heartbeat requests to every node in the area. After that, the root node calculates trust and runs the attack detection algorithm. Periodically, heartbeats are sent; if a node does not react within a predetermined threshold, additional detection steps are initiated in case of a probable black hole or selective forward attack. In addition, the system broadcasts the malicious node list via the heartbeat mechanism, which is then restricted by other nodes during routing. To prevent specific attacks like selective forwarding, sinkhole attacks, and Sybil attacks, the system implements security measures including making sure all nodes forward incoming data packets and looking for rank flaws to avoid selective forwarding. Attackers who use cloned attackers are identified by counting the number of instances of each node and taking geographic location into account.

To identify and mitigate multiple types of attacks in an RPL-based IoT network, the proposed IDS uses a variety of methods. A centralised node (root or 6BR) receives the observations that each node makes of its neighbours as part of the detection process. Specific assaults like Sybil and selective forward attacks are catered for by the detection algorithms. Network topology analysis prevents sinkhole attacks, in which a hacker modifies the route graph of the network. To reduce false positives, the detection algorithm uses parameters like FaultThreshold to look for differences in the rank between nodes and their parents.

Each node determines and sends to the root node regularly its packet forwarding ratio in the event of a selective forward attack, in which malicious nodes drop packets. A decrease in this ratio indicates potential packet drops, leading to trust value adjustments. Sybil attacks, where a node impersonates another, are detected by considering the geographical location of nodes. Nodes report their location and address to the root node, which checks for inconsistencies in the reported location for nodes with the same address. The proposed heartbeat process involves the root node sending requests to all nodes, expecting responses with information collected for trust calculation. Failure to respond prompts further detection, and if a node is marked malicious, the malicious list is broadcasted using the heartbeat request message.

These algorithms collectively form the foundation of the intrusion detection and prevention framework for RPL-based IoT networks, addressing specific attack scenarios and ensuring the network's security and integrity.

Algorithm 1 Rank Inconsistency Check

```
Data: Result: p, n; NodeList - list of nodes in the
      network at root node
foreach Node in NodeList do
    // For each node, compare the
       rank with their parent for
       checking inconsistencies.
    if Node.rank + MinHopRankIncrease <
     Node.parent.rank then
        Node.n = Node.n + 1;
    else
        Node.p = Node.p + 1;
```

Algorithm 1 handles sinkhole detection, aims to identify potential sinkhole attacks, where an attacker manipulates the network's routing graph by advertising a false rank. The algorithm analyzes the network's topology by checking for inconsistencies in the rank between nodes and their parents. The parameter FaultThreshold represents a global parameter representing the minimum number of consecutive inconsistencies required before taking action. It helps minimize false positives. MinHopRankIncrease is a parameter specific to RPL, representing the minimum rank increase between any host and its parent.

Algorithm 2 addresses the correction of inconsistent information identified in the network. It ensures that the reported rank information for nodes is consistent and minimizes false positives. It distinguishes between valid and invalid inconsistencies using the number of reported faulty ranks and the difference between reported ranks. The main parameter is FaultThreshold which represents the threshold determining when a node is classified as faulty based on reported disagreements.RankDifferenceThreshold is the threshold for the difference between reported ranks to distinguish between valid and invalid inconsistencies.

Algorithm 2 Correcting Rank Inconsistency

```
Data: Result: p, n; NodeList - list of nodes in the
      network at root node
foreach Node in NodeList do
    foreach Neighbour in Node.neighbours do
        // For checking
           inconsistencies, if the
           difference between the ranks
           is greater than 20% of the
           average rank.
        diff = Neighbour.rank −
         Node.neighbour.rank;
        average = (Neighbour.rank +
         Node.neighbour.rank) / 2;
        if diff < average * 0.2 then
            Node.n = Node.n + 1;
        else
            Node.p = Node.p + 1;
```

Algorithm 3 deals with selective forward detection, aims to identify nodes engaged in selective forwarding attacks, dropping some of the packets they receive. The algorithm utilizes the packet forwarding ratio, calculated by nodes and periodically reported to the root node, to detect changes indicative of selective forwarding attacks.

Algorithm 3 Selective Forward Detection

```
Data: Result: p, n; NodeList - list of nodes in the
      network at root node
foreach Node in NodeList do
    Node.FRnew = Node.forwarded / Node.received;
    // Selective Forward detection
    if (Node.FRnew < 1 and
    Node.FRnew < Node.FRold) then
        Node.n = Node.n + 1;
    else
        Node.p = Node.p + 1;
```

The Sybil detection algorithm 4 focuses on identifying nodes involved in Sybil attacks, where a malicious node

impersonates other nodes to gain control over the network. The algorithm checks the inconsistencies in the reported locations of nodes with the same address. The heartbeat process algorithm 5 outlines the procedure for the root node sending heartbeat requests to all nodes, expecting responses with information necessary for trust calculation. It plays a crucial role in maintaining an up-to-date network picture and detecting ongoing attacks. The parameters, threshold is the maximum time allowed for a node to respond to a heartbeat request before initiating further detection.

Algorithm 4 Sybil Detection

```
Data: Result: p, n; NodeList - list of nodes in the
      network at root node
foreach Node in NodeList do
    Node.FRnew = Node.forwarded / Node.received;
    // Sybil detection
    if (Node[i].x ≠ Node[j].parent.x or
    Node[i].y ≠ Node[j].parent.y) then
        Node.n = Node.n + 1;
    else
        Node.p = Node.p + 1;
```

Algorithm 5 Heartbeat Process

```
Data: Result: MaliciousList; NodeList - list of nodes
      in the network at root node; HBList - list of
      nodes for which heartbeat request is sent
foreach Node in NodeList do
    sendHBRequest(Node);
    HBList.add(Node);
foreach Response in Responses do
    HBList.removeItemFromList(Response.node);
if no response received from node then
    foreach Node in HBList do
        raiseAlert("Node.id not reachable, possible
        attack!");
```

Trust is calculated based on the subjective logic explained above. For evaluating the opinion the input parameters positive (pos) and negative (neg) which we get from the detection algorithms. Disbelief (d) is calculated based on Equation 2 and it is rated whether malicious or not based on the rating table in Table 2.

TABLE 2. Rating table.

Belief (b)	Disbelief (d)	Uncertainty (u)	Value
> 0.5	< 0.5	< 0.5	Trusted
< 0.5	> 0.5	< 0.5	Marked malicious

If the disbelief is greater than a threshold of 0.5 the node is marked as malicious. The node status received at the network nodes as malicious will not be considered for routing and also the DIO is ignored. The pseudo-code for rate node status notification and Isolation process are shown in Algorithm 6 and Algorithm 7.

Algorithm 6 Rate Node Status and Notify

```
Data: Result: MaliciousList; NodeList - list of nodes
      in the network at root node
MaliciousList = [];
foreach Node in NodeList do
    if d > 0.5 then
        Node.status = malicious;
    else
        Node.status = trusted;
foreach N in NodeList do
    if N.status = malicious then
        MaliciousList.add(N);
if MaliciousList not empty then
    Notify all the nodes by broadcasting the malicious
    list;
```

Algorithm 7 Isolate Malicious Node at Nodes

```
Data: Result: Isolate/Consider for routing;
      MaliciousList - List of malicious nodes
foreach Parent in MaliciousList do
    // During Parent Selection
    if Parent in MaliciousList then
        ignore the parent;
    else
        Consider for routing;
foreach DIO.message from malicious node do
    // Ignore DIO messages from
       malicious node
    if DIO.node.ip in MaliciousList then
        ignore the DIO;
```

V. RESULTS AND DISCUSSION

A. EXPERIMENTATION AND SIMULATION

This section stipulates an outline of the configuration and execution of the Intrusion Detection System (IDS). Firstly, the process of gathering the measurements will be outlined. Subsequently, the entire IDS is put into action. The contiki OS operating system is used for the implementation and experimentation. It is a freely available operating system designed specifically for IoT. It is configured for miniature devices with limited memory capacity. It functions as a powerful tool for building wireless networks and is a great way to get these devices connected to the internet.

Contiki is capable of replicating the behaviour of a physical device due to an integrated simulation tool called cooja. Contiki is compatible with 6LoWPAN networks, RPL, power awareness, as well as complete IP networking. With this tool, developers may test and simulate networks of any size before deploying them on a real hardware platform. Its goal is to find and confirm issues with functionality.

The wide range of useful applications for Cooja and Contiki OS across numerous industries shows how adaptable these platforms are in addressing actual IoT challenges.Contiki OS is utilised to enable efficient monitoring and control systems in industrial settings by enabling smooth communication across restricted equipment. Furthermore, the utilisation of Contiki OS and Cooja has demonstrated their value for smart city applications. The key reason for this is because the simulation tool can enhance the performance of sensor networks and assess the effectiveness of infrastructure based on the Internet of Things (IoT). Contiki's reduced resource needs are helpful for the healthcare industry and make it suitable for wearable technology and health monitoring applications. This feature makes it easier to come up with affordable and energy-efficient alternatives. Additionally, the applications of Cooja's simulation capabilities and Contiki OS's flexibility have improved resource management and the adoption of precision agricultural techniques. Practical uses of Cooja and Contiki OS demonstrate the importance of these technologies in fostering innovation in several industries. They thereby improve IoT technology's ability to deal with real-world issues.

For the creation, implementation, and assessment of Internet of Things (IoT) applications in resource-constrained environments, Cooja and Contiki OS together provide a powerful toolkit. The main target audience for Contiki OS, an open-source operating system, is IoT devices with limited RAM and computing capability. Portability and efficiency are key design features. Efficiency was a primary consideration in the design of this product, which makes it an excellent option for devices used in extensive linear networks. Power monitoring, broad IP networking, RPL for routing, and compatibility with 6LoWPAN are just a few of the connectivity features that Contiki OS offers that are appropriate for IoT. Integration into different IoT contexts is made easier by these features.

Contiki OS serves as an extremely efficient Internet access solution for small devices because of how efficiently it utilises its resources. Developers can intentionally choose and include only the components required for a particular application, enhancing memory and energy efficiency, thanks to the system's modular architecture. Developing Internet of Things applications is made simpler with Contiki OS's vast library and pre-built protocols.

The simulation tool Cooja is an integral part of Contiki OS and is a crucial part of the development life cycle. Engineers can use this tool to simulate and assess networks of different sizes and complexity before implementing IoT networks on real devices. The simulation environment makes it possible to identify potential issues, assess the effectiveness of protocols, and verify overall system operation. The ability to replicate communication in a real-world environment is one of Cooja's simulation capabilities that enables developers to watch and analyse node behaviour inside the network. Because it helps predict issues with scalability, resource allocation, and network dynamics, this tool is very useful for conducting studies on a large scale.

The integration of Contiki OS and Cooja provides a full solution for IoT developers, merging a resource-efficient operating system with a dynamic simulation environment. The integration of these two components improves the progression and examination procedures, finally resulting in the production of resilient and effective IoT applications, as exemplified in the Trust-Based Intrusion Detection System for RPL (TIDSRPL).

B. SYSTEM DEVELOPMENT

The development of TIDS involves multiple sequential stages. Prioritize the initial step of devising a comprehensive strategy for the system's functionality. The IDS gathers trust parameters from every node and transmits them to the root node. The primary node will thereafter compute the trustworthiness of each node and, relying on the trust metrics, it determines whether a node is malicious or not. This makes the intended system a hybrid approach. Since all the trust-related computation and heartbeat system is implemented in the root node and acts as a centralized module. Other network nodes collect the information from the neighborhood for trust computation and send it to the root node along with the heartbeat reply, thus the nodes act as a distributed module.

The simulation is done on a virtual machine with 1GB RAM with Ubuntu 14. Since the network is a Low-power Lossy Network the transmission ratio is set to 100% and the reception ratio is set from 30% to 100%. Thus they produce a loss at the reception layer. The transmission range is 50m and the interference range is 55m. The simulation environment is set as shown in Table3 and the simulation of nodes is done on cooja as shown in the simulation window in Fig.7. Node 1 is the root node and nodes 2 to 30 are client or normal nodes. The simulation is done in such a way that 10% of the total nodes are malicious. So out of 30 nodes 28, 29, and 30 are malicious nodes with transmission and reception ratio of 100%. Simulation for each test case is conducted for 60 minutes. To validate our mechanism we considered attack detection time, packet loss ratio, and power consumption rate.

1) ATTACK IMPLEMENTATION

In implementing the sinkhole attack, the rank calculation method within the contiki/core/net/rpl/rpl-mrhof.c file is modified. This entails introducing a flag that, after 5 to 10 seconds of normal behavior, triggers the attack. When

TABLE 3. Parameters and their values.

Parameter	Value
Simulation Tool	Contiki 3.0
Simulation Area	90 m × 90 m
Nodes	30
Malicious Nodes	3 (28, 29, 30)
Malicious: Legitimate	1:10
Reciever Ratio	30-100%
Transmission Ratio	100%
Transmission Range	50 m
INT Range	55 m
Routing Protocols	MRHOF, Trust Based Routing
Network Protocol	IP based
Simulation Time	60 minutes

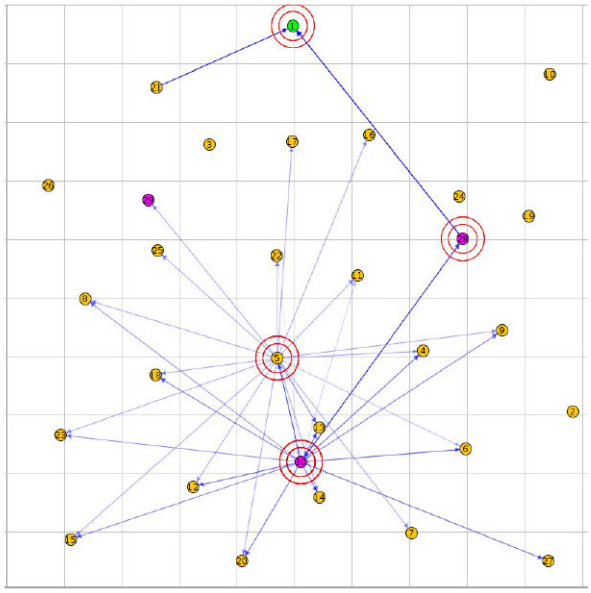

FIGURE 7. Topology in cooja simulator.

active, the rank is manipulated to be lower than all nodes, excluding the border router, compelling other nodes to route their traffic through the malicious node. The attack is strategically designed to disrupt normal traffic patterns while maintaining an appearance of legitimacy. Meanwhile, the selective forward attack is orchestrated in the contiki/core/net/ipv6/uip6.c file. This attack is executed by dropping every data packet after a brief period of normal operation, causing a targeted disruption in the forwarding of data packets not intended for the perpetrating node.

In the context of a Sybil attack, the malicious one assumes the identity of a nearby authentic node and communicates with others in the network, leveraging false identities. Implementation involves using multiple logical entities on the same physical node to interact under different guises. Detection strategies include monitoring the number of instances of each identity and verifying geographical locations. To counteract Sybil attacks, it is essential to deploy mechanisms that can identify cloned identities and distinguish between genuine and malicious nodes. These manipulations and attacks highlight the importance of robust intrusion detection systems and countermeasures in securing IoT networks against various threats. Threat modeling is the systematic process of identifying the threats to a set of identified sensitive assets, and vulnerabilities that make the threats a necessary concern. The threat modeling is very essential in defining security requirements that mitigate the threats and the development of the necessary architecture to ensure right-size security requirements for securing devices, the network, and data in an IoT system and its use cases. The pseudo-codes for the implementation of the sinkhole attack and Sybil attack are exhibited in Algorithm 8 and Algorithm 9 respectively.

Algorithm 8 Pseudocode for Sinkhole Attack

```
Data: Inf_rank, baserank, rankincrease, attackflag, p
if Inf_rank − baserank < rankincrease then
    // The maximum rank has reached.
    newrank = Inf_rank;
else
    // Determine the rank by
       exploiting the updated rank
       information obtained from DIO
       or preserved elsewhere.
    if attackflag then
        // Calculate a new rank that is
           higher than the border
           router (256) but less than
           all nodes if the sinkhole
           attack is activated.
        newrank = (int)(p →rank * 7 / 20);
        if newrank < 256 then
            newrank = 256 + 20;
    else
        // In the absence of a sinkhole
           attack, function as a
           regular node.
        newrank = baserank + rankincrease;
```

C. RESULTS

This section presents the outcomes achieved from the implementation discussed in section IV. We conducted a comparative analysis between our methodology and MRHOF. MRHOFRPL uses an optimisation function (OF) to choose routes that minimize a specific measure. It operates using the additive metrics that are advertised in the RPL DIO messages. The RPL protocol utilizes a default routing mechanism. However, it lacks support for assault detection. To make comparisons, we solely looked at the packet loss ratio and energy usage. The results produced demonstrate superior outcomes to the chosen parameters.

Algorithm 9 Pseudocode for Implementing Selective Forward Attack

```
Data: selectiveforwardattackflag, UIP, IPBUF
if selectiveforwardattackflag then
    // Navigate to the section
       dedicated to dropped packets
       in the event of an attack
    goto drop;
else
    // The packet is forwarded
    UIP IPBUF → ttl = UIP IPBUF → ttl − 1;
    goto send;
```

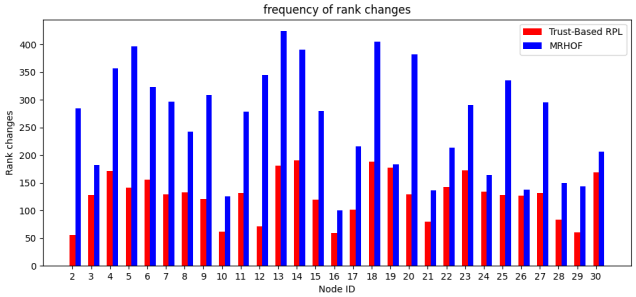

FIGURE 8. Frequency of rank changes under sinkhole attack.

1) FREQUENCY OF RANK CHANGES

Nodes ensure the prevention of routing loops by consistently maintaining a low rank. As a result, the rank of a child node changes when it aligns with its preferred parent that has a lower rank. RPL allows high-rank attackers to broadcast their presence to neighboring nodes to lure and deceive them. The neighboring nodes then switch parents to the malicious node. This creates segmented RPL networks separately from the sink node. Fig. 8 compares the changes in the rank frequency of nodes in MRHOF-RPL and TIDS. The MRHOF-RPL algorithm is more vulnerable to node rank changes than TIDS, making it vulnerable to Rank attacks. Unlike MRHOF, TIDS had negligible node rank variations during the simulation.

2) PACKET LOSS RATIO

While the TIDS is capable of identifying and isolating malicious nodes, specifically Rank assaults, the TIDS must not excessively impact the network performance. The purpose is to quantify the packet loss rate to evaluate the effectiveness of TIDS compared to MRHOF-RPL in maintaining acceptable network performance while also protecting against Rank attacks. Fig 9 demonstrates that the average packet loss rate for TIDS was 45.58%, while MRHOF-RPL had 66%. Network segmentation is the main cause of MRHOF-RPL packet loss. Therefore, TIDS may defend against Rank attacks.

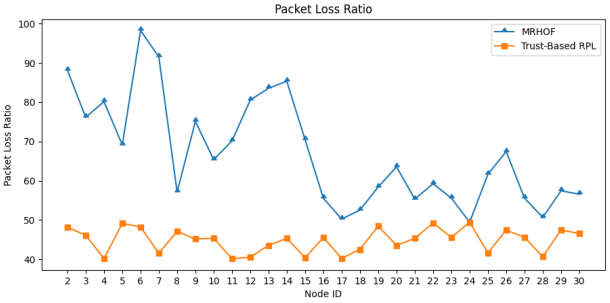

FIGURE 9. Packet loss ratio under sinkhole attack.

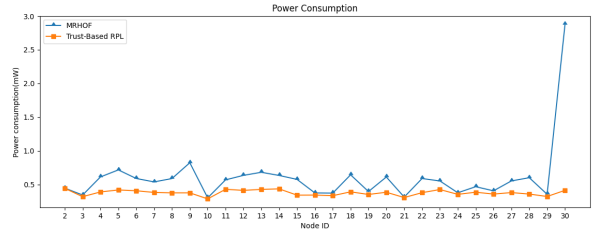

FIGURE 10. Power consumption under sinkhole attack.

3) POWER CONSUMPTION

The MRHOF-RPL and the TIDS power consumption rates are compared and shown in Fig.10. From the result is ts very clear that the average rate of MRHOF is slightly higher than that of TIDS. Also, the higher consumption rate is for node 30, under MRHOF as it attacks all the nodes (2, 4, 5, 6, 7, 20, etc). These nodes are not in range of the root node and lead to higher consumption of power which gets attacked, and has higher data loss. The average power consumption in TIDS and MRHOF is 0.28 and 0.61 respectively, where TIDS has 33% less power consumption than MRHOF-RPL under Sinkhole attack.

D. SELECTIVE FORWARD AND SINKHOLE ATTACK

Since the attack can also occur as a combination of attacks it's better to evaluate the same. Here the attacks considered are Selective forward along with sinkhole attacks. Sinkhole attracts the traffic for selective forward to drop the packets. The evaluation is done based on packet loss ratio and power consumption.

1) PACKET LOSS RATIO

TIDS promptly identified and countered a sophisticated attack involving both selective forwarding and sinkhole, targeting a rogue node within the network. By reducing the excessive burden resulting from the attack-induced packet loss rate, it effectively cut overhead. Fig. 11 provides a detailed analysis of the packet loss rate to compare the performance of the TIDS mechanism with that of MRHOF, showing that the TIDS mechanism outperformed MRHOF. The mechanism's packet loss ratio is 38% lower than that

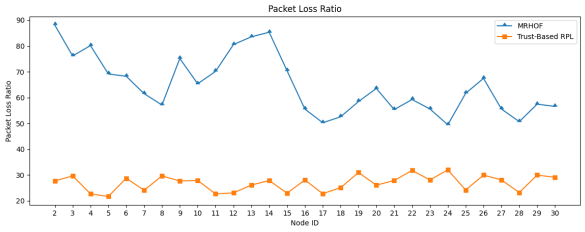

FIGURE 11. Packet loss ratio under selective forward and sinkhole attack.

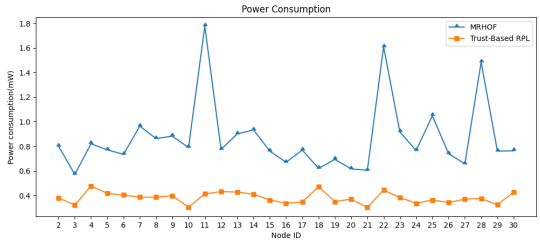

FIGURE 12. Power consumption under selective forward and sinkhole attack.

of MRHOF, even when identical network characteristics are used. Interestingly, a limited number of patterns in both methods appear to be unaffected by the utilization of identical network parameters. The average packet loss ratio for TIDS was 26.37%, while for MRHOF it was 64.48%. Therefore, our method offers superior protection against attacks with the lowest rate of packet loss.

2) POWER CONSUMPTION

The trust-based method proposed for LLNs exhibits a 45% enhancement in energy efficiency as compared to MRHOF, particularly in the scenario of concurrent selective forward and sinkhole attacks. This is seen in Figure 12. The results demonstrate that the mean power consumption rates for TIDS and MRHOF are 0.35 mW and 0.8 mW, correspondingly. The consumption rate of MRHOF is higher than other approaches considering the fact that, it lacks these mitigation strategies to handle the packet losses generated by malicious nodes in the network. IoT devices spend a substantial amount of energy since they lack a centralized detecting module. The trust mechanism functions at the individual node level, where a node with restricted resources performs all computations. As a result of this, the affected nodes undergo a reduction in their lifespan, ultimately failing the network due to the exhaustion of energy in these nodes. Therefore, the suggested method effectively utilized network resources.

E. SYBIL ATTACK

A Sybil node communicates with other nodes using the identity of a valid node in the network. This will give the impression that a node is appearing at different locations. Therefore, during each instance, the address of the node will be the same but its rank and location will be different. Thus a Sybil attack can be detected by checking for inconsistencies in the rank or location of the node. The rank inconsistencies will be detected at the time when sinkhole attacks are detected. When a malicious node assumes the identity of a victim node, it retains its rank. It then uses this address to communicate with the neighbors of the victim. This however causes an inconsistency between the ranks of the parent and child nodes, indicating the presence of an attack. Furthermore, we can check the location of the node whenever it sends a message against its previously known location. Any location change indicates the presence of a Sybil attack. This is used to raise an alarm notifying of the attack and the sink node detects the attacker using the DODAG information. In this way, any Sybil attacks in the network are detected.

VI. CONCLUSION

This research work addresses the security vulnerabilities like Sink hole, selective forwarding, and Sybil attacks in the routing layer of IoT devices, specifically those using LLNs and RPL. The study identifies shortcomings in RPL, highlighting its susceptibility to various routing attacks, potentially leading to severe consequences. To mitigate these risks, the research proposes a trust-based framework integrated into the RPL routing protocol. The suggested framework calculates trust values for each node based on observed packet flow between nodes, aiming to identify and isolate malicious nodes, such as those involved in sinkholes, selective forwarding, and Sybil attacks. The trust mechanism effectively identifies and segregates malevolent nodes, resulting in improved energy efficiency and lower average disparity in packet loss ratio.

The research emphasizes the importance of energy conservation, scalability, and decentralization in developing security solutions for distributed IoT devices. The proposed trust mechanism is designed to accommodate a large number of interconnected nodes, addressing scalability concerns. Additionally, the study advocates for the exploration of vulnerabilities associated with the trust model. The research concludes by suggesting an IDS as a reliable platform for ensuring secure data routing in IoT.

VII. FUTURE SCOPE

The future trajectory of research and development for the proposed IDS involves several key avenues. First, there is a focus on expanding the IDS coverage to encompass a wider array of attacks, including Rank attacks, version number attacks, and neighbor attacks. Additionally, the integration of multi path routing in wireless ad hoc networks is explored to enhance load balancing and resilience to mobility. To address power consumption concerns, a cluster-based approach is proposed, distributing network monitoring activities among nodes to reduce reliance on the root node. Furthermore, the implementation of blockchain-based logging is suggested to ensure data integrity and prevent loss in IoT environments. Finally, to optimize energy consumption, the migration of the centralized module to a Border Router is recommended, offering a strategic shift from the root node. These future

developments collectively aim to fortify the IDS, making it more adaptable and efficient in mitigating evolving security threats in RPL-based IoT networks.

REFERENCES

[1] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for Internet of Things: A survey," *J. Netw. Comput. Appl.*, vol. 66, pp. 198–213, May 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516300133

[2] K. Zhao and L. Ge, "A survey on the Internet of Things security," in *Proc. 9th Int. Conf. Comput. Intell. Secur.*, Dec. 2013, pp. 663–667.

[3] G. Xu, Y. Ding, J. Zhao, L. Hu, and X. Fu, "Research on the Internet of Things (Iot)," *Sensors Transducers*, vol. 160, p. 463, Jan. 2013.

[4] S. Park, N. Crespi, H. Park, and S.-H. Kim, "IoT routing architecture with autonomous systems of things," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 442–445.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[6] S. Sicari, S. Hailes, D. Turgut, S. Sharafeddine, and U. B. Desai, "Security, privacy and trust management in the Internet of Things era—SePriT," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2623–2624, Nov. 2013.

[7] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[8] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proc. Int. Conf. Pervasive Comput. (ICPC)*, Jan. 2015, pp. 1–6.

[9] V. Pai and U. K. K. Shenoy, "6LoWPAN—Performance analysis on low power networks," in *Proc. Int. Conf. Comput. Netw. Commun. Technol.*, S. Smys, R. Bestak, J. I.-Z. Chen, and I. Kotuliak, Eds. Singapore: Springer, 2019, pp. 145–156.

[10] H. A. Abdul-Ghani, D. Konstantas, and M. Mahyoub, "A comprehensive IoT attacks survey based on a building-blocked reference model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 3, pp. 355–373, 2018.

[11] N. H. A. Ismail, R. Hassan, and K. W. M. Ghazali, "A study on protocol stack in 6LoWPAN model," *J. Theor. Appl. Inf. Technol.*, vol. 41, pp. 220–229, Mar. 2012.

[12] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, *The Trickle Algorithm*, document RFC 6206, 2011, pp. 1–13.

[13] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL)," in *Proc. IEEE 7th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2011, pp. 365–372.

[14] H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 964–979, Apr. 2017.

[15] J. Hui, J. Vasseur, D. Culler, and V. Manral, *An IPv6 Routing Header for Source Routes With the Routing Protocol for Low-Power and Lossy Networks (RPL)*, document RFC 6554, 2012, pp. 1–13.

[16] O. Gnawali and P. Levis, *The Minimum Rank With Hysteresis Objective Function*, document RFC 6719, 2012, pp. 1–13.

[17] J. Ko, A. Terzis, S. Dawson-Haggerty, D. E. Culler, J. W. Hui, and P. Levis, "Connecting low-power and lossy networks to the internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 96–101, Apr. 2011.

[18] 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.

[19] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Wireless Network Security*. The Netherlands: Springer, 2007, pp. 103–135.

[20] 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, pp. 37–45, Jun. 2019.

[21] A. Dvir, T. Holczer, and L. Buttyan, "VeRA–version number and rank authentication in RPL," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2011, pp. 709–714.

[22] M. Landsmann, M. Wahlisch, and T. C. Schmidt, "Topology authentication in RPL," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2013, pp. 73–74.

[23] K. Iuchi, T. Matsunaga, K. Toyoda, and I. Sasase, "Secure parent node selection scheme in route construction to exclude attacking nodes from RPL network," in *Proc. 21st Asia–Pacific Conf. Commun. (APCC)*, Oct. 2015, pp. 299–303.

[24] D. Airehrour, J. Gutierrez, and S. K. Ray, "A trust-aware RPL routing protocol to detect blackhole and selective forwarding attacks," *Austral. J. Telecommun. Digit. Economy*, vol. 5, no. 1, pp. 50–69, 2017.

[25] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for Internet of Things (IoT)," *J. ISMAC*, vol. 2, no. 4, pp. 190–199, 2020.

[26] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. Spirito, "DEMO: An IDS framework for Internet of Things empowered by 6LoWPAN," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 1337–1340.

[27] M. N. Napiah, M. Y. I. Bin Idris, R. Ramli, and I. Ahmedy, "Compression header analyzer intrusion detection system (CHA–IDS) for 6LoWPAN communication protocol," *IEEE Access*, vol. 6, pp. 16623–16638, 2018.

[28] E. Kfoury, J. Saab, P. Younes, and R. Achkar, "A self organizing map intrusion detection system for RPL protocol attacks," *Int. J. Interdiscipl. Telecommun. Netw.*, vol. 11, no. 1, pp. 30–43, Jan. 2019.

[29] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2013, pp. 600–607.

[30] P. Shukla, "ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 234–240.

[31] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Netw.*, vol. 11, pp. 2661–2674, Nov. 2013.

[32] D. Shreenivas, S. Raza, and T. Voigt, "Intrusion detection in the RPL-connected 6LoWPAN networks," in *Proc. 3rd ACM Int. Workshop IoT Privacy, Trust, Secur.*, 2017, pp. 31–38.

[33] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in Internet of Things," *Int. J. Comput. Appl.*, vol. 121, no. 9, pp. 1–9, Jul. 2015.

[34] A. Mayzaud, R. Badonnel, and I. Chrisment, "Detecting version number attacks in RPL-based networks using a distributed monitoring architecture," in *Proc. 12th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2016, pp. 127–135.

[35] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "Using the RPL protocol for supporting passive monitoring in the Internet of Things," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2016, pp. 366–374.

[36] F. Gara, L. Ben Saad, and R. Ben Ayed, "An intrusion detection system for selective forwarding attack in IPv6-based mobile WSNs," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 276–281.

[37] F. Gara, L. B. Saad, and R. B. Ayed, "An efficient intrusion detection system for selective forwarding and clone attackers in IPv6-based wireless sensor networks under mobility," *Int. J. Semantic Web Inf. Syst. (IJSWIS)*, vol. 13, no. 3, pp. 22–47, 2017.

[38] 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.

[39] G. Glissa, A. Rachedi, and A. Meddeb, "A secure routing protocol based on RPL for Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.

[40] Z. A. Almusaylim, N. Jhanjhi, and A. Alhumam, "Detection and mitigation of RPL rank and version number attacks in the Internet of Things: SRPL-RP," *Sensors*, vol. 20, no. 21, p. 5997, Oct. 2020.

[41] 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.

[42] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 606–611.

[43] C.-M. Chen, S.-C. Hsu, and G. Lai, "Defense denial-of service attacks on IPv6 wireless sensor networks," in *Proc. ICGEC*, 2015, pp. 319–326.

[44] A. Jøsang and T. Bhuiyan, "Optimal trust network analysis with subjective logic," in *Proc. 2nd Int. Conf. Emerg. Secur. Inf., Syst. Technol.*, Aug. 2008, pp. 179–184.

[45] E. G. Ribera, B. Martinez Alvarez, C. Samuel, P. P. Ioulianou, and V. G. Vassilakis, "Heartbeat-based detection of blackhole and greyhole attacks in RPL networks," in *Proc. 12th Int. Symp. Commun. Syst., Netw. Digit. Signal Process. (CSNDSP)*, Jul. 2020, pp. 1–6.
[46] L. Wallgren, S. Raza, and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of Things," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 8, Aug. 2013, Art. no. 794326.
[47] H. Bostani and M. Sheikhan, "Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach," *Comput. Commun.*, vol. 98, pp. 52–71, Jan. 2017.
[48] S. O. Amin, M. S. Siddiqui, C. S. Hong, and S. Lee, "RIDES: Robust intrusion detection system for IP-based ubiquitous sensor networks," *Sensors*, vol. 9, no. 5, pp. 3447–3468, May 2009.

S. REMYA received the Ph.D. degree in computer science and engineering from Vellore Institute of Technology, Vellore Campus. She is currently an Assistant Professor with the Department of Computer Science and Engineering, School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri Campus, Kollam, Kerala, India. Her research interests include deep learning, data science, computer vision, network security, and smart environments.

MANU J. PILLAI received the Ph.D. degree in computer science and engineering from the National Institute of Technology, Kozhikode. He is currently an Associate Professor with the Department of Computer Science and Engineering, TKM College of Engineering, Kollam, Kerala, India. His research interests include wireless networks, deep learning, and smart environments.

C. ARJUN received the B.Tech. degree (Hons.) from TKM College of Engineering, Kollam. He is currently a Software Engineer. His professional expertise extends to intricate aspects of software development, specializing in backend infrastructure, database design, and user interface development. His research interests include wireless networks, the IoT, and smart environments, reflecting his commitment to advancing technology in these domains.

SOMULA RAMASUBBAREDDY received the Ph.D. degree in computer science and engineering from VIT University, Vellore, India, in 2022. He was a Postdoctoral Researcher with the Department of Information and Communication, Sunchon National University, South Korea, in 2024. He is currently an Assistant Professor with the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad. He has more than 40 publications in reputed journals and conferences. His research interests include mobile cloud computing, the IoT, machine learning, and edge computing.

YONGYUN CHO received the Ph.D. degree in computer engineering from Soongsil University. He is currently a Professor with the Department of Information and Communication Engineering, Sunchon National University. His main research interests include system software, embedded software, and ubiquitous computing.