This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

LiSec-RTF: Reinforcing RPL Resilience Against Routing Table Falsification Attack in 6LoWPAN

Shefali Goel, Vinod Kumar Jain, *Senior Member, IEEE*, and Abhishek Verma*

***Abstract*—Routing Protocol for Low-Power and Lossy Networks (RPL) is an energy-efficient routing solution for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), recommended for resource-constrained devices. While RPL offers significant benefits, its security vulnerabilities pose challenges, particularly due to unauthenticated control messages used to establish and maintain routing information. These messages are susceptible to manipulation, enabling malicious nodes to inject false routing data. A notable security concern is the Routing Table Falsification (RTF) attack, where attackers forge Destination Advertisement Object (DAO) messages to promote fake routes via a parent node's routing table. Experimental results indicate that RTF attacks significantly reduce packet delivery ratio, increase end-to-end delay, and leverage power consumption. Currently, no effective countermeasures exist in the literature, reinforcing the need for a security solution to prevent network disruption and protect user applications. This paper introduces a Lightweight Security Solution against Routing Table Falsification Attack (LiSec-RTF), leveraging Physical Unclonable Functions (PUFs) to generate unique authentication codes, termed "Licenses." LiSec-RTF mitigates RTF attack impact while considering the resource limitations of 6LoWPAN devices in both static and mobile scenarios. Our testbed experiments indicate that LiSec-RTF significantly improves network performance compared to standard RPL under RTF attacks, thereby ensuring reliable and efficient operation.**

***Index Terms*—IoT, 6LoWPAN, Routing Table Falsification, RPL, PUF**

I. Introduction

The Internet of Things (IoT) is a fast-growing technology that comprises several physical devices, sensors, and software for exchanging data across networks, spanning from local area networks to the broader Internet [1]. This connectivity empowers devices to communicate and share information among themselves. According to the McKinsey Global Institute report, IoT's economic influence is projected to range between 3.9 to 11.1 trillion by the end of the year 2025 [2]. One of the primary benefits of IoT is its ability to facilitate communication among devices that are situated at far distant locations by employing IPv6 addressing. Nevertheless, when deploying IoT in industrial sectors (IIoT) like industrial automation and advanced metering infrastructure (AMI) there is a specific need for an infrastructure that requires small power yet supports a longer lifetime while supporting IPv6 capabilities [3], [4]. This requirement is satisfied through the implementation of IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN). The devices operating within 6LoWPAN are characterized by their limited computational ability, limited storage, memory, and energy-efficient attributes [5], [6]. The key benefit of these resource-constrained devices is their ability to run exceptionally low-voltage and consume minimal energy. This enables them to operate for extended periods, often spanning several years. A diverse array of such devices, commonly called ultra-low-powered microcontrollers, are easily available in the market.

Routing is an integral component of 6LoWPAN networks, facilitating communication between devices situated at a distance from each other [7]. The conventional routing protocols such as Adhoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Open Shortest Path First (OSPF) are well-suited for Wireless Sensor Networks due to the resource-rich nature of these networks [8]. However, in the case of 6LoWPAN, embedded devices are inherently resource-constrained. Consequently, these traditional routing protocols are not recommended for 6LoWPAN. Therefore, there is a need for an energy-efficient routing protocol to conserve the resources of 6LoWPAN. Addressing this challenge is quite complex in 6LoWPAN networks [9], [10]. To meet this requirement, the Internet Engineering Task Force (IETF) introduced the Routing Protocol for Low Power and Lossy Networks (RPL) [11]. RPL is explicitly designed for IPv6-based Low-Power and Lossy Networks (LLNs), which include 6LoWPAN. It functions as a proactive protocol, establishing and maintaining a routing topology in advance to enable efficient and reliable communication among nodes in the network. It is important to emphasize that RPL is still in its developmental phase [11], and its specifications are presented in RFC 6550. RPL demonstrates diverse attributes, including the capacity to modify control packet frequencies, dynamically regulate control packet transmission rates via the trickle algorithm, and compute routing metrics utilizing an Objective Function to accommodate multi-path topologies [12], [13], [14], [15]. These characteristics of RPL make it well-suited for deployment in 6LoWPAN [16]. Nevertheless, it is important to remember that RPL and IIoT are susceptible to many types of attacks that target the privacy and security of users [17], [18]. This vulnerability arises from the fact that RPL employs a shared key for security, making it susceptible to compromise if an attacker gains access to the key [19]. Furthermore, RPL does not provide packet confidentiality, which means that an attacker with the capability to eavesdrop

* Corresponding author

Shefali Goel is with Computer Science & Engineering Discipline, PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Madhya Pradesh, India (e-mail: 21pcso08@iiitdmj.ac.in).

Vinod Kumar Jain is with Department of Computer Science and Engineering, ABV-Indian Institute of Information Technology and Management Gwalior, Morena Link Road, Gwalior, Madhya Pradesh, India, 474015 (e-mail: vkjain@iiitdmj.ac.in).

Abhishek Verma is with Department of Information Technology, Babasaheb Bhimrao Ambedkar University, Lucknow 226025, Uttar Pradesh, India (e-mail: abhiverma866@gmail.com).

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

on communication can potentially obtain sensitive information [20], [21].

A common cyber threat in 6LoWPAN networks is insider or outsider routing attacks. These attacks exploit vulnerabilities in RPL to target legitimate nodes, potentially causing substantial disruptions to the overall performance of the network [22]. The objective of this paper is to explore the Routing Table Falsification Attack (RTF). This attack can occur when malicious nodes tamper with Destination Advertisement Object (DAO) control messages or create forged DAO to establish fictitious downward routes. This type of attack is feasible only when the network enables the storing mode [23], [24]. The consequence of such an attack can include longer paths, increased network delays, higher packet drop rates, and even network congestion. It is essential to consider that this specific vulnerability and its impact on RPL networks have not been extensively studied or documented [6], [25] and does not have any defense solution to address RTF attack [26]. This paper shows that the RTF attack decreases the packet delivery ratio, and increases the packet delay and power consumption. To counteract Routing Table Falsification (RTF) in RPL, we have introduced a secure variant of RPL known as LiSec-RTF. While RPL is susceptible to a wide range of attacks, the proposed LiSec-RTF framework specifically focuses on mitigating the RTF attack (an Identity spoofing attack). This attack involves the injection of malicious routing information, enabling an attacker to manipulate the network's routing tables. LiSec-RTF aims to detect and prevent such falsified updates by integrating lightweight trust-based validation mechanisms suited for resource-constrained IoT environments. In the proposed solution (LiSec-RTF), we introduce a modified DAO message, referred to as $DAO_{modified}$, which incorporates an authentication mechanism for verifying sensor nodes at the 6LoWPAN Border Router (6LBR). The Reserved field of the DAO packet is utilized to encapsulate a License, which consists of a unique bit sequence generated by performing an exclusive OR (XOR) operation on the node's Physical Unclonable Function (PUF) data. When a child node unicasts a $DAO_{modified}$ message to the 6LBR, the router extracts the License from the Reserved field and verifies it against the License generated during the node's registration phase. This process enables the 6LBR to authenticate whether the License is genuine i.e., derived from the registered PUF data. Very few changes have been made in the standard RPL implementation to incorporate LiSec-RTF. We have only updated the existing DAO processing mechanism of standard RPL implementation to perform authentication of source DAO's.

Some of the benefits of the proposed LiSec-RTF approach include: (1) accurate detection of attack; (2) mitigation of attack which improves the network's performance in static and mobile network environments; (3) the proposed solution does not introduce any memory overhead on the resource-constrained nodes. The novelty of the proposed LiSec-RTF framework lies in its specific focus on detecting and mitigating RTF attack in RPL-based networks. In the literature, there is no specific defense mechanism exists that focused on mitigating RTF attack in RPL-based IoT networks. Although several IDS can detect and mitigate a range of RPL-based attacks, they are not effective against RTF attack due to fundamental differences in attack characteristics. RTF attack exhibit distinct behavioral patterns compared to attacks such as Forwarding Misbehavior, DAO Inconsistency, Hatchetman, DIO Suppression, Energy Depletion, Spam DIS, and Advanced Vampire attacks. The proposed LiSec-RTF framework specifically focuses on mitigating the RTF attack. This attack involves the injection of malicious routing information, enabling an attacker to manipulate the network's routing tables. LiSec-RTF aims to detect and prevent such falsified updates by integrating lightweight trust-based validation mechanisms suited for resource-constrained IoT environments. The key points of the contribution are outlined below:

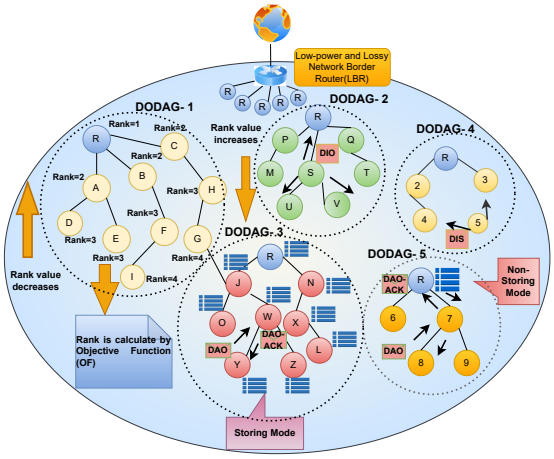


Fig. 1: Overview of RPL

- Analyze the impact of Routing Table Falsification attack on both static and mobile environment.
- An effective solution named "LiSec-RTF" is proposed to mitigate the effect of Routing Table Falsification attack.
- The effectiveness of "LiSec-RTF" is compared against the standard RPL protocol using simulation under Routing Table Falsification attack.
- A simulation study is performed comparing "LiSec-RTF" against the standard RPL protocol in delivering good network performance while under Routing Table Falsification attack on both static and mobile scenarios.
- A testbed experimentation is also performed to validate the reported simulation results.

The organization of the paper is as follows: Section II presented the RPL protocol, the Routing Table Falsification attack. SectionIII presented the literature of some RPL specific attacks. Section IV presents the preliminary requirements, which are essential to design our proposed solution. Moreover, our proposed approach is presented in Section V. Section VI elaborates on the simulation parameters and presents an analysis of the network's performance based on various metrics. Further, section VII presents an encrypted variant of the proposed solution. The final Section VIII summarizes the conclusions drawn in the paper.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

II. Background

A. Overview of the RPL protocol

Routing Protocol for Low Power and Lossy Networks (RPL) operates on the principle of distance-vector and source routing. RPL aims to establish routes between sensor nodes within the topology. RPL creates a Directed Acyclic Graph (DAG) structure to manage the routing of data packets. The DODAG comprises sensor nodes that represent the devices in the network and the edges that represent the links between the devices. The DAG structure enables RPL to efficiently route packets even in networks with low bandwidth and high loss rates. The sensor nodes are structured to converge towards a single destination, known as Destination-Oriented Directed Acyclic Graphs (DODAG). The DODAG is an acyclic structure in which each sensor node forwards its data toward the sink. There may be multiple DODAGs running simultaneously in the network to achieve fault tolerance. The sink node of the DODAG is called the 6LoWPAN Border Router (6LBR). Figure 1 represents the overview of RPL. There are four control messages available in RPL for topology construction and maintenance: (a) DODAG Information Solicitation (DIS), (b) DODAG Information Object (DIO), (c) Destination Advertisement Object (DAO), and (d) DAO-Acknowledgment (DAO-ACK).

The new node broadcasts the DIS message to request DIO messages when it wants to join an existing network. Upon receiving a DIS message, a sink node or intermediate node responds with a DIO message. The DIO message is periodically broadcast to advertise its existence and provide routing information to other nodes in the network. The DIO contains information about the DODAG Version, DODAG ID, rank, supported Objective Function (OF), and other parameters. The rank value serves to indicate the position of a node relative to the sink node. Upon receiving the rank in the DIO packet, the new node calculates its rank and manages the parent list accordingly [27]. The sensor node with the lowest rank value among the surrounding nodes is selected as the preferred parent node. The supported OF in the DIO message is used to calculate the rank. OF is used to select the best path based on the rank value and routing metrics. The RPL has the availability of two modes, i.e., storing and non-storing modes. In storing mode, after selecting the preferred parent, the DAO message is unicast to the selected preferred parent for route registry in their routing tables, and the message is forwarded up to the sink. In the non-storing mode of RPL, the child unicasts the DAO message to the sink. The DAO-ACK message is used to acknowledge the DAO message. In RPL, a timer mechanism is present to dynamically adjust the transmission of control packets[28] known as Trickle Timer.

B. RPL-specific attacks

Mayzaud *et al.* [6] proposed a taxonomy of routing attacks against the RPL protocol. Some of the RPL-specific attacks are discussed here:

- Forwarding Misbehavior attack: The malicious node accepts data packets from neighboring or upstream nodes but deliberately fails to forward them to their destination, instead discarding them.
- DAO Inconsistency attack: A malicious node deliberately discards received data packets and responds with a forwarding error message, misleading the parent node into removing valid downward routes from its routing table.
- Hatchetman attack: A malicious node modifies the source header of control packets and generates a large volume of invalid packets containing incorrect routing information, aiming to disrupt legitimate nodes.
- DIO-Suppression attack: Malicious node deliberately block or suppress DIO messages, preventing legitimate nodes from receiving critical routing updates and causing disruptions in the network routing process.
- Energy Depletion attack: A malicious node floods legitimate nodes with a large volume of packets, aiming to exhaust their energy resources.
- Spam DIS attack: A malicious node generates and broadcasts a large number of DIS messages using spoofed or fake source identities, overwhelming the network and triggering unnecessary routing updates.
- Sybil attack: A malicious node broadcasts DIS messages using numerous fake identities throughout the network in an attempt to manipulate and take control of the routing process.
- Advanced Vampire attack: A malicious node alters the source routing information, causing legitimate nodes to reject the packet and respond with error messages, ultimately leading to routing instability and disruption of network services.

C. Overview of PUF

Physical Unclonable Function (PUF) is a hardware security mechanism that leverages the inherent manufacturing variations of physical devices to generate unique, unpredictable, and tamper-resistant identifiers. These identifiers are used for applications such as secure key storage, device authentication, and cryptographic operations [29]. PUFs are essential for addressing security challenges inherent in IoT Networks. IoT devices are often resource-constrained and deployed in untrusted environments. PUFs provide a lightweight, hardware-based solution that enhances security without requiring significant computational overhead [30]. There are several applications of PUFs in IoT:

1 Device Authentication: A PUF generates a unique response for a given challenge based on the device's physical properties. This challenge-response mechanism ensures that only authorized devices can participate in the IoT network.
2 Cryptographic Key Generation: PUFs generate cryptographic keys dynamically from hardware properties, eliminating the need for storing sensitive keys in non-volatile memory.
3 Secure Communication: PUF-derived keys are used for encrypting data transmitted between IoT devices, ensuring confidentiality and integrity in communication.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

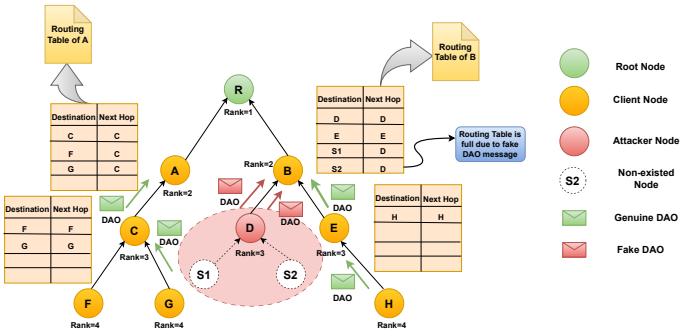


Fig. 2: An Illustration of Routing Table Falsification Attack

4 Anti-Tampering Measures: The physical attack on a PUF typically alters the underlying hardware properties, invalidating the responses and alerting the system.

D. Routing Table Falsification Attack

The DAO control message is utilized in RPL to establish downward routes from the sink node to the leaf nodes. RPL offers two modes: storing mode and non-storing mode. In non-storing mode, the child node unicasts the DAO message to the sink via the parent node, which is selected with the aid of OF. The parent node adds its address to the header and forwards the DAO message to the sink. Upon receiving the DAO, the sink node sends a DAO-ACK to the child node, which originated the DAO message. However, in the storing mode of RPL, the child node unicasts DAO messages directly to the parent. Upon the establishment of a new route between the parent and child nodes, the parent's routing table is updated with an entry for this route. Following this, the route must be registered at the sink node by forwarding the DAO message through the preferred parent node. In response to the DAO message, the sink node subsequently unicasts a DAO-ACK, which is forwarded to the child node through the downward routing path. The acknowledgment in the form of DAO-ACK confirms the registration of the child node at the sink. In this mode, the intermediate routers in the network maintain routing tables to store information about routes. The routing table of the standard RPL is exploited to mount the RTF attack. The intruder may compromise an insider node to perform this attack. By introducing an RTF attack in RPL, the malicious nodes forge the DAO message of the standard RPL to promote fake routes to the parent to disrupt the resource-constrained nature of sensor nodes.

The RTF attack is introduced in Mayzaud *et al.* [6]. During an RTF attack, the malicious node is an internal node and serves as the child of the parent node. The malicious node unicast forged DAO messages to the parent to fill the routing table of the parent node. The impact of this attack on RPL networks has not been studied or documented yet [25]. Figure 2 represents the RTF attack in RPL. In the typical scenario of RPL, Nodes F, and G unicast DAO messages to the parent node C, and the entry is shown in the routing table of C. Node C forwards the DAO message of their child node to its parent node A for route registry in the routing table of A. This is the storing mode operation of the standard RPL. In the attack scenario, an insider node D is compromised to perform an RTF attack. The malicious node D unicasts the forged DAO message of Nodes S1 and S2 (non-existing nodes) to the parent node B. Node B registers the route entries of Nodes S1 and S2 by considering them as genuine nodes. Afterward, when Node E unicasts the DAO message of their child Node H, the routing table of B is filled by the non-existing routes unicast by the malicious node D. As a result, the routing table entries are full, thus blocking the creation of routes towards new legitimate nodes (such as H) [31]. The attack impacts longer paths, increased network delays, higher packet drop rates, and even network congestion.

III. Related Work

Recognizing the significance of security concerns in RPL, various defense mechanisms have been proposed to address some of the RPL-based attacks. Pu et al. proposed a solution against Forwarding misbehavior attack known as CMD. In this mechanism, each node monitors its preferred parent's forwarding behavior, specifically tracking packet loss. The node compares its own observations with packet loss rates reported by one-hop neighbors to detect potential misbehavior. Pu et al. addresses the DAO Inconsistency attack. The attacker node deliberately drops some data packets and replies with an error message to discard the entry of the node from the routing table. The proposed defense mechanism suggests that each parent dynamically adjusts the threshold of the error packet at a particular period. The author proposed a security solution against Energy Depletion attack known as misbehavior-aware threshold detection. In this technique, each node monitors the number of packets received from its child nodes over a set time window and compares this count against a dynamically determined threshold to identify possible malicious behavior. The authors proposed a security system against Advance vampire attack. In which, each node track the destination MAC addresses of the incoming data packets and evaluates the evenness of the distribution using the Theil index, a statistical measure of inequality. Under normal conditions, destination addresses show a certain pattern, but during an advanced vampire attack, a malicious node randomly injects packets having fictitious MAC addresses. This causes a significant increase in distribution randomness, leading to an abnormally high Theil index value. The countermeasure activates to mitigate the attack by identifying and eliminating the malicious activity. Pu et al. proposed a lightweight security solution against the sybil attack called LiteSAD, based on Bloom Filter. Bloom filters are probabilistic data structures that efficiently test whether an element is a member of a set. In this solution, each node maintains a record of legitimate neighbors using Bloom Filter. When a node receives multiple DIS messages and identities that do not match the filter, it suspects a Sybil attack. The authors validated LiteSAD through simulation as well as on real-world testbeds.

Although existing solutions can detect and mitigate a range of RPL-based attacks, they are not effective against RTF

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

TABLE I: A comparison of various attacks with their countermeasure

Reference	Targeted attack	Description of attack	Defense approach	Mobility support	Type of validation
Perazzo *et al.* [32]	DIO Suppression attack	Malicious nodes selectively block or suppress DIO messages that cause legitimate nodes may not receive important routing updates, leading to routing disruptions.	No solution exist	No	Simulation.
Pu *et al.* [33]	Forwarding misbehavior attack	A malicious node silently drop incoming data packets instead of forwarding them.	Monitoring based approach known as CMD	No	Simulation.
Pu *et al.* [33]	DAO Inconsistency attack	A malicious node intentionally drops the received data packet and replies the forwarding error packet to cause the parent node to discard valid downward routes from the routing table.	Threshold based approach	No	Simulation.
Pu *et al.* [34]	Hatchetman attack	A malicious node alter the source header of the control packet and then produces a huge number of invalid packet with erroneous routes targeting legitimate nodes.	No solution exist	No	Simulation.
Pu *et al.* [35]	Spam DIS attack	A malicious node sends a flood of DIS messages with fake source identities.	No solution exist	No	Simulation.
Pu *et al.* [36]	Energy Depletion attack	A malicious node sends excessive packets to legitimate nodes in order to drain the energy of the legitimate nodes.	Misbehaviour Aware Detction (MAD)	No	Simulation.
Pu *et al.* [37]	Advance Vampire attack	A malicious node manipulates source route, the legitimate nodes immediately drops the packet and reply with error message that cause routing instability and service disruption.	Theil-index based approach	No	Simulation.
Pu *et al.* [30]	Sybil attack	Malicious node multicast DIS messages with multiple fake identities within the network to gain control over the routing process.	Bloom filter based approach known as liteSAD	No	Simulation as well as on testbed.
Our paper	Routing Table Falsification attack	A malicious node unicast forged DAO messages to the parent to overflow the routing table of the parent node.	PUF based security solution known as LiSec-RTF	Yes	Simulation as well as on testbed.

attack due to fundamental differences in attack characteristics. RTF attack exhibit distinct behavioral patterns compared to attacks such as Forwarding Misbehavior, DAO Inconsistency, Hatchetman, DIO Suppression, Energy Depletion, Spam DIS, and Advanced Vampire attacks. These existing attacks primarily target packet forwarding, control message manipulation, or energy exhaustion, whereas RTF attack specifically compromise the integrity and consistency of routing information within the RPL topology. As a result, current IDS mechanisms fail to detect RTF attack. Since a Sybil attack is a form of identity spoofing, existing countermeasures focus primarily on detecting anomalies in DIS messages exchanged during the network discovery phase. However, these solutions are limited in scope and are not designed to detect RTF attack, which involve manipulation of routing information during the parent registration phase. As a result, existing Sybil attack detection mechanisms are insufficient for identifying or mitigating RTF attack.

In this paper, we have focused only on the study of Routing Table Falsification attack, i.e., an identity spoofing attack. Moreover, we propose a defense mechanism to address this attack. Table I provides a detailed comparison of the some RPL specific attacks in the literature. LiSec-RTF aims to detect and prevent such falsified updates by integrating lightweight trust-based validation mechanisms suited for resource-constrained IoT environments. The proposed LiSec-RTF mechanism is effective in detecting forged DAO control messages that mislead routing decisions. However, to address the above mentioned attacks, the proposed approach can be further enhanced and extended. Since sybil attack is a kind of identity spoofing attack, it can also be detected by LiSec-RTF.

IV. Preliminary

A. System and Threat Model

The system and threat model considered in this paper is depicted in Figure 3. Table II represents the notations and their corresponding definition for the proposed model. The assumption of the system model are outlined as follows:

- The study focuses on a PUF-enabled IoT network comprising a set S = S_1, S_2, S_3,..., S_n of n nodes. These nodes are IoT sensors characterized by limitations in communication range, memory capacity, processing capability, and battery capacity.
- The sink node, also known as the 6LoWPAN Border Router (6LBR), possesses abundant resources. The 6LBR stores the PUF data ($\mathcal{CH} - \mathcal{R}$) of each sensor node generated at the registration phase.
- As shown in Fig. 3, the DAO message is modified ($DAO_{modified}$) to use the 8-bit $\mathcal{R}eserved$ field to incorporate our proposed approach.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

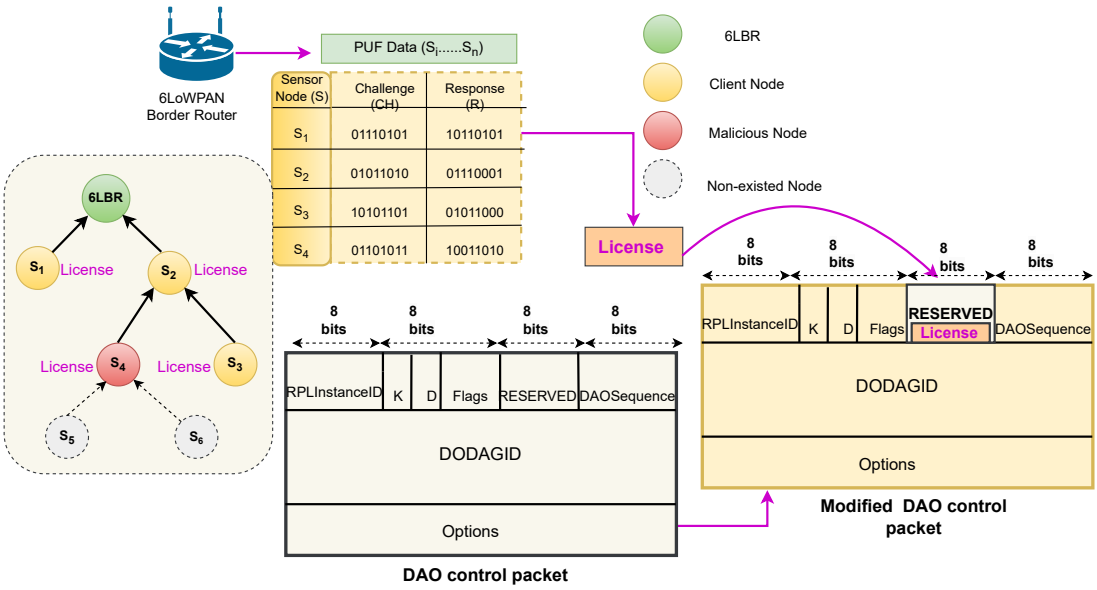


Fig. 3: System and Threat model

TABLE II: Notations and Definitions

Notation	Definition
$\mathcal{R}eserved$	Reserved field of DAO message.
P	Preferred parent node.
$\mathcal{C}$	Child node.
Max_{node}	Maximum count of sensor nodes.
$\mathcal{RT}$	Routing Table.
$\mathcal{R}_i$	Response of sensor node i.
$\mathcal{L}_i$	License of sensor node i.
$\mathcal{CH}$	Challenge.
r_i	Calculate response of sensor node i.
$DAO_{modified}$	Modified DAO control packet.
$\mathcal{N}_{bl}$	Count of Blacklist nodes.
$\mathcal{B} \leftarrow [1, \ldots, Max_{node}]$	Number of entries in the Blacklist table.
$\mathcal{N} \leftarrow [1, \ldots, Max_{node}]$	Number of entries in the Neighbor table
$\mathcal{PUF}_{data} \leftarrow [< \mathcal{S}_i >, < \mathcal{CH} >, < \mathcal{R} >]$ $i = 1,\ldots, Max_{node}$	Structure of PUF data at 6LBR.
$\mathcal{RT} \leftarrow [R_1, R_2, \ldots, R_n]$	Structure of Routing Table.
$\mathcal{M}_i \leftarrow [<Src_{ip}>]$ $i = 1,\ldots, Max_{node}$	Framework of malicious node in the Blacklist table.
$\oplus$	exclusive OR
Src_{ip}	Source address of DAO.

- Each sensor node has its own License ($\mathcal{L}_i$) that is provided by the administrator using the node's PUF data ($\mathcal{CH} - \mathcal{R}$).
- The assumed system model is applicable in those applications where an administrator provides an authentication code before deployment, e.g., military weapons, Advanced Metering Infrastructure (AMI) [38].
  This ensures the secure operation of LiSec-RTF.

In the assumed network scenario, the malicious user is considered to have the following features.

- The malicious user can compromise the legitimate node and reprogram it to behave maliciously. [20], [39], [40].
- The malicious node is the insider node of the network. For experimentation purposes, the topology has a number of malicious nodes.
- In the assumed IoT scenario, the malicious node pretends that it is the parent of several child nodes. The malicious node unicast the DAO packet with the falsified route to 6LBR through its parent.
- The goal of a malicious user is to overload the routing table of the parent node with fake routes to create congestion on the network.
- As shown in Figure 3, the insider node S_4 is registered in the network and has its License generated at the time of registration. Later on, it is captured by an attacker to mount an RTF attack.

This work focuses on an non-encrypted lightweight authentication for applications like smart agriculture, IIoT (Machine Monitoring), smart street lightening, wildlife tracking, smart parking systems, and conservation, where the data itself may not be highly sensitive (i.e. confidentiality is not a major concern) and has a need of low computational overhead based solution, requires only simple device identity verification due to regulatory compliance, or have open-field deployment. Such applications demand low-overhead, low-cost, simple deployment, and limited security. However, for applications which require more security against smart adversary and eavesdropping attack a encrypted variant of the proposed solution is discussed further in Section VII.

V. Proposed Approach

The current specification of RPL lacks a mechanism to counteract the RTF attack. We have proposed an approach

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

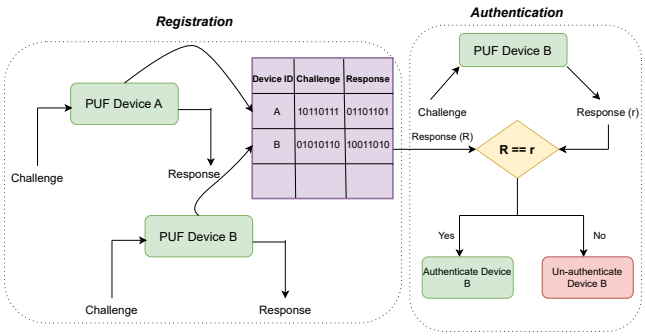


Fig. 4: Overview of PUF for device authentication

named LiSec-RTF to address this limitation of RPL. Figure 6 illustrates the overview of our proposed approach. In the proposed solution, we employ a $DAO_{modified}$ message, for authentication of nodes at 6LBR. The $\mathcal{R}eserved$ field encapsulates the Licence in the modified DAO ($DAO_{modified}$) packet. The License specifies a sequence of bits achieved by eXclusive OR (XOR) of the PUF data of a sensor node. The proposed LiSec-RTF leveraging strong PUFs to generate a large number of challenge-response pairs which makes them suitable for applications like authentication and anti-cloning mechanisms in large-scale networks. When a child node unicasts the $DAO_{modified}$ to the 6LBR, the 6LBR extracts the License from the $\mathcal{R}eserved$ field of the $DAO_{modified}$ for validation. Afterward, the 6LBR validates the License, whether it is a genuine License generated at the time of registration using their PUF data or a fake License provided by the malicious node to overload the routing table through fake routes. Figure 7 illustrates the proposed architecture of LiSec-RTF. The License generation procedure is described in Section V-B.

The algorithm LiSec-RTF is illustrated in Algorithm 1. If the License is validated, it confirms that the node unicasted the DAO packet with a genuine route. For the generation of the License at the time of registration, we used PUF information of a node [30].

A. Physical Unclonable Function (PUF)

A Physical Unclonable Function (PUF) is a hardware security primitive that uses the physical variations available in Integrated Circuits (ICs) to create a unique authentication code [41]. These physical variations are nearly impossible to replicate precisely to generate an identifier that resembles a fingerprint for every distinct chip. When the binary sequence of inputs named "Challenge" is applied to the PUF, it will produce the corresponding output named "Response". The Challenge-Response pair of each IC is unique due to the feature of PUF, and no two IC will produce an identical response for the same Challenge.

Device authentication based on PUF can be implemented in two phases: (1) Registration; (2) Authentication. In the registration phase, Devices A and B are registered in the trusted environment, as depicted in Figure 4. In the next phase to authenticate Device B, the same Challenge is passed as

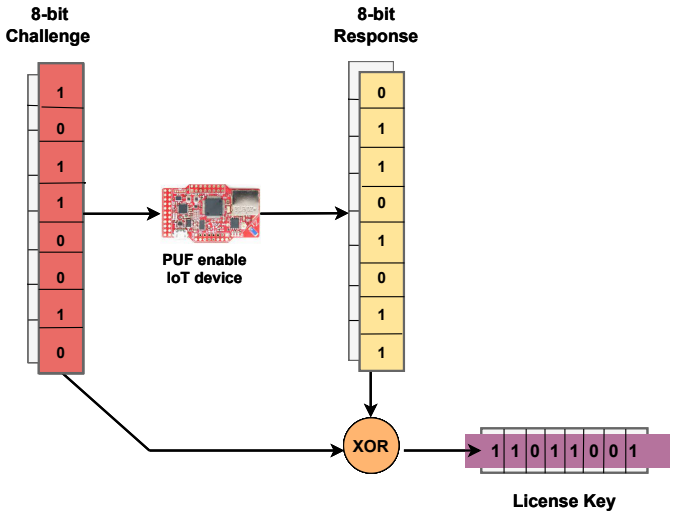


Fig. 5: License Generation Procedure

the input to Device B, and it will produce the Response. If both responses are the same, we consider that Device B is authenticated. Otherwise, it is a malicious device by keeping the identity of Device B.

B. License Generation Procedure

Figure 5 illustrates the License generation procedure of a PUF-enabled sensor device. When a 8-bit random value called as Challenge ($\mathcal{CH}$) is given as input to the PUF-enabled sensor node, the $\mathcal{CH}$ is passed to the function of PUF which produces an 8-bit output known as Response ($\mathcal{R}$). This process is represented with the help of Eq. 1.

$$\mathcal{R} = f_{PUF}(\mathcal{CH}) \tag{1}$$

License ($\mathcal{L}_i$) is generated at the time of registration and represented by Eq. 2. Each sensor node is configured with the generated License during node deployment phase of RPL.

$$\mathcal{L}_i = \mathcal{CH}_i \oplus \mathcal{R}_i \tag{2}$$

C. Detection mechanism of LiSec-RTF

Fig. 7 represents the framework of LiSec-RTF. When the 6LBR receives the modified version of DAO message ($DAO_{modified}$). It extracts the License ($\mathcal{L}_i$) from the $\mathcal{R}eserved$ field of the DAO message. For the purpose of authentication, 6LBR extracts the Challenge ($\mathcal{CH}_i$) from the PUF data ($\mathcal{PUF}_{data}$) of that sensor node and decrypts the $\mathcal{L}_i^{enc}$ from the $DAO_{modified}$.

$$r_i = \mathcal{CH}_i \oplus \mathcal{L}_i \tag{3}$$

In Eq.3, the calculated response (r_i) is the XOR of $\mathcal{CH}_i$ and $\mathcal{L}_i$. At the 6LBR, if the calculated response (r_i) is equal to the Response ($\mathcal{R}_i$) that is stored in their ($\mathcal{PUF}_{data}$), then we state that the License ($\mathcal{L}_i$) is validated and the source address (Src_{ip}) of the DAO sender is authenticated; otherwise, it is a

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

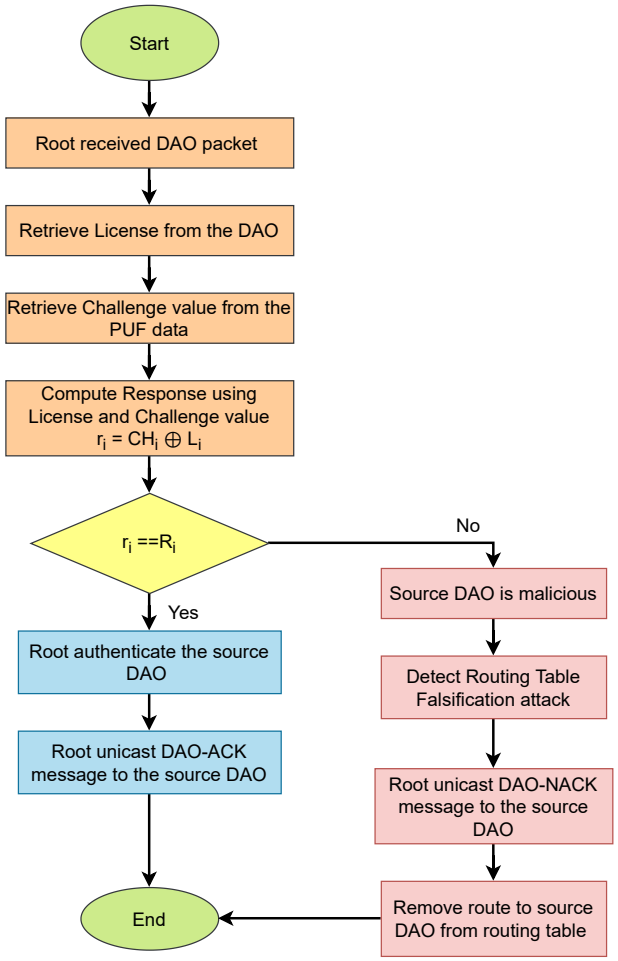


Fig. 6: Overview of proposed approach

Algorithm 1 Pseudocode of LiSec-RTF

procedure INITIALIZATION
  $\mathcal{PUF}_{data} \leftarrow [1, \ldots, Max_{node}]$
  $\mathcal{B} \leftarrow [1, \ldots, Max_{node}]$
  $\mathcal{N} \leftarrow [1, \ldots, Max_{node}]$
  $\mathcal{RT} \leftarrow [R_1, R_2, \ldots, R_n]$
  $\mathcal{L}_i = \mathcal{CH}_i \oplus \mathcal{R}_i$ ▷ Eq. 2
end procedure
procedure ON_$Sender(pkt)$
  if $(pkt[type] == pkt[DAO_{modified}])$ **then**
    $\mathcal{R}eserved \leftarrow \mathcal{L}_i$
    Send $DAO_{modified}$ to P
  end if
end procedure
procedure ON_$Receiver(pkt)$
  if $(pkt[type] == pkt[DAO_{modified}])$ **then**
    Receive $DAO_{modified}$ from $\mathcal{C}$
    Add route to $\mathcal{C}$ in $\mathcal{RT}$ ▷ Add route in routing table
    Forward $DAO_{modified}$ to 6LBR
  else if $(pkt[type] == pkt[\mathcal{D}AO - ACK])$ **then**
    Forward $\mathcal{D}AO - ACK$ to $\mathcal{C}$
  else if $(pkt[type] == pkt[\mathcal{D}AO - NACK])$ **then**
    Forward $\mathcal{D}AO - NACK$ to $\mathcal{C}$
    $\mathcal{B} \leftarrow Src_{ip}$
    Remove route to $\mathcal{C}$ from $\mathcal{RT}$
    Remove Src_{ip} from $\mathcal{N}$
    $\mathcal{N}_{bl}++$
  else
    Do nothing
  end if
end procedure
procedure ON_6LBR(pkt)
  if $(pkt[type] == pkt[DAO_{modified}])$ **then**
    $\mathcal{L}_i \leftarrow \mathcal{R}eserved$
    $(\mathcal{CH} - \mathcal{R})_i \leftarrow \mathcal{PUF}_{data}(1, \ldots, Max_{node})$
    $r_i = \mathcal{CH}_i \oplus \mathcal{L}_i \cdots$ by Eq. 3
    if $(\mathcal{R}_i$ **equals** $r_i)$ **then**
      unicast $\mathcal{D}AO - ACK$
    else
      unicast $\mathcal{D}AO - NACK$
    end if
  end if
end procedure

fake License generated by the malicious node for registering the fake routes to overload the routing table of the parent node. LiSec-RTF is depicted in Algorithm 1.

D. Description of LiSec-RTF

The Algorithm 1 illustrates the pseudocode for LiSec-RTF, which is integrated into the DAO processing function within the $rpl - icmp6.c$ file. The DAO control message plays a pivotal role in registering routes at the 6LBR node and maintaining the network topology. LiSec-RTF is triggered whenever the sender unicasts a DAO message to the root node via its preferred parent to facilitate route registration and topology maintenance.

The Algorithm 1 contains three procedures. Procedure 1 is On_$Sender$, which denotes the Sender procedure. In this procedure, the sender is the child node, which prepares the $DAO_{modified}$ packet by inserting the $\mathcal{L}_i$ in the $\mathcal{R}eserved$ field of the DAO and sending it to the preferred parent P. The second procedure is On_$Receiver$, which denotes the receiver procedure. In this procedure, if $pkt[type]$ is $DAO_{modified}$, then it receives the packet from the child node, adds the route to the child in the routing table of P, and forwards the packet to the 6LBR. Other than this, if $pkt[type]$ is DAO-ACK, it states that it receives the DAO-ACK from 6LBR and forwards it to the child node $\mathcal{C}$. When $pkt[type]$ is DAO-NACK, it states that it receives the DAO-NACK from 6LBR. The parent node forwards this packet to the child, inserts the source address in the blacklist table, and removes the route to the child from the routing table and neighbor table. In the last procedure, On_6LBR, if $pkt[type]$ is $DAO_{modified}$, it receives the packet from P. The 6LBR extracts $\mathcal{L}_i$ from the $\mathcal{R}eserved$ field of the DAO. It extracts the child node PUF_{data}. The 6LBR calculates r_i using Eq. 3. If both response values are the same, then it unicasts the DAO-ACK; otherwise, it unicasts DAO-NACK to the child through P.

E. Mathematical formulation of LiSec-RTF

This section considers the assumed system and threat model as shown in Figure 3. The topology consists of a 6LBR

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

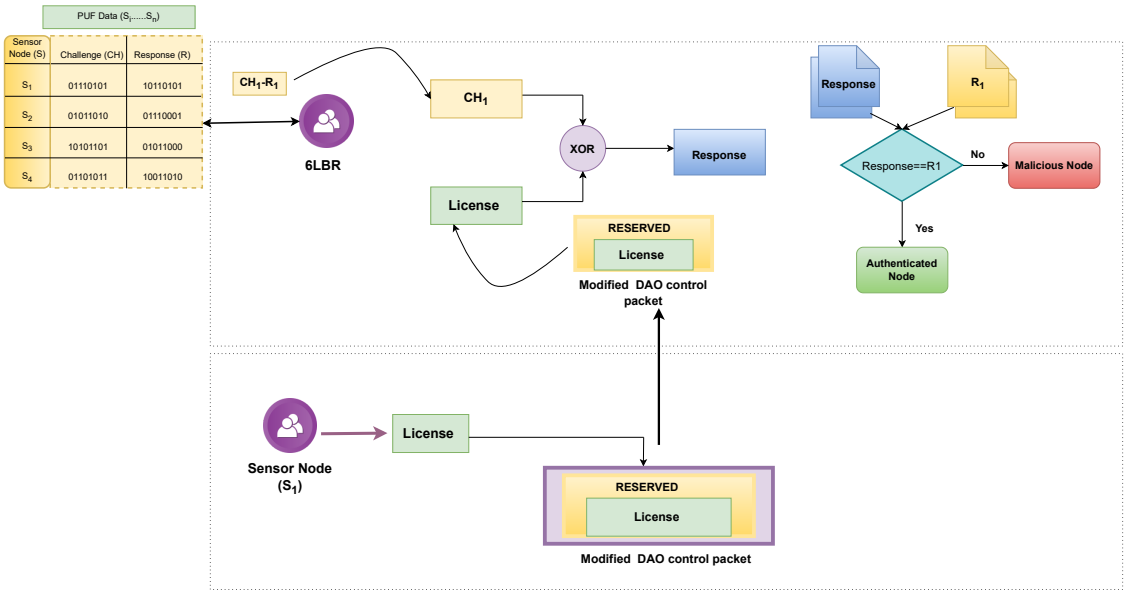


Fig. 7: Proposed Architecture of LiSec-RTF

and four sensor nodes (S_1, S_2, S_3 and S_4). The PUF data ($\mathcal{PUF}_{data}$) of these sensor nodes is deployed at 6LBR by using Eq. 1. The sensor node is configured with its License at the phase of deployment by using Eq. 2.

Each sensor node S_i has a unique PUF response, denoted as $\mathcal{PUF}_{data}$, represented as $(CH - R)_i$.

1) The License L_i of a sensor node is generated using $\mathcal{PUF}_{data}$:
$$L_i = CH_i \oplus R_i \tag{4}$$
2) The sensor node unicasts L_i in the $DAO_{modified}$ message.
3) Upon receiving the $DAO_{modified}$ message, the 6LBR extracts L_i and uses its stored CH_i to compute:
$$r_i = CH_i \oplus L_i$$
Substituting L_i from Eq. 4:
$$r_i = CH_i \oplus (CH_i \oplus R_i)$$
Using the XOR properties:
$$A \oplus A = 0, \quad \text{and} \quad A \oplus 0 = A$$
Applying this to simplify:
$$r_i = (CH_i \oplus CH_i) \oplus R_i$$
$$r_i = 0 \oplus R_i$$
$$r_i = R_i$$

Since r_i (computed at 6LBR) equals R_i (original response stored at 6LBR), the authentication condition $R_i = r_i$ holds **true**. The proof establishes that if the $DAO_{modified}$ message is not tampered with, the authentication check $R_i = r_i$ will always be **true** due to the properties of XOR. However, if a malicious node sends an incorrect L_i, the computed r_i at 6LBR will not match R_i, leading to authentication failure.

The main feature of this mitigation approach is to authenticate each sensor node at the 6LBR. The purpose of using PUF is to identify malicious nodes, as PUF generates a device's unique bit pattern (Response). The 6LBR sends an arbitrary bit pattern (Challenge) to the PUF-enabled device, which then produces an output bit pattern (Response). If the Response matches the stored Response, it confirms that the particular node is authentic. Therefore, if an attacker unicasts a DAO message with fake identities, the 6LBR can detect the malicious node and respond with a NACK.

In a brute-force attack scenario, an attacker may try all possible License combinations to gain authentication at the root node. Our proposed solution is secure against brute-force attacks, provided that a sufficiently large License size is used (16, 32, 64, 128 bits). In this paper, we use an 8-bit License stored in the unused RESERVED field of the DAO message to demonstrate how the proposed approach can be implemented in an RPL-based Contiki-NG environment. However, for better security against brute-force attacks, we have the option to store the License in the OPTIONS field of the DAO, which supports storing variable-sized information. Therefore, depending on the security requirements, a network administrator has the flexibility to utilize either the OPTIONS field or the RESERVED field to store Licenses for resisting brute-force attacks, based on the application or deployment scenario.

F. Analysis of LiSec-RTF

The $(\mathcal{CH} - \mathcal{R})_1$ for S_1 node is $(01110101 - 10110101)$. By using Eq. 2:

$$\mathcal{L}_1 = (01110101 \ \oplus \ 10110101)$$
$$\mathcal{L}_1 = 11000000$$

The $\mathcal{L}_1$ is inserted in the $\mathcal{R}eserved$ field of the $DAO_{modified}$ message and unicast to the 6LBR through the parent node (P). When the $DAO_{modified}$ received at the 6LBR, it extracts the $\mathcal{L}_1$ from the $\mathcal{R}eserved$ field. The 6LBR also extracts the $\mathcal{PUF}_{data}$ $(\mathcal{CH} - \mathcal{R})_1$ of the S_1 node. By using Eq. 3:

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

$$r_1 = (01110101 \oplus 11000000)$$
$$r_1 = 10110101$$

In this example, as we refer line number 36 of Algorithm 1 is true i.e., $(\mathcal{R}_1 equals\ r_1)$. Then, we state that sensor node S_1 is authenticated. Alternatively, the malicious node unicasts the DAO packet with the falsified route.

TABLE III: SIMULATION SCENARIO

Parameters	Values
Simulator	Cooja on Contiki-NG
Radio Model	Unit Disk Graph Medium (UDGM)
Mobility model	Random Waypoint Mobility Model
Size of Grid	$200m \times 200m$
Objective function	Minimum Rank with Hysteresis Objective Function (MRHOF)
Range of transmission	$50m$
Time of simulation	$1800s$ (30 minutes)
Node type	z1 mote
Speed of node	1-2 m/sec
Size of data packet	30 Bytes
Number of 6LBR node	1
Number of client nodes	29
Number of malicious nodes	1, 2, 3

VI. Performance Evaluation

A. Experimental Setup

In the simulations, the RTF attacker is implemented by making modifications to the existing file of rpl-icmp6.c in Contiki-NG. In the file, the attacker node unicast DAO message to its parent node with fake ip address in order to fill the routing table of its parent node. LiSec-RTF is implemented in Contiki-NG operating system. We evaluated the performance of the LiSec-RTF in Cooja simulator. The proposed solution is implemented by making modifications to the existing files within the rpl-classic mode of the Contiki-NG. The proposed solution can detect the identity spoofing attacks (A node impersonates another to mislead routing decision). Table III shows the details of simulation parameters. We have repeated the experiments with 10 different random seeds for statistically accurate results. Subsequently, we utilized the mean values of these results along with their errors, computed at a 95% confidence interval, to mitigate any potential biases in the findings.

B. Evaluation metrics

We employed Packet Delivery Ratio (PDR), Average End-to-End Delay (AE2ED), and Average Power Consumption (APC) to investigate the influence of an RTF attack on RPL and to verify the efficacy of the LiSec-RTF in both static and mobile environments. Additionally, the memory overhead of the LiSec-RTF on the Zoletria z1 mote are also examined. These measures are described as follows:

1) PDR: It represents the proportion of packets received successfully at 6LBR compared to the number of packets sent by each sensor node. It can be quantified using the Eq. 5

$$PDR = \frac{D_{receieved}}{\sum_{i=1}^{N} D_{sent_i}} \tag{5}$$

where $D_{received}$ represents the total count of data packets received at the sink, and D_{sent_i} denotes the total count of data packets sent by client node i.

2) AE2ED: AE2ED refers to the average time taken for data packets to travel from the client node to the 6LBR in a network. It is represented by Eq. 6

$$AE2ED = \frac{\sum_{i=1}^{N} D_{received_i}}{D_N} \tag{6}$$

where $D_{received_i}$ denotes the cumulative number of packets received by each client node i, while D_N signifies the time delay experienced by the data packet.

3) APC: It refers to the average power consumed by each client in a specified time. Eqs. 7 and 8 denote energy and power, respectively.

$$Energy(mJ) = (TX + RX + CPU + LPM) \tag{7}$$

TX represents the transmission, RX represents the receiving, LPM represents the low power mode, and CPU represents CPU time [22].

$$Power(mW) = \frac{Energy}{Tst} \tag{8}$$

where Tst represents the total simulation time in seconds.

C. Results and Discussion

In this section, we analyze the performance of the network through simulation. For comparison, we have chosen RPL, $RPL_{Under\ Attack}$, and LiSec-RTF. Where RPL denotes the standard RPL, it serves as a baseline for performance comparison, representing the unaltered behavior of the RPL in ideal scenarios. $RPL_{Under\ Attack}$ denotes the standard RPL that is being attacked by the insider node. The idea behind using this benchmark is to simulate RTF attacker nodes, which attempt to degrade network performance. LiSec-RTF refers to the secure version of RPL designed to counteract the RTF attack. The performance of these three scenarios has been validated on PDR, AE2ED, and APC metrics.

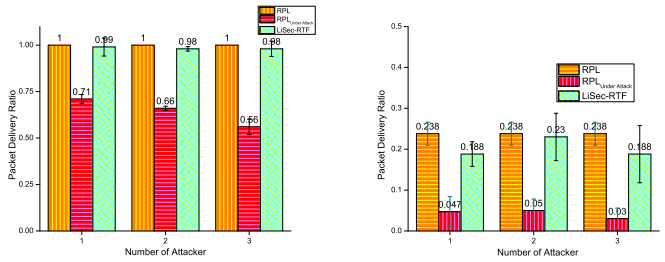
1) Analysis on PDR: PDR serves as a fundamental metric for evaluating the network's performance, reliability, and efficiency by successfully delivering data packets from client nodes to the sink node. We calculate PDR using Eq. 5. To measure the effect of an RTF attack in an IoT network, we have considered three cases: RPL, $RPL_{Under\ Attack}$, and LiSec-RTF. Figures 8a and 8b depict the values of PDR obtained while varying the count of malicious nodes in a static as well as mobile environment. In standard RPL (RPL), the sensor nodes are legitimate. The mean value of PDR in the case of RPL is 1 in static and 0.238 in mobile scenarios. The value of PDR is lesser in mobile scenarios because sensor nodes may move unpredictably. The routes between mobile nodes may be less stable due to node mobility, leading to frequent route changes in network topology. In the case of an RTF attack ($RPL_{Under\ Attack}$), the mean value of PDR is approximately 0.55 to 0.71 in the static scenario and 0.03 to 0.05 in a mobility environment while varying the number

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

of malicious nodes. By examining both scenarios, it becomes evident that the attack has a substantial impact on the mobile environment because the routes of the sensor nodes change very frequently in the network topology. When we simulate our proposed defense mechanism, the average value of PDR is 0.98 in the static scenario and 0.19 to 0.23 in the mobile scenario. As seen in Figures 8a and 8b, our proposed approach, LiSec-RTF, increased the PDR as the standard RPL in static as well as in the mobile environment as it detects fake DAO packets unicast by the malicious insider nodes.

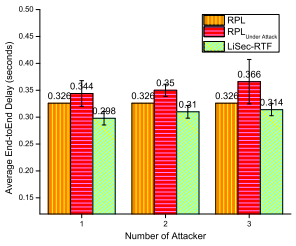
2) Analysis on AE2ED: We calculate the AE2ED by using Eq. 6. Figures 8c and 8d show the effect of AE2ED on RPL, $RPL_{Under\ Attack}$, and LiSec-RTF in the static and mobile environments while varying the number of malicious insider attackers. In the static scenario, the AE2ED for standard RPL (RPL) is approximately 0.326, while in a mobile environment, it decreases to around 0.162. The delay of the mobile scenario is less compared to static due to the proximity of the sensor to others, and sensor nodes can dynamically change their positions and network topology. The mean range of AE2ED in $RPL_{Under\ Attack}$ is 0.344 to 0.366 in a static environment and 0.104 to 0.134 in a mobile scenario. The delay in the static scenario is more than mobile because the sensor nodes may choose suboptimal or even nonexistent paths for forwarding packets. The mean values of AE2ED in the case of LiSec-RTF are 0.298 to 0.314 in static and 0.156 to 0.194 in mobile environments. The delay of LiSec-RTF is approximately the same as the RPL. Therefore, our proposed solution successfully addresses the RTF attack without imposing a delay on the network.

3) Analysis on APC: The RPL protocol is predominantly utilized in LLNs due to its energy-efficient routing capabilities. Hence, it is crucial to analyze the power consumption of nodes prior to deploying any new defense mechanism. As we compute the APC using Eq. 8, Figures 8e and 8f illustrate the comparison of the APC on standard RPL (RPL), $RPL_{Under\ Attack}$, and LiSec-RTF in static and mobile scenarios with varying the number of malicious nodes. The average power consumed in standard RPL (RPL) is 2.11mW in static and 2.28mW in mobile environments. The APC is more suitable for mobile scenarios compared to static because the continuous movement of the sensor nodes and the active communication between sensor nodes drain the battery faster in a mobility environment. The mean range of APC in $RPL_{Under\ Attack}$ is 2.85 to 3.01 in static and 2.49 to 2.71 in mobile scenarios with varied malicious nodes. The power consumption of sensor nodes increases in static and mobile environments because, during the RTF attack, sensor nodes may need to exchange more control messages to update and synchronize their routing tables. The increased control traffic results in higher power consumption. The mean range of LiSec-RTF is 2.88 to 3.03 in static and 2.76 to 2.85 in mobility environments. Our proposed defense mechanism, LiSec-RTF, slightly increases power consumption because the nodes need to recalculate their routes and update the routing table. Therefore, the processing overhead slightly increases the power on the network.

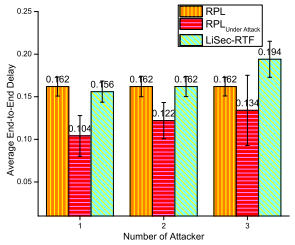


(a) PDR in static scenario

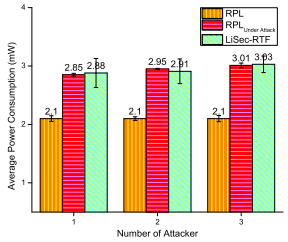
(b) PDR in mobile scenario



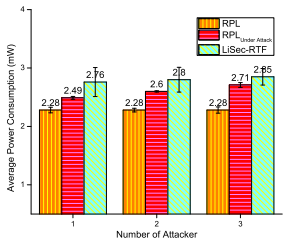
(c) AE2ED in static scenario



(d) AE2ED in mobile scenario



(e) APC in static scenario



(f) APC in mobile scenario

Fig. 8: PDR, AE2ED and APC obtained in different scenarios

4) Memory Overhead: Table IV illustrates the memory requirements of both standard RPL and LiSec-RTF. It is generally discouraged to employ a resource-consuming defense approach within the RPL protocol. Therefore, lightweight defense mechanisms have been proposed to facilitate the creation of resource-efficient networks. Using the msp-430 size tool, this study examines the effects of integrating LiSec-RTF on RAM and ROM usage. Table IV illustrates the memory requirements of $Mote_{RPL}$ (Contiki-NG firmware with RPL implemented) and $Mote_{LiSec-RTF}$ (Contiki-NG firmware with the proposed solution implemented). Based on the findings, there has been less than 1% increase in the RAM and ROM requirements for LiSec-RTF. It is important to remember that the $92KB$ is the maximum storage of the standard Z1 Mote. Thus, without causing significant overhead, LiSec-RTF is appropriate for z1 motes.

TABLE IV: Memory Requirements

File	RAM (Bytes)	ROM (Bytes)
udp-client.z1 (RPL)	6610	71653
udp-client.z1 (LiSec-RTF)	6610(+**762**)	71653 (+**290**)

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

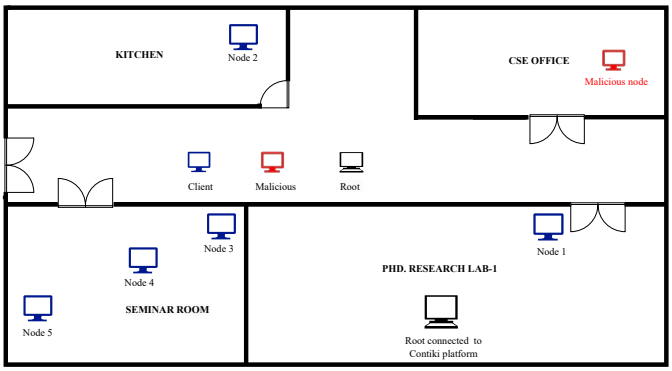


Fig. 9: A topological view of CC26X2R1 mote deployment consisting of 1 root attached to a desktop, 5 client nodes and 1 malicious node.

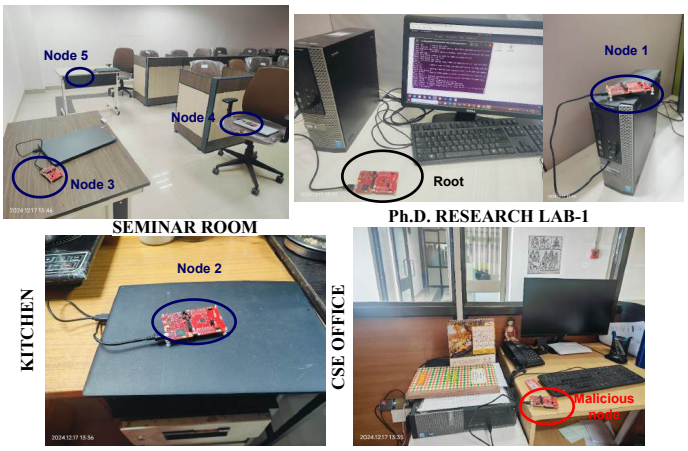
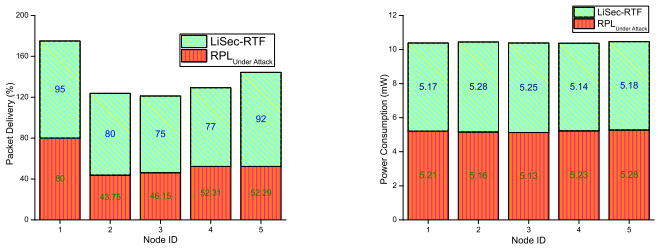


Fig. 10: Experimental testbed

D. Experimental Evaluation on Testbed

To examine the effect of the RTF attack on the DODAG of RPL networks, we conducted a series of testbed experiments. The experiments were designed using the open-source Contiki-NG operating system to generate the necessary binary executable files. These files were then flashed onto Texas Instruments (TI) CC26X2R1 LaunchPad devices [42]. In our experimental setup, one launchpad device was configured as the RPL root node, another as a malicious node, and the remaining devices were set up as legitimate nodes. For our testbed experiment, we utilized the PhD Research Lab, located on Level 3 of the CC building at IIITDM Jabalpur. Fig.9 shows the topological view of the testbed in the lab, with 7 CC26X2R1 sensor motes deployed for the experiment. Among them, 4 were sender motes, including 1 malicious mote, while 1 mote served as the root, connected to a desktop PC running the Contiki-NG Cooja emulation program. The client and malicious motes were evenly distributed throughout the 3rd floor of the CC building. Fig. 10 illustrates the experimental setup of the testbed on the third floor of the CC building at IIITDM Jabalpur. The root mote was connected to the Contiki-NG platform on the PC, and 1 additional node was deployed in the PhD Research Lab on the third floor. Additionally, the malicious mote was placed in the CSE office, while 3 client motes were deployed in the seminar room, and 1 mote was positioned in the kitchen area. We executed each set of experiments five times and used the mean values of the obtained results. The root initiates the network formation, and the malicious node joins the network as a legitimate node. Figure 11a shows the network's successful transmission of data packets. As a result of the attack, the PDR decreased by 55%. However, our proposed approach (LiSec-RTF) improved the PDR by up to 40% by successfully identifying the fake unicast DAO packets sent by malicious insider nodes.

Figure 11b shows the power consumption of each node throughout the experiment. We derived the node power consumption based on its radio duty cycle, using data collected from the 'energest' module of Contiki-NG. Due to RTF attack, already-joined nodes unnecessarily transmit DAO packets to overload the parent node's routing table, leading to increased control traffic and increased power consumption. LiSec-RTF integration imposes a very minor overhead to nodes in terms of power consumption due to the need for route recalculation and routing table updates.



(a) Packet Delivery Ratio (b) Power Consumption

Fig. 11: Experimental testbed results on packet delivery and power consumption by the nodes

VII. Enhanced LiSec-RTF with Lightweight Encryption

Several applications deal with sensitive data and have a low tolerance for security breaches. Assuming DAO messages are not encrypted, an adversary could easily obtain knowledge of the License through an eavesdropping attack. In such a case, an unencrypted LiSec-RTF would not perform well. Therefore, to address this issue, an encrypted LiSec-RTF model is also proposed. At present, various IoT hardware devices (e.g., TI CC26X2R1, CC2650, or similar low-power devices) support Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) due to built-in hardware accelerators [42] Moreover, embedded operating systems like Contiki-NG and TinyOS can be easily integrated with lightweight ECC variants such as TinyECC (Tiny Elliptic Curve Cryptography), MicroECC, NanoECC, and Curve25519 [43], [44], [45], [46]. Thus, LiSec-RTF can be integrated with lightweight ECC to secure License transmission, making it difficult for an attacker to extract the exact License through eavesdropping. Consequently, this prevents unauthorized authentication at the

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3581561

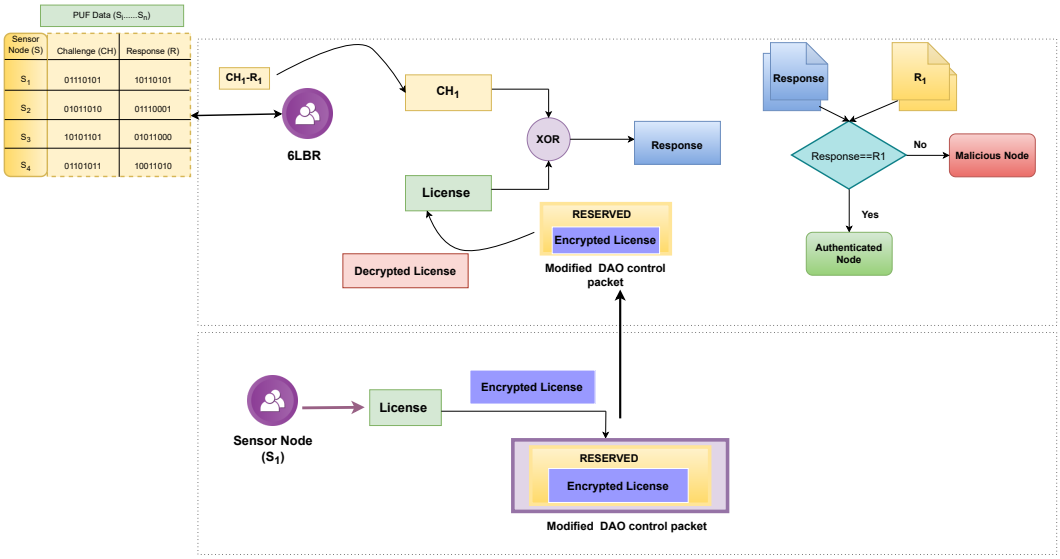


Fig. 12: Architecture of LiSec-RTF with Lightweight Encryption

root. The major modifications made to the unencrypted variant of LiSec-RTF are outlined below.

Instead of transmitting $\mathcal{L}_i$ directly, we encrypt it using a lightweight ECC before transmission. The encrypted License can be represented as (Eq. 9):

$$\mathcal{L}_i^{enc} = Encrypt(K_{shared}, \mathcal{L}_i) \tag{9}$$

where K_{shared} is a pre-shared symmetric key or dynamically generated session key between the sensor node and the 6LBR. As shown in Fig. 12, upon receiving the modified DAO message, the 6LBR decrypts $\mathcal{L}_i^{enc}$ before proceeding with validation. The equation for extracting r_i at the 6LBR as (Eq. 10):

$$r_i = \mathcal{CH}_i \oplus Decrypt(K_{shared}, \mathcal{L}_i^{enc}) \tag{10}$$

Therefore by encrypting License ($\mathcal{L}_i$) before transmission and decrypting it at the 6LBR, we ensure that an attacker cannot extract the License from intercepted DAO's. Additionally, the use of pre-shared symmetric key (K_{shared}) enhances the confidentiality of the data involved in the authentication process and mitigates the risk of eavesdropping attack.

VIII. Conclusion and Future Scope

The Routing Table Falsification (RTF) attack, an understudied routing threat initiated by an insider node, involves disseminating false information through DAO packets, leading to parent node routing table overflow. Extensive experiments reveal its adverse impact on packet delivery ratio. Given RPL's developmental stage, it lacks a defense mechanism against RTF. To address this, our paper proposes LiSec-RTF, a lightweight detection and mitigation solution. It utilizes a Physical Unclonable Function (PUF) to generate a unique authentication code (License) validated at the sink node. The widely used Cooja simulator for 6LoWPAN network analysis is used for carrying out experiments on Contiki-NG. The experimental observations confirm the effectiveness of LiSec-RTF in mitigating RTF impact in resource-constrained static and mobile sensor node environments. Moreover, LiSec-RTF does not impose any significant memory overhead on Zolertia z1 motes. In this research, we have also suggested an encrypted variant of LiSec-RTF to defend against eavesdropping attacks and prevent attackers from being maliciously authenticated. Our future goal is to extend this defense solution to counteract collaborative attacks and evaluate the encrypted variant of LiSec-RTF on a testbed.

References

[1] P. I. R. Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019.

[2] H. Espinoza, G. Kling, F. McGroarty, M. O'Mahony, and X. Ziouvelou, "Estimating the impact of the Internet of Things on productivity in Europe," *Heliyon*, vol. 6, no. 5, p. e03935, 2020.

[3] B. Safaei, A. M. H. Monazzah, and A. Ejlali, "ELITE: An Elaborated Cross-Layer RPL Objective Function to Achieve Energy Efficiency in Internet-of-Things Devices," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1169–1182, 2021.

[4] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial internet of things monitoring solution for advanced predictive maintenance applications," *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, 2017.

[5] 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. 16 623–16 638, 2018.

[6] A. Mayzaud, R. Badonnel, and I. Chrisment, "A Taxonomy of Attacks in RPL-based Internet of Things," *International Journal of Network Security*, vol. 18, no. 3, pp. 459–473, 2016.

[7] S. Murali and A. Jamalipour, "A Lightweight Intrusion Detection for Sybil Attack Under Mobile RPL in the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 379–388, 2020.

[8] G. Sharma, J. Grover, and A. Verma, "Performance evaluation of mobile RPL-based IoT networks under version number attack," *Computer Communications*, vol. 197, pp. 12–22, 2023.

[9] D. Airehrour, J. Gutierrez, and S. K. Ray, "Secure routing for internet of things: A survey," *Journal of Network and Computer Applications*, vol. 66, pp. 198–213, 2016.

[10] T. A. Al-Amiedy, M. Anbar, B. Belaton, A. A. Bahashwan, I. H. Hasbullah, M. A. Aladaileh, and G. A. Mukhaini, "A systematic literature review on attacks defense mechanisms in rpl-based 6lowpan of internet of things," *Internet of Things*, p. 100741, 2023.

[11] 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," Tech. Rep., 2012.

[12] A. Musaddiq, Y. B. Zikria, S. W. Kim *et al.*, "Routing protocol for low-power and lossy networks for heterogeneous traffic network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–23, 2020.

[13] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[14] J. Vasseur, N. Agarwal, J. Hui, Z. Shelby, P. Bertrand, and C. Chauvenet, "RPL: The IP routing protocol designed for low power and lossy networks," *Internet Protocol for Smart Objects (IPSO) Alliance*, vol. 36, 2011.

[15] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function," *Ad Hoc Networks*, vol. 96, p. 102001, 2020.

[16] S. M. Muzammal, R. K. Murugesan, and N. Jhanjhi, "A comprehensive review on secure routing in internet of things: Mitigation methods and trust-based approaches," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4186–4210, 2020.

[17] C. Ni and S. C. Li, "Machine learning enabled industrial iot security: Challenges, trends and solutions," *Journal of Industrial Information Integration*, p. 100549, 2024.

[18] A. Zilberman, A. Dvir, and A. Stulman, "Sprinkler: A multi-rpl man-in-the-middle identification scheme in iot networks," *IEEE Transactions on Mobile Computing*, 2024.

[19] I. Butun, P. Österberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 616–644, 2019.

[20] A. O. Bang and U. P. Rao, "EMBOF-RPL: Improved RPL for early detection and isolation of rank attack in RPL-based internet of things," *Peer-to-Peer Networking and Applications*, vol. 15, no. 1, pp. 642–665, 2022.

[21] A. A. R. A. Omar, B. Soudan *et al.*, "A comprehensive survey on detection of sinkhole attack in routing over low power and lossy network for internet of things," *Internet of Things*, p. 100750, 2023.

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

[23] H. Albinali and F. Azzedin, "Replay attacks in rpl-based internet of things: Comparative and empirical study," *Computer Networks*, vol. 257, p. 110996, 2025.

[24] A. Krari and A. Hajami, "Rpl-shield: A deep learning gnn-based approach for protecting iot networks from rpl routing table falsification attacks," in *International Conference on Digital Technologies and Applications*. Springer, 2024, pp. 117–127.

[25] A. O. Bang, U. P. Rao, P. Kaliyar, and M. Conti, "Assessment of Routing Attacks and Mitigation Techniques with RPL Control Messages: A Survey," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 44:1–44:36, 2023.

[26] H. Albinali and F. Azzedin, "Towards rpl attacks and mitigation taxonomy: Systematic literature review approach," *IEEE Transactions on Network and Service Management*, 2024.

[27] S. Goel, A. Verma, and V. K. Jain, "Cra-rpl: A novel lightweight challenge-response authentication-based technique for securing rpl against dropped dao attacks," *Computers & Security*, vol. 132, p. 103346, 2023.

[28] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," *Internet Engineering Task Force, RFC6206*, pp. 1–13, 2011.

[29] A. Al-Meer and S. Al-Kuwari, "Physical unclonable functions (puf) for iot devices," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–31, 2023.

[30] C. Pu and K.-K. R. Choo, "Lightweight Sybil attack detection in IoT based on bloom filter and physical unclonable function," *Computers & Security*, vol. 113, p. 102541, 2022.

[31] A. Kamble, V. S. Malemath, and D. Patil, "Security attacks and secure routing protocols in rpl-based internet of things: Survey," in *2017 International Conference on Emerging Trends & Innovation in ICT (ICEI)*. IEEE, 2017, pp. 33–39.

[32] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "Dio suppression attack against routing in the internet of things," *IEEE Communications Letters*, vol. 21, no. 11, pp. 2524–2527, 2017.

[33] C. Pu, "Mitigating dao inconsistency attack in rpl-based low power and lossy networks," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2018, pp. 570–574.

[34] C. Pu and T. Song, "Hatchetman attack: A denial of service attack against routing in low power and lossy networks," in *2018 5th IEEE international conference on cyber security and cloud computing (CSCloud)/2018 4th IEEE International conference on edge computing and scalable cloud (EdgeCom)*. IEEE, 2018, pp. 12–17.

[35] C. Pu, "Spam dis attack against routing protocol in the internet of things," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 73–77.

[36] ——, "Energy depletion attack against routing protocol in the Internet of Things," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–4.

[37] C. Pu, J. Brown, and L. Carpenter, "A theil index-based countermeasure against advanced vampire attack in internet of things," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2020, pp. 1–6.

[38] N. Cam-Winget, J. Hui, and D. Popa, "Applicability statement for the routing protocol for low-power and lossy networks (rpl) in advanced metering infrastructure (ami) networks," Tech. Rep., 2017.

[39] P. Kaliyar, W. B. Jaballah, M. Conti, and C. Lal, "Lidl: localization with early detection of sybil and wormhole attacks in iot networks," *Computers & Security*, vol. 94, p. 101849, 2020.

[40] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future iot applications," *Ieee Access*, vol. 5, pp. 3028–3043, 2017.

[41] J. Guajardo, *Physical Unclonable Functions (PUFs)*. Boston, MA: Springer US, 2011, pp. 929–934.

[42] T. Instruments, "CC2652RSimpleLink Multiprotocol 2.4 GHz Wireless MCU datasheet (Rev. J)," https://www.ti.com/lit/ds/swrs207j/swrs207j.pdf, 2025.

[43] M. Rana, Q. Mamun, and R. Islam, "Lightweight cryptography in iot networks: A survey," *Future Generation Computer Systems*, vol. 129, pp. 77–89, 2022.

[44] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: a solution to secure iot," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.

[45] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks," in *Wireless Sensor Networks: 5th European Conference, EWSN 2008, Bologna, Italy, January 30-February 1, 2008. Proceedings*. Springer, 2008, pp. 305–320.

[46] T. Fadia and L. Toufik, "Elliptic curves cryptography for lightweight devices in iot system," *Brazilian Journal of Technology*, vol. 7, no. 4, pp. e73 725–e73 725, 2024.