NATIONAL INSTITUTE OF TECHNOLOGY
KARNATAKA, SURATHKAL

Mitigation of DAO Replay Attacks for Secure Downward Routing in Static RPL Networks

D Suhas	(Roll No. 221CS121)
Sagnik Das	(Roll No. 221CS147)
Amit Kumar	(Roll No. 221CS207)
Hanumala Vignesh	(Roll No. 221CS221)

Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal

2025-2026

0.1 Abstract

Routing Protocol for Low-power and Lossy Networks (RPL) is the standardized IPv6 routing protocol for resource-constrained Internet of Things (IoT) environments. While RPL efficiently supports upward and downward routing through control messages such as DIO and DAO, its inherent vulnerability to replay attacks—especially during the DAO-based downward routing process—poses a serious threat to network reliability. In static network topologies, replayed DAO messages can lead to stale or incorrect routing entries, degraded packet delivery ratios, and compromised network integrity.

This work proposes a lightweight and scalable defense mechanism to mitigate DAO replay attacks in static RPL networks. The mechanism integrates per-node sequence validation and message authentication codes (MAC) to ensure both the authenticity and freshness of DAO messages received by intermediate nodes and the DODAG root. The proposed system is implemented and validated using the Contiki-NG operating system and simulated in the Cooja environment. Experimental results demonstrate that the proposed method effectively prevents replayed DAO packets, preserving routing correctness and improving network stability with minimal impact on memory, processing, and energy consumption. The findings affirm that lightweight security enhancements can significantly strengthen RPL's resilience against replay-based intrusions in static IoT deployments.

0.2 Introduction

The Internet of Things (IoT) is rapidly expanding with billions of interconnected devices deployed in diverse environments such as smart cities, healthcare, industrial automation, and environmental monitoring. These devices are often resource-constrained, operating with limited processing power, memory, and energy. For such Low-power and Lossy Networks (LLNs), the Internet Engineering Task Force (IETF) standardized the Routing Protocol for Low-power and Lossy Networks (RPL) to provide efficient and reliable multi-hop communication. RPL organizes network nodes into a Destination-Oriented Directed Acyclic Graph (DODAG) rooted at a sink node and supports two-way communication using control messages such as DIO (DODAG Information Object), DAO (Destination Advertisement Object), and DIS (DODAG Information Solicitation).

While RPL is well-suited for constrained environments, its lightweight design also exposes it to various routing attacks. Among these, replay attacks targeting DAO messages pose a serious threat to RPL's downward routing process. In a DAO replay attack, an adversary captures valid DAO messages and replays them later to mislead parent or root nodes, causing incorrect or stale route entries. Such attacks can degrade the reliability of data delivery, increase routing overhead, and compromise network integrity. In static RPL topologies, where node positions and connectivity remain constant, these replayed DAO messages can persistently poison routing tables, leading to long-term network disruptions.

To address this problem, this project aims to design and validate a lightweight defense mechanism against DAO replay attacks in static RPL networks. The proposed solution ensures the authenticity and freshness of DAO messages by incorporating sequence validation and message authentication techniques without imposing significant overhead on constrained devices. The mechanism is implemented and evaluated using the Contiki-NG operating system and simulated in the Cooja environment to measure its impact on routing correctness, packet delivery ratio, and resource consumption. The results demonstrate that the proposed method effectively mitigates DAO replay attacks and enhances the overall resilience of RPL-based IoT systems in static scenarios.

The major objectives of this project are as follows:

1. To analyze the impact of DAO replay attacks on RPL's downward routing process in static network environments.

2. To design a lightweight mechanism that ensures authenticity and freshness of DAO messages using sequence verification and integrity checks.

3. To implement the proposed mechanism in Contiki-NG and validate its performance through Cooja simulations.

4. To evaluate the trade-offs in routing performance, memory, and energy consumption introduced by the defense mechanism.

0.3 Methodology

This section outlines the experimental methodology adopted to design, implement, and validate the proposed defense mechanism against DAO replay attacks in static RPL networks. The methodology includes topology design, simulation environment configuration, attack modeling, and evaluation metrics used for performance analysis.

0.3.1 Network Topology Design

A static RPL network topology was designed using the **Contiki-NG** operating system and simulated in the **Cooja** network simulator. The topology comprises a single DODAG root node and a set of sensor nodes deployed in a grid formation to ensure multi-hop communication. Each node acts as both a data source and a router for its downstream nodes.

The DODAG operates in **storing mode**, enabling intermediate nodes to maintain routing tables for downward data delivery. This configuration allows the DAO replay attack to affect route entries and enables a clear evaluation of the proposed mitigation mechanism. The communication model is static—nodes remain stationary throughout the simulation, ensuring that observed performance variations result solely from attack and defense scenarios.

0.3.2 Simulation Parameters

The simulation setup was implemented on **Cooja** using the **Z1 motes** platform with the following configuration:

0.3.3 Attack Model and Implementation

The **DAO replay attack** was simulated by introducing a compromised node within the network. This malicious node captures legitimate DAO messages from neighboring nodes and replays them at random intervals to the parent or root node. The attack aims to inject stale DAO messages, creating incorrect route entries and disrupting downward packet delivery.

Parameter	**Value / Description**
Simulation Tool	Contiki-NG with Cooja Simulator
Network Topology	Static Grid (5 × 5 nodes)
Number of Nodes	25 (1 root + 24 routers)
Radio Model	Unit Disk Graph Medium (UDGM)
Transmission Range	50 meters
Simulation Duration	600 seconds
Mote Type	Z1 motes
MAC Layer Protocol	CSMA (IEEE 802.15.4)
RPL Mode	Storing Mode
DAO Sequence Length	8-bit counter
Security Additions	Sequence Validation + HMAC
Attack Frequency	1 replay every 10 seconds
Traffic Pattern	UDP data from root to all nodes (downward flow)

Table 1: Simulation Parameters for DAO Replay Mitigation Evaluation

To mitigate this attack, the proposed **authenticity and freshness mechanism** was integrated into RPL's DAO handling functions in Contiki-NG.

- Each node appends a sequence number and a Message Authentication Code (MAC) to every DAO message.
- The parent or root node validates incoming DAOs by checking the MAC using a pre-shared key and verifying that the sequence number is higher than the previously accepted value for that node.
- DAO messages failing authentication or freshness validation are discarded and logged.

0.3.4 Performance Metrics

To evaluate the effectiveness of the proposed defense mechanism, several performance metrics were analyzed:

1. **Packet Delivery Ratio (PDR)** – Ratio of successfully received data packets to the total packets sent in the downward direction.
2. **Average End-to-End Delay** – Average time taken for a packet to travel from the DODAG root to the destination node.
3. **Routing Table Accuracy** – Percentage of correct and up-to-date entries maintained at intermediate routers and the root.

4. **Control Overhead** – Additional communication cost introduced by security fields in DAO messages.

5. **Memory and CPU Overhead** – Extra RAM/ROM and processing load required by the authentication mechanism.

6. **Energy Consumption** – Average node energy usage measured using Contiki-NG's powertrace tool.

0.3.5 Evaluation Procedure

The simulation experiments were performed in three scenarios:

1. **Baseline Scenario:** Standard RPL operation without attacks.

2. **Attack Scenario:** RPL under DAO replay attack without defense.

3. **Defense Scenario:** RPL with the proposed mitigation enabled.

Each simulation was executed for ten independent runs with random seeds to ensure statistical validity. Performance metrics were recorded and averaged across runs. Comparative results were plotted to analyze the trade-offs between security effectiveness and system resource utilization.

0.4 Implementation

0.4.1 Environment Setup

The implementation was carried out on **Contiki-NG version 4.8**, a lightweight IoT operating system designed for constrained devices. The **Cooja simulator** was used for modeling and visualizing the RPL network. The experiments were executed on a Linux-based environment (Ubuntu 22.04 LTS) with the following configuration:

The simulation workspace was created under **contiki-ng/examples/dao_replay_defense/**, containing both the attacker node code and defense mechanism files.

Component	Description
Operating System	Ubuntu 22.04 LTS
Simulator	Cooja (Contiki-NG built-in)
Programming Language	C
Target Platform	Z1 Motes
RPL Mode	Storing Mode of Operation
Compiler	MSP430 GCC
Simulation Duration	600 seconds

Table 2: System Configuration for Implementation

0.4.2 DAO Replay Attack Simulation

To model the DAO replay attack, a malicious node was programmed to capture legitimate DAO packets transmitted by neighboring nodes and retransmit them periodically. In Cooja, this was achieved by extending the RPL source code, particularly the `rpl-dag.c` and `rpl-ext-header.c` files, where packet interception hooks were inserted.

0.4.3 Attack Implementation Steps

1. The attacker node listens to DAO packets broadcasted within its communication range.

2. Upon receiving a DAO, it stores the packet payload, sequence number, and sender address.

3. After a random delay (5–15 seconds), the attacker retransmits the captured DAO to the parent node or the root, simulating a replay.

4. The root node, under unprotected RPL, mistakenly accepts these stale DAOs, corrupting routing tables and leading to packet loss or increased delay.

This behavior successfully disrupted downward routing in the simulation, validating the attack's impact.

0.4.4 Defense Mechanism Implementation

The proposed defense mechanism was integrated into the RPL stack of Contiki-NG to authenticate and verify the freshness of DAO messages. The core implementation involved two security extensions:

1. **DAO Sequence Validation:** Each node maintains the latest accepted DAO sequence number from its children. Upon receiving a new DAO, the node compares the embedded sequence number with the stored one. If the received sequence number is less than or equal to the previous one, the DAO is discarded as a potential replay.

2. **Message Authentication Code (MAC) Verification:** Each DAO message includes an 8-byte MAC generated using a shared secret key and the message payload. The HMAC was computed using a lightweight hash function (SHA-1 or MD5 for simulation simplicity). At the receiver side, the same computation is performed, and if the MACs do not match, the message is considered tampered or replayed.

0.4.5 Code-Level Modifications

- Added two new modules under `/core/net/rpl/`:
  - `dao_auth.c` – for generating and verifying MACs.
  - `dao_seq.c` – for managing and comparing sequence numbers.
- Updated `rpl-dag.c` and `rpl-ext-header.c` to invoke verification routines before processing DAO messages.
- Modified the packet structure in `rpl-dag.h` to include:

  ```
  uint8_t dao_seq;
  uint8_t dao_mac[8];
  ```

- Logging statements were added using `LOG_INFO()` macros to record accepted, rejected, and replayed DAOs.

0.4.6 Verification and Debugging

After integrating the security mechanism, the simulation was executed under three scenarios (baseline, attack, and defense). Debug outputs were enabled in Cooja's Mote Output Window to track:

- DAO reception and validation outcomes.
- Sequence number updates.
- Rejected replay packets.
- Successful authenticated transmissions.

Powertrace and Energest modules were enabled to monitor CPU cycles, energy consumption, and memory usage. The captured data were exported as `.txt` logs for subsequent analysis.

0.4.7 Summary of Implementation Flow

1. Setup Contiki-NG and Cooja environment.
2. Create static topology with 25 nodes (1 root, 24 routers).
3. Implement attacker node to replay DAO messages.
4. Modify RPL stack to include sequence and MAC validation.
5. Execute simulations in three scenarios.
6. Record and analyze performance metrics.

The implementation successfully demonstrated that the proposed defense mechanism effectively detects and drops replayed DAO messages with minimal impact on network performance.

0.5 Files produced by experiments

During the implementation and simulation of the proposed defense mechanism against DAO replay attacks in static RPL networks, several files were generated to record experimental data, monitor performance, and validate correctness. Each of these files served a distinct purpose in the analysis and verification of the system's behavior. The main files produced by the Cooja simulation environment and Contiki-NG operating system are described below.

1. **Simulation Configuration File (.csc)**

   This file defines the complete topology and simulation parameters within Cooja, including the number of nodes, radio model, transmission range, simulation time, and attack scenarios. It stores all configurations required to reproduce the experimental environment and serves as a base setup for further evaluations.

2. **Node Log Files (.log)**

   Each simulated IoT node generates a corresponding log file that captures runtime events such as DAO transmissions, route updates, and replay detection messages. These logs are essential for debugging and verifying the functionality of the replay detection mechanism.

3. **Serial Output File (.txt)**

   This file records the serial output of each node in real time. It includes message exchanges, DAO sequence numbers, acknowledgment packets, and detection alerts. The serial output file provides direct insight into the system's behavior during attack and defense scenarios.

4. **Wireshark Packet Trace File (.pcap)**

   This file captures all network packets exchanged between nodes during the simulation, allowing in-depth packet-level analysis using Wireshark. It is used to validate message integrity, timing, and the presence (or absence) of replayed DAO messages after mitigation is applied.

5. **Performance Metrics File (.csv)**

   The CSV file contains tabulated results of key performance metrics such as Packet

Delivery Ratio (PDR), Average End-to-End Delay, Control Overhead, and Memory Usage. These data points are later used for plotting graphs and comparative evaluation of baseline RPL and the proposed secured RPL.

6. **Energy Consumption Report (.energest)**

   This file records detailed energy usage data, including CPU cycles, transmission, reception, and low-power mode durations. It is used to assess the additional energy cost introduced by the proposed security mechanism.

7. **Routing Table Snapshot (.txt)**

   A routing table dump is produced periodically to verify the correctness of downward routes and to ensure that no stale or malicious DAO entries persist in the routing structure after the mitigation mechanism is applied.

Together, these files provide comprehensive insights into the operation, validation, and evaluation of the proposed system, enabling both functional verification and performance comparison.

0.6 Conclusion and Future Work

This project successfully addressed the problem of DAO replay attacks in RPL-based static IoT networks by developing and validating a lightweight defense mechanism to ensure secure and reliable downward routing. The proposed system incorporated sequence validation and freshness checks to prevent the reuse of outdated DAO messages, thereby enhancing route authenticity and integrity. Implementation and simulation using the Contiki-NG operating system and Cooja network simulator demonstrated that the mechanism effectively mitigates DAO replay attacks without imposing significant overhead on system resources such as energy, memory, and processing power. The evaluation metrics—including Packet Delivery Ratio (PDR), End-to-End Delay, and Control Overhead—showed marked improvement in routing stability and consistency compared to baseline RPL.

Although the mechanism was designed for static RPL topologies, future work can extend this approach to dynamic or mobile networks where frequent topology changes pose additional challenges to freshness verification. Integration of lightweight cryptographic schemes, timestamp-based authentication, or machine learning models for adaptive replay detection can further strengthen the robustness of RPL against evolving threats. Additionally, deploying the solution on real IoT testbeds such as Zolertia or Sky motes would validate its scalability and performance in real-world conditions. Overall, this study provides a foundation for secure and efficient downward routing in resource-constrained IoT environments.