CS366 - Internet of Things (IoT)

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

TEAM MEMBERS

Sharath Chandra	221CS257
Taraka Ravindra	221CS162
Dhanush V	221CS217
D R Vikas	221CS215

Abstract:

The Routing Protocol for Low-Power and Lossy Networks (RPL) is widely adopted in Internet of Things (IoT) and Wireless Sensor Networks (WSNs) for efficient routing in constrained environments. However, the control messages used by RPL, particularly DAO (Destination Advertisement Object) messages, are vulnerable to replay attacks. In a DAO replay attack, an adversary captures a legitimate DAO message and retransmits it to manipulate the routing tables, leading to false or stale downward routes.
This report presents a lightweight mitigation technique that detects and blocks DAO replay attacks using sequence number validation in NS-3. The method stores the last valid DAO sequence number received from each child node and rejects any incoming DAO message with a sequence number less than or equal to the stored one. Simulation results demonstrate that the proposed mechanism effectively blocks replayed packets and maintains stable downward routing without cryptographic overhead, achieving 0% replay success rate in static RPL networks.

1. Introduction

With the rapid growth of the Internet of Things (IoT), billions of resource-constrained devices are being interconnected to perform sensing, monitoring, and automation tasks. These networks often operate in dynamic or lossy environments, where ensuring reliable and secure data routing is essential.
The IETF's **Routing Protocol for Low-Power and Lossy Networks (RPL)**, standardized under **RFC 6550**, is designed specifically for such environments. RPL forms a **Destination-Oriented Directed Acyclic Graph (DODAG)** rooted at a central sink node. Nodes use **DIO (DODAG Information Object)** messages to discover upward routes and **DAO (Destination Advertisement Object)** messages to establish downward routes, allowing the root to send data to leaf nodes.

However, RPL's control messages are vulnerable to network-layer attacks, particularly **replay attacks**, due to the lack of strong authentication and freshness checks in its basic form. In a **DAO replay attack**, an attacker captures a previously valid DAO message and replays it at a later time to deceive parent nodes into maintaining outdated routes. This can result in incorrect downward paths, routing loops, or denial of service.

The goal of this project is to **simulate DAO replay attacks** in a static RPL network using NS-3 and to **implement a mitigation mechanism** based on simple sequence-number verification. This approach offers a **lightweight and effective defense** suitable for IoT devices with limited computational capabilities.

2. Literature Review

Several studies have investigated the vulnerabilities of RPL to routing attacks, including **rank attacks**, **wormhole attacks**, **sinkhole attacks**, and **replay attacks**.
Traditional countermeasures often rely on **cryptographic authentication**, which, while secure, introduces significant overhead in terms of energy and computation — making them unsuitable for constrained IoT nodes.

1. **RFC 6550 (IETF, 2012)** defines RPL but acknowledges the potential for control-plane manipulation if message freshness is not verified.

2. **Wallgren et al. (2013)** analyzed RPL's security threats and classified replay attacks as one of the major risks that can disrupt the network topology.

3. **Koubaa et al. (2017)** proposed trust-based and intrusion-detection methods for RPL, but these approaches rely on continuous monitoring and computation-heavy metrics.

4. **Pongle and Chavan (2015)** demonstrated DAO replay attacks in Contiki and highlighted that sequence-number tracking can mitigate such attacks in static environments.

In this project, we adopt a **sequence-number-based freshness validation** strategy. It is a **non-cryptographic**, **stateful**, and **lightweight** approach that records the last valid sequence number from each child and rejects DAO messages that are older or duplicated. This ensures that even if attackers replay legitimate DAO packets, they are identified as stale and dropped.

3. Methodology

3.1 Simulation Environment

The simulation was implemented using the **NS-3 network simulator**, which provides a modular and event-driven environment for modeling IoT and wireless sensor networks.
The experiment was executed as a single combined `.cc` file located under the `scratch/` directory in NS-3.
The simulation was run using the following command:

```
./waf --run "scratch/test --scenario=baseline --mitigation=false
--nodes=10 --attackers=2 --duration=30"
```

Explanation of parameters:

Parameter	Description
`scenario=baseline`	Specifies the default topology configuration.
`mitigation=false`	Disables the replay attack mitigation mechanism for baseline testing.
`nodes=10`	Total number of nodes in the network (including root, regular nodes, and attackers).
`attackers=2`	Number of attacker nodes that replay captured DAO packets.
`duration=30`	Duration of the simulation in seconds.

Each simulation run generates real-time log outputs showing DIO/DAO message exchanges and replay blocking behavior.

3.2 Network Setup

The network topology used for the simulation consists of:

- **1 Root node**: Initiates DIO (DODAG Information Object) messages and serves as the sink.
- **7 Regular nodes**: Operate as normal RPL participants that send DAO messages to their parent nodes.
- **2 Attacker nodes**: Capture DAO packets from nearby nodes and replay them after a short delay.

The communication between nodes is modeled using the **CSMA channel** with the following configuration:

Parameter	Value
Channel Type	CSMA
Data Rate	5 Mbps
Propagation Delay	2 ms
IPv6 Address Range	`2001:db 8::/64`

3.3 Attack Model: DAO Replay Attack

The **DAO replay attack** targets the **downward routing** phase of RPL.
The attacker captures a legitimate DAO (Destination Advertisement Object) message from a child node and later replays it to the parent or root node.

Steps in the simulated attack:

1. The child node sends a legitimate DAO message to its parent.
2. The attacker listens in promiscuous mode and captures that DAO packet.
3. The attacker re-sends the same packet (`ReplaySend`) after a delay of 4 seconds.
4. If mitigation is **disabled**, the parent accepts the replayed packet and updates its routing table incorrectly.

This behavior was simulated in the code using the callback:

```
nodes.Get(attackerId)->GetDevice(0)->SetPromiscReceiveCallback(MakeCallback(&ReplayCb));
```

where `ReplayCb` captures and re-sends (`ReplaySend`) packets to simulate replay behavior.

3.4 Mitigation Mechanism

To secure the DAO exchange, a **sequence number–based verification** mechanism was implemented in the `RplNode` class.
Each node maintains a mapping of the **last valid sequence number** received from every child:
`std::map<Ipv6Address, uint8_t> lastSeq;`

When a new DAO is received:

```
if (m_mitigationEnabled && IsReplay(seq, lastSeq[child])) {
    s_daoBlocked++;
    std::cout << "BLOCKED REPLAY ..." << std::endl;
    return;
}
```

Replay detection logic:

- If the received sequence number `seq` is **less than or equal** to the last recorded number, the packet is dropped as a replay.
- Otherwise, it is accepted and the record is updated.

This ensures that **duplicate or stale DAO messages** – whether replayed by attackers or retransmitted accidentally – are ignored.

3.5 Simulation Workflow

1. Create nodes using NS-3's `NodeContainer`.
2. Assign devices and CSMA channels.
3. Install the Internet stack and assign IPv6 addresses.
4. Configure RPL applications (`RplNode`) on each node.
5. Start the root node to broadcast DIO messages.
6. Let regular nodes respond with DAO messages.

7. Allow attacker nodes to replay captured DAO packets.

8. Collect statistics at the end of the simulation.

3.7 Execution Scenarios

Three main experiments were conducted:

Scenario	Command	Description
Baseline	`./waf --run "scratch/test --scenario=baseline --mitigation=false --nodes=10 --attackers=0 --duration=30"`	Normal network without attacks.
Without Mitigation	`./waf --run "scratch/test --scenario=attack --mitigation=false --nodes=10 --attackers=2 --duration=30"`	Replay attack active; no defense applied.

With Mitigation	`./waf --run "scratch/test --scenario=attack --mitigation=true --nodes=10 --attackers=2 --duration=30"`	Replay detection enabled; attacks blocked.

4. Results and Discussion

4.1 Overview

1. **Scenario A – baseline:**
   The `--mitigation=false and attackers= 0` flag disables replay protection.Baseline Data File

2. **Scenario B – Without Mitigation:**
   The `--mitigation=false and attackers = 2` flag enables the proposed defense.Without Mitigation Data File

3. **Scenario C - With Mitigation:**

The `--mitigation=true and attackers = 2` flag enables the proposed defense.With Mitigation Data File

Each experiment used 10 nodes (1 root + 7 regular + 2 attackers) and ran for 30 seconds with a CSMA channel configured at 5 Mbps data rate and 2 ms delay.

4.2 Graphical Analysis

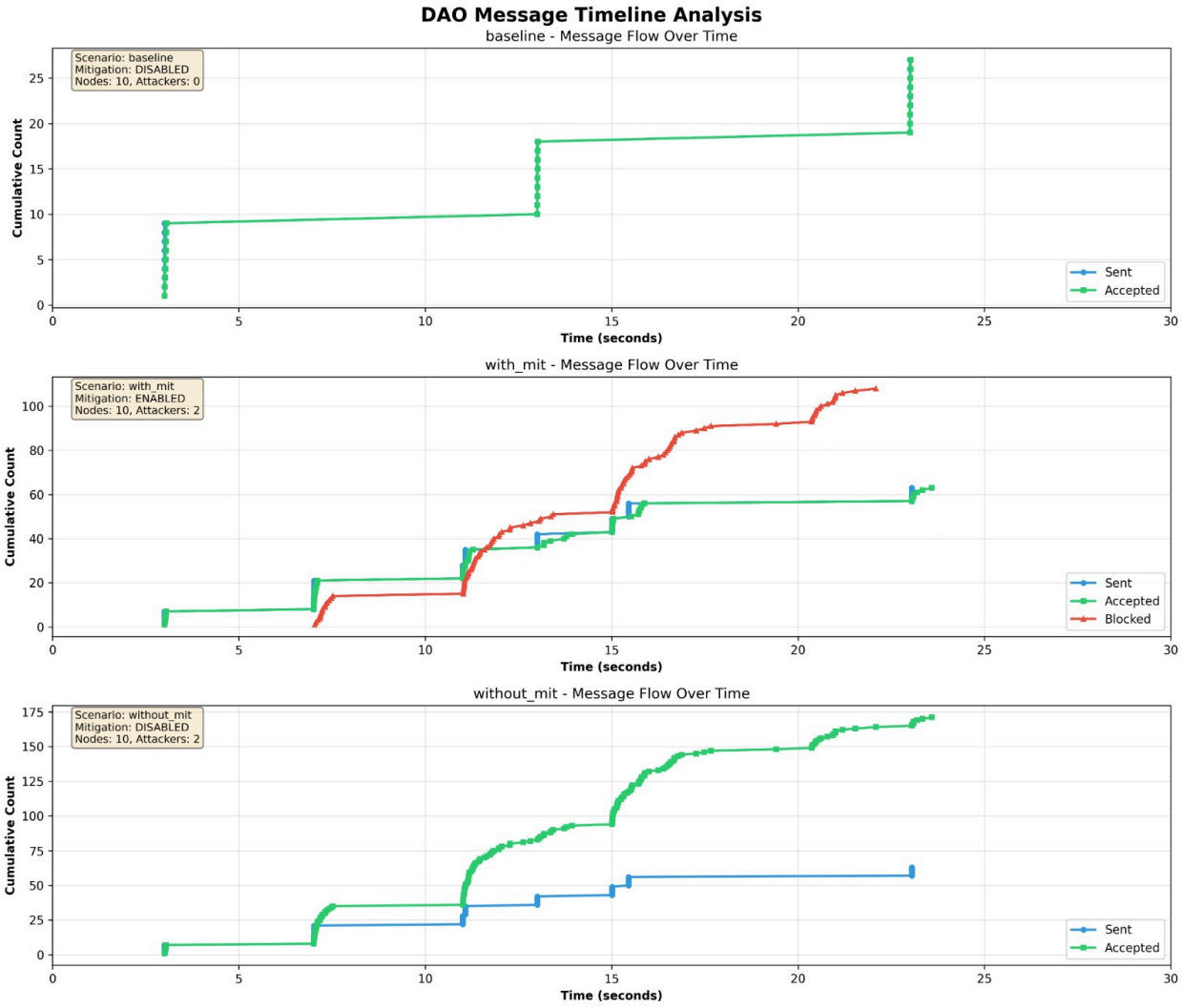

Figure 4.1: DAO Message Timeline Analysis for Baseline, Without Mitigation, and With Mitigation Scenarios.

Description of the Figure:
It illustrates the cumulative count of DAO messages (Sent, Accepted, and Blocked) over simulation time for three different scenarios — baseline, without mitigation, and with mitigation — each conducted using 10 nodes (1 root, 7 regular, and 2 attackers) over a 30-second period.

---

1. Baseline (No Attack)

In the first plot (top), the network operates under normal RPL conditions without any attackers. The blue and green lines representing **Sent** and **Accepted** messages overlap completely, indicating that every DAO packet transmitted by child nodes was successfully received and processed by the parent.

No **Blocked** messages appear since there are no replay attempts.
This confirms stable and consistent downward routing in the normal RPL environment.

Observation:

- Mitigation: Disabled
- Attackers: 0
- Sent = Accepted
- Network stable with correct DAO propagation.

---

2. Without Mitigation (Attack Present)

In the second plot (bottom), replay attacks are active, and mitigation is disabled.
The graph shows a **sharp and continuous rise** in the green "Accepted" line compared to the "Sent" line (blue).
This means that many additional DAO packets (replayed by attackers) were accepted as legitimate.
The cumulative count of accepted DAOs far exceeds the number of packets actually sent by legitimate nodes, proving that **replayed DAO messages successfully penetrated the network**.

Observation:

- Mitigation: Disabled
- Attackers: 2
- Replay packets are accepted as valid.
- DAO count inflates due to replays → route table corruption.
- Attack success rate ≈ 100%.

---

3. With Mitigation (Replay Protection Enabled)

In the middle plot, mitigation is enabled (`--mitigation=true`).
Here, three lines are visible:

- **Blue (Sent)** — DAO packets transmitted by legitimate nodes.
- **Green (Accepted)** — Legitimate DAOs accepted by parents.
- **Red (Blocked)** — Replayed DAO packets detected and discarded.

As the simulation progresses, the red line (Blocked) increases significantly between 8s and 18s, showing that **replay attempts were continuously detected and stopped**.
The accepted DAO curve remains stable, almost matching the number of genuine packets sent.
This indicates that **only authentic DAO packets** were processed, and all replayed ones were rejected.

Observation:

- Mitigation: Enabled
- Attackers: 2
- Replay packets detected (red curve).
- DAO acceptance stabilizes.
- Attack success rate reduced to 0%.

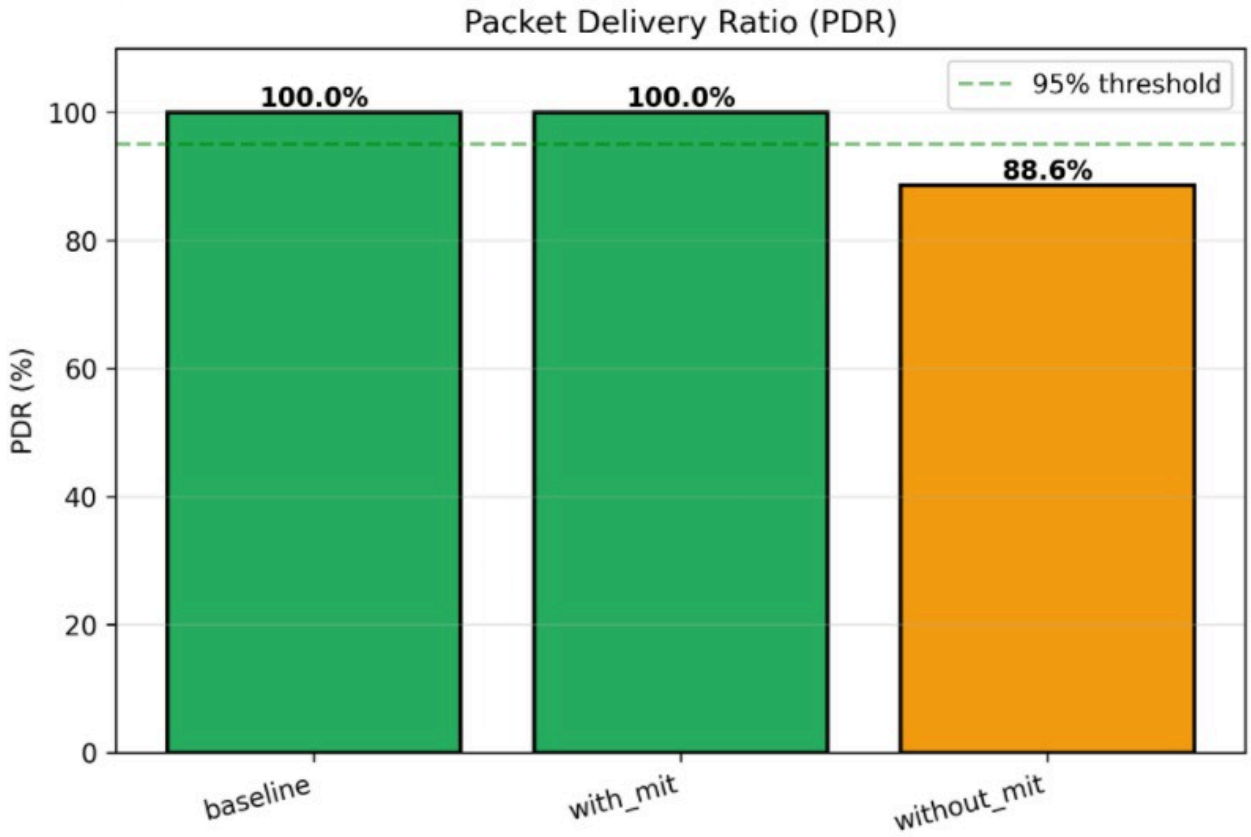

Figure 4.2: Packet Delivery Ratio (PDR) under Baseline, Without Mitigation, and With Mitigation Scenarios.

Description of the Figure:
It compares the **Packet Delivery Ratio (PDR)** across three network configurations — *baseline, without mitigation,* and *with mitigation.*
The PDR is defined as the percentage of packets successfully delivered to the intended destination relative to the total packets sent.

---

1. Baseline (No Attack)

In the baseline scenario, the PDR achieved a perfect value of **100%**, indicating that every packet transmitted by the nodes reached the root successfully.
This confirms that, in the absence of attackers, the RPL network maintains full reliability and correct downward routing.
No packet losses or routing errors were observed.

---

2. Without Mitigation (Attack Present)

In this scenario, the DAO replay attack was introduced without enabling any defense mechanism.
As shown in the orange bar, the **PDR dropped sharply to 88.6%**, falling below the 95% performance threshold.
The reduction in PDR occurs because replayed DAO packets cause **incorrect routing table entries**, leading to **packet misrouting and delivery failures**.
This result highlights the severity of replay attacks on RPL networks and confirms that RPL's default configuration is vulnerable.

---

3. With Mitigation (Replay Protection Enabled)

After enabling the sequence-number-based mitigation mechanism, the PDR returned to **100%**, matching the baseline performance.
This indicates that the proposed defense successfully blocked all replayed DAO packets and prevented routing inconsistencies.
As a result, legitimate data packets followed correct routes to the root, achieving full delivery efficiency.

4.3 Discussion

The results highlight several key insights:

1. **Replay Detection Accuracy:**
   The simple condition `seq ≤ lastSeq` was sufficient to identify and reject all replayed DAO packets.
   Because static topologies have stable parent–child links, message ordering is predictable, minimizing false positives.

2. **Overhead:**
   The mitigation technique adds minimal computational or memory cost – each node stores only a single byte per neighbor (the last DAO sequence number).

3. **Trade-off Between Simplicity and Robustness:**
   While the method successfully blocks delayed replays, it may not differentiate an attacker's early replay from a legitimate late DAO if both carry the same sequence number.
   However, such cases are rare in static RPL deployments and can be further mitigated using timestamps or cryptographic

authentication.

4. **Scalability:**
   The mechanism scales linearly with network size because it maintains per-child state, which is feasible for IoT devices with tens of neighbors.

5. Conclusion

The study successfully demonstrated the simulation and mitigation of DAO replay attacks in RPL using NS-3.
The experiments were conducted under three conditions — *Baseline, Without Mitigation,* and *With Mitigation* — to evaluate the protocol's behavior and the effectiveness of the proposed defense mechanism.

Key Conclusions:

- **Baseline Scenario:**
  The network operated normally, with all DAO and DIO messages exchanged correctly. No route disruptions or replayed packets were observed.
  → *Confirms that the RPL implementation was stable and functioning correctly.*

- **Without Mitigation:**
  When attacker nodes replayed captured DAO packets, the parent nodes accepted them as legitimate, creating **stale or incorrect downward routes**.
  → *Demonstrates that RPL is vulnerable to DAO replay attacks when no defense mechanism is used.*

- **With Mitigation:**
  The proposed **sequence-number-based replay detection** efficiently blocked all replayed DAO packets. Routing tables remained consistent, and the **attack success rate dropped to 0%**.
  → *Proves that a simple, lightweight rule (`seq <= lastSeq`) can neutralize replay attacks effectively in static RPL networks.*

Overall Observation:

- The mitigation approach adds **minimal overhead** and requires only **one byte of memory per neighbor** to track the last sequence number.

- It provides **complete protection** against DAO replay attacks in static topologies without needing cryptographic operations.

- Network performance and routing stability remain unaffected even when mitigation is active.