

Detection and Mitigation of Version Number Attacks in RPL Based IoT Networks: A Review of Techniques, Evaluation Gaps, and Real World Readiness

Fahad Alfaify - 262000500

1. Introduction

RPL, the Routing Protocol for Low Power and Lossy Networks, is widely used as a default routing protocol for a number of IoT constrained network applications. Unfortunately, the version number field is not covered by any authentication value as required by RFC 6550. Such a single malicious node is able to trigger a global repair for the whole network, resulting in a big increase of control overhead, energy waste of nodes, decreased delivery ratios and increased latencies. Therefore, we call this type of attack a Version Number Attack (VNA).

When it comes to defending against attacks, the cost of launching one is relatively low, but the cost of dealing with it is extremely high. This review takes a closer look at six studies from 2017 to 2023 that cover a wide range of design approaches.

2. Literature Review

2.1 Distributed Passive Monitoring (Mayzaud et al., 2017)

The main idea proposed in the paper consists in the offloading the detection task to the RPL routing protocol. If the version number of the packet sent by a node is greater than the version number currently used by the node, the monitoring node reports the sender of that packet as well as the set of neighbors of that node to the root of the network. The root of the network then collects the information received from the monitoring nodes, it thus is able to identify the attacking node as well as the set of relaying nodes affected by that attacking node without disturbing the other nodes. The paper also sets up the problem of where to place the monitoring nodes as an optimization problem and solves it using an Integer Linear Programming approach with the CPLEX solver. Also, the ratio of false positives is decreased from 20% in the worst case to almost zero as soon as two monitoring nodes are watching a node in an appropriate way. In the last part of the paper the previously introduced model is used for studying sensor networks up to 1000 nodes.

The work has a strong mathematical model that backs up the authors' claims to scalability. The detection experiments were performed on one small grid graph (a 'test bed' graph), using a fixed radio model with no packet loss, with all nodes static. For this work, detection is basically the end of the day for it, and it has not work with basic point of attack (i.e. injecting version numbers).

2.2 Machine Learning Detection with LightGBM (Osman et al., 2021)

Osman et al. (2021) transformed anomaly detection in IoT sensor networks into a classification problem and called the resulting system as ML LGBM. After training the feature set of 113 instances of raw features by Light Gradient Boosting Machine (LGBM) ML algorithm on the database, the number of features were reduced to 17 after cleaning, scaling and pruning by the Sequential Forward Selection (SFS) search algorithm. The resulting system attained high accuracy of 99.6% with very low false positives of 0.009% in ten fold cross validation experiments. But the numbers do not support the label “lightweight”. The model is 348 KB in size, which is far too large to fit in the 8 KB of RAM on the Zolertia Z1 motes that the authors used for their simulation. The data set used for evaluation is roughly 16 % attack traffic and left imbalanced, so the accuracy numbers are therefore flat. The system detects the attack, but as with Mayzaud’s team, no action to stop it.

2.3 The Claim Algorithm (Nandhini et al., 2021)

In the IEEE ICOSec 2021 conference paper, Nandhini et al. (2021) proposed a cryptography free “Claim algorithm” for securing topology changes in RPL. A node that detects a DIO message with higher version number than its current information temporarily holds off updating its information. A polled node then determines whether the version number increment of the DIO message detected by the node is genuine by a majority vote. The paper, mainly, surveys the security approaches for securing RPL networks, namely, VeRA, TRAIL and SecRPL. But of all the papers reviewed, this one is by far the thinnest in actual content, i.e., evidence of the claims made. Not a single detection metric is reported for the actual proposed algorithm. The basic parameters for the experiments (e.g., number of nodes, radio models, run length of experiments, etc.) are not even provided so that the experiments cannot be repeated.

2.4 CDRPL: Lightweight Detection and Mitigation (Alsukayti & Singh, 2022)

In addition to the theoretical contribution outlined above, the CDRPL (Collaborative and Distributed scheme for RPL) proposed by Alsukayti and Singh (2022) in the IEEE Access journal, offers a very practical solution for global repair. A big change in the behavior of nodes participating in the global repair is achieved by first checking whether the repair is in fact started by the sink instead of directly joining the repair as soon as a higher version number is seen by a node.

Any node that first hears of the update from a child, that node reports the attacking child to the root. The authors tested CDRPL on 10 runs (averaged over 3 topologies with 25, 40 and 65 nodes respectively) and the results showed that CDRPL is able to accurately and quickly detect the spreading of false information by malicious nodes with an average accuracy of 99.3% within 1 second. CDRPL is also able to handle multiple attackers.

CDRPL is cheap for detection and mitigation of version number attacks. The evaluation of the system is only done in simulation, i.e., no real hardware and no real link variations are tested. The mitigation approach of CDRPL only works if the sink itself is assumed to be safe.

2.5 QSec RPL: Detection under Mobility (Sharma et al., 2023)

Sharma, Grover and Verma (2023), in Elsevier's Ad Hoc Networks journal, are the only ones in this review that take into account the node mobility in their design. Each node in the network keeps track of the DIO messages with different versions and based on the learning done by Q Learning, it may send a 16 bit summary to the border router from time to time. The border router waits for 120 seconds to collect all the summaries and then applies Tukey's inter quartile range test to find the statistical outliers that represent the attackers in the network. The protocol fits within a real Z1 mote of 8 KB of RAM. First, it actually tests mobility, contrary to the majority of papers which hide their bad results for larger scale networks by not testing mobility. The average accuracy of this paper for the different settings is about 90%, but in reality it is as low as 0.75 with 0.64 true positives and 0.20 false positives (even slightly decreased for this paper). This paper is for detection and there is nothing related to prevention.

2.6 Parent Supervision (Rouissat et al., 2023)

In the preprint "Parent supervision lightweight solution against version number attacks for IoT networks" (2023) by Research Square, Rouissat, Belkheir and Mokaddem have taken the lightweight approach to its limits. In a parent supervised IoT sensor network, the fixed job of each parent is to supervise the version number of the information announced by their children. Once verified by the root, the child node and all of its child nodes (including preferred parents) are placed on a black list which the root broadcasts to all of the nodes in the network. Simulation results for 16 node scenario simulated with the Cooja simulator are provided which show that the attack raised the control overhead from approximately 673 messages to 4,736 messages and decreased the percentage of deliveries from 98.6% to 8.5% while the announced Parent Supervision scheme reduced the control overhead by 77.1% while delivery percentage was returned to 98.5% at the cost of only approximately 5.8 KB or an increase of 11.8% of memory.

Their approach is a very tiny design: a single edge only topology, with one fixed attacker and one static graph topology. The design is to detect and fix, and they do point out the memory cost of the design. The alert tags are not authenticated, so a node can falsely accuse an honest parent of misbehavior.

3. Connections and Conflicts Across the Studies

The attack studied in the six papers are similar in that an insider attack uses the version number field to force a network wide repair. In Sharma et al. (2023), the increase in DIO traffic is 583% while the delivery of data packets falls from 98.6% to 8.5% in Rouissat et al. (2023).

Three of the existing defenses for tree based versioning have a deeper connection: all of them check for a “higher version” in the form of a fork, but a real version increment always goes down from the root of the tree. Sharma et al. (2023) and Rouissat et al. (2023) in their work implement the detection within the memory of a real mote, unlike the heavier or offline approaches such as Mayzaud et al. (2017) and Osman et al. (2021).

A huge difference between the accuracy reported in the papers (near perfect scores even up to 100% for Osman et al. for 99.6% accurate anomaly detection on tiny sensor motes, above 99.3% for detection of several attackers by Alsukayti and Singh) and the real world performance, that is likely to suffer greatly from the use of the small, static test cases that the papers were based on. Specifically, the accuracy of the network of Sharma et al. starts dropping down to 0.75 as their network is getting filled up with more and more attackers, unlike the simpler cases considered by Alsukayti and Singh (several attackers), and by the other papers (a single, non colluding attacker).

4. Summary of Reviewed Papers

Study (Year)	Venue	Method / Approach	Key Findings	Main Weaknesses / Limitations
Mayzaud et al. (2017)	IEEE TNSM	Distributed passive monitoring on a separate RPL instance plus an ILP monitor placement model; detection and finding the attacker	Attacker always found; false positive rate ~20% down to ~0 with double coverage; placement projected to 1000 nodes	Detection only (no mitigation); single attacker; 20 node grid, idealised radio; scalability guessed from ILP, not measured; no energy/latency/precision reported
Osman et al. (2021)	IEEE Access	Supervised ML (LightGBM) ; ~1.05M record self built dataset; 113 to 17 features via SFS; 10 fold CV; centralised/offline	99.6% accuracy, FPR ~0.009; model ~348 KB; training ~140 s; beats five ML baselines	Detection only; 348 KB model cannot fit 8 KB motes ("lightweight" unproven on device); ~16% class imbalance untreated; internal metric inconsistencies; feature leakage risk
Nandhini et al. (2021)	IEEE ICOSEC	Claim algorithm root assisted cooperative neighbour verification plus majority vote; crypto free; non storing mode	Shows attack raises energy and hop count; useful survey of prior schemes	No detection metrics reported for own scheme; key parameters omitted (non reproducible); results only as plots; majority vote fails under colluders; misattribution risk
Alsukayti & Singh (2022)	IEEE Access	CDRPL in protocol cooperative verification (S	Over 99.3% accuracy; ~1 s detection; over 60%	Simulation only, no testbed; composite tests

Study (Year)	Venue	Method / Approach	Key Findings	Main Weaknesses / Limitations
		DIO _{nv} /S DAO _{nv}) exploiting the downward propagation invariant; detection and mitigation (blacklist)	energy and at least 75% overhead reduction; handles composite multi attacker	on one 40 node topology only; trusted sink dependency; own control flags not stress tested against forgery; ~8 s (45–55%) attack free convergence tax
Sharma et al. (2023)	Ad Hoc Networks (Elsevier)	QSec RPL Q Learning weighted DIO count plus Tukey IQR outlier test at the border router; mobility aware ; detection	~583% DIO surge quantified; fits 8 KB RAM; ~90% average accuracy; reports its own degradation	Detection only; accuracy falls to 0.75 / precision to 0.38 at scale; thin "RL" framing; 120 s delayed detection; self reported counts forgeable; no colluding attackers
Rouissat et al. (2023)	Research Square (preprint)	Parent supervision parent watches child's advertised version; reuses DAO/DIO tag fields; detection and mitigation plus recovery	Overhead reduced 77.1%; PDR restored 8.5% to 98.5%; only +5.8 KB (11.8%) memory	Single 16 node topology, one run (no repetition/CIs); no baseline comparison; forgeable alert tags (framing attack); slow recovery for some nodes; garbled tables; unrefereed preprint

5. Synthesis and Research Gap

Based on the approaches of the studies, it can be seen that there are 2 main approaches currently been used by researchers to detect an unauthenticated insider in an RPL network. But there is not much of a mitigation for the problem currently faced by the networks, and in the case of ML based approaches for detecting an unauthenticated insider in an RPL network, there is a lot of off path computation that is required, and therefore, these types of approaches cannot be used on actual devices under consideration in the current scenario. On the other hand, there are also a lot of studies (from the year 2022 and 2023) which are trying to mitigate the problem currently faced by RPL networks from within the network itself.

But the ways in which these mechanisms have been tested have not kept up with the increased complexity in the body of research into detection and prevention of data manipulation in Wireless Sensor and Ad Hoc Networks over the last 7 years. First, all current research into the detection and prevention of data manipulation in Wireless Sensor and Ad Hoc Networks is based on Cooja simulation and therefore the networks used are small, mostly static, and there is no shared public dataset of results, only Sharma et al. have included some form of mobility in their simulation. Second, all current research into the detection and prevention of data manipulation in

Wireless Sensor and Ad Hoc Networks places complete trust in the sink or root of the network and therefore the root never fails. Third, each individual defense for data manipulation in Wireless Sensor and Ad Hoc Networks introduces new, typically unprotected, control signals into the protocol. Also, in most cases, these additional control signals can be forged by an adversary, or a group of adversaries working together in order to attack the network. Fourth, and finally, most of the current research into the detection and prevention of data manipulation in Wireless Sensor and Ad Hoc Networks cannot be repeated.

The problem with current solutions for attack detection in distributed systems is that they have been tuned for one specific scenario: one honest majority attacker in a fixed simulation. How would they perform in a number of harder scenarios such as colluding or message forging attackers? This is the problem and the objective of the rest of this work.

References

1. Mayzaud, A., Badonnel, R., & Chrismont, I. (2017). A distributed monitoring strategy for detecting version number attacks in RPL based networks. *IEEE Transactions on Network and Service Management*, 14(2), 472–486.
2. Osman, M., He, J., Mokbal, F. M. M., Zhu, N., & Qureshi, S. (2021). ML LGBM: A machine learning model based on Light Gradient Boosting Machine for the detection of version number attacks in RPL based networks. *IEEE Access*, 9, 83654–83665.
3. Nandhini, P. S., Srinath, P., Veeramanikandan, P., & Malliga, S. (2021). Version attack detection using Claim algorithm in RPL based IoT networks: Effects and performance parameters evaluation. *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 209–215.
4. Alsukayti, I. S., & Singh, A. (2022). A lightweight scheme for mitigating RPL version number attacks in IoT networks. *IEEE Access*, 10, 111115–111133.
5. Sharma, G., Grover, J., & Verma, A. (2023). QSec RPL: Detection of version number attacks in RPL based mobile IoT using Q Learning. *Ad Hoc Networks*, 142, 103118.
6. Rouissat, M., Belkheir, M., & Mokaddem, A. (2023). Parent supervision lightweight solution against version number attacks for IoT networks. *Research Square* (preprint).