

Assignment 2

Research Methodology in Computer Science (AI800)

Student: Fahad Alfaify (262000500) **Lecturer:** Dr. Mohammed Anbar

Research Title: Detection and Mitigation of Version Number Attacks in RPL-Based IoT Networks

1. Problem Statement

RPL is a routing protocol for Low Power and Lossy Networks such as used in IoT networks and 6LoWPAN networks. In these types of networks, all nodes in the network form a tree structure. In this tree structure, each node has a parent node except for the root of the DODAG, which is typically a border router. In order to perform repairs in the network, the root of the DODAG increases the version number in the DIO messages and all nodes in the network have to recompute their position in the DODAG. This makes it possible for small energy constrained devices with very limited memory to maintain the routing information in the network.

The main problem of the Version Number Attack (VNA) is that the RFC 6550 does not protect the version-number field of the DIO messages. These messages are spread by all nodes in a network and any of them is able to change the version number contained in these messages. As a result of such an attack, the network is forced into a number of re-constructions that are running in order to fix the network from time to time, as it is performed by the root of a DODAG (e.g. border router) increasing the global version number contained in the DIO messages that it is sending. While fixing a network from time to time is an acceptable action, the number of restrictions running in order to fix the network from time to time should be as small as possible. In case of a VNA attack, the network is running in a number of restrictions that are running in order to fix the network from time to time in order to find a stable tree. This results in an energy wasting, in a high amount of control traffic, in a low packet delivery ratio (PDR) and in high delay. Because of the weak IoT devices which are used in very critical areas and which can be purchased for very cheap amounts of money even a single attacker is able to cause a lot of damage in a network. Therefore the detection and the prevention of the Version Number Attack is a problem which has to be solved.

Several attempts to counteract the Version Number Attack have been made but they failed for several reasons. Most of the proposed approaches for detecting misbehavior in a node do not also try to stop the misbehavior. An approach to detect and handle misbehavior in RPL is proposed in the work of Mayzaud et al. (2017). In this approach, one node monitors other nodes in the network and reports any misbehavior to the root of the DODAG. The root then handles the reported misbehavior but the approach of Mayzaud et al. (2017) does not provide any protection

for the network. Additional nodes for monitoring would consume a lot of energy and thus reduce the lifetime of the devices in the IoT network, which is not acceptable. The approach of Mayzaud et al. (2017) is limited to detecting an increase of the version number and does not consider any decrease of the version number. A malicious node could also reduce the version number. Their approach works only if all valid nodes in the network send DIO messages regularly. But some nodes may not send DIO messages, for example, if they are switched off temporarily. The root then cannot determine the parent of these nodes and thus cannot construct the DODAG in the network. An approach based on machine learning is proposed in the work of Osman et al. (2021) to distinguish between normal DIO messages and malicious messages that are sent by an attacker. Their approach requires a large amount of training data for learning and also a lot of memory for the classification. The used model in the work of Osman et al. (2021) is an ML-LGBM model that requires 348 KB of memory. The used motes in the work of Osman et al. (2021) have only 8 KB of RAM. The RAM of the motes is sufficient for the program of the mote, for the stack and for the buffers for receiving and sending messages. But the RAM of the motes is not sufficient for the model of Osman et al. (2021). Therefore, the approach of Osman et al. (2021) cannot be implemented on the used motes and thus is not applicable. An approach based on Claim algorithm is proposed in the work of Nandhini et al. (2021). In the Claim algorithm, each node that detects an increased version number sends a Claim message to the parent of the node. The parent then checks whether it has already sent a DIO message with

There is a major gap of existing work in testing RPL-based networks for Version Number Attacks. Most of the proposed solutions use small static graphs, rely on the sink and use only one attacker at a time. Each of the above solutions proposes additional control signals such as alert tags or self-reported counts by neighbors. These additional control signals are not protected in any way and can easily be spoofed by an attacker (Alsukayti and Singh, 2022; Sharma et al., 2023; Rouissat et al., 2023). To the best of the authors' knowledge, no work has been done to test and evaluate RPL-based networks for such attacks on real hardware, with mobile nodes that are mobile, with fake signals and with several attackers at the same time. Therefore, more research is required to see how a network behaves in real world scenarios.

2. Research Objectives

The purpose of this work is to evaluate current VNA based detection and mitigation approaches for more and more realistic attacks (e.g. faked signals, several attackers at the same time, untrusted sink etc.) and to compare them fairly.

To reach this goal, the objectives are:

1. Evaluate 6 existing solutions for VNA (detection, mitigation, memory usage, attacker models) and compare them.

2. For each of the VNA solutions under investigation (six in total), in addition to the additional control signals they send (i.e., the signals sent by a VNA over and above those sent by a normal controller), define an attacker, who may either: (a) alone spoof the additional control signals sent by a VNA (e.g., inject false control packets onto the network); or (b) in conjunction with other attackers spoof the additional control signals sent by a VNA (e.g., jointly inject false control packets onto the network).
3. Develop test cases that investigate the performance of VNAs in scenarios such as node mobility, larger networks and an untrusted sink.
4. Evaluate and compare the performance of the solutions under the defined attacker and report the results using accuracy, PDR, control overhead, energy and delay as metrics.
5. To compare the approaches in a fair manner and select approaches that can be applied in practice.

3. Literature Review (Summary)

The full literature review was done in Assignment 1. The table below summarizes the six reviewed papers by their method, key findings, and main limitations.

Study (Year)	Method / Approach	Key Findings	Main Limitations
Mayzaud et al. (2017)	Distributed passive monitoring + ILP monitor placement; detection and finding the attacker	Attacker always found; false positive rate around 20% down to Around 0 with double-coverage; scales to 1000 nodes	Detection only (no mitigation); single attacker; small static grid; no real hardware
Osman et al. (2021)	Machine learning (LightGBM), ML-LGBM; centralised/offline	99.6% accuracy; model Around 348 KB; Around 140 s training	Detection only; 348 KB cannot fit 8 KB nodes; imbalanced data; not tested on device
Nandhini et al. (2021)	Claim algorithm; root-assisted majority-vote verification; crypto-free	Shows attack raises energy and hop count; useful survey	No detection metrics reported; majority vote fails with colluding nodes; not reproducible
Alsukayti & Singh (2022)	CDRPL; in-protocol	Over 99.3% accuracy; ~1 s detection; handles multiple attackers	Simulation only; trusts the sink; new control flags not tested

Study (Year)	Method / Approach	Key Findings	Main Limitations
	cooperative verification; detection and mitigation		against forging
Sharma et al. (2023)	QSec-RPL; Q-Learning + Tukey-IQR outlier test; mobility-aware	~583% DIO surge measured; fits 8 KB RAM; ~90% average accuracy	Detection only; accuracy drops to 0.75 at scale; 120 s delay; self-reported counts forgeable
Rouissat et al. (2023)	Parent supervision; reuses DAO/DIO tags; detection, mitigation, recovery	Overhead reduced 77.1%; PDR restored 8.5% to 98.5%; only +5.8 KB memory	Single 16-node run; trusts the sink; alert tags not protected; no comparison with others

4. References

1. Mayzaud, A., Badonnel, R., & Chrisment, 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. <https://doi.org/10.1109/TNSM.2017.2705290>
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. <https://doi.org/10.1109/ACCESS.2021.3087175>
3. Nandhini, P. S., Srinath, P., Veeramanikandan, P., & Malliga, S. (2021). Version attack detection using Claim algorithm in RPL based IoT networks. *ICOSEC 2021*, 209-215. <https://doi.org/10.1109/ICOSEC51865.2021.9591735>
4. Alsukayti, I. S., & Singh, A. (2022). A lightweight scheme for mitigating RPL version number attacks in IoT networks. *IEEE Access*, 10, 111115-111133. <https://doi.org/10.1109/ACCESS.2022.3215460>
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. <https://doi.org/10.1016/j.adhoc.2023.103118>
6. Rouissat, M., Belkheir, M., & Mokaddem, A. (2023). Parent supervision lightweight solution against version number attacks for IoT networks. *Research Square* (preprint).