An Effective Approach to Protecting Low-Power and Lossy IoT Networks Against Blackhole Attacks

by

Kent Patrick Sanders

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2021 by the
Graduate Supervisory Committee:

Stephen S. Yau, Chair
Dijiang Huang
Arunabha Sen

ARIZONA STATE UNIVERSITY

December 2021

ABSTRACT

Realizing the applications of Internet of Things (IoT) with the goal of achieving a more efficient and automated world requires billions of connected smart devices and the minimization of hardware cost in these devices. As a result, many IoT devices do not have sufficient resources to support various protocols required in many IoT applications. Because of this, new protocols have been introduced to support the integration of these devices. One of these protocols is the increasingly popular routing protocol for low-power and lossy networks (RPL). However, this protocol is well known to attract blackhole and sinkhole attacks and cause serious difficulties when using more computationally intensive techniques to protect against these attacks, such as intrusion detection systems and rank authentication schemes. In this paper, an effective approach is presented to protect RPL networks against blackhole attacks. The approach does not address sinkhole attacks because they cause low damage and are often used along blackhole attacks and can be detected when blackhole attaches are detected. This approach uses the feature of multiple parents per node and a parent evaluation system enabling nodes to select more reliable routes. Simulations have been conducted, compared to existing approaches this approach would provide better protection against blackhole attacks with much lower overheads for small RPL networks.

DEDICATION

I would like to dedicate this to my mother and father, Stephanie Sanders and James Sanders for their support and guidance of my academic and life efforts. I would further like to dedicate this to Annie Wood for keeping me semi-sane and being a wonderfully supportive friend during my graduate studies. And finally, to Sam Marple for being an excellent group member on so many projects and surviving so many classes with me.

ACKNOWLEDGEMENTS

I would like to express my gratitude for Dr. Stephen S. Yau for allowing me to research under his guidance. He has taught me the research process, reviewed my work and progress many, many times, and provided critical insights and direction.

Thank you to Dr. Dijiang Huang and Dr. Arunabha Sen for taking the time to be a part of my thesis committee.

My deep appreciation to the National Science Foundation SFS CyberCorps Scholarship Program for funding my academic endeavors. It has been a huge blessing for me over the past two and a half years of my collegiate studies.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

I. INTRODUCTION

The IoT paradigm envisions the pairing of objects such as household appliances, medical devices, industrial devices, urban control mechanisms and the like with computing and connectivity to achieve a more automated and efficient world [8]. The realization of this vision relies on the deployment of billions of connected devices. This creates a demand for hardware that can be produced at a lower cost, and thus minimizing the resources of each device to what is necessary to perform the desired functions. The lack of hardware resources makes many networking and security protocols too expensive for these devices. These limitations make it necessary to minimize operations that place a burden on these resources such as cryptographic functions and transmissions.

This has driven the creation of new standards and protocols that are better suited to IoT devices that are light on resources. One such standard is the Routing Protocol for Low-Power and Lossy Networks (RPL). RPL is a protocol designed to connect groups of lightweight devices in an ad hoc manner, while supporting connectivity to IPv6 networks through a gateway called a border router (BR) or a root node. To do this a data structure called a destination oriented directed acyclic graph (DODAG) to form its routes. A DODAG is a graph of points and directed edges that converge on a given destination without forming any loops. In RPL the destination is the BR or root, and the other nodes are points. Edges in the graph represent a single hop along a route. Transmissions traveling away from the root of the DODAG are downward and transmissions that travel toward the root are upward. This structure allows for efficient multipoint-to-root and root-to-point routing for Low-Power and Lossy Networks (LLNs) [1].

However, IoT devices are generally considered easy to compromise [19], and the DODAG structure makes blackhole attacks more effective against RPL. There are also known vulnerabilities in the RPL protocol that make sinkhole attacks easy to perform and further increase the possible damage of a blackhole attack [4]. There are several existing solutions for detecting or preventing sinkhole attacks in RPL networks [8, 17, 15, 16, 18], but few [11, 12] directly address blackhole attacks.

In this paper, an effective approach is presented to protect RPL networks against blackhole attacks. This approach uses the feature of multiple parents per node and a parent evaluation system enabling nodes to select more reliable routes. Simulations have been conducted, compared to existing approaches this approach would provide better protection against blackhole attacks with much lower overheads for small RPL networks.

In this paper, I will first discuss the background matters in section II, including RPL route formation mechanisms, blackhole attacks and other related attacks. Section III discusses the state of the art for protecting against these attacks in RPL networks. In Section IV, the overall approach to protecting low-power and lossy IoT networks against blockhole attacks will be presented. In Section V I will discuss implementation and simulation details and evaluate the proposed approach. Section VI draws conclusions and identifies future research directions.

II. BACKGROUND

A. Route Formation

To form the DODAG structure Destination Advertisement Object (DAO), DODAG Information Object (DIO), and DODAG Information Solicitation (DIS) control messages are used alongside a rank metric and an objective function. DAO messages are sent by a node that has selected a parent and is joining the network. In mode of operation 1, the DAO message is routed to the root node, where it stores the routing information contained in the DAO. This information later allows the root node to look-up downward routes to nodes and embed them in packets being sent downward. Mode of operation 1 is the mode this paper will assume, though the ideas discussed can be applied to modes of operation 2 and 3 where each node stores downward routes and downward messages do not need to have the route embedded in them. DIO messages are sent to advertise the parameters of the network as well as information necessary for the sending node to be selected as a routing parent. DIS messages are sent by nodes that want to join the network and are responded to with DIO messages. Rank is a metric advertised by node's when they send DIO messages that is used by nodes joining the network to evaluate the suitability of its neighbors as routing parents. Rank strictly increases with each hop along a downward route. Rank is also contained in every message as part of a header that is updated with each hop. The strictly increasing nature of rank along downward routes allows nodes to check if messages are travelling in the correct direction and also ensures that loops do not form in the topology. Several possible rank metrics are outlined in [2]. Finally, the objective function is also included in DIO messages and is the function used by the nodes to evaluate rank metrics. [3] is a standard for one of the more common

objective functions. When initializing the network, the root node begins by sending a DIO message to its neighbors, providing them network information and allowing them to connect. Nodes that choose to connect to the root node send a DAO message and then send their own DIO message to their neighbors. Their neighbor may then choose to join, send a DAO to the root, and then send their own DIO message. This process continues until the network has formed. Nodes that do not wish to act as routers can signify a refusal to route in their DIO messages. Nodes that later wish to join the network may send DIS message to solicit DIO messages and find a parent to route through. [1]

As an example, let the number of hops away from the root node be the rank metric and minimizing the number of hops along a route be the objective function. Then a network begins initializing with the root node multicasting a DIO message to all its neighbors. Included in the DIO message is the DODAG ID, the root node's rank of 0 hops away from the root, and the objective function to minimize the number of hops. Neighbors that receive the DIO message and wish to join the network will send a DAO message to the root node to confirm joining the network and then multicast their own DIO messages to their neighbors. The DIO messages will advertise a rank of 1 since they are now 1 hop away from the root node. Nodes that receive these DIO messages and wish

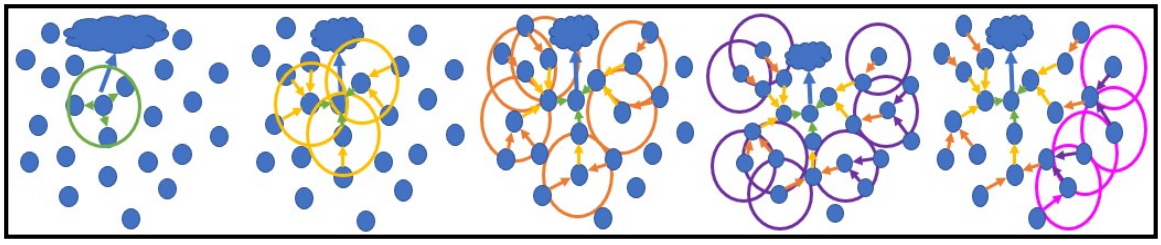


Fig. 1 Route Formation in RPL Network
The cloud is an external IPv6 network. Blue dots are RPL nodes. Circles indicate DIO broadcasts messages. Arrows indicate a route from a node to its selected parent. Color represents rank.

to join the network will then select a parent from among the advertising nodes and send a DAO message to the root to confirm joining the network. These nodes then send DIO messages with a rank of 2 and this process continues until route formation is completed. If during this process a node receives a DIO message from a node advertising a rank of 2 and another node advertising a rank of 3, then it would choose the node advertising a rank of 2, since the objective function is to choose parents that are a minimum number of hops from the BR. Nodes that do not wish to serve as nodes do not need to send DIO messages during route formation. They may advertise the network to other nodes with DIO messages and indicate that they will not serve as routers. These nodes will simple be leaves in the DODAG structure and are not considered during this paper. This process is shown in Fig. 1.

B. Blackhole Attacks and Related Attacks

Blackhole attacks are the primary attacks that this work is concerned with and occur when a node drops the messages forwarded to it for routing. The primary impact of this attack is the loss of availability and is thus a kind of DoS attack [5]. Packet filtering or selective forwarding are similar attacks in which the attacking node drops some traffic outed to it, but not all of it. These are the kind of attacks that this paper is primarily concerned with, though other related attacks are discussed, because they are important to the state of the art. Blackhole attacks are a significant threat to RPL networks, because of how easy they can be to enact and the severe impact they can have on the network. Any malicious node that serves as a routing node can carry out a blackhole attack. In some cases, RPL networks are publicly accessible and allow nodes to join and participate in routing freely. In many cases participation in routing is restricted, but even so IoT devices

are known for being vulnerable to compromise often as a direct result of efforts to reduce manufacturing costs. The OWASP Internet of Things Top 10 list [19] identifies the most common attack vectors against IoT devices including “insecure network services”, “outdated components”, and “lack of physical hardening” among others. Thus, in many IoT networks it is feasible to hack devices with routing permission remotely or via physical access. Attackers that have physical proximity to the network may be able to circumvent firewalls and intrusion detection systems (IDSs) supported by better resourced devices at the network’s perimeter. After compromising devices, the attacker must simply have devices drop packets that are forwarded to it. This will cut of access to the portion of the RPL network that normally routes through the compromised node and result in denial of service. This attack will generally be more potent if the node is closer to the root. An example of a blackhole attack is shown in fig. 2.

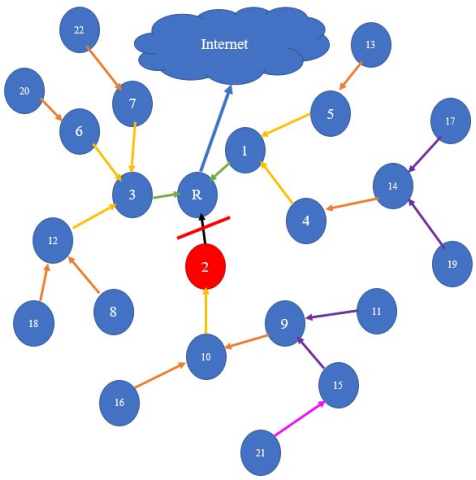


Fig. 2 Blackhole Attack in an RPL Network
Node 2 acting as a malicious node commits a blackhole attack and drops all traffic forwarded to it. Nodes 9, 10, 11, 15, 16, 21 are cut off from the network.

Sinkhole attacks occur when a node advertises itself as a particularly desirable routing parent to draw large amounts of network traffic to it. In RPL networks this can be accomplished by advertising a lower rank. For example, if a node is 4 hops away from the root node, but advertises a rank of 2, then it will appear preferrable to its neighbors as a parent compared to other nearby nodes that will probably be advertising ranks between 3 and 5. This can be used to amplify the impact of a blackhole attack by increasing the number of packets it can drop. This specific type of sinkhole attack is called a decreased rank attack [6]. Many existing solutions focus on sinkhole attacks instead of blackhole attacks which leaves a gap in the existing research. Fig 3. depicts a sinkhole attack.

Wormhole attacks occur when two nodes appear closer to each other in a network than they are physically. This can be achieved by various means, such as by having access to a separate network, a direct ethernet connection, or by creating a tunnel. This allows attacking nodes to share information and collude in a manner that can undermine

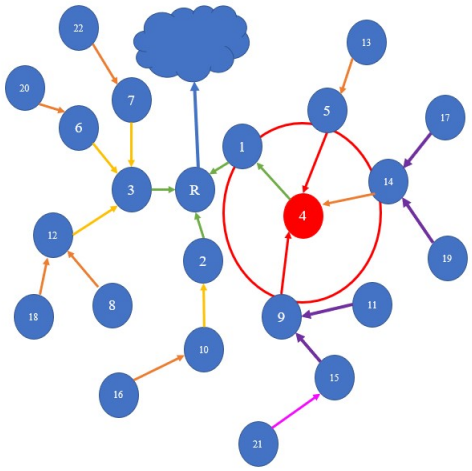


Fig. 3 Sinkhole Attack in an RPL Network
Node 4 acting as a malicious node executes a sinkhole attack by falsifying its rank as rank 1 instead of 2 and gains 6 additional children.

various defenses against sinkhole and rank attacks, such as efforts to authenticate rank [5]. It should be noted that a wormhole is not bad for a network on its own and can improve the network in some circumstances but can also be used to enhance and perpetrate other attacks.

Blackmail attacks occur when an attacker attempts to portray another participant in the network as the perpetrator of its malicious behavior. This can be problematic when defense mechanisms impose retaliation on bad actors, such as blacklisting malicious nodes. In this case a malicious node could drop packets from specific nodes and report false metrics to have certain well-behaved nodes blacklisted. An effective solution must be able resist such attacks to maintain network efficiency.

III. State of the Art

Almusaylim et al. [7] defines two main types of existing solutions that address blackhole attacks and rank attacks, namely modification solutions and IDS solutions. Recently blockchain solutions have emerged as part of the state-of-the-art and I include them as a smaller third classification. The major deficiency in existing research is that almost no solutions directly address blackhole attacks, but rather tend to address sinkhole attacks that commonly occur alongside blackhole attacks

A. Modification Solutions

Modification solutions change existing aspects of the RPL standard or add additional steps. Generally, the focus of these efforts is to authenticate the rank values of each node because this prevents decreased rank attacks; however, authenticating rank alone will not protect against a blackhole attack.

Dvir, Holczer, and Buttyan [9] is amongst the earliest of solutions that address rank attacks. During route formation the BR initializes a hash chain and transmits the second value of the hash chain along with its rank. Nodes that select the BR as their parent then hash the value received and transmit the new value along with their rank. This continues with nodes hashing the hash value received from their selected parent and transmitting it along with their own rank. Once route formation is complete the BR floods the network with the initial value of the hash chain. Each node can then calculate the values of the hash chain and verify that the rank advertised by its chosen parent matches the hash value advertised. Since hash chains are one-way malicious nodes cannot artificially reduce their rank lower than the rank corresponding to the hash value, they

receive from adjacent nodes without the falsification being detected. This still allows nodes to advertise a rank that matches the one received.

In the case of using hops as the rank metric nodes can reduce their rank by one, which may be sufficient to execute a potent blackhole attack. Further this scheme places limitations on what metrics can be effectively used as routing metrics. The scheme works well with metrics such as hop distance where rank increments by one. Several metrics may increment by variable amounts with each hop and thus associating rank values with hash values becomes challenging in these cases. Also, if multiple nodes in the network are malicious and one that is naturally close to the BR and it shares its low hash value with other malicious nodes in the network that are farther from the root, then those nodes can falsify their rank more effectively. Thus, the scheme can be broken by collusion.

Building on this solution Perrey et al. [10] proposes a challenge response mechanism that can be used in combination with [9] to verify that a node has issued a valid rank. This approach can be used to detected if a node replays the hash value and rank of its parent but can still be broken if the parent of the challenged node colludes with it or if the challenged node has a wormhole to a parent significantly closer to the root. Further if the challenged node is benign and the parent is malicious, then the parent could use the challenge response mechanism to perform a blackmail attack. In terms of cost, this solution requires several extra transmissions and hash operations for each challenge response, thus it could be expensive to perform for every node, every time routes are formed.

Nikravan et al. [8] uses an efficient identity-based signature scheme to authenticate rank. The mechanism is similar to that in [9], but with fewer vulnerabilities. The scheme is

still vulnerable to the collusion of multiple nodes and as a signature scheme it is more expensive to operate than a hash chain, but does not require certificates, so it is more efficient than other signature schemes. This scheme does require the distribution of cryptographic primitives, which is an area of research unto itself.

Ahmed and Ko [17] impose upper and lower bounds on how much nodes are allowed to alter their rank during network operation. The longer the network operates, the more stringent these thresholds become. This assumes that all devices are benign upon route initialization as a malicious node could start with an abnormally low rank in order to carry out a sinkhole attack. Additionally, this limits the mobility of nodes in the network or the ability of the network to adapt to changes in conditions. If for example a node moves in the network or if the physical environment changes in a way that impacts the usability of routes, then the thresholds can prevent nodes from adjusting their rank accordingly. This mechanism also limits the welcomingness of the network and can prevent new nodes from being used to their full potential as a routing node. The ability to reset or re-initialize the network is also limited if thresholds are enforced. Thus, this is a low-cost solution that is effective in relatively static networks with little need to adapt.

Each of these solutions shares the weakness that they do not address blackhole attacks directly, but rather seek to prevent sinkhole attacks that usually occur alongside them.

B. Intrusion Detection Solutions

Weekly and Pister's [11] solution is a hybrid of modification and intrusion detection solutions. It employs a hash chain like Dvir et al. [9] to authenticate rank and its own "parent fall-over" mechanism. The parent fall-over mechanism expects the BR to

receive a certain minimum number of messages from each node in the network over a given period of time. Any node from which the minimum number of messages is not received is added to a list. The BR will then broadcast the list. Nodes that find themselves on the list then blacklist their parents and routes are reformed. This combination of mechanisms performed well, though attacker behavior was not clearly specified as to whether nodes colluded to circumvent the hash chain. However, the solution could easily lead to suboptimal routes as a single blackhole can cause all of its children and their "descendants" to be listed and all affected nodes would blacklist their parents and create inefficiencies despite only one node needing to be blacklisted. This could also be used to blacklist certain parts of the network with a blackmail attack if a malicious node selectively drops traffic from only certain nodes.

Wallgren, Raza and Voigt [12] develop a simple mechanism for detecting the presence of blackholes in the network. The scheme requires end-to-end encryption on most messages. Then the BR can periodically send encrypted echo request packets to various nodes. If the BR receives an echo response, then it assumes the route does not have a blackhole on it. If it does not receive an echo response, then it assumes that the traffic along that route is being dropped. Since the echo packets are encrypted, it is assumed that they are indiscernible from regular traffic. However, depending on the type of network and the nature of its operation there may be ways to discover the echo packets and allow them while dropping normal traffic. For example, analyzing packet length might show that echo packets are distinct from regular traffic. In some networks it may be much more common for traffic to flow from nodes to BR rather than BR to nodes and filter traffic based on that assumption. If traffic normally flows upward, then allow

downward traffic to flow normally and when a possible echo request arrives record the destination address. When a possible echo response arrives from that address allow it to pass or simple allow all traffic from that node for a short period of time. Additionally, this approach does not identify the exact location of a blackhole, but rather its presence somewhere along a given route. Finally, the requirement for end-to-end encryption could be an extra cost if the network does not normally require such a measure.

Le et al. [13] and Krontiris et al. [14] develop intrusion detection systems (IDS) in which dedicated monitoring nodes observe route formation and network operation to detect the presence of blackholes. Mayzaud et al. [20] has a similar scheme in which naturally existing resource rich devices monitor the network for blackhole attacks. These schemes have the advantage of being able to monitor for many other types of attacks with a thorough IDS, but place requirements on the resources of various devices in the network and the distribution of such devices across the network. Such requirements could increase hardware cost or limit use cases. The topology also becomes constrained as monitoring nodes must be distributed throughout the network in a manner that allows the entire network to be monitored.

Ngai, Liu, and Lyu [15] and Shafique et al. [16] each used IDSs that monitor the network from the BR. This preferred to assuming the availability of well-resourced nodes distributed across the network. Each scheme uses symmetric end-to-end encryption between each node and the BR to ensure the integrity of messages. This use of encryption may be an extra overhead in some networks where confidentiality is not required and requires the distribution of cryptographic primitives. [16] simply observes the node ID, parent ID, and current rank in DAO messages and checks for changes in rank that may

indicate a blackhole attack. This approach does not account for the collusion of multiple nodes, which may allow blackmail attacks. [15] is similar in that it gathers node ID, parent ID, and routing metric, but is for sensor networks and not RPL networks. There is considerable overlap between these two types of networks, but neither is a superset of the other. [15] also uses localization methods to approximate the location of each node relative to each other, estimates the affected area of the attack, requests information, and attempts to identify the attacking node. This approach does account for the collusion of malicious nodes. This method does not account for mobile nodes. Each of these methods do well at detecting sinkhole attacks, which are commonly paired with blackhole attacks, but do not address packet dropping from malicious nodes that make only minor adjustments to rank in an effort to remain disguised. Thus, a malicious node in a strategic location would have the ability to commit a blackhole attack without using a sinkhole attack and remain undetected.

C. Blockchain Solutions

Sahay, Geethakumari, and Mitra [18] develop the primary blockchain-based solution for defending against blackhole attacks in RPL networks. This approach deploys an IDS on the Ethereum blockchain that monitors for events relating to version number and rank. These events are sent to the blockchain and evaluated by smart contract to determine if malicious actions have occurred, and which node is the perpetrator. This work addresses attack scenarios involving only one malicious node and small network size. Further use of the Ethereum network requires payment for each transaction and transaction speed may be too slow for busier RPL networks. The mechanism achieves reasonable results and serves as a good model for deploying an IDS on blockchain for IoT

but is ultimately not tested against more sophisticated scenarios and has various overheads.

IV. APPROACH

In this section, the overall approach for protecting RPL networks against blackhole attacks is presented. This approach is based on using the feature of multiple parents per node and a new parent evaluation system enabling nodes to select more reliable routes. The approach can be presented in five steps. Each of these steps will be described in detail as follows:

1) Initialization: Initialize the network with p parents where p≥2, excepting nodes adjacent to the BR and nodes that only have one neighbor.
2) Operation - Nodes: Nodes store a bitstring for each selected parent that indicates which messages were routed through each parent.
3) Operation - BR: On the other end the BR tracks what message indices have been from each node using a bitstring for each node. The BR will eventually send this record back to the node.
4) Parent Evaluation: Upon receiving the bitstring from the BR the nodes compare it to their own records and determine the reliability of each route.
5) Adaptation: Nodes then choose to favor the most efficient routes.

A. Step 1 – Initialization

The proposed approach has all nodes that are not adjacent to the root node attempt to select 2 or more nodes as routing parents. In doing so it is important that the following relationship hold $\text{rank} = \text{MAX}(\text{PR}) + \text{n};\ \text{n} > 0$ where PR is the set of ranks for a node's parents. This approach allows the network to adapt more easily when attacks occur and prevents rank attacks from causing a cascade effect throughout the entire network. That is, normally if an attacker spoofs a lower rank, then nodes that select it as a parent will also

advertise lower ranks than normal as will their children. However, if instead that node selects two parents, one with a falsified rank and another with a legitimate rank, its own rank will be based largely on the legitimate rank.

Example: The root node advertises a rank of 0. Nodes 1 and 2 join the network with only the BR as a parent and a rank of 1, because the BR is considered a secure and reliable parent. Nodes 3, 4, and 5 each select 2 parents for routing. Node 4 will have a rank of 2, since both of it parents have a rank of 1. Nodes 3 and 5 will have a rank of 3 since their parent with the highest rank is node 4 with a rank of 2. See Fig. 4.

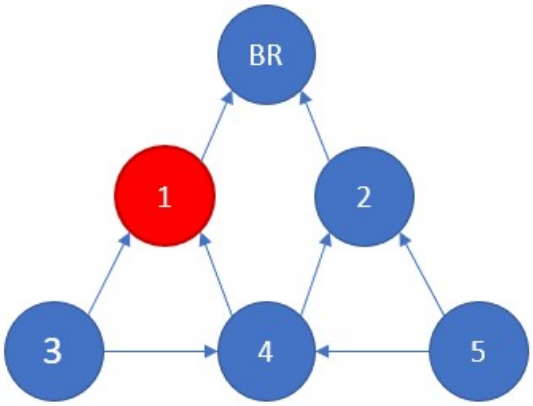


Fig. 4 An example for Step 1 – Initialization
Nodes not adjacent to root select 2 routing parents.

B. Step 2 – Operation - Nodes

To track which messages are routed through each parent nodes will store bitstrings for each parent internally. When sending a message, the node includes an index number in the packet and in the bitstring of the parent it chooses to route through it stores a 1 at that index. For other parents, a 0 is stored at that index. Periodically the node will increment the index by more than 1 and 0's will be stored at the skipped indices for all parents.

Example: Node 4 sends messages 0 and 2 through node 1 and messages 1 and 4 through node 2. Index 3 is skipped. The bitstring that node 4 stores for node 1 is 10100. The bitstring that node 4 stores for node 2 is 01001. The table stored by node 4 is shown in fig. 5.

Parent ID	Bitstring
1	10100
2	01001

Fig. 5 An example for Step 2 – Operation Nodes
This table indicates that the first and third messages were sent via node 1, the second and fourth messages were sent via node 2, and index number 3 is skipped.

C. Step 3 – Operation - BR

The BR will in turn store a bitstring for each node. When a message is received the BR will check the source address and the index. The BR then sets the entry in the bitstring for that node to 1 at that index. If the BR does not receive a message with a given index from a node, then that entry is a 0. When the bitstring reaches a certain length, the BR sends the bitstring to the node. The specific length required can be adjusted dynamically depending on the stability of the network. Sending the data less frequently will lower overheads but cause parents to be evaluated less frequently. This step serves the same purpose as sending ACK messages, but the innovative use of bitstrings allows for the confirmation of many messages at once and significantly reduces the number of transmissions required.

Example: Node 1 is malicious and drops messages 0 and 2, so the BR does not receive these and stores 0's at these indices. Index 3 is skipped, so this index also has a 0. Node 2 forwards the messages and the BR receives them and stores 1's. The bitstring at the BR will be 01001. The BR then sends this bitstring back to node 4. Such a table is shown in fig. 6.

Node ID	Bitstring
…	…
4	01001
…	…

Fig. 6 An example for Step 3 – Root Operation
This figure shows the entry for node 4 stored by the root node. It shows that the messages with indices 1 and 4 were received from node 4 and no other indices were received.

D. Step 4 – Parent Evaluation

Using the bitstring from the BR nodes can evaluate the reliability of their parents. If the bitstring from the BR contains a 1, then it interprets it as the message having been successfully delivered by the parent that has a 1 at the same index in its bitstring. If there is a 0 in the bitstring from the BR, then the node interprets it as the message being dropped by the parent that has a 1 in its bitstring at the same index. If a 1 is at an index that was skipped, then an alert is raised that the integrity of the data has been compromised. This mechanism of skipping indices is not intended to replace classical means of authentication and integrity verification but rather is a lightweight additional check.

Example: Node 4 receives 01001 from the BR via node 2. Node 4 compares the bitstring it has stored for node 1 (10100) with the bitstring from the BR. The entries with 1's in node 1's bitstring match up with 0's in the BR's bitstring, so node 4 assigns node 1 a reliability rating of 0.00. On the other hand, the bitstring for node 2 (01001) has matching 1's in the bitstring from the BR, so node 2 is assigned a rating of 1.00. The table stored by node 4 during this step is shown in fig. 7.

Node ID	Bitstring	Rating
1	10100	0.0
2	01001	1.0
Root Data	01001	-

Fig. 7 An example for Step 4 – Parent Evaluation
This table shows the table stored by node 4 after receiving feedback data from the root node and the resulting ratings of each parent after evaluating them with the feedback data.

Example Alternative: Node 4 receives 10110 from the BR via node 1. Node 1 flipped the bits of the bitstring before forwarding the data to try an improve its own rating. Node 4 detects that index 3 in the string from the BR is set to 1, which should be 0 since index 3 was skipped. Node 4 raises an alert.

E. Step 5 – Adaptation

Using the reliability ratings from the previous step as well as the rank of each parent, nodes form biases toward more efficient and reliable parents. It is generally best to delay this step for nodes that are far from root, since attacks occurring close to root

may impact nodes further away; however, it is best to allow nodes closest to the attack address the problem.

Example: Node 4 selects node 2 as a preferred parent since it proves more reliable than node 1. Node 5 would choose node 2 as a preferred parent since it is closer to the BR and reliable. Node 3 would choose node 4 as preferred parent even though node 1 is closer to the BR, because node 1 would have a reliability of 0.00 compared to a reliability of about 0.50 from node 4. See Fig. 8.

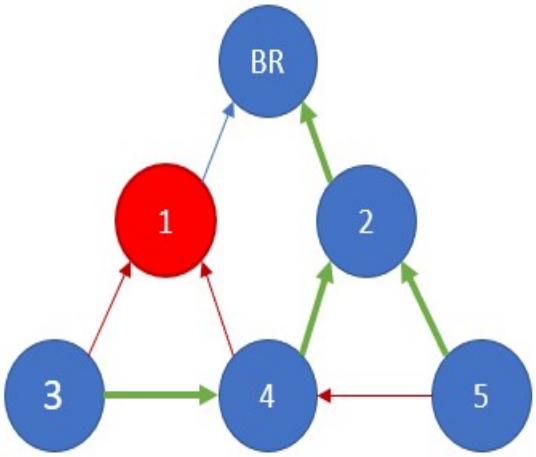


Fig. 8 An example for Step 6 – Adaptation
Based on data gathered, nodes evaluate routes and select more reliable and efficient routes for future use. Green arrows indicate favored routes. Red arrows indicate disfavored routes.

This approach has several advantages over existing solutions. One of the primary advantages is that it is reliability oriented rather than focusing on rank or routing metrics. This enables the approach to counter packet filtering that is not paired with rank attacks or other sinkhole attacks. In the example given a blackhole attack was avoided from a node adjacent to the root. This is the most advantageous location for packet filtering and

requires no falsification of routing information to draw large amounts of traffic. Other methods may prove ineffective against this because of their reliance on detecting or preventing the falsification of routing metrics. Additionally, it can address other routing problems such as the failure of a node along a route, noisy environments, and poor links. The use of bitstrings allows for efficient storage of information and reduces the number of transmissions necessary to communicate information. ACK messages could serve the same purpose but would require more bandwidth and resources. The proposed bitstring encoding helps keep transmission overheads low.

V. IMPLEMENTATION

A. Algorithms

To implement the presented approach several key algorithms and systems were developed. These algorithms include rank calculation, storing and evaluating bitstrings, and choosing a next hop when routing upward. See Fig. 9.

Rank in RPL networks is a 16-bit number. In this implementation rank is handled as a 5-digit number between 0 and 65535 in which the two highest-order digits and the three lowest-order digits are handled separately. This is represented as $XXYYY$. The XX portion represents the maximum number of hops a node is from the root, while the YYY portion represents the average number of hops from root times ten. The algorithm to calculate rank is shown in Algorithm 1. This supports a rank of up to 64 hops from the root node, maintains a strictly increasing rank relationship from parent to child, and allows nodes to represent both their worst route and their overall efficiency. The 64-hop limitation may prove quickly restrictive of network size in some low density and more linear topologies; however, a network size of roughly 4000 nodes is supported in a topology and density similar to that of the largest network shown in the evaluation section. Alternatively, allocating 7 bits to the maximum route and 9 bits to the average route allows for 128 hops and precision of .25 for the average. Ultimately, the specific algorithm can be adjusted based on topology and the chosen rank metric.

The storage of bitstrings requires careful tracking to ensure that the bitstrings stored at the root and at the node are properly aligned. A 16-bit sequence number is used

Algorithm 1

$n = node$

$P = Set\ of\ n's\ parents, p \in P$

$max, avg = \emptyset, \emptyset$

$ForEach\ p\ in\ P:$

$max.append(p.rank - (p.rank\ mod\ 1000))$

$avg.append\left[(p.rank\ mod\ 1000) + \left(\frac{10}{p.rating}\right)\right]$

$XX = MAX(max) + 1000$

$YYY = AVERAGE(avg)$

$n.rank = XX + YYY$

Algorithm 2

$n = node$

$n.startSeq = starting\ sequence\ number\ stored\ by\ n$

$R.startSeq = starting\ sequence\ number\ stored\ by\ n$

$P = Set\ of\ n's\ parents, p_i \in P$

$p_i.r = reliability\ rating\ of\ p_i$

$p_i.b = bitstring\ stored\ by\ n\ pertaining\ to\ p_i$

$R.B_n = bitstring\ sent\ from\ root\ pertain\ to\ n$

$T = threshold$

$offset = n.startSeq - R.startSeq$

$R.B_n.shiftBy(offset)$

$ForEach\ p_i:$

$Compare(p_i.b, B_n)$

$p_i.r = \left(\frac{\Delta(packets\ sent\ and\ lost\ via\ p_i)}{packets\ sent\ via\ p_i}\right)$

$P.sortBy\_p_i.r\_Descending()$

$n.preferredParent = null$

$If\ p_0.r > T:$

$n.preferredParent = p_0$

Algorithm 3

$if\ n.hasPreferredParent()$

$then\ forward\ via\ preferred\ parent\ and\ return$

$P.sortBy\_p_i.rank\_Ascending()$

$ForEach\ p_i:$

$if\ randomFloat(0,1) < MIN(.7, p_i.r)$

$then\ forward\ via\ p_i\ and\ return$

$Forward\ via\ random\ p_i$

Fig. 9 Three Key Algorithms

Algorithm 1 used for calculating rank. Algorithm 2 used for evaluating parents and select a preferred parent. Algorithm 3 used for selecting the next hop when routing packets.

to align the bitstrings. The sequence number is initialized at a node to a random integer between 0 and 30000 and is attached to each data packet, then incremented. The root node stores the lowest sequence number from a given node as the starting sequence number that it stores and uses it to place bitflags in the proper order regardless of the order in which data packets arrive. It also allows the root to place 0's in the appropriate positions. The root sends the starting sequence number with packets containing feedback data. If the starting sequence number in the packet is different than the starting sequence number stored by the node, then the difference between the two is used to align the bitstrings before comparing them. The general process for evaluating feedback is shown in Algorithm 2. Each time a node receives feedback from the root, the sequence number is re-initialized, and the highest order bit is flipped from its previous state. The node resets its bitstrings and the root does likewise upon receiving a sequence number with a different high-order bit.

Finally, routing is done with Algorithm 3. Nodes with a preferred parent default to that parent. Otherwise, parents are sorted based on best ranking and then given a random chance to be selected as the next hop, with more favorable odds for nodes with a good rating. This allows more efficient nodes to be prioritized, but also favors more reliable nodes.

B. Simulation Details

The approach has been implemented and tested in a custom simulator written in Python. The simulator places nodes in a network space, which then connect to each other using DIO and DAO control messages. Once connected to the network nodes send data to the root node and the root node sends feedback data after accumulating sufficient data on

a given node. The network operates in mode of operation 1, in which nodes only store routing information for their parents and the root node stores routes to the whole network. Nodes are given a radio range of 50 units after which, signal strength decreases exponentially over distance resulting in potential data loss. The signal strength is used in place of a rating if a rating has not yet been calculated for a given parent. During testing it was found that data delivery was greatly improved if nodes delayed parent selection in favor of finding parents with a strong signal strength. Nodes also have limited packet caches, but by carefully scheduling the frequency of sending data, data loss from a full cache can be avoided.

Malicious nodes are programmed to drop any data packets routed through them as well as any packets carrying feedback from the root node. Malicious nodes also select parents slightly faster than other nodes, but otherwise behave the same as a benign node. Malicious nodes do not falsify their rank. This makes them less damaging to the network as fewer nodes will select them as parents but makes them effectively immune to most existing solutions.

VI. Evaluation

A. Simulation Details

After surveying a variety of solutions aimed at addressing blackhole, sinkhole, and rank attacks in RPL networks the need for a solution that effectively addresses blackhole attacks without relying on falsified rank is shown.

Six test simulations were run on three different networks configured two different ways each. The results are shown in table 2 and the networks are shown in Fig. 10. All results are an average of 30 consecutive executions of the different configurations.

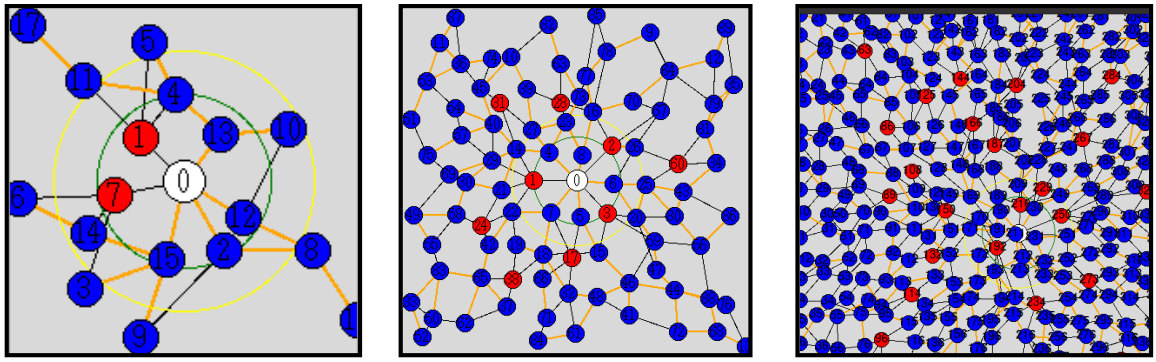
TABLE I. SIMULATION RESULTS COMPARISON

Network Composition	Percentage of Data Delivered		Transmissions Made		Maximum Rank	
	1 Parent	*2 Parents*	*1 Parent*	*2 Parents*	*1 Parent*	*2 Parents*
1 Root, 2 Malicious, 15 Benign	60.71%	96.86%	127833	168050	-	-
1 Root, 17 Benign	99.82%	99.81%	175203	173787	-	-
1 Root, 9 Malicious, 80 Benign	66.38%	95.31%	299261	371092	-	-
1 Root, 89 Benign	99.76%	98.48%	394650	408984	-	-
1 Root, 30 Malicious, 370 Benign	32.98%	80.79%	256772	396400	11.53	16.06
1 Root, 400 Benign	99.27%	98.30%	421306	471668	11.76	16.93

The first network consisted of a network of 18 nodes including 1 root node and 2 malicious nodes. The malicious nodes are placed adjacent to the root node in an optimal location to perform a blackhole attack. This network was run while selecting 1 parent per node as a base case, and then selecting 2 parents per node. While using two parents, the network delivered 1.60 times as much data and made 1.31 times as many transmissions. The extra transmissions are explainable by the fact that when data is lost or dropped it does not make all the hops to the root node. Thus, delivering more data requires more transmissions.

The second network consisted of 90 nodes including 1 root node, 9 malicious nodes, and 80 benign nodes. While using two parents the network delivered 1.44 times as much data and made 1.24 times as many transmissions.

The third and largest network consisting of 401 nodes including 1 root node, 30 malicious nodes, and 370 benign nodes delivered 2.45 times as much data and made 1.54 times as many transmissions. Finally, the same networks were configured with no malicious nodes to analyze overheads under normal operation. The smaller network delivered 1.00 times as much data and made .99 times as many transmissions while using



(a) (b) (c)

Fig. 10 Simulation Network Layouts
Small (a), medium (b), and large (c) simulation networks. Node 0 (white) is the root node. Red nodes are attacking nodes. Blue nodes are benign nodes. Orange lines are favored links.

two parents. Thus, in a small network the proposed approach delivers almost exactly as much data and does so slightly more efficiently. The mid-size network delivered .99 times as much data while making 1.04 times as many transmissions. The largest network delivered .99 times as much data and made 1.12 times as many transmissions.

B. Discussion

These results show some inefficiencies in larger networks, that could prove prohibitive in extremely large networks; however, with further refinement the approach may be able to deliver overall improved efficiency in some networks as was the case in the smallest network.

These results demonstrate a proof of concept for the proposed approach. The approach delivers data much more effectively under blackhole attacks than an RPL network using 1 parent. The approach also remains efficient when not under attack with no overheads in a smaller networks and growing overheads in larger networks.

The main overhead is due to the extra transmissions, which arise from extra DAO messages during initialization, feedback messages, and possibly longer routes. Slowing

down the initialization process to spread out the DAO messages across more time may reduce congestion during route formation. Feedback messages are a necessary overhead, but the frequency of them that are sent can be adjusted during runtime based on network needs. The possibility of longer routes can be addressed by optimizing route formation to select parents that are both reliable and efficient.

The scalability of this approach is constrained by the increasing overhead with more nodes. This problem may be addressed by using methods in the last paragraph to address overheads and avoid congestion from extra transmissions. RPL itself has scalability issues with large volumes of traffic, since messages converge at the root nodes and create congestion. Adding additional root nodes or external tunnels to the root may be used increase the scalability of RPL and this approach.

To compare this approach with the existing approaches, we consider the two approaches [11, 18] because they directly address blackhole attacks. Our approach and [11] offer similar levels of protection when under similar circumstances; however, [11] is vulnerable to blackmail attacks, requires hash operations, and may create network instability. [18] has significantly higher overheads in the form of transaction delays and transaction costs.

Several more general comparisons can also be made. First, the typical size of test networks for other solutions is about 50 nodes with as many as 200 and as few as 15. The simulations for our approach show excellent results in small networks and promising results for networks as large as 401 nodes. Second, the reliability-oriented approach allows it to address blackhole attacks directly as well as counteract other sources of data loss related to routing. Third, the only cryptographic algorithm algorithms necessary are

those related to authentication and integrity, which will prove necessary in almost any network to ensure a basic level of security, while several other solutions require hash or encryption algorithms. While this approach does not specifically enhance node mobility it does not place limitations and may improve mobility, since nodes have a second parent to route through should one relocate. Our approach does not require specialized hardware, dedicated monitoring nodes, or assume the presence of resource rich devices. It only makes the common assumption that the root node is well secured. Finally, this approach is flexible for various topologies, though it would suffer in topologies with low node density as would several IDS based solutions.

VII. Conclusion and Future Work

After surveying a variety of solutions aiming at addressing blackhole, sinkhole, and rank attacks in RPL networks, the need for an effective approach to protecting RPL networks against blackhole attacks without relying on it being paired with sinkhole attacks has been identified. In this paper, we have presented such an effective approach. This approach has been evaluated and compared favorably with the existing approaches in terms of providing better protection with low overhead.

Further refinement of the approach may be able to reduce the already small overheads that occur in larger networks. Specifically focusing on optimizing the initialization process can help to reduce congestion from extra DAO messages and ensure that routes are as efficient as possible and that nodes are selecting the best two or more parents available. Solutions to reduce congestion near the root node in RPL networks generally are likely the best method for reducing overheads and congestion resulting from sending feedback data from the root to nodes. This may include adding wormhole from the root node to more remote nodes via ethernet or an external network to reduce the amount of data converging on nodes around the root node, combining data messages at intermediate nodes to reduce the number of packets, or extending the approach to work for multiple root nodes and multiple DODAGs at once. These future research directions seem to be some more effective options for decreasing overheads and congestion and increasing viability for this approach in large networks or in high traffic networks.

There are also several possible research directions that could be taken to improve the usefulness of this scheme. Collecting data on scenarios in which the approach is deployed against attacking nodes that also commit sinkhole attacks may provide insight

for improvements given that they are frequently paired with blackhole attacks. Development of an intrusion detection system that builds on this approach by making use of the data that is naturally gathered by the root node would allow the scheme to identify attacks in addition to being able to route around them. Possible approaches could be to take note of nodes with high amounts of 0's in their corresponding bitstrings and compare their routes for common nodes or have nodes report poorly performing parents to identify potential attackers. An IDS may also be able to detect various other attacks using the information gathered by this scheme. Additionally, research on pairing the approach with a rank authentication mechanism as was done in [11] may be beneficial. [11] was able to achieve high protection even under heavy attacks by pairing rank authentication with simple intrusion detection. Given some similarities between my approach and that of [11], it is possible that adding rank authentication would prove particularly strong against blackhole attacks paired with sinkhole attacks. Finally, looking for other network types or applications where the scheme could be adapted to fit a different use case may prove fruitful.

WORKS CITED

[1] Winter, T., Thubert, P., & Brandt, A. (2012). RPL: IPv6 routing protocol for low-power and lossy networks. IETF, RFC 6550. https://tools.ietf.org/html/rfc6550. Accessed 8 July 2020.

[2] Vasseur, J., Kim, M., & Pister K.(2012). Routing Metrics Used for Path Calculation in Low Power and Lossy Networks. IETF, RFC 6551. https://tools.ietf.org/html/rfc6551. Accessed 8 July 2020.

[3] Gnawali, O. & Levis, P. (2012). The Minimum Rank with Hysteresis Objective Function. IETF, RFC 6719. https://tools.ietf.org/html/rfc6719. Accessed 8 July 2020.

[4] Tsao, T., Alexander, R., & Dohler, M.(2015). A Security Threat Analysis for the Routing Protocol for Low Power and Lossy Networks. IETF, RFC 7416. https://tools.ietf.org/html/rfc7416. Accessed 8 July 2020.

[5] Mayzaud, R. Badonnel and I. Chrisment, "A taxonomy of attacks in RPL-based Internet of Things", Int. J. Netw. Security, vol. 18, no. 3, pp. 459-473, 2016.

[6] Le et al., "The impact of rank attack on network topology of routing protocol for low-power and lossy networks", IEEE Sensors J., vol. 13, no. 10, pp. 3685-3692, Oct. 2013.

[7] Z. A. Almusaylim, A. Alhumam and N. Z. Jhanjhi, "Proposing a secure RPL based Internet of Things routing protocol: A review", Ad Hoc Netw., vol. 101, Apr. 2020.

[8] M. Nikravan et al., "A Lightweight Defense Approach to Mitigate Version Number and Rank Attacks in Low-Power and Lossy Networks", Wireless Personal Communications, vol. 99, no. 2, pp. 1035-1059, 2018.

[9] Dvir, T. Holczer and L. Buttyan, "VeRA - Version Number and Rank Authentication in RPL," 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011, pp. 709-714, doi: 10.1109/MASS.2011.76.

[10] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch and T. C. Schmidt, "TRAIL: Topology authentication in RPL", Proc. Eur. Workshop Wireless Sensor Netw. (EWSN), pp. 59-64, 2016.

[11] K. Weekly and K. Pister, "Evaluating sinkhole defense techniques in RPL networks," 2012 20th IEEE International Conference on Network Protocols (ICNP), 2012, pp. 1-6, doi: 10.1109/ICNP.2012.6459948.

[12] L. Wallgren, S. Raza and T. Voigt, "Routing attacks and countermeasures in the RPL-based Internet of Things", Int. J. Distrib. Sensor Netw., vol. 2013, 2013.

[13] Le, J. Loo, Y. Luo and A. Lasebae, "Specification-based IDS for securing RPL from topology attacks," 2011 IFIP Wireless Days (WD), 2011, pp. 1-3, doi: 10.1109/WD.2011.6098218.

[14] Krontiris, T. Dimitriou, T. Giannetsos and M. Mpasoukos, "Intrusion Detection of Sinkhole Attacks in Wireless Sensor Networks", LNCS, vol. 4837, pp. 150-161, 2008.

[15] E. C. H. Ngai, J. Liu and M. R. Lyu, "On the Intruder Detection for Sinkhole Attack in Wireless Sensor Networks," 2006 IEEE International Conference on Communications, 2006, pp. 3383-3389, doi: 10.1109/ICC.2006.255595.

[16] U. Shafique, A. Khan, A. Rehman, F. Bashir and M. Alam, "Detection of rank attack in routing protocol for low power and lossy networks", Ann. Telecommun., vol. 73, no. 7, pp. 429-438, Aug. 2018.

[17] F. Ahmed and Y.-B. Ko, "A distributed and cooperative verification mechanism to defend against dodag version number attack in rpl", Proceedings of the 6th International Joint Conference on Pervasive and Embedded Computing and Communication Systems, pp. 55-62, 2016.

[18] Sahay, R., Geethakumari, G. & Mitra, B. A novel blockchain based framework to secure IoT-LLNs against routing attacks. Computing 102, 2445–2470 (2020). https://doi.org/10.1007/s00607-020-00823-8

[19] OWASP. “OWASP Top Ten IoT List 2018”, https://owasp.org/www-project-internet-of-things/. 2021.

[20] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment and J. Schönwälder, "Using the RPL protocol for supporting passive monitoring in the Internet of Things", Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS), pp. 366-374, Apr. 2016.