The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks

Anhtuan Le, Jonathan Loo, Aboubaker Lasebae, Alexey Vinel, Yue Chen, and Michael Chai

***Abstract*— Routing protocol for low power and lossy networks (RPL) is the underlying routing protocol of 6LoWPAN, a core communication standard for the Internet of Things. RPL outperforms other wireless sensor and *ad hoc* routing protocols in quality of service (QoS), device management, and energy saving performance. The Rank concept in RPL serves multiple purposes, including route optimization, prevention of loops, and managing control overhead. In this paper, we analyze several different types of internal threats that are aimed at the Rank property and study their impact on the performance of the wireless sensor network. Our analysis raises the question of an RPL weakness, which is the lack of a monitoring parent in every node. In RPL, the child node only receives the parent information through control messages, but it cannot check the services that its parent provide hence it will follow a bad quality route if it has a malicious parent. Our results show that different types of the Rank attacks can be used to intentionally downgrade specific QoS parameters. This paper also reveals that attack in a high forwarding load area will have more impact on network performance than attack in other areas. The defenders can use the knowledge of such correlation between attack location and its impact to set higher security levels at particular positions by monitoring sensitive network parameters and detecting the anomalies**

***Index Terms*— RPL, Rank attack, performance, security, internal threat.**

I. Introduction

IN RECENT few years, Internet of Things (IoT) has gradually become a hot topic in the area of Wireless Sensor Network (WSN) with a lot of promising applications. One of the most challenged issues for IoT is to enable the convergence of WSN with the IP world, or in other words, the connectivity of smart objects to the Internet. Most of the core technology solutions for this issue has been conducted by the Internet Engineering Task Force (IETF) Working Group IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [1].

Manuscript received January 30, 2013; revised May 7, 2013; accepted May 21, 2013. Date of publication June 6, 2013; date of current version August 28, 2013. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xufei Mao.

A. Le, J. Loo, and A. Lasebae are with the School of Engineering and Information Sciences, Middlesex University, London NW4 4BT, U.K. (e-mail: a.le@mdx.ac.uk; j.loo@mdx.ac.uk; a.lasebae@mdx.ac.uk).

A. Vinel is with the Tampere University of Technology, Tampere 33720, Finland, and also with Halmstad University, Halmstad 302 60, Sweden (e-mail: valexey.vinel@tut.fi).

Y. Chen and M. Chai are with the Department of Electronic Engineering, Queen Mary University of London, London E1 4NS, U.K. (e-mail: yue.chen@elec.qmul.ac.uk; michael.chai@elec.qmul.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2013.2266399

As part of 6LoWPAN, the Routing Protocol for Low Power and Lossy networks (RPL) [2] has recently been standardized by IETF to efficiently handle the Layer 3 functions when providing Internet connectivity for WSN. Although RPL provides optional cryptography mechanisms to secure its control messages for providing network confidentiality, integrity, and authenticity, the attackers can still get control of the legitimate nodes by taking advantage of the fact that sensor devices are not tamper resistant and are weakly secured. These compromised nodes can be used to create attacks, which may affect the Quality of Service (QoS) of real-time WSNs-based applications [3].

Previous works have focused on internal threats to WSNs, most of which can also be applied in RPL use cases. Some applicable threats can be listed: Sybil attack, which uses packet forging mechanism to work as multiple identities as a base for initiating other attacks; Sink Hole attack, which attracts traffic to a specific node and then drops; or Selective Forwarding and Black Hole to drop and to add delay to the transmissions [4], [5].

However, unique RPL internal threats have not yet been well studied. To study such unique threats is important because they may have different nature to the traditional threats and therefore are difficult for the defending system to detect. The only similar work on this topic, to the best of our knowledge, is [6], which considers the ability of the network to recover from anomaly behavior. In this paper, we introduce so-called Rank Attack (RA), which is a new specific RPL internal threat aiming at its Rank property, and analyze its consequences.

The paper is organized as follows. Section II describes the principles of RPL operation and introduces several variations of the RA. Section III describes in detail our performance evaluation results and characterizes the influence of the RA on the WSN performance. Section IV concludes the paper.

II. Rank Attack on RPL

A. RPL Operation

RPL organizes the communication of network devices as a Directed Acyclic Graph (DAG). This topology is then divided into multiple DODAGs (Destination Oriented Acyclic Graphs); each DODAG includes many sensor nodes and a sink to collect data from them. DODAG sinks are connected together through a backbone. Each DODAG is differentiated by four parameters: RPL Instance ID; DODAG ID; DODAG

1530-437X © 2013 IEEE

Version Number; and Rank. The route inside each DODAG is chosen based on the selected link and the node cost information, such as available energy resources, workload, throughput, latency, or reliability. In order to generate the topology, each node first chooses a set of parents, which includes nodes which has equal or better quality of the paths towards the sink than the node has itself. From that set, the node which offers the best route will be selected as the preferred parent. All the communication of a node towards the sink node will, by default, be done through its preferred parent.

In order to establish and manage the routing, RPL utilizes three types of control messages: DIO (DODAG Information Object) for setting and updating the topology, DAO (DODAG Destination Object) for propagating destination information upwards during route updating progress, and DIS (DODAG Information Solicitation) for a new node to ask for topology information before joining the network. While DAO and DIS are mainly used for the purpose of starting a topology change process, DIO is commonly used for setting and maintaining the topology. A DIO message is broadcast by each node to signal its routing condition to other nodes through information such as Rank, and Objective Function. Node *Rank* is a specific concept in RPL which indicates the quality of the path to the sink node. Each node has to calculate its *Rank* according to the Rank of its preferred parent, and from the Objective Function. Every time a node updates its *Rank* or preferred parent, it needs to inform other nodes by sending the updated information in the next DIO. RPL uses the Rank rule that a node in the parent should always have lower rank than its children to prevent the loop creation.

To optimize the resource, instead of sending DIO frequently, RPL uses the trickle algorithm for scheduling it. In that algorithm, each node maintains a trickle time and a DIO counter which serves as the monitor for the topology stable. The "trickle time" interval will decide when the node has to send its next DIO messages. Each time a node receives a DIO without a change compared to the previous DIO, its DIO counter will be increased. Later if the DIO counter exceeds a pre-set value called the "redundancy threshold", the node will reset its DIO counter and double the trickle time. The reason for increasing the trickle time is that the threshold value of the DIO counter ensures the stability of the topology over an acceptable period of time, so there is no need to make frequent topology updates. This mechanism helps to reduce the number of DIOs generated in order to save network resources. On the other hand, if there is any change in the incoming DIO, the node will reset its DIO counter to 0 and minimize its trigger time. This will allow the network to quickly update its topology through fast DIO generation.

B. Rank Attack (RA)

The Rank property plays a crucial role which is related to almost all RPL operations. Its three main benefits are to create optimal topology, prevent loop formation and to manage the control overhead. However, the drawback is that any attack which aims at the Rank property can also achieve multiple impacts on RPL performance. RPL assumes that all the nodes are reliable and that they are following the protocol rules so it provides no mechanism to check node behavior. Therefore, having once compromised the cryptography defense, the internal attackers can control the nodes so as to downgrade the performance through intercepting the rank.

The literature does not show many ways to use the node Rank for compromising network performance. To the best of our knowledge there are only the work in [6], [7]. Author in [6] describes how changing Rank can affect the network performance. In this work, after running for a pre-set time, a node increases its Rank to equal the highest Rank value of its neighbors. The result of this is that there may be some loops between the node and its child so that the network becomes unstable and more control messages are then generated for recovering the optimized topology. The work in [7] describes but without assessing potential RPL Rank attack, in which an attacker can exploit the Rank value to attract and manipulate the network traffic. Our Rank attack is different by compromising the way a node process the Rank information, but not by changing itself. Therefore, the cryptography solution given in [7] cannot prevent such a kind of Rank attack because the Rank information is keeping no change.

In this paper, we present a different RA that aims at a mechanism to process information relating to the rank of other neighbors in each node. Rank information is used to select the parent set and the preferred parent according to the Rank rule, which states that the Rank of the parent always has to be smaller than the Rank of the child and the preferred parent should be the parent with best Rank. The malicious node is programmed to compromise the Rank rule so that instead of choosing the best node for its preferred parent, it chooses the worst one. RPL checks the Rank rule through DAO messages that a node informs to its preferred parent [2], however, this can be easily bypassed while compromised nodes skip informing those DAOs. As a result, more delay will be added to all the traffic routed through the malicious nodes. The topology around the malicious node is also expected to change so as more DIO messages will be generated, which leads to more control overhead as well as more packet collisions.

The attackers can act against the Rank rule permanently, or it can flip between for and against the Rule over a period of time. The purpose of flipping is to disrupt the stability of the topology by continually changing the preferred parent. Attackers can also choose to provide their updated DIO information to their neighbors or not. If they update the routing information in the DIO, their neighbors will have to update their topology as well, so more control overheads will be created, although the topology around that node may still be optimized. On the other hand, if compromised nodes do not update their routing information, no additional control messages will be generated, but since the topology is kept non-optimal, it silently adds delay to all the traffic that goes through them. By combining these options, we have investigated the four variations of the RA, which are summarized in Table I. The pseudo-code for implementing different Rank attack types is also given in Table II.

TABLE I

TYPES OF RA STUDIED

	Attack against rank rule (permanent /non-permanent)	
DIO information (update / no update)	RA I: permanent, update	RA II: permanent, no update
	RA III: non-permanent, update	RA IV: non-permanent, no update

TABLE II

PSEUDO CODE FOR RANK ATTACK

```
//1. compromising best parent function by selecting worst parent
1. best_parent(parent p1, parent p2) {
2.   if (node->attack_flag == TRUE) {
3.     return p1_metric > p2_metric ? p1 : p2;
4.     }
5. /* p_metric: the cost of the path towards the sink */
6. }
//2. DIO update flag: only update DIO if the flag is disabled
1. If (node->DIO_update_attack_flag == FALSE) {
2.   Update_DIO;
3. }
//3. Disabling DAO when having new illegitimate preferred parent
1.  Select_preferred_parent() {
2.  If (node->attack_flag == FALSE) {
3.    Dao_output; //inform the selected preferred parent
4.    }
5.  }
```

II(a). General Rank attack implementation

```
//Type I, start attack after t seconds
1.   node->attack_flag = FALSE;
2.   node->DIO_update_attack_flag = FALSE;
3.   If (time >= t)  {
4.  node->attack_flag = TRUE;
5.   }
```

```
//Type II, start attack after t seconds, disable DIO_update
1.   node->attack_flag = FALSE;
2.   node->DIO_update_attack_flag = FALSE;
3.   If (time >= t)  {
4.   node->attack_flag = TRUE;
5.   node->DIO_update_attack_flag = TRUE;
6.   }
```

```
/* Type III, start attack after t seconds, then time
is divided by multiple p-seconds period, attack happens
in only first half of each period */
1.   node->attack_flag = FALSE;
2.   node->DIO_update_attack_flag = FALSE;
3.   if (time >= t)
4.   {  period = (time - t)/p;
5.     check = period – int(period);
6.     If (check < 0.5) {
7.     node->attack_flag = TRUE;
8.     } else {
9.     node->attack_flag = FALSE;
10.    }
11.  }
```

```
/* Type III, start attack after t seconds, then time
is divided by multiple p-seconds period, attack happens
in only first half of each period. The DIO information is
not updated. */
1.   node->attack_flag = FALSE;
2.   node->DIO_update_attack_flag = FALSE;
3.   if (time >= t)
4.   {  period = (time - t)/p;
5.     check = period - int(period);
6.     If (check < 0.5) {
7.     node->attack_flag = TRUE;
8.     node->DIO_update_attack_flag = TRUE;
9.     } else {
10.      node->attack_flag = FALSE;
11.      node->DIO_update_attack_flag = FALSE;
12.    }
13.  }
```

II(b). Detail implementations for each RA type

III. PERFORMANCE EVALUATION RESULTS

A. Simulation Setup

We use Contiki 2.5 and Cooja [8], which includes the implementation of RPL IPv6, to simulate the network performance. We did not try the real testbed experiments because the study requires a large number of tests, which will take a lot of time and cost, while the simulation-based can provide similar and reliable results with much lower required resources. In each scenario, we made several modifications to the legitimate RPL code to allow compromised nodes to trigger the attack on the time required and behave like the RA as described in Section II.

The following choices of parameters and features are common to all the simulations in our study: the nodes use the beaconless IEEE 802.15.4 MAC/PHY operating with a default configuration in the 2.4 GHz range and they organize

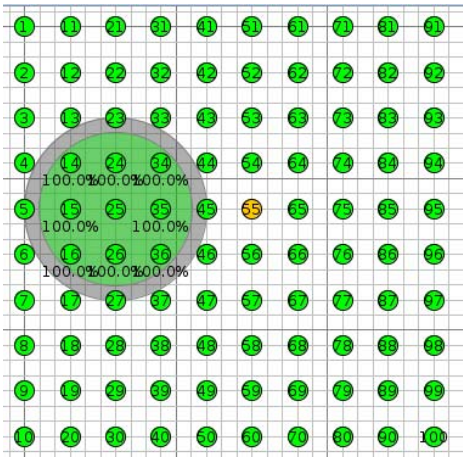


Fig. 1. Topology set up – the Sink is node 55.

themselves in a DODAG. The grid topology of 100 nodes in the region of 300 × 300 m is considered, see Fig. 1. Each node has a communication range of 50 m and the interference range is 60 m. The topology is set up so that every node can have (multi-hop) communication with the sink. We use free space propagation with no external noise to exclude the impact of the environment on the results so as to give a better view of the attack impact. In each simulation run, every node is set to send 1 packet to the sink every 10 second. In order to measure the network performance, we use the following parameters: average end-to-end delay of all the delivered packets, delivery ratio, namely the percentage of all sent packets which are delivered to the sink.

We ran 20 simulations under the normal network conditions (where all nodes ran legitimate RPL source code) and collected the performance results. These gave an end-to-end delay of ~1100 ms and a delivery ratio of ~97.6%. These results will be used later as a benchmark for the performance evaluation.

We created 4 scenarios for each type of attack as described in Section II. In each scenario, we first ran 99 non-root attack simulations by implementing malicious code in every non-root node, consecutively one by one. The purpose of triggering the attack in every location of the network is to investigate which areas create the most impact and which factors determine the level of the attack damage.

In each scenario, the triggering of the attack is set up as follows. For RA I & II the compromised node always uses the corresponding malicious code 50 seconds (t = 50) after the simulation starts. The system operates in normal mode for 50 seconds to ensure that the network topology becomes stable. For RA III & IV the first attack is triggered 50 seconds after the simulation starts and nodes switch from normal operation to malicious operation every 20 seconds (p = 40 seconds).

B. General Impact

We first compare and highlight the general impact of each variation of Rank attack on the network performance.

Figure 2 shows the average end-to-end delay (arranged in increasing order) and the corresponding delivery ratio

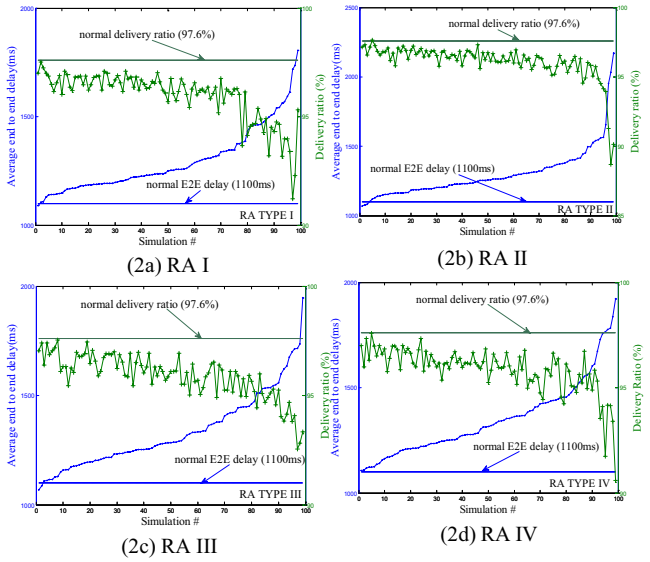


Fig. 2. Comparison of average end-to-end delay and delivery ratio performance between normal and compromised simulations in four types of RA - Ordered E2E delay performance versus the corresponding delivery ratio of 99 attack simulations.

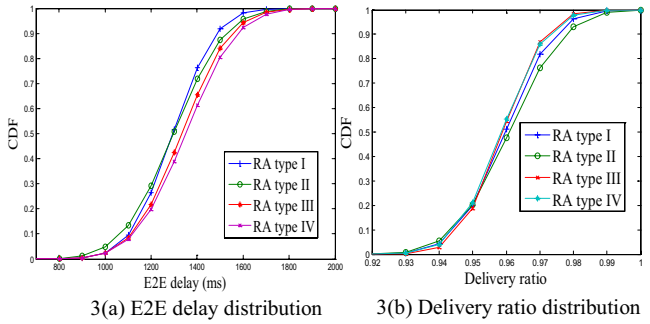


Fig. 3. Comparison of network performance distribution between 4 RAs: Each line indicates the distribution of network performance (E2E delay or delivery ratio) measured in sets of 99 similar RA scenarios - in each scenario a single non-root node triggers this particular RA.

performance of all the 99 attacks from single non-root node in scenario of RA I, II, III and IV consecutively. Figure 3 shows in more detail, a comparison of the corresponding cumulative distribution function (CDF) and Table III shows the worst impact that each type of attack can bring to the network performance. It can be seen clearly that in most of the attack cases, the performance of the network is downgraded in terms of both average end-to-end delay and delivery ratio performance. In 20% of the compromised case, the average end-to-end delay increases to more than 1.5 seconds (a 30% downgrade compared with the normal performance) and up to 2.2 second (a 100% downgrade). On the other hand, in 15% of the attack case, the delivery ratio decreases to less than 93% (a 5% downgrade compared to normal performance) or down to 88% (a 10% downgrade).

These results show that in a particular case, RA II may cause the worst impact on average end-to-end delay or delivery ratio, but in general, it has the least impact on the delivery ratio performance of the four types. This means that if attackers implement RA II randomly on the network, the probability

TABLE III

THE WORST PERFORMANCE COLLECTED AMONG ALL THE ATTACK SIMULATIONS AND THE CORRESPONDING ID OF THE ATTACKER

RA type #	Worst impact on E2E delay – location of attacker	Worst impact on delivery ratio –location of attacker
I	1.8s – ID:53	91% – ID:66
II	2.2s – ID:44	88% – ID:65
III	1.9s – ID:67	92% – ID:53
IV	1.9s – ID:88	90% – ID:88

that it will have the least impact on network performance is higher in comparison to other RA types.

Among the four types, type I and type II have the least impact, but type II has more effect on the end-to-end delay while type I has more impact on the delivery ratio. The nature of RA I and II is very similar, the only difference is that type I allows other nodes around the malicious source to optimize the topology through updating the DIO messages while type II does not allow this. As a result, the average end-to-end delay in type I is likely to be smaller than in type II. On the other hand, type I requires more additional control messages to maintain the optimized topology, so it may cause more packet collisions. This in turn reduces the delivery ratio so as to be less than for type II.

RA IV on the other hand, has more probability of creating the greatest impact on network performance in both the delivery ratio and the end-to-end delay. Type III of Rank attack has the same impact on delivery ratio, but a slightly smaller impact on average end-to-end delay compared with type IV. For these two types of attacks, the reason for having a higher delay than the first two types is that these two types create a lot of changes in the topology by frequently changing the preferred parent of the malicious node. The nodes around the affected area also have to spend more time updating the route, which adds more delay to overall performance. The delivery ratio is decreased because more control overhead is generated, which leads to more packet collisions. RA IV has a larger impact on end-to-end delay compared with type III because it does not allow updating of the optimized topology, so there will be more non-optimized routes in the network.

C. Specific Impact at Particular Simulations

We have also attempted to provide insight into the impact of the behavior of the attackers by studying particular cases for each of the attack types. The main concern of this part is what happens after the attack was triggered and which factor decides the level of impact on network performance.

The following cases with the highest impact on the performance were selected: RA type I at node 53, type II at node 44, type III at node 67, and type IV at node 88. These cases will be analyzed more thoroughly in terms of other performance parameters related to route stability, and factors that impact upon performance, as described below:

- Number of DIO messages generated: the number of DIOs messages generated every 10 seconds.
- Number of affected nodes: Number of nodes that update its rank or change its preferred parent every 10 seconds

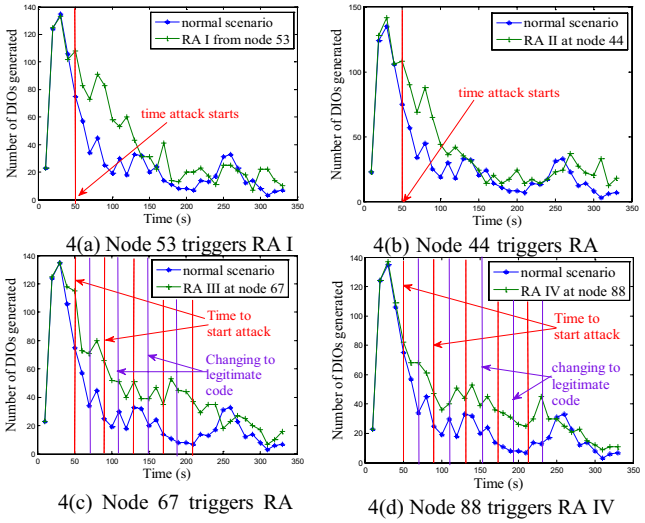


Fig. 4. Number of DIOs generated every 10 seconds in particular simulations.

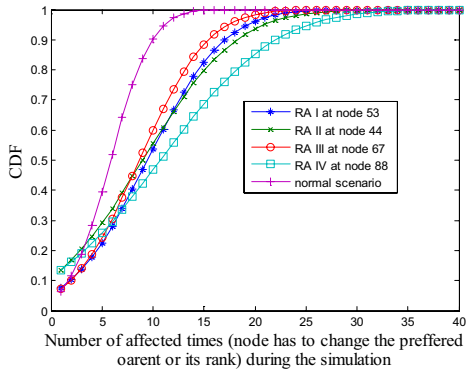


Fig. 5. Comparison of the cumulative distribution of the number of affected nodes between different particular RA simulations.

- Forwarding load of a node: the number of packets that a node needs to forward for its neighbors during the simulation time
- Forwarding load of an area: the sum of the forwarding load of all nodes in that area during the simulation time

Figure 4 shows the number of DIOs generated during the simulation time and Figure 5 shows the distribution of the times that a node needs to update its topology through changing the preferred parent or updating the rank. In the case of RA types I and II, the number of DIOs generated is increased compared with the normal performance, but mostly for a short time after the attack happens. On the other hand, in the case of RA type III and type IV, the difference between the number of DIOs in the attack case and normal case remained the same during the whole attack time. This result is interesting because from Figure 5, type III did not have many nodes that need to change the topology; however, it still showed a lot of DIOs generated. This suggests that the increase in the number of DIOs can have some other causes, for example, the change in the forwarding load of the neighbors around the malicious node makes those nodes distribute different routing information during this period. This, therefore, prevents the increase of DIO counter and DIO trickle time interval, and as a result, will make the node generate more DIOs than in normal case.

Now we will go into more detail for each of these four cases to show why the impact on performance occurred.

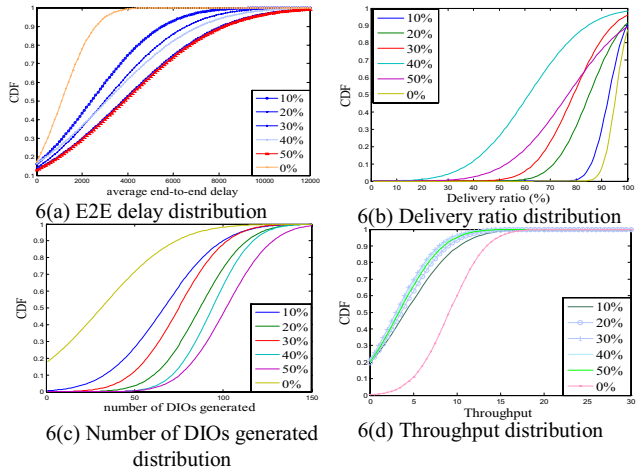
i) Node 53 triggers RA I: before the attack, node 53 forwarded the packets for 6 nodes (41, 42, 51, 52, 61, and 62) through route 53-44-Sink (2 hops). After the attack was triggered, packets from node 53 to the Sink changed to the new route 53-52-63-54-Sink (4 hops) instead of the previous 2-hops route so they contributed to the overall delay in the network performance. Besides that, none of node 53's previous children continued to choose it as their preferred parent anymore, so their forwarding loads were then shared with other neighbors of 53. We recorded a significant increase in the forwarding loads of node 44 and 45 (44 is a neighbor of 53 and 45 is the preferred parent of 44), which in turn affected their average forwarding delay making it much higher than for any other nodes (packets through node 44 or 45 have to wait almost 2 seconds to be forwarded, while the corresponding time for other nodes is from 0.1 to 0.5 seconds). This suggests that RA I not only affects the traffic through the compromised nodes, but also has an impact on the performance of other nodes around them and their preferred parents.

ii) Node 44 triggers RA II: before the attack, node 44 had to forward packets from 30 nodes to the Sink from the route 44-Sink (1 hop). After the attack, this route changed to 44-43-54-Sink (3 hops). Only 2 nodes (24, 33) continued to choose 44 as the preferred forwarder, which made their forwarding delay much higher (3.4 and 1.4 seconds respectively) while the delivery ratio decreased substantially (to 72% and 60% respectively). This was more than for the other nodes and contributed to the fall of overall network performance. Performance of node 44's neighbor – node 45 is also affected due to the increasing forwarding load diverted from 44.

iii) Node 67 triggers RA III: after the attack occurred, we recorded a frequent change of node 67's preferred parent. As a result, a lot more DIOs messages were generated in the area around this malicious node. The performances of these preferred parents were also affected so that the nodes with higher forwarding loads increased the delay more than those nodes with lower forwarding loads.

iv) Node 88 triggers RA IV: before the attack, node 88 forwarded the packets from 4 children: node 98, 99, 89, 100. After the attack, we found that there were loops between node 88 and its children which caused a significant delay and packet loss in this area. The loops were obviously caused by the non-updated routing information in DIO messages of the malicious node. The reason why RA IV is affected the most on node 88 may be because this area is the corner of the network so node 88's children may not be able to find other alternative non-affected parents to resolve the loops. This suggests that implementing RA in the area where nodes have limited choices of preferred parents will create higher impact on network performance.

From analysis of these specific simulations, it can be seen that nodes with a high forwarding load or in a high forwarding load area are more likely to have a high impact on network performance when the attack is initiated through it. The reason



6(a) E2E delay distribution

6(b) Delivery ratio distribution

6(c) Number of DIOs generated distribution

6(d) Throughput distribution

Fig. 6. Comparison of network performance distribution for scenarios with various numbers of attackers: Each line indicates the distribution of network performance metrics measured in scenarios with the same percentage of attackers (RA type employed in each attacking node may be varied).

is that RA through such a node will degrade not only its own performance but also the performance of its neighbors, which in turn affects the quality of traffic through them.

D. Impact When Multiple Attackers Cooperate

We extended our study by increasing the number of attackers and attack type randomly through the network to see how much the attack can damage network performance if they cooperate. The number of attacking nodes in each scenario was increased from 0 to 10, 20, 30, 40 and 50. In each multiple attacker scenario, the attacking positions and attack types were distributed randomly across the grid. In each of these scenarios, we ran 5 simulations to compare their performances.

Figure 6(a) and 6(b) respectively shows the impact on end-to-end delay and delivery ratio in the case of scenarios with multiple attackers, which are distributed randomly and with mixed attack types. They show a significant impact when the number of malicious nodes is increased. With 50 compromised nodes (50% of the network), the end-to-end delay performance increase 5 times compared with normal performance rising to between 4.8 and 5.5 seconds, while the delivery ratio is from 50% to 70%. With 30% or 40% malicious nodes, the average end-to-end delay is about 3 to 4.8 seconds while the delivery ratio is from 65 to 80%. In the case of 10 distributed attackers, the delay is slightly higher than the worst case of one node RA III, with delay from 1.5 to 2.6 seconds and delivery ratio from 85% to 93%. In general, 50% of the nodes in those simulations decrease their delivery ratio to less than 75%. Figure 6(c) indicates the number of DIO messages that are generated during network operation. It shows that in most of the multiple attack cases, the DIO overhead rises so as to become two to three times larger than in the normal case. Figure 6(d) presents the distribution of network throughput, which is the number of packets that are received at the sink every 10 seconds. It demonstrates that in 80% of the multiple attack cases, the throughput decreases to become half of its normal performance. All of these results suggest that the cooperated attacks can be so severe as to make the network performance unacceptable. It therefore raises the question of the need to protect RPL from such internal threats.

On the other hand, some of the results show that scenarios with a smaller number of attackers may create more severe impact than the scenario with a larger number of nodes. For example, in Figure 6(b), the distributions show that attack with 40 malicious nodes has more impact on the delivery ratio than attack with 50 malicious nodes. This suggests that the point where the attack is initiated is also important. Sometimes attack at a few crucial points can have as much impact as those initiated at many non-crucial points.

E. Summary of Main Findings

The main findings of these simulation results can be summarized as follows:

- RA III and IV have greater impact on the delivery ratio performance in comparison with type I and type II because they create more control packets (DIO) and make the topology change more frequently. Type II and IV have more effect on the end-to-end delay because they silently raise the use of non-optimized routes by disabling the routing update.
- There is a strong correlation between the forwarding load of a node or area around it and the impact of the attack initiated there. This suggests that if the malicious attacker implements the attack in a high forwarding load area, it is likely that it will downgrade the network performance more than in the case of implementation in other low load forwarding areas. In this case, the impact will not only appear in this local area but also expand through the malicious children or sub-child nodes.
- There is no mechanism for nodes in RPL to monitor the behavior of its parent. Currently, nodes rely on their parents so that in many attack cases it has to follow the non-optimized route provided by the malicious parents. All the information that a node knows about its parent is obtained via the broadcast DIO messages, but under the internal RPL threats, those packets cannot be properly trusted. This raises the question about the performance security of RPL under internal threats.
- Some parameters are sensitive to the attack, for example, the number of affected nodes, the number of DIO messages generated, the average end-to-end delay and the delivery ratio. These parameters will change significantly when the attack happens, and become stable again a short time after the attack stops. It means that the Rank attack can be used intentionally to attack specific performance parameters of the network. It also suggests that the system can be defended by monitoring these parameters to keep track of the node behavior so as to detect any anomaly in network performance.
- The cooperation of attackers can create severe damage to network performance, especially if they are put in the right positions. It is therefore necessary to put more security on those locations where network performance will be significantly downgraded if they are attacked.

IV. Conclusion

RPL is currently applied in the role of a main routing protocol for large scale low-power and lossy networks, and therefore becomes a good candidate to bring the application of WSN into many areas such as Internet of Things or Smart Grid. Securing the network performance is crucial and willsoon become a major requirement in order to integrate into such applications. The unique concept of "Rank" in RPL can make its performance become vulnerable to the internal threats. Attack on Rank may create un-optimized paths, more overhead, and more packet collisions, thus downgrading the network performance by, for example, increasing the end-to-end delay and decreasing the delivery ratio. In this paper, we first analyze some possible Rank attack threats that can be implemented to downgrade RPL performance. We then study the impact of the attack by simulating attack on different locations within the network. The results reveal that attack may have a severe impact on the network performance, especially when it is implemented in a high forwarding load area, or in multiple attacker cases. Different types of Rank attack behavior are analyzed in terms of hiding the non-optimized routing information or flipping the preferred parents. Our study also reveals that the number of affected nodes, number of DIO generated, end-to-end delay and delivery ratio are the most sensitive to this particular type of attack. In addition, the results indicate a weakness in the security design of RPL so that the children have to rely on its parent's routing information through DIO packets but they have no other mechanism to verify the services of their parents. It is crucial because once the preferred parent is compromised and no other node discovers its malicious behavior, the performance of all the surrounding area will be affected. The findings also suggest that it is important to introduce more security resources in some crucial parts of the network than others because the attack has a different level of impact in different network areas. In the future, we would like to expand the results of this study for implementing an anomaly intrusion detection system, which can diagnose the internal attacks based on monitoring some attack-sensitive performance parameters.

References

[1] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43. New York, NY, USA: Wiley, 2011.

[2] *RPL: IPv6 Routing Protocol for Low Power and Lossy Networks*, RFC Standard 6550, Mar. 2012.

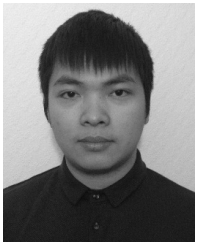
[3] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6LoWPAN: A study on QoS security threats and countermeasures using intrusion detection system approach," *Int. J. Commun. Syst.*, vol. 25, no. 9, pp. 1189–1212, Sep. 2012.

[4] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 38–43, Dec. 2004.

[5] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: A survey," *J. Inf. Assurance Security*, vol. 5, no. 1, pp. 31–44, 2010.

[6] W. Xie, M. Goyal, H. Hosseini, J. Martocci, Y. Bashir, E. Baccelli, and A. Durresi, "Routing loops in DAG-based low power and lossy networks," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 888–895.

[7] A. Dvir, T. Holczer, and L. Buttyán, "VeRA—Version number and rank authentication in RPL," in *Proc. 8th IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2011, pp. 709–714.



Anhtuan Le received the B.Sc. degree from the Hanoi University of Technology, Hanoi, Vietnam, in 2006, and the M.Sc. degree in computer network security from Middlesex University, Middlesex, U.K., in 2009, where he is currently pursuing the Ph.D. degree with the Computer Communication Department, School of Computing. His current research interests include communication protocols, quality of service, network security, wireless communication, and the Internet of things.

Jonathan Loo received the M.Sc. degree in electronics (with distinction) and the Ph.D. degree in electronics and communications from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and 2003, respectively. Currently, he is a Reader of communication and networking with the School of Science and Technology, Middlesex University, Middlesex, U.K. He leads a research team in the area of communication and networking. His current research interests include network architecture, communication protocols, network security, wireless communications, embedded systems, video coding and transmission, digital signal processing, and optical networks. He has successfully graduated 11 Ph.D.s as a Director of studies in the aforementioned specialist areas. He has been an Associate Editor for *Wiley International Journal of Communication Systems* since 2011.

Aboubaker Lasebae is currently a Director of postgraduate programmes with Middlesex University, Middlesex, U.K. He received the B.A.Sc. degree from the University of Regina, Regina, SK, Canada, the master's degree from Southampton University, Southampton, U.K, and the Ph.D. degree from Middlesex University. His current research interests include mobile and wireless communications, wireless sensor networks, telecommunication security, networking and computer security, Internet of things, quality of service, and telemedicine applications and LTE.

Alexey Vinel (M'07–SM'12) received the bachelor's (Hons.) and master's (Hons.) degrees in information systems from St. Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia, in 2003 and 2005, respectively, and the Ph.D. (candidate of science) degree in technical sciences from the Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia, in 2007. He is currently a Researcher with the Department of Electronics and Communications Engineering, Tampere University of Technology, Tampere, Finland, and a Guest Professor with the School of Information Science, Computer and Electrical Engineering, Halmstad University, Halmstad, Sweden. He has been an Associate Editor for the IEEE COMMUNICATIONS LETTERS since 2012. His current research interests include multiple-access protocols and intelligent transportation systems.

Yue Chen received the B.Sc. and M.Sc. degrees in radio engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in electronic engineering from Queen Mary University of London, London, U.K., in 2003. Her current research interests include intelligent radio resource management for wireless networks, cognitive and cooperative wireless networking, energy harvesting, smart energy systems, and Internet of things.

Michael Chai received the B.Eng. (Hons.), M.Sc., and Ph.D. degrees in 1998, 1999, and 2007, respectively. From 2002 to 2008, he was a Senior Lecturer with Staffordshire University, Staffordshire, U.K. He joined the School of Electronic Engineering and Computer Science, Queen Mary University of London (QMUL), London, U.K., in 2008, as a Joint Programme Lecturer in QMUL and Beijing University of Posts and Telecommunications, Beijing, China. He is a member of the Networks Research Group, QMUL. His current research interests include dynamic resource management wireless communications, machine to machine communications and networks, and the Internet of things in intelligent transport systems and home automation networks.