2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)

Hatchetman Attack: A Denial of Service Attack Against Routing in Low Power and Lossy Networks

Cong Pu Tianyi Song
Weisberg Division of Computer Science
Marshall University
Huntington, WV 25755, USA
{puc, songt}@marshall.edu

***Abstract*—Low power and lossy networks (LLNs) are rapidly burgeoning as an important part of ubiquitous communication infrastructure, and serving as a major building block for emerging Internet-of-Things (IoT) applications. A novel routing protocol for low power and lossy networks, referred to as *RPL*, has been standardized to provide efficient and reliable communication in LLNs, and enable the integration of resources-constrained devices into the Internet. However, due to the lack of resources, physical protection, and security requirements of inherent routing protocol, RPL-based LLNs are admittedly vulnerable to Denial-of-Service (DoS) attacks that primarily disrupt network protocols and interfere with on-going communications. In this paper, we investigate a new type of DoS attack, called *hatchetman attack*, in promptly emerging RPL-based LLNs. In hatchetman attack, the malicious node manipulates the source route header of the received packets, and then generates and sends a large number of invalid packets with error route to legitimate nodes, which cause the legitimate nodes to drop the received packets and reply an excessive number of Error messages back to the DODAG root. As a result, a great number of packets are dropped by legitimate nodes and excessive Error messages exhaust the communication bandwidth and node energy, which lead to a denial of service in RPL-based LLNs. We conduct extensive simulation experiments for performance evaluation of hatchetman attack and comparison with jamming attack and original RPL without adversary. The simulation results indicate that the hatchetman attack is an extremely severe attack in RPL-based LLNs.**

***Index Terms*—Hatchetman attack, denial-of-service (DoS) attack, RPL, low power and lossy networks.**

I. Introduction

A rapidly growing number of physical objects being connected to the Internet are realizing the idea of Internet-of-Things (IoT) and its applications, where a myriad of multi-scale sensors and devices (later nodes) are seamlessly blended and communicate with each other [1]. It is predicted that 20.4 billion wirelessly connected devices will be available for IoT applications by 2020, nearly triple the number that exists today [2]. As a part of speedily emerging IoT, low power and lossy networks (LLNs) are playing a remarkable role in building a ubiquitous computing and communication infrastructure, where a set of resources-constrained nodes with the limited processing power, energy capacity, and memory communicates directly or indirectly via lossy links. With the increasing demand of connecting resources-constrained nodes to the Internet, the Internet Engineering Task Force (IETF) Working Group [3] has proposed a novel routing protocol for low power and lossy networks, referred to as RPL [4], as the communication standard for IP smart object networks. With the prevalence of cloud computing and social networking paradigms as well as the recent progress in communication technologies, embedded devices, and sensor networks, we envision that wirelessly connected IP smart nodes under IoT will enhance information accessibility and availability as well as improve our lives further.

However, due to the shared medium and the lack of resource, physical protection and security requirements of inherent network protocols, LLNs are undoubtedly vulnerable to Denial-of-Service (DoS) attacks [5]. For example, a legitimate node compromised by an adversary can easily overhear, duplicate, corrupt, alter, or drop an on-flying packet. Although the RPL standard includes the optional security mechanisms to ensure the confidentiality and integrity of control messages as well as the availability of routing information, however, current RPL implementations choose not to enable these secure operation modes due to resource consumption, which greatly affects the performance of resource-constrained devices [6], [7]. In addition, threat analysis for securing RPL presented in [8] only identify the well-known security issues with fundamental countermeasures, thus, this leaves RPL open to new attack wherein a malicious node can manipulate the content of packet header to disrupt routing protocol or interfere with on-going communications.

In this paper, we present a new type of denial-of-service attack, called *hatchetman attack*, in RPL-based LLNs. In hatchetman attack, a malicious node manipulates the source route header of the received packet, and then generates and sends the invalid packets with error route to legitimate nodes. When the legitimate node receives the invalid packets with error route, the packets will be dropped since the receiving node cannot forward the packets with the piggybacked error route. The receiving node also will reply an Error message back to the DODAG root to report the error in source route header. If the malicious node generates and sends a large number of invalid packets with error route to legitimate nodes, this will cause the legitimate nodes to drop the received packets and reply an excessive number of Error messages, which eventually lead to a denial of service in RPL-based LLNs. Our major contribution is briefly summarized in twofold.

- We identify and present a new and severe denial-of-service attack, called *hatchetman attack*, in RPL-based

978-1-5386-5850-5/18/$31.00 ©2018 IEEE
DOI 10.1109/CSCloud/EdgeCom.2018.00012

LLNs. This is the first in-depth work to investigate the performance impact of hatchetman attack in RPL-based LLNs.
- We revisit and implement the well-known jamming attack and the original RPL without adversary for performance comparison. The original RPL without adversary is used as the upper and lower bound of packet delivery ratio and packet delivery latency, respectively.

We develop a customized discrete event-driven simulation framework by using OMNeT++ [9] and evaluate its performance impact through extensive simulation experiments in terms of packet delivery ratio, throughput, packet delivery latency, energy consumption, the number of attack packets, and attack energy inefficiency. The simulation results indicate that the hatchetman attack is an extremely severe attack in RPL-based LLNs.

The rest of the paper is organized as follows. An overview of relevant work is provided in Section II. The basic RPL operations and its potential vulnerabilities are summarized and analyzed in Section III. The hatchetman attack and its performance impact evaluation with extensive simulation experiments are presented in Sections IV and V, respectively. In Section VI, we analyze the hatchetman attack in terms of four criteria. Finally, concluding remarks and future research direction are provided in Section VII.

II. Related Work

While the study of RPL security is relatively new, many researchers have investigated security issues in similar environments. Potential forwarding misbehaviors and its corresponding countermeasures in energy harvesting motivated networks are discussed in [10], [11]. In [12], an explore-based active detection scheme (EBAD) running with DSR is proposed to detect routing attack in MANETs. In the SCAD [13], a lightweight countermeasure to selective forwarding attack is proposed by deploying a single checkpoint node integrated with timeout and hop-by-hop retransmission techniques. An optimal monitoring node selection algorithm is proposed to protect the network against denial-of-service attacks in wireless sensor networks in [14].

In the last few years, a significant amount of research efforts have been focusing on security in RPL-based LLNs. The SVELTE [15] proposes a novel intrusion detection system to secure Low-Power Wireless Personal Area Network (6LoWPAN) running with RPL from network layer and routing attacks. The CMD [16] proposes a monitor-based approach to mitigate the forwarding misbehaviors in LLNs running with RPL, where each node monitors the forwarding behaviors of the preferred parent node to observe the packet loss rate, compares the observation result with the collected packet loss rate from one-hop neighbor nodes, and detects the forwarding misbehaviors of the preferred parent node. In [17], a dynamic threshold mechanism is proposed to mitigate destination advertisement object (DAO) inconsistency attack in RPL-based LLNs. [18] designs and implements an intrusion detection system that can be modified to employ RPL routing protocol in neighborhood area network. In [19], a rank attack that aims at the rank property in RPL and its impact on the performance are investigated in wireless sensor networks, where the adversary can compromise the rank rule to downgrade the RPL performance. Four adversarial scenarios motivated by violating rank rule permanently and non-permanently and their potential performance impact are analyzed. In the VeRA [20], a version number and rank authentication security scheme based on one-way hash chains are proposed to secure the RPL in LLN, where the misbehaving nodes illegitimately increase the version number of DIO message and compromise illegal rank values. In order to protect against the attackers that send DIO messages with higher version number values or that publish a high rank value, the version numbers are binded with authentication data and signatures. A security threat analysis of RPL has been performed in [8], where potential security issues and fundamental countermeasures are presented. [21] analyzes the security capability of the IEEE 802.15.4 MAC protocol as well as the limitations thereof in the context of Internet-of-Things. A more detailed survey of denial-of-service attacks on IoT can be found in [22], [23]. In [5], the history of research efforts in RPL and future research directions on which RPL should evolve have been reviewed and discussed, respectively.

III. The RPL Routing Protocol

RPL [4] is a novel distance vector and source routing protocol designed for low power and lossy networks operating on IEEE 802.15.4 PHY and MAC layers. The basic idea of RPL is to construct one or more Destination-Oriented Directed Acyclic Graphs (DODAGs) to maintain the network state information, where DODAGs are differentiated by RPL Instance ID, DODAG ID, and DODAG Version Number. Each DODAG is associated with a set of nodes and one DODAG root (i.e., base station or gateway node), where nodes can generate and forward data traffic and DODAG root is responsible for collecting the data measured by other nodes, controlling these nodes, and bridging the DODAG with IPv6 networks.

RPL relies on four types of control messages to establish and manage the network topology and routing information: DAG Information Object (DIO), DAG Information Solicitation (DIS), Destination Advertisement Object (DAO), and Destination Advertisement Object Ack (DAO-Ack). In order to construct a DODAG and build upward routes directed from other nodes to the DODAG root, the DODAG root will issue a DIO control message, which includes the DODAG root's ID, the rank of the DODAG root, and an Objective Function which describes the routing metrics and constraints. Any node that receives the DIO message and is willing to join the DODAG should add the DIO message sender to its parent list, computes its own rank according to the piggybacked Objective Function, and passes on the DIO message with the updated rank information. Here, the rank is used to imply the node's position relative to other nodes with respect to a DODAG root, and the rank of nodes along any upward route to the DODAG root should be monotonically decreasing to avoid

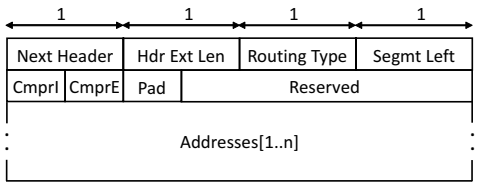


Fig. 1. The format of RPL source route header, where the route information is piggybacked in Address[1..n] field. Here, the length is shown in byte.

any routing loop. The node that has the lowest rank among all the nodes in the parent list is selected as the preferred parent node. After the DODAG is constructed, each node will be able to forward sensory data to the DODAG root by choosing its most preferred parent node as the next-hop forwarding node.

If a new node wants to join the existing network, it can request topology information from the neighbor nodes in the adjacent DODAGs by broadcasting a DIS control message. To build downward routes from the DODAG root to other nodes, the node needs to issue a DAO control message to propagate reverse route information and record the nodes visited along the upward routes. After passing the DAO message to the DODAG root, a complete downward route between the DODAG root and the node is established. Finally, the DODAG root replies a DAO-Ack message as a unicast packet to the source of DAO message as a response.

Unlike prior source routing protocols (i.e., DSR), where each intermediate node can quickly learn the routes of other nodes by aggressively overhearing on-flying packets and caching the piggybacked route information in its routing table, RPL heavily relies on source routing mechanism to forward packet and maintain reachability to destinations within the LLNs. In particular, nodes do not store any information about downward routes to other nodes and only the DODAG root possesses such information. If the DODAG root generates a packet to send, it first searches its routing table for the downward route to the destination node and sends the packet with the cached source route. If a node has a packet to other node, the packet must be first sent through the upward route to the DODAG root, which will forward the packet to its destination node through downward route. If the intermediate node fails to forward the packet with the piggybacked source route, the packet should be dropped. And then the intermediate node replies an Error message back to the DODAG root. RPL implements a strict source routing policy where each and every hop between the source and destination of the source route is specified within the source route header of the packet. Here, the format of RPL source route header is shown in Fig. 1. However, the source routing mechanism can be exploited by an adversary to attack the network as well. For example, a malicious node along the forwarding path can manipulate the source route header of the received packet to disrupt network protocols and interfere with on-going communications.

IV. Hatchetman Attack

In this section, we present our newly discovered attack, called hatchetman attack, in RPL-based LLNs. The basic idea

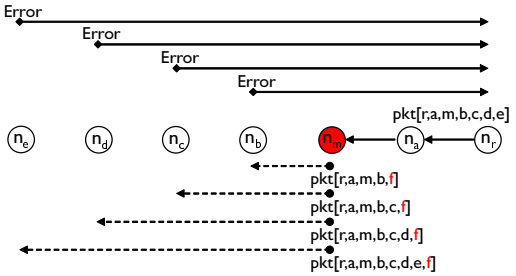


Fig. 2. A snapshot of the network, where a malicious node n_m sends the manipulated packets piggybacked with invalid source route to legitimate nodes. Here, f is the fictitious node address that does not exist in the network.

of hatchetman attack is that the malicious node manipulates the source route header of the received packets, and then generates and sends a large number of invalid packets with error route to legitimate nodes, which cause the legitimate nodes to drop the received packets and reply an excessive number of Error messages back to the DODAG root. As a result, a great number of packets are dropped by legitimate nodes and excessive Error messages exhaust the communication bandwidth and node energy, which lead to a denial of service in RPL-based LLNs. In this paper, we assume that an adversary is able to capture and compromise legitimate node, gain access to all stored information including public and private keys, and reprogram it to behave maliciously [24].

First, when the DODAG root generates a packet to send, it first searches its downward routing table for the route to the destination node, and then piggybacks the cached source route into the packet. Any legitimate node that receives the packet will forward it to the next-hop node according to the piggybacked source route. However, when a malicious node receives the packet, it may manipulate the source route header of the received packet by replacing the post-hops of a legitimate node with a fictitious destination, and then generates and sends the invalid packet with error route to the legitimate node. When the invalid packet reaches the legitimate node that is one-hop prior to the fictitious destination, the receiving node has to drop the packet and reply an Error message back to the source node of the packet, which is the DODAG root. This is because the receiving node cannot forward the packet further to the next-hop node, which is the fictitious destination, based on the piggybacked source route.

For example, suppose the DODAG root n_r sends a packet with the cached source route ($[r, a, m, b, c, d, e]$) to destination node n_e as shown in Fig. 2. When the malicious node n_m receives the packet, $pkt[r, a, m, b, c, d, e]$, it manipulates the source route header by replacing all the post-hops (i.e., $[c, d, e]$) of the legitimate node (i.e., n_b) with a fictitious destination (i.e., n_f), and then sends the invalid packet with error route ($[r,a,m,b,f]$) to the next-hop node, n_b. Here, f is the fictitious node address that does not exist in the network. When n_b receives the packet, $pkt[r,a,m,b,f]$, it drops the received packet and replies an Error message back to the DODAG root. This is because n_b cannot forward the packet to destination node n_f specified in the source route.

Second, if the malicious node generates multiple invalid

Notations:

• $pkt[seq, sr, type]$: A packet with a sequence number, seq, piggybacked source route, sr, and packet type, $type$. Here, $type$ is *Data* or *Error*.

• S_{atk}: The set of intermediate nodes after the malicious node along the source route. E.g., S_{atk} is [b,c,d,e] based on the packet $pkt[seq, [r, a, m, b, c, d, e], Data]$. Here, n_e and n_m is the destination node and malicious node, respectively.

• n_f and frc: A fictitious node that does not exist in the network and an invalid source route, respectively.

• $DRT_r[i]$: A cached source route to node n_i in the downward routing table of DODAG root n_r.

◇ When the DODAG root n_r has a data packet to node n_e:
Send out $pkt[seq, DRT_r[e], Data]$;

◇ When the malicious node n_m receives $pkt[seq, DRT_r[e], Data]$:
Extract S_{atk} from $DRT_r[e]$;
for $n_i \in S_{atk}$
Replace the post-hop node(s) of n_i in $DRT_r[e]$ with n_f;
Send $pkt[seq, frc, Data]$ to n_i;

Fig. 3. The pseudo code of hatchetman attack.

packets with error route, and sends them to each post-hop node of itself along the forwarding path, all the receiving nodes will drop the received packet and reply an Error message back to the DODAG root. For example in Fig. 2, the malicious node n_m can generate and send multiple invalid packets with error route to each post-hop node, n_b, n_c, n_d, and n_e, respectively. And all the receiving nodes will drop the received packet and reply an Error message back to the DODAG root, which cause each intermediate node along the forwarding path to receive and forward a large number of Error messages. As a result, excessive Error messages can significantly exhaust communication bandwidth and node energy, and finally result in a denial of service in RPL-based LLNs. The major operation of hatchetman attack is summarized in Fig. 3.

V. Evaluation

We conduct extensive simulation experiments using the OMNeT++ [9] to evaluate the performance impact of hatchetman attack in RPL-based LLNs. 50 nodes are uniformly distributed in a 150×150 m^2 square network area, where a single DODAG root is deployed. The communication range of each node is 30 (m). The radio model simulates CC2420 with a normal data rate of 250 Kbps, and 802.15.4 MAC/PHY operates with a default configuration in the 2.4 GHz band [25]. To emulate low data rate scenario, packet injection rate is set to 0.1 pkt/sec. A set of malicious nodes are randomly located in the network. And the total simulation time is 5000 seconds, and each simulation scenario is repeated 5 times to obtain steady state performance metrics. In this paper, we measure the performance in terms of packet delivery ratio, throughput, packet delivery latency, energy consumption, the number of attack packets, and attack energy inefficiency by changing key simulation parameters, including channel error rate (r_{cer}), jamming frequency (r_{jf}), and the percentage of attackers (r_{ap}). We compare the performance impact of hatchetman attack with the well-known jamming attack and original RPL without adversary.

First, the packet delivery ratio (PDR) is measured by changing channel error rate (r_{cer}), jamming frequency (r_{jf}), and the percentage of attackers (r_{ap}) in Fig. 4. The RPL with

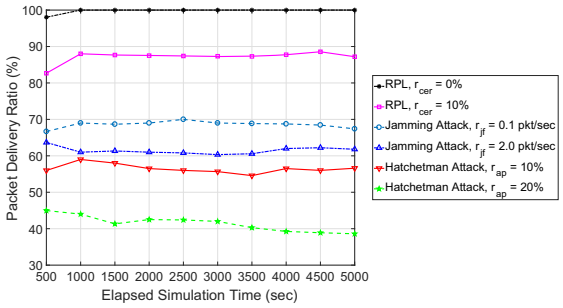


Fig. 4. The performance of PDR against elapsed simulation time.

r_{cer} = 0% achieves the highest PDR, this is because every node cooperatively and faithfully forwards the received packets to the destination node under the ideal channel condition. However, the RPL without adversary is very sensitive to bad channel condition and the PDR is fluctuating around 76% with r_{cer} = 10% because the packets could get lost due to bad channel quality. Under jamming attack, the PDR decreases to 69% and 61% with different jamming frequency, r_{jf} = 0.1 pkt/sec and 2.0 pkt/sec, respectively. Since the packets have more chances to be collided with the jamming packets which are frequently generated by the malicious nodes, the lower PDR is achieved than that of original RPL without adversary. The hatchetman attack with different r_{ap} = 10% and 20% shows the lowest PDR than that of jamming attack and original RPL without adversary. This is because the malicious nodes can frequently manipulate the source route header of the received packets, and send the invalid packets to multiple legitimate nodes, which cause the legitimate nodes to drop the packets. With more number of malicious nodes r_{ap} = 20%, the hatchetman attack causes the PDR to drop below 45%. This is because more number of malicious nodes can generate more invalid packets and send them to legitimate nodes, more packets will be dropped.

Second, we measure the throughput of an intermediate node along the forwarding path by changing r_{cer}, r_{jf}, and r_{ap} in Fig. 5. The jamming attack shows the lowest throughput with different r_{jf} than that of hatchetman attack and RPL without adversary. This is because a large number of packets collide with the jamming packets, less number of packets are received and forwarded by intermediate node, the lowest throughput is achieved. Since more jamming packets are generated with larger jamming frequency r_{jf} = 2.0 pkt/sec, more packets could collide with jamming packets, lower throughput is achieved. RPL without adversary shows higher throughput than that of jamming attack. This is because more packets are received and forwarded by intermediate nodes, higher throughput is achieved. However, since more number of packets could get lost due to bad channel quality, RPL with r_{cer} = 10% shows lower throughput than that of RPL with r_{cer} = 0%. The hatchetman attack with different r_{ap} achieves the highest throughput, this is because a large number of Error messages are generated and forwarded by intermediate nodes along the forwarding path, the throughput are significantly increased. The hatchetman attack with r_{ap} = 20% achieves

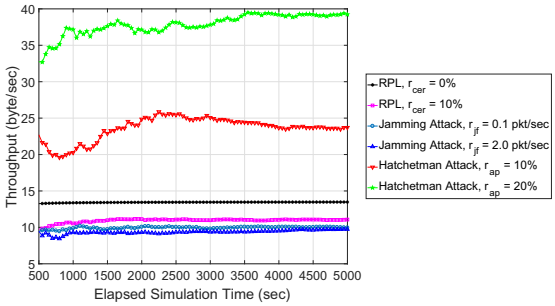
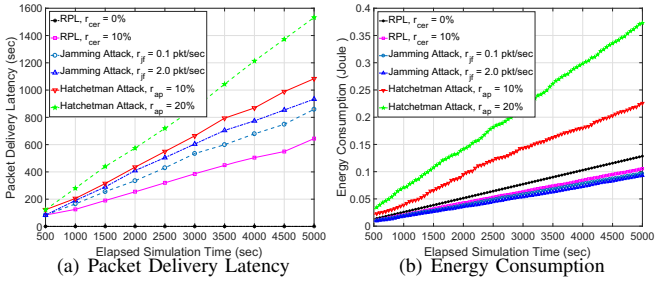


Fig. 5. The performance of throughput against elapsed simulation time.

the higher throughput than that of hatchetman attack with r_{ap} = 10%. This is because more invalid packets with error route can be generated and sent to legitimate nodes, and more Error messages will be generated and replied back to the DODAG root.

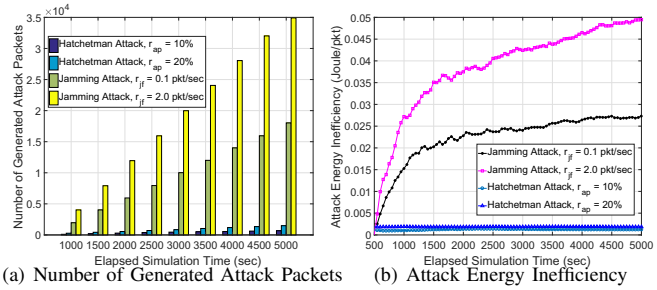
Third, the packet delivery latency is measured by changing r_{cer}, r_{jf}, and r_{ap} in Subfig. 6(a). In this experiment, if the packet is lost due to bad channel quality or forwarding misbehavior, the packet delivery latency is calculated by using the currently elapsed simulation time. The RPL with r_{cer} = 0% achieves the lowest packet delivery latency (around 0.35 sec in average), this is because all the intermediate nodes cooperatively forward the received packets and most of the packets can reach the destination node quickly. However, the packet delivery latency of RPL with r_{cer} = 10% significantly increases as the simulation time elapses, compared to that of RPL with r_{cer} = 0%. This is because some packets could get lost due to bad channel condition, longer latency is achieved. Under jamming attack, since more packets will collide with frequently generated jamming packets, the lost packets will experience a longer delivery time, longer latency is achieved than that of original RPL. The hatchetman attack with r_{ap} = 20% achieves the largest packet delivery latency, this is because the malicious nodes can generate more number of invalid packets with error route to cause the legitimate nodes to drop the packets, more packets will experience a longer delivery latency.

Fourth, we measure the energy consumption of intermediate node along the forwarding path in terms of the number of received and forwarded packets [26] in Subfig. 6(b). The hatchetman attack with different r_{ap} can achieve the higher energy consumption than that of original RPL without adversary and jamming attack. This is because the malicious nodes can generate and send a large number of invalid packets with error route to multiple legitimate nodes, which cause the legitimate nodes to reply an excessive amount of Error messages back to the DODAG root. As a result, each intermediate node along the forwarding path has to receive and forward a high volume of Error messages, thus, the higher energy consumption is achieved. The jamming attack achieves the lowest energy consumption because the packets could be collided with the jamming packets, the number of received and forwarded packets is significantly reduced. The RPL without adversary shows higher and lower energy consumption than that of jamming attack and hatchetman attack, respectively.



(a) Packet Delivery Latency (b) Energy Consumption

Fig. 6. The performance of packet delivery latency and energy consumption against elapsed simulation time.



(a) Number of Generated Attack Packets (b) Attack Energy Inefficiency

Fig. 7. The performance of the number of generated attack packets and attack energy inefficiency against elapsed simulation time.

Fifth, we measure the number of generated attack packets, which are invalid packet with error route and jamming packet in hatchetman attack and jamming attack, respectively, by changing r_{cer}, r_{jf}, and r_{ap} in Subfig. 7(a). The hatchetman attack generates an invalid packet with error route whenever the malicious node receives a packet to other node. Since the low data rate (0.1 pkt/sec) is adopted in the experiments, the less number of attack packets will be generated by hatchetman attack. However, the jamming attack frequently generates the jamming packets to cause the packet collision, thus, excessive number of attack packets are observed.

Finally, the attack energy inefficiency is measured by changing r_{jf} and r_{ap} in Subfig. 7(b). Here, the attack energy inefficiency is calculated as the total energy consumption of sending the attack packets divided by the total number of generated attack packets observed in Subfig. 7(a). And the attack energy inefficiency indicates how energy-efficiently the malicious nodes can attack the network. The hatchetman attack shows the lowest attack energy inefficiency, this is because the less number of attack packets are generated by the malicious nodes. However, the jamming attack achieves much higher attack energy inefficiency than that of hatchetman attack. This is because more number of jamming packets are generated and more energy are consumed by the malicious nodes. This simulation result also indicates that the hatchetman attack can severely attack the network with less energy consumption.

VI. Discussion

In this section, we analyze the hatchetman attack in terms of attack method, stealthiness, attack energy inefficiency, and level of denial of service. The basic idea of hatchetman attack is that the malicious node manipulates the source route header of the received packet to generate the invalid packets with

error route, and then selects the legitimate nodes as target nodes and sends the invalid packets to these target nodes. According to the RPL standard, the legitimate nodes will drop the received invalid packets and reply an excessive number of Error messages back to the source of the packet, which is the DODAG root. Based on the above described attack method, the hatchetman attack has high stealthiness and more difficult to detect. This is because the malicious node acts like a normal node, but sends the invalid packets to legitimate nodes to make them attack network, for example dropping the received packets and replying a large number of Error messages. In addition, the hatchetman attack shows the lower attack energy inefficiency compared to that of jamming attack because the less number of attack packets are generated by the malicious node as shown in Fig. 7. In terms of the level of denial of service, since an excessive number of Error messages are generated and forwarded by each intermediate node along the forwarding path, which exhaust the communication bandwidth and node energy, channel condition will get worse and the legitimate nodes consume a significant amount of energy. Eventually, the hatchetman attack can lead to an extremely severe denial of service in RPL-based LLNs.

VII. Conclusion and Future Work

In this paper, we investigate the hatchetman attack, which is a new and severe denial-of-service attack in RPL-based low power and lossy networks (LLNs). In hatchetman attack, the malicious node manipulates the source route header of the received packets, and then generates and sends the invalid packets with error route to legitimate nodes to cause the legitimate nodes to drop the received packets and reply an excessive number of Error messages back to the DODAG root, which eventually lead to a denial of service in RPL-based LLNs. We analyze the hatchetman attack and compare it with the well-known jamming attack and original RPL without adversary. Extensive simulation results indicate that the hatchetman attack is a severe denial-of-service attack, which significantly decreases the PDR and increases the packet delivery latency, energy consumption, and throughput.

As a future work, we plan to propose a light-weight countermeasure to mitigate the hatchetman attack in RPL-based LLNs. For example, each intermediate node along the forwarding path can maintain a threshold to limit the rate of forwarding Error messages within a time period. If the number of forwarded Error messages exceeds the threshold, all further Error messages will be rejected. In order to dynamically react to different attack patterns under varying network conditions, the threshold should be adaptively adjusted based on the number of forwarded Error messages as well as the estimated normal Error message rate. To see the full potential of the proposed countermeasure, we plan to develop a small-scale testbed for the experimental study and implementation.

Acknowledgment

This research was supported by Startup grant in Weisberg Division of Computer Science at Marshall University.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] *Gartner Research*, 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016, February 2017.

[3] *The Internet Engineering Task Force (IETF)*, https://www.ietf.org.

[4] T. Winter and P. Thubert, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *RFC Standard 6550*, March 2012.

[5] H. Kim, J. Ko, D. Culler, and J. Paek, "Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey," *IEEE Commun. Surveys Tuts.*, Sep 2017.

[6] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, "Management of Resource Constrained Devices in the Internet of Things," *IEEE Commun. Mag.*, vol. 50, no. 12, p. 144149, 2012.

[7] A. Sehgal, A. Mayzaud, R. Badonnel, I. Chrisment, and J. Schnwlder, "Addressing DODAG Inconsistency Attacks in RPL Networks," in *Proc. IEEE GIIS*, 2014, pp. 1–8.

[8] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, and M. Richardson, "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)," *RFC Standard 7416*, January 2015.

[9] A. Varga, *OMNeT++*, 2014, http://www.omnetpp.org/.

[10] C. Pu and S. Lim, "Spy vs. Spy: Camouflage-based Active Detection in Energy Harvesting Motivated Networks," in *Proc. IEEE MILCOM*, 2015, pp. 903–908.

[11] C. Pu, S. Lim, J. Byungkwan, and M. Manki, "Mitigating Stealthy Collision Attack in Energy Harvesting Motivated Networks," in *Proc. IEEE MILCOM*, 2017, pp. 575–580.

[12] C. Pu, S. Lim, C. Jinseok, and J. Byungkwan, "Active Detection in Mitigating Routing Misbehavior for MANETs," *Wireless Network (2017)*, https://doi.org/10.1007/s11276-017-1621-z.

[13] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, pp. 1–9, 2016.

[14] Q. Monnet, L. Mokdad, and J. Ben-Othman, "Energy-balancing method to detect denial of service attacks in wireless sensor networks," in *Proc. IEEE ICC*, 2014, pp. 106–111.

[15] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time intrusion detection in the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[16] C. Pu and S. Hajjar, "Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks," in *Proc. IEEE CCNC*, 2018, pp. 1–6.

[17] C. Pu, "Mitigating DAO Inconsistency Attack in RPL-based Low Power and Lossy Networks," in *Proc. IEEE CCWC*, 2018, pp. 570–574.

[18] N. Beigi-Mohammadi, J. Misic, H. Khazaei, and V. B. Misic, "An Intrusion Detection System for Smart Grid Neighborhood Area Network," in *Proc. IEEE ICC*, 2014, pp. 4125–4130.

[19] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks," *IEEE Sensors J.*, vol. 11, no. 10, pp. 3685–3692, 2013.

[20] A. Dvir, T. Holczer, and L. Buttyan, "VeRA-Version Number and Rank Authentication in RPL," in *Proc. IEEE MASS*, 2011, pp. 709–714.

[21] S. M. Sajjad and M. Yousaf, "Security analysis of IEEE 802.15. 4 MAC in the context of Internet of Things (IoT)," in *Proc. IEEE CIACS*, 2014, pp. 9–14.

[22] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, "Denial-of-Service detection in 6LoWPAN based Internet of Things," in *Proc. IEEE WiMob*, 2013, pp. 600–607.

[23] A. Rghioui, A. Khannous, and M. Bouhorma, "Denial-of-Service attacks on 6LoWPAN-RPL networks: Threats and an intrusion detection system proposition," *Journal of Advanced Computer Science & Technology*, vol. 3, no. 2, pp. 143–152, 2014.

[24] S. Challa, M. Wazid, A. Das, N. Kumar, A. Reddy, E. Yoon, and K. Yoo, "Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[25] A. Boulis, *Castalia*, 2014, http://castalia.forge.nicta.com.au.

[26] K. Zeng, K. Ren, W. Lou, and P. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.