DIO Suppression Attack Against Routing in the Internet of Things

Pericle Perazzo, Carlo Vallati, Giuseppe Anastasi, and Gianluca Dini

Department of Information Engineering
University of Pisa, 56122 Pisa, Italy
e-mail: [name].[surname]@unipi.it

***Abstract*—Recent standardization efforts are consolidating the role of RPL as the standard routing protocol for IPv6-based Wireless Sensor and Actuator Networks (WSANs). Investigating possible attacks against RPL is a top priority to improve the security of the future Internet of Things (IoT) systems. In this paper, we present the *DIO suppression attack*, a novel degradation-of-service attack against RPL. Unlike other attacks in the literature, the DIO suppression attack does not require to steal cryptographic keys from some legitimate node. We show that the attack severely degrades the routing service, and it is far less energy-expensive than a jamming attack.**

***Index Terms*—Internet of Things, RPL, secure routing, routing attacks, Trickle algorithm.**

I. Introduction

The Internet of Things (IoT) vision foresees a future in which information systems will be seamlessly integrated with smart objects, i.e., common objects empowered with communication capabilities [1]. IoT applications are expected to penetrate our daily lives in areas as diverse as e-health (e.g., remote patient monitoring), smart home (e.g., smart lighting and heating), and smart city (e.g., smart traffic and parking applications). In this context, Wireless Sensor and Actuator Networks (WSANs) will represent a key enabler for IoT deployments. WSANs guarantee rapid installation of smart objects to cover large areas, so keeping the deployment costs low. Data delivery through wireless links in a multi-hop fashion reduces the need for complex network infrastructure and guarantees the flexibility required for expansion and evolution. In this context, securing the routing functionalities will be a major challenge to protect IoT systems against malicious actions aimed at disrupting network operations [2].

The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [3], standardized by the IETF ROLL working group in 2012, is currently considered the most mature option to connect IPv6-enabled resource-constrained devices over lossy links [4]. Since its proposal, the security of RPL has been analyzed by the research community [5], [6], [7]. Dvir et al. [5] studied the impact of the *sinkhole attack*, in which a malicious node injects in the network RPL messages carrying fake information, which attracts traffic from surrounding honest nodes. The malicious node can then intercept and/or discard a large amount of traffic. Mayzaud et al. [7] studied the impact of the *DODAG Version attack*, which has a similar effect. Both sinkhole and DODAG Version attacks can degrade or break completely the routing service. The authors of [5], [8], and [9] proposed and evaluated countermeasures to these attacks. Le et al. [6] studied the impact of the *rank attack*, in which a malicious node misbehaves in selecting next-hop nodes and advertises a wrong distance to the border router. Although all the above attacks have quite a severe impact on the routing functionalities, they require the adversary to successfully forge bogus RPL messages. This could be infeasible if such messages are cryptographically authenticated as specified by the RPL standard [3], unless the adversary steals keys from some legitimate node.

In this paper, we present the *DIO suppression attack*, which can severely degrade the routing service in RPL. The DIO suppression attack induces victim nodes to suppress the transmission of *DIO messages*, which are the RPL messages necessary to build the routing topology. This causes a general degradation of the routes' quality that can lead, eventually, to network partitions. Unlike other RPL attacks in the literature, the DIO suppression attack does not require the adversary to forge bogus RPL messages. It is sufficient that she periodically replays previously heard messages. The attack can thus be mounted without stealing cryptographic keys from legitimate nodes. The DIO suppression attack uses the replay technique, which is a classic attack technique, for a radically different purpose. Indeed, the replay technique is usually used to make a victim accept old information as new. On the other hand, in the DIO suppression attack it is used to make a victim believe that the routing information it is about to send is already being transmitted many times by other nodes. We show that the attack severely degrades the routing service, and it is far less energy-expensive than a jamming attack.

The rest of the paper is organized as follows. Section II describes the RPL routing protocol. Section III describes the DIO suppression attack. Section IV evaluates experimentally the impact and the cost of the attack. The paper is concluded in Section V.

II. RPL Protocol and Trickle Algorithm

RPL [3], [4] is a distance-vector routing protocol that takes into account the unreliable nature of wireless communication and the limited available power of devices by minimizing the complexity of its functionalities, and by reducing the signaling overhead. Its design assumes that the majority of the application traffic is upward, i.e., generated by nodes and directed towards a single node acting as a border router. Downward traffic, i.e., generated by the border router towards other nodes, is assumed to be sporadic, while node-to-node interactions are considered rare. For this reason, RPL builds and maintains a logical topology for upstream data delivery, while downward routes are established only when required. Specifically, the topology is a *Destination Oriented Directed*

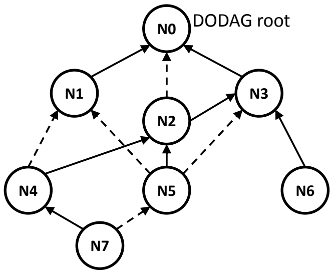

Fig. 1. Example of DODAG. Solid arrows point to preferred parents, dashed arrows point to other parents in the parent set.

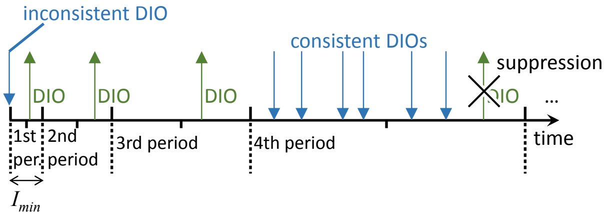

Fig. 2. Example of Trickle algorithm with $k = 6$. Upward arrows represent emitted DIOs, downward arrows represent received ones, and the black cross represents a DIO suppression.

Acyclic Graph (DODAG), an example of which is shown in Fig. 1. In a DODAG, every node selects a set of neighbors, called *parent set*, as candidates for upstream data delivery. One of the nodes within the parent set is selected as the *preferred parent*, which is exploited for the actual data forwarding. The DODAG is rooted in a single node, the *DODAG root*, to which all upstream data is directed. The DODAG root acts also as a border router for other networks. The DODAG root triggers the RPL topology formation by emitting *DODAG Information Object* (DIO) messages. Non-root nodes listen for DIOs and use the included information to join the DODAG. Upon joining the DODAG, a node also starts emitting DIOs to advertise its presence and its distance to the root.

The emission of DIOs is regulated by the *Trickle algorithm* [10]. Trickle was originally designed for polite gossiping in wireless networks, to reduce the power consumption of the nodes by minimizing the redundant messages and by dynamically adapting the transmission rate. In particular, the emission rate of DIOs is tuned according to the stability of routing information. If the information included in DIOs from the neighbors is *consistent* with internal routing information, then the emission rate is reduced. Otherwise, if *inconsistent* DIOs are received, then the emission rate is increased. RPL specifies the conditions to determine if a DIO is consistent. For example, a DIO that causes no changes in the parent set, the preferred parent, and the distance to the root must be considered consistent.

The Trickle algorithm divides time in periods of variable length (Fig. 2). The node schedules the transmission of a DIO message at a random time t in the second half of each period. Until t, the node listens for messages and keeps track of the consistent DIOs. At time t, the scheduled DIO message is broadcast only if the number of consistent DIOs received within the current period is below a given *suppression threshold* (k). Otherwise, the transmission of the DIO is suppressed, as it happens during the 4th period in the example of Fig. 2. At the end of the period, if only consistent DIOs have been received, the length of the next period is doubled, until a maximum length I_{max} is reached. At any time, if an inconsistent DIO is received, the current period is interrupted and the algorithm starts again from a period of a minimum length I_{min}. The mechanism of DIO suppression is an essential part of the Trickle algorithm, since it makes DIO traffic scale logarithmically with the number of the nodes. Disabling this mechanism is not recommended, because it can lead to congestion in dense networks [10].

III. DIO Suppression Attack

The goal of the DIO suppression attack is to interrupt or slow down the transmission of the DIO messages in the network. To this aim, the DIO suppression mechanism of the Trickle algorithm is exploited. In this attack, the adversary transmits repeatedly a DIO message that is considered consistent by the receiving nodes. If the nodes receive enough consistent DIOs, they will suppress their own DIO transmission. Since DIO messages are exploited to discover neighbors and the network topology, their continuous suppression can cause some nodes to remain hidden and some routes to remain undiscovered. The effect is a general degradation of the routes' quality or, in the worst case, a partition of the network.

A simple way to mount a DIO suppression attack is to eavesdrop a DIO message from a legitimate node and then replay it many times with a fixed frequency. The surrounding legitimate nodes will consider the replayed DIOs consistent. Indeed, receiving a DIO equal to the last received one will cause no changes in their parent set, their preferred parent, or their distance to the root. Let us illustrate this attack by means of an example. Consider the network depicted in Fig. 1 and suppose that the adversary places a malicious device in the proximity of node N0. As soon as the malicious device eavesdrops a DIO emitted by N0, it starts broadcasting the message with a fixed interval. If the number of messages is sufficient to activate the suppression threshold of N1, N2 and N3, they all suppress the emission of DIOs. If the number of replayed DIOs is enough to cause the suppression of all the legitimate DIOs, then the network will be partitioned, since some nodes (e.g., N4) will not receive any routing information. If instead the number of replayed DIOs are sufficient, the emission of the legitimate DIOs will be significantly reduced, thus impairing the proper network formation and the propagation of fresh routing information. For instance, let us suppose that node N4 changes its parent using a more convenient route, then the propagation of the updated route towards N7 can be delayed by the attack. This results in the use of suboptimal routes, e.g., N7 might select N5 as preferred parent for a certain period.

Note that the DIO suppression attack is more convenient in terms of power consumption compared to the simple jamming of DIO messages. Indeed, according to Trickle, the DIO messages are sent at random times. Unless the adversary can predict these times, she has to jam the channel continuously. In contrast, the DIO suppression attack requires the adversary to transmit only k DIO messages at each Trickle period. This significant power saving allows the adversary to maintain the attack for a longer time.

Simply appending a Message Integrity Code (MIC) to the messages, as specified by the security features of RPL [3], is not effective in preventing the DIO suppression attack. Indeed, to mount the attack it is sufficient to replay previously heard DIOs, including their MIC, without modifications.

Another possible countermeasure is to enable MAC-layer encryption in order to impede the adversary from identifying DIO messages and distinguishing them from data messages or other types of routing messages. Unfortunately, this countermeasure alone may not be effective to prevent the attack. Indeed, the adversary can exploit some specific features to identify DIO messages. DIOs are sent as multicast frames, which can be distinguished from the unicast ones from the MAC header which is never encrypted. Among the multicast frames, the DIO messages can be identified through their payload size. This is possible because many widespread MAC protocols like IEEE 802.15.4 uses the CCM cryptographic mode of operation, which does not change the size of the frame payload when encrypting. As a practical example, in the standard Contiki RPL implementation with the default settings and the 802.15.4 MAC encryption enabled, DIO messages are the only multicast frames having a payload of 80 bytes[1]. Even if the adversary fails in identifying the DIO messages directly by their size, for example because of the presence of variable-length options, she can undertake other actions. For example, she can identify and replay a multicast *DODAG Information Solicitation* (DIS) message, which causes a legitimate receiving node to reset its Trickle timer. After a wait shorter than I_{min}, such a node will send a DIO, which can be captured by the adversary. Identifying a DIS message from its size is simpler than a DIO message, because a DIS is quite small (10 bytes) and can only have one fixed-size option (Solicited Information option, 21 bytes).

On the other hand, a replay protection mechanism can be effective to counteract the attack, because it allows the legitimate nodes to detect and discard the replayed DIO messages. Although these mechanisms can be handled by recent platforms in terms of CPU and memory, they result in a significant overhead in terms of additional control messages. For example, the RPL standard includes an optional replay protection mechanism [3] which, to be fully secure, needs a cryptographic challenge-response handshake to assess the freshness of the first message received from each new neighbor. Such handshake can be implemented by means of *Consistency Check* (CC) messages, whose format is specified by the RPL standard. Then, the freshness of the messages after the first one is guaranteed by an integrity-protected incremental counter field. The initial challenge-response handshake significantly increases the signaling overhead and consequently the energy consumption, but also introduces a significant delay in the routing operation. Such overhead delays the overall network formation and makes the routing protocol cumbersome to react to topology changes. Alternatively, a replay protection mechanism can rely on a tight synchronization between the nodes' clocks, which avoids the need for challenge-response handshakes. However, a secure tight synchronization may not be possible or cost-affordable in many WSANs.

IV. Experimental Evaluation

In order to evaluate the effects of the DIO suppression attack on a real RPL network, a set of experiments have been run using the Contiki operating system, a popular open-source operating system for constrained devices. The operating system has been modified in order to implement the DIO suppression attack on malicious devices. Specifically, a malicious device is programmed to wait for the emission of a DIO message from a given legitimate node (*replay source node*), and then to replay this message repeatedly with a fixed interval (*replay interval*). Experiments have been run using Cooja [11], a network emulator which is available as part of the Contiki distribution. We used Cooja to emulate Tmote Sky sensor mote, an MSP430-based board with an IEEE 802.15.4-compatible CC2420 radio chip. We used the same emulated hardware for both the legitimate nodes and the malicious devices. To simulate a realistic channel, we adopted the Multipath Ray-tracer Medium (MRM) model, a propagation model that implements ray-tracing techniques with various propagation effects.

The scenario considered is a network composed of one root and 30 non-root nodes, placed randomly on a 20m×20m playground. The nodes are programmed to send one data packet of 30 bytes every 60 seconds to the root node as application data. A set of 5 malicious devices implementing the DIO suppression attack are introduced in the network.

Note that in this scenario, replaying DIOs has a negative impact on the routing service even without considering the DIO suppression. Indeed, if the replayed DIOs are received by nodes that are not in the communication range of the replay source node, these nodes could believe that the replay source node is reachable when it is not. If the victim nodes select such unreachable node as their preferred parent, their route towards the DODAG root will include a non-existing link and traffic will not be delivered. This effect is similar to a HELLO flood attack [2]. The combined effect of HELLO flood and DIO suppression on a network is not evaluated in this paper and is left as future work. In order to isolate only the effect of the DIO suppression attack, in this evaluation each malicious device is placed in the *close proximity* of its replay source node. In this way, the replay source node is reachable by every node that receives its replayed DIOs, thus excluding the HELLO flood effect.

In order to assess the impact and the cost of the attack, the following metrics are adopted:

- *Network path stretch*, defined as the fraction of nodes having a path stretch greater than one [12]. The path stretch of a node is the difference between its actual route cost and the cost of the shortest path.
- *Network packet delivery ratio*, defined as the average value of the packet delivery ratios experienced by the nodes. The packet delivery ratio of a node is the ratio between the number of application packets received by root and the overall number of packets sent by the node.

[1] DIS messages are 31 bytes, Neighbor Solicitations 46 bytes, Neighbor Advertisements 44 bytes, Router Solicitations 28 bytes, Router Advertisements 46 bytes.

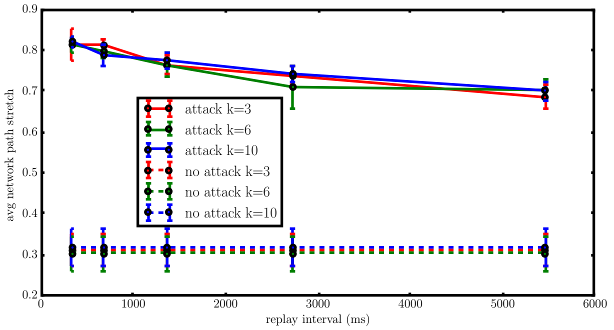

Fig. 3. Average network path stretch.

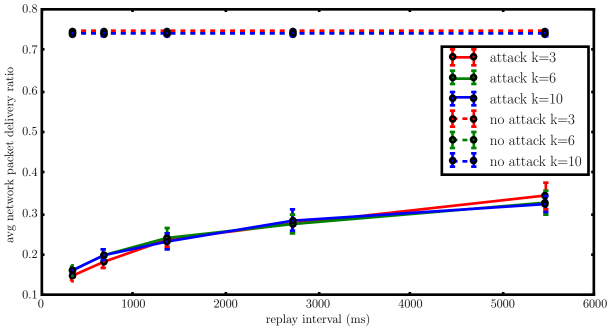

Fig. 4. Average network packet delivery ratio.

- *Adversarial power consumption*, defined as the average power consumption of the malicious devices. This metric expresses the cost of the attack.

In order to obtain statistically sound results, 30 independent replications with different seeds are run for each scenario. The average values with its 95%-confidence interval are reported.

Fig. 3 illustrates the average network path stretch with different replay intervals. Different values of the suppression threshold are considered, namely $k = 3, 6, 10$. In order to provide a term of comparison, the results obtained with the unattacked network are also shown. Of course, the replay interval is not meaningful in case of unattacked network. It can be seen that the attack causes a significant degradation of the route quality for a significant number of nodes in the network. As expected, the routes' quality degradation strictly depends on the replay interval: the shorter the replay interval is, the higher is the network path stretch. This can be explained considering that a shorter replay interval increases the likelihood of DIO suppression on the victim nodes.

In order to show the practical effects of the degradation of RPL performance, in Fig. 4 the average network packet delivery ratio is shown. As can be seen, the attack significantly impairs network ability to successfully deliver application traffic to destination, reducing the delivery ratio from an average value of 0.75 to a value in between 0.15 and 0.35. It is important to highlight that even with large replay intervals, the delivery ratio is halved by the attack.

In Fig. 5 the average adversarial power consumption is shown. The power consumption of a node performing a continuous jamming is also reported, to compare the cost of the DIO suppression attack with an attack that completely disrupts the network formation. The cost of the jammer is evaluated as the power consumption of a node that keeps its wireless transceiver always in transmit state. As can be seen, the power consumption of a DIO suppression attack is on average five times lower than a jamming one. As shown in Fig. 5, the larger the replay interval is, the lower is the adversarial power consumption. This suggests a possible trade-off between the cost of the attack and its impact on routing.

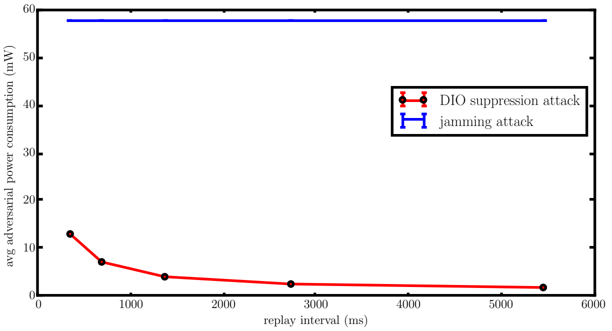

Fig. 5. Average adversarial power consumption.

V. Conclusions

In this paper, we presented the DIO suppression attack, an attack that induces victim nodes to suppress the transmission of DIO messages. This causes a general degradation of the routes' quality that can lead, eventually, to network partitions. Unlike other RPL attacks in the literature, the DIO suppression attack does not require the adversary to forge bogus RPL messages. The attack can thus be mounted without stealing cryptographic keys from legitimate nodes. We showed that the attack severely degrades the routing service, and it is far less energy-expensive than a jamming attack.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[3] T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Requests for Comments, RFC 6550, 2012.

[4] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Computer Networks*, vol. 56, no. 14, pp. 3163–3178, 2012.

[5] A. Dvir, T. Holczer, and L. Buttyan, "VeRA - version number and rank authentication in RPL," in *IEEE 8th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, 2011, pp. 709–714.

[6] 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 Journal*, vol. 13, no. 10, pp. 3685–3692, 2013.

[7] A. Mayzaud, A. Sehgal, R. Badonnel, I. Chrisment, and J. Schönwälder, "A study of RPL DODAG version attacks," in *IFIP 8th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, 2014, pp. 92–104.

[8] K. Weekly and K. Pister, "Evaluating sinkhole defense techniques in RPL networks," in *IEEE 20th International Conference on Network Protocols (ICNP)*, 2012, pp. 1–6.

[9] H. Perrey, M. Landsmann, O. Ugus, M. Wählisch, and T. C. Schmidt, "TRAIL: Topology authentication in RPL," in *International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2016, pp. 59–64.

[10] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle algorithm," Internet Requests for Comments, RFC 6206, 2011.

[11] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *IEEE 31st Conference on Local Computer Networks (LCN)*, 2006, pp. 641–648.

[12] C. Vallati and E. Mingozzi, "Trickle-F: Fair broadcast suppression to improve energy-efficient route formation with the RPL routing protocol," in *IFIP 3rd Sustainable Internet and ICT for Sustainability (SustainIT)*, 2013, pp. 1–9.