Using Provenance to Detect Selective Forwarding Attack in RPL-Based Internet of Things
Provenance를 사용한 RPL기반 사물인터넷 상 선택적 포워딩 공격 감지

저자
(Authors)
Sabah Suhail, Shashi Raj Pandey, Choong Seon Hong

출처
(Source)
정보과학회 컴퓨팅의 실제 논문지 26(1), 2020.1, 20–25(6 pages)

발행처
(Publisher)
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09289754

APA Style
Sabah Suhail, Shashi Raj Pandey, Choong Seon Hong (2020). Using Provenance to Detect Selective Forwarding Attack in RPL-Based Internet of Things. 정보과학회 컴퓨팅의 실제 논문지, 26(1), 20–25

이용정보
(Accessed)
경희대학교 국제캠퍼스
163.***.116.156
2020/03/28 12:43 (KST)
Abstract In the Internet of Things (IoT), resource–constrained things can connect to the Internet via IPv6 and 6LoWPAN networks. The Routing Protocol for Low–Power and Lossy Networks (RPL) has enabled such interconnection. However, the data transportation using RPL is vulnerable to various attacks due to the interaction between unattended things with the unreliable Internet. For instance, the data generated by sensors are vulnerable to attacks (for instance, selective forwarding attack). Therefore, error–free and reliable information cannot be assured in the decision–making process. During data transmission from source to destination, provenance can be used to track data acquisition and data traversal. In this paper, we use provenance to evaluate the network performance by computing the packet delivery ratio (PDR) at each forwarding node in the packet path. Furthermore, to identify the faulty nodes, we counted the packets received from the respective child nodes in the routing table at each parent node participating in the network. We have evaluated the proposed approach for RPL–based IoT in terms of provenance size, provenance generation time, and memory consumption.

Keywords: anomaly–based detection, IoT, provenance, RPL, selective forwarding attack, 6LoWPAN
1. Introduction

IoT is deployed in various application areas; for instance, environmental monitoring, energy management, health-care system, industrial automation, surveillance, and military [1]. To enable communication between the resource-constrained things with the Internet, the RPL routing protocol has been standardized for constrained environments (6LoWPAN networks). RPL [2] is a gradient-based routing protocol for low power and lossy network (LLN) such as sensor networks. The RPL protocol works by constructing a Destination-oriented Directed Acyclic Graph (DODAG) in which the nodes (root, source, forwarding) exchange control messages\(^1\) to establish packet path proactively. Firstly, the root node multicast DIO messages to its neighboring nodes. Upon reception, the neighboring nodes compute their rank\(^2\), join DODAG, and choose a preferred parent. A node can also request DIO message from its neighboring node via DIS message if it cannot receive a DIO within a specific time interval. The child nodes unicast DAO messages to their respective parent node to establish point-to-multipoint and point-to-point connectivity. Each node maintains a routing table (RT) to maintain information about its child node(s) and parent node(s). The DIOs are sent aperiodically based on the trickle timer for maintaining the overall network topology. In order to send a data packet, the source node forwards the data packet to its preferred parent node which in turn keeps on forwarding the data packet until the packet arrives at the root node for further processing and analysis of sensor data. The forwarding decision is based on routing information maintained at RT of each node in the network.

In RPL implementation, the secure operation modes are usually not enabled making it vulnerable to various attacks such as selective forwarding attack. In selective forwarding attack [3], an attacker forwards only selected packets. For instance, an attacker may forward only routing or control messages and drop all other packets to disconnect legitimate nodes from the network. To identify the source of an attack, it is important to find the source causing the data loss or network interruption.

In this regard, provenance can be deployed to keep track of data source and the actions performed by the other participating entities during the data propagation and processing [4]. Though provenance has been used extensively in various application areas including databases, scientific workflows, distributed systems, and networks [5]. However, provenance management in the IoT domain requires consideration on constraints, for instance, storage, energy, and processing [6]. The use of provenance in WSN has been studied by [6, 8] to detect anomalies in the network. However, in comparison to WSNs, the idea of provenance in the IoT is yet to be explored keeping in view the challenging requirements coupled with the unique architecture of the IoT, for instance, things are globally accessible, operate in an unattended environment, resource-constrained in nature, and connected through the lossy links [9]. Furthermore, the detection of selective forwarding attack is discussed in [7,9], however, constructing provenance information based on packet traversal is not given due attention. In this paper, we introduce the idea of provenance for RPL-connected IoT. The main contributions of this paper are as follows.

We propose a provenance-enabled scheme to identify selective forwarding attack in RPL-based IoT environment. To identify an anomaly in the network, we compute PDR at each forwarding node and embed PDR as provenance data in the payload. For further investigation of the malicious node, we insert the packet count received by the child node in RT of the parent node. We implement the proposed scheme in Contiki OS [10] and evaluate it in terms of provenance size, provenance generation time, and memory consumption.

The rest of the paper is organized as follows. Section 2 discusses the system model. Section 3 explains the working of the proposed provenance scheme. Section 4 presents the results. Finally, we conclude the paper with future research directions in Section 5.

\(^1\) Types of control messages: DODAG Information Solicitation (DIS), DODAG Destination Advertisement Object (DAO), and DODAG Information Object (DIO).

\(^2\) rank defines the distance of a node from the root node.
2. System Model

In this section, we explain the main components of our proposed framework in terms of the network model, data model, and provenance model. We also discuss how an attacker can drop random data packets in our attacker model.

2.1 Network Model

The network is modeled as a directed acyclic graph (DAG) $G(N, R)$, where $N$ represents the set of nodes (including source node and forwarding node) responsible for generating and forwarding the data packet based on routing information $R$ maintained by the Routing Table (RT). Fig. 1 presents a typical system model where a border router collects sensor data from an RPL-DODAG and ultimately sends the sensor data to the Internet host for further processing.

2.2 Data Model

In our data model, we consider that a source node ($n_s$) sends a data packet $d_p$ to its neighboring node at regular intervals. Before sending the packet, $n_s$ increments a counter $c$ to keep track number of sent packets by $n_s$. A $d_p$ consists of the following three entities (as shown in Fig. 2): i) a unique sequence number Seq#, ii) payload, and iii) Average PDR (as provenance information).

2.3 Provenance Model

The provenance data ($P_{data}$) consists of average PDR computed at each of the forwarding node ($n_f$) in the packet path with respect to the number of packets sent by the source node ($n_s$). The average PDR can be calculated as:

$$\text{Average PDR} = \frac{\text{packet_by} \ n_f}{\text{packet_by} \ n_s}$$  \hspace{1cm} (1)

Thus, each $n_f$ embeds $P_{data}$ in the payload as:

$$P_{data} = \text{payload} + \text{Average PDR @} n_f$$  \hspace{1cm} (2)

Furthermore, each $n_f$ is also responsible for maintaining information about the total number of packets received ($\text{packet}_v$) by its subsequent child node at its RT.

2.4 Attacker Model

In our attacker model, we assume that an attacker node can impersonate as a malicious forwarding node ($M$) in the RPL network. Upon packet reception from a legitimate source or forwarding node, $M$ may randomly drop some of the data packets while forwarding control packets i.e., performing a selective forwarding attack. Under such circumstance, the PDR at subsequent forwarding nodes is affected which eventually causes data quality degradation at the root node.

3. Provenance Scheme

In this section, we discuss the provenance encoding and decoding scheme that compute PDR at each forwarding node in the packet path to identify the anomaly (high packet drop rate) occurred either by the malicious nodes or by network disruptions. We have adopted anomaly-based detection mechanism that tries to detect anomalies in the system by determining the ordinary behavior and using it as a baseline. Therefore, any deviations from that baseline are considered an anomaly.

![Data packet entities](image_url)

**Algorithm 1 Provenance Embedding**

**Input:** $d_p$

**Output:** $d_p \leftarrow \text{PDR, RT@} n_f \leftarrow \text{packet}_v$

1. if (node = $n_s$) then
   1. $n_s$ sends $d_p$ to its parent node
   1. increment $c$  \hspace{1cm} $\triangleright$ count number of sent packets.
2. else for each forwarding node $n_f$ do
   1. $n_f$ computes PDR \hspace{1cm} $\triangleright$ PDR is computed as mentioned in eq. 1
   1. $n_f$ forwards $d_p$ to its parent node
   end for

end if
3.1 Provenance Embedding

The process of provenance embedding consists of two main steps. Firstly, upon packet reception, the forwarding node \( n_f \) computes PDR with respect to the number of packets sent by the source node \( n_s \) and embeds the PDR in the \( d_p \) along with payload. Secondly, \( n_f \) inserts the total number of received packets \( \text{packet}_f \) from its immediate child node in its routing table (RT) against respective child node entries.

Fig. 3 shows the provenance embedding workflow diagram. Let us consider an example scenario (as shown in Fig. 4) to explain the provenance encoding process. Suppose the path traversed by packet is \( <l,n,q,r> \), \( l \) as a source node sends data to its preferred parent \( n \). \( n \) will compute the PDR (as in eq. 1) and inserts it in the \( d_p \) along with the payload. \( n \) computes PDR and embeds it in the payload. It also inserts the count \( c \) of \( \text{packet}_f \) in the RT against the routing entry of its child node \( l \) and then forwards the packet to \( q \). Similarly, the next node in the packet path i.e., \( q \) updates the payload by inserting its PDR and updates its RT entries with the count of \( \text{packet}_f \) from its child node \( n \). This process continues until the packet arrives at the root node \( r \).

3.2 Provenance Decoding

The root node \( n_r \) verify the \( P_{data} \) in two incremental steps. Step 1: it extracts the PDR from the payload and compare it with the minimum baseline PDR threshold \( \tau \). If PDR is less than \( \tau \), then it leads to some suspicious anomaly in the network and hence, \( n_r \) proceed to next step. Step 2: \( n_r \) checks the RT of the \( n_f \) in the packet path and recompute the PDR with respect to the total number of sent packets \( \text{packet}_f \) to identify \( M \). \( n_r \) may also carry out the data trustworthy assessment process after some pre-defined interval \( l \). Other reasons affecting the PDR can be any form of network discrepancy (congestion or link loss). We may consider those factors as a part of our future work.

4. Simulation Results

We implement the proposed idea in Cooja which is a Contiki-based simulator. We use Tmote sky having 48 KB of ROM and 10 KB of RAM as things. Each simulation is performed for 600 seconds excluding the topology convergence time. Other simulation parameters used in the proposed approach are given in Table 1. We evaluate our scheme with respect to the following performance metrics:

4.1 Provenance Generation Time

Provenance generation time can be defined as the amount of time required by each forwarding node to compute and embed provenance, and then forward the data packet to the next neighboring node.

---

3) The value of \( \tau \) can be decided on the basis of data rate depending on application.
4) \( n_f \) has acquired the knowledge of the packet path during topology formation.
Fig. 5 shows the packet processing time (including the provenance generation time) in case of RPL-only and provenance-based RPL. It can be seen that the provenance generation time (min) is almost negligible.

4.2 Provenance Size

Provenance size can be defined as the amount of extra metadata included in the payload as provenance data. Since we are only embedding the PDR in the payload as $P_{\text{data}}$, the provenance size remains constant (2 bytes) irrespective of the number of hops as shown in Fig. 6.

4.3 RAM and ROM Consumption

We compute RAM and ROM consumption for RPL without and with provenance as shown in Table 2. It can be concluded that the extra overhead for provenance-enabled RPL in case of ROM and RAM is 4206 bytes and 4 bytes respectively. It is important to note that the additional required RAM and ROM are under the available capacity of Tmote sky.

4.4 Attack Detection

To carry out selective forwarding attack, we set up a simulation environment consisting of 8-hop network. We select a forwarding node (ID 3) as a malicious node that performs the selective forwarding attack. The malicious node keeps on dropping data packets at random intervals while forwarding control messages. Consequently, as the packet drop rate increases more than the natural packet loss rate (set to 1%), the effect of attack is reflected through the PDR enclosed in the payload.

Similarly, the packet loss rate becomes more obvious as the packet drop increases from 3% to 7% and can be detected as an abnormal behavior in comparison to natural packet loss as shown in Fig. 7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulator</td>
<td>Coja</td>
</tr>
<tr>
<td>Network Layer</td>
<td>RPL</td>
</tr>
<tr>
<td>Simulation time</td>
<td>600 sec</td>
</tr>
<tr>
<td>Packet size (excluding header)</td>
<td>50 bytes</td>
</tr>
<tr>
<td>Data rate</td>
<td>1 packet/10 sec</td>
</tr>
<tr>
<td>Baseline PDR threshold ((\tau))</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Table 1 Network parameters used in the simulation**

<table>
<thead>
<tr>
<th>Comparison</th>
<th>ROM consumption (Bytes)</th>
<th>RAM consumption (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPL(Without provenance)</td>
<td>44162</td>
<td>7246</td>
</tr>
<tr>
<td>RPL(With provenance)</td>
<td>48368</td>
<td>7250</td>
</tr>
</tbody>
</table>

**Table 2 Comparison between RPL with and without provenance**

Fig. 7 Packet loss rate
5. Conclusion

In this paper, we present a provenance-based scheme for detection of malicious nodes performing selective forwarding attack in RPL-connected networks. In order to identify the malicious node responsible for causing disruptions in the network i.e., performing selective forwarding attack, we compute PDR at each forwarding node in the packet route and add it as provenance information in the payload. Furthermore, we add the total number of packets received by the forwarding node in the RT against its respective child node route entry. Based on the received packet count at the forwarding node, we compute PDR to identify the faulty node. If the PDR of the forwarding node is less than natural link loss rate=1%, then we can detect the anomaly in the network. In the future, we will extend the proposed scheme to identify other malicious attacks in the RPL network.

References


Sabah Suhail is a Ph.D. scholar at Kyung Hee University, Korea. She did her MS in Information Security from NUST, Pakistan. Her research interests include secure provenance, security and privacy issues in Internet of things, distributed ledger technology, and cryptography.

Shashi Raj Pandey received the B.E degree in Electrical and Electronics with specialization in Communication from Kathmandu University, Nepal in 2013. After graduation, he served as a Network Engineer at Huawei Technologies Nepal Co. Pvt. Ltd, Nepal from 2013 to 2016. Since March 2016, he is working for his Ph.D in Computer Science and Engineering at Kyung Hee University, South Korea.

Choong Seon Hong received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Minato, Japan, in 1997. He served Korea Telecom Telecommunications Network Laboratory as a Director of Networking Team from 1988 to 1999. Since 1999, he has been Professor, Department of Computer Science and Engineering, Kyung Hee University. His research interests include future Internet, ad hoc networks, network management, and network security.