Chapter 6

Camouflaging Wormhole Attack in OLSR

6.1 Introduction

In the previous chapter, we discussed that the routing protocols, which are in existence for a reasonable duration of time, are secure from simple attacks, because appropriate Intrusion Detection Systems (IDSs) have been developed and deployed. AODV and blackhole were considered as the case studies for routing protocol and attack, respectively. Statistical based IDSs were found good for detecting the simple blackhole attack with acceptable accuracy and detection rate. However, such IDSs have failed when the blackhole was made a bit more intelligent, termed as “active blackhole attack”. Finally, a more powerful IDS was developed for the intelligent attack.

On similar lines, in this chapter, we study the impact of intelligent or modified attacks on the well accepted proactive protocol “OLSR”. Regarding the attack, wormhole is considered as the case study. In this attack, two colluding malicious nodes are placed strategically in the network [43]. The attacker nodes sniff packets at one location in the network and tunnel them to another location. Several IDSs have developed and deployed which can detect the wormhole attack [11, 134, 135]. Broadly speaking, such IDSs work by detecting the delay of packets traversing through the tunnel [136]. If the delay is
higher than a threshold, attack is detected. The basic idea is, as a large number of routes are maliciously established through the wormhole tunnel, it incurs higher delay than the genuine routes. In this work, we first develop a modified version of the wormhole attack called “Camouflaging wormhole attack”, and show that it can easily defy the existing IDSs for OLSR. Following that, we design an IDS for this modified attack and then show that accuracy of the proposed IDS scheme is much higher than the ones reported in the literature. Finally, NS-2 experimental results show the efficacy of the proposed IDS scheme in terms of better throughput, end-to-end delay, lifetime etc.

The chapter is organized as follows. Section 6.2 discusses briefly the proactive routing protocol OLSR. In the same section, we illustrate the wormhole attack on OLSR. Section 6.3 presents the proposed camouflaging wormhole attack. Section 6.4 describes the proposed IDS which can detect camouflaging wormhole attack on OLSR. Section 6.5 discusses performance evaluation of OLSR under traditional wormhole attack and the proposed camouflaging wormhole attack. This section also compares the effect of camouflaging wormhole attack on OLSR, with and without the proposed IDS. The chapter is concluded in Section 6.6.

6.2 Preliminaries and background

6.2.1 Optimized Link-State Routing (OLSR)

The Optimized Link-State Routing (OLSR) protocol is the most popular proactive routing protocol for WSNs [14]. It is a table-driven link-state routing protocol designed for mobile ad-hoc networks. It is well suited to large and dense WSNs. OLSR uses the stability of the link state algorithm and provides the advantage of having routes immediately available when needed. OLSR protocol is an optimization over the original link state protocol for WSNs.
6.2. Preliminaries and background

OLSR minimizes the overhead from flooding of control packets by using only a subset of its neighbor nodes, called MultiPoint Relays (MPRs). A MPR set for a particular node is the subset of its 1-hop neighbors and it covers all 2-hop neighbors of that node. Only the MPRs of a node are able to retransmit its broadcast messages. This protocol is suitable for large and dense networks because the optimization done using MPRs works well in this context. If network is more dense and large then OLSR achieves better optimization than that of the usual link state algorithm.

Each node selects its MPR set from its 1-hop nodes and this makes it easy to forward control packets to its 2-hop nodes. Instead of every node forwarding every message, only MPR nodes relay messages that are intended to be forwarded. The MPR mechanism makes the process of network transmission more scalable and efficient. In a normal broadcast, a node forwards control packets to all its 1-hop neighbors and these 1-hop neighbors again forward the received packets to all their 1-hop neighbors. So, in normal broadcast, a number of duplicate data packets are created. Traffic overhead is the main drawback of this approach. MPR forwarding is used in OLSR as a solution of the above problem. Each node maintains information about the set of neighbors that have selected it as MPR. This set is called the “Multipoint Relay Selector set” (MPR selector set) of a node. Each node stores the information of its 1-hop neighbors in the neighbor table. Neighbor table also contains the status of the links with these neighbors and a list of 2-hop neighbors that these 1-hop neighbors can access. The link status can be uni-directional, bi-directional or MPR.

In MPR approach, HELLO messages and TC messages are two types of routing messages, periodically used by OLSR. HELLO messages are broadcasted after a certain interval to trace neighbors and maintain communication with them. TC messages are broadcasted in the entire network periodically by each MPR node to declare its MPR selector set, i.e., the neighbor nodes which have selected the sender as its MPR. A TC message contains the
MPR selector set of its originator. A node never sends or retransmits any TC message if it has an empty MPR selector set. In this manner OLSR provides communication between source and destination with minimal flooding of control packets.

### 6.2.2 Wormhole attack

Wormhole attack is a severe threat for a WSN [43]. In this attack, two colluding malicious nodes are placed strategically in the network. The adversary nodes record packets at one location in the network and tunnel them to another location. Attacker nodes advertise in the network that their path is the shortest path for transmitting data from one node to another. The tunnel can be established in different ways, like, in-band channel and out-of-band channel [137]. In out-of-band channel, two malicious nodes employ a physical channel between them by either a dedicated wired link or long range wireless link that is not generally available throughout the network. Another type of wormhole attack is known as in-band wormhole attack. In this type of attack the attacker builds a tunnel using a number of nodes over the existing wireless medium.

In-band wormhole is much more popular than out-of-band wormhole attack because in the in-band wormhole attack, adversary does not require additional specialized hardware. In-band wormhole attack consumes network bandwidth continuously. As a result, network losses its capacity very quickly. So, in-band wormhole attack is potentially more harmful than out-of-band wormhole attack and is the most preferred choice for the attacker. This work focuses on in-band wormhole attack in WSNs on the OLSR protocol.

In case of in-band wormhole attack two remote nodes falsely advertise a 1-hop symmetric link between them by forwarding HELLO messages. This creates an illusion that the two colluding attacker nodes are in 1-hop distance. Figure 6.1 shows a scenario of WSN, where OLSR protocol is affected by in-band wormhole attack. In this example, M₁, M₂, M₃ and M₄ are the malicious nodes. When node N₁ broadcast a HELLO message,
attacker node $M_1$ copies this message and sends it to another attacker node $M_2$ and $M_2$ transmits this HELLO packet to $M_3$. In this way the attacker $M_1$ tunnels the HELLO message to $M_4$.

\[ \begin{array}{c}
\text{Sensor Node} \\
\text{Victim Node} \\
\text{Malicious Node}
\end{array} \]

\textbf{Figure 6.1: An example of in-band wormhole attack}

$M_4$ receives $N_1$’s HELLO message and then replays it. This replayed message is received by node $N_{17}$. Node $N_{17}$ thinks that node $N_1$ is 1-hop neighbor of $N_{17}$. When $N_{17}$ sends the HELLO reply packet, it is also tunneled by the attacker node $M_4$. By the same process $N_1$ gets the packet and node $N_1$ also believes that $N_{17}$ is a 1-hop neighbor of node $N_1$. So, a symmetric link can be established between node $N_1$ and $N_{17}$. Through this false route, nodes transmit their data packets. Actually all these data packets are tunneled by the attacker nodes. From Figure 6.1, it is observed that $N_1$ can reach $N_{17}$ by a number of genuine paths; among them $N_1$, $N_5$, $N_9$, $N_{12}$, $N_{17}$ or $N_1$, $N_5$, $N_9$, $N_{16}$, $N_{17}$ is the shortest (4-hop) path. $N_1$ chooses the 1-hop path instead of those 4-hop paths. It may be noted
that by choosing the 1-hop wormhole route the packets pass through the malicious nodes, thereby leading to packet sniffing. Several approaches have been developed to detect and defend against wormhole attacks in WSNs [138, 43, 11, 139, 135, 140, 136, 8]. Among these techniques, the most efficient and widely used schemes apply the following two steps:

- The IDS sends a request packet (HELLO\textsubscript{req}) to its neighbor nodes and waits for a reply message (HELLO\textsubscript{rep}) within a certain time span. If the sender node does not get any reply within that scheduled timeout, then the link between the sender and receiver is detected as a suspicious link.

- However, the slowness may also be due to genuine congestion. So, a second level of check is required. The sender node uses digital signature and sends a probing packet to the receiver node. This probing packet also contains timing information. When the receiver node gets the probing packet it verifies the identification. If the verification is successful the receiver node sends a probing acknowledgement. After getting the acknowledgement the sender node also verifies the identification and the timing information.

### 6.3 Proposed camouflaging wormhole attack for OLSR

Camouflage is the use of different coloration or illumination for concealment. The proposed attack basically works on the following “Camouflage” philosophy– as the link verification technique is based on cryptography, it is difficult to camouflage the time out check by sending the reply early from an attacker node (i.e., before the genuine node’s reply). So our proposed attack technique performs the “camouflage” in the first step i.e., when the detection system checks for delay using unsigned HELLO\textsubscript{req}-HELLO\textsubscript{rep} messages. In other words, in the proposed camouflaging wormhole attack, whenever there is a HELLO\textsubscript{req} packet sent by a source and it is destined to a node that needs to pass though
the attack channel, some attacker node near to the originator sends a $HELLO_{rep}$. As this attacker node is near to the source node, this $HELLO_{rep}$ would arrive before time out and there would be no suspicion generated. So, the attack detection system would not run the verification scheme. Thus, it is clear that the proposed attack creates a camouflaging effect that there is no delay and hence there is no wormhole attack.

**Algorithm 11**: Create an $HELLO_{rep}$ packet for victim node

**Input**: $HELLO_{req}$ packet  
**Output**: $HELLO_{rep}$ packet

1. Receive a $HELLO_{req}$ packet.
2. Assign $Origin_{HELLO_{rep}} := Origin_{HELLO_{req}}, Dst_{HELLO_{rep}} := Dst_{HELLO_{req}}.$
3. CREATE a reply packet $HELLO_{rep}$ ($Origin_{HELLO_{rep}}, Dst_{HELLO_{rep}}$).
4. Unicast $HELLO_{rep}$ to the victim node ($Origin_{HELLO_{req}}$) which sent the $HELLO_{rep}$ packet.
5. Send $HELLO_{req}$ to last attacker node of the tunnel.
6. STOP

The proposed attack scheme is described in three parts. Algorithm 11, Algorithm 12 and Algorithm 13 present the complete procedure of launching the camouflaging wormhole attack. Algorithm 11 is designed for the first attacker node of the wormhole tunnel. Similarly, Algorithm 12 and Algorithm 13 work for the last attacker node of the tunnel. The first attacker node receives a $HELLO_{req}$ packet and sends a $HELLO_{rep}$ packet to the originator node of the $HELLO_{req}$ packet ($Origin_{HELLO_{req}}$). Then the first attacker node of the wormhole tunnel transmits the $HELLO_{req}$ packet to the last attacker node of the wormhole tunnel.

Algorithm 12 describes the second part of the scheme. The last attacker node of the wormhole tunnel receives the $HELLO_{req}$ packet, sent by the first attacker node through the tunnel. Then the last attacker node transmits that $HELLO_{req}$ packet to the victim
destination node. It may be noted that the last attacker node and the victim destination node are in 1-hop.

**Algorithm 12:** Transmit $HELLO_{req}$ packet to the destination victim node

**Input:** $HELLO_{req}$ packet

**Output:** $HELLO_{req}$ packet

1. Receive a $HELLO_{req}$ packet.
2. Transmit $HELLO_{req}$ packet to the destination victim node.
3. HALT

Finally, Algorithm 13 completes the procedure of launching camouflaging wormhole attack. When the last attacker node of the wormhole tunnel receives an $HELLO_{rep}$ packet from the victim destination node (or any other nodes), it drops the $HELLO_{rep}$ packet.

**Algorithm 13:** Drop original $HELLO_{rep}$ packet coming from victim node

**Input:** $HELLO_{rep}$ packet

**Output:** Drop original $HELLO_{rep}$ packet

1. Receive a $HELLO_{rep}$ packet.
2. DROP $HELLO_{rep}$ packet.
3. STOP

Now we describe the proposed camouflaging wormhole attack with the help of the example in Figure 6.1. As shown in the figure, $M_1$ ($M_4$) is the first (last) attacker node of the tunnel. Let $N_1$ broadcast a $HELLO_{req}$ packet for node $N_{17}$ to check the delay. Node $N_1$ waits for $HELLO_{rep}$ packet to arrive within a certain interval of time. But originally $N_{17}$ is not within the 1-hop neighbourhood of $N_1$. So $HELLO_{rep}$ will take time to reach $N_1$ which can create a doubt and lead to execution of the cryptography based verification module. So in the camouflaging wormhole attack the malicious node tries to avoid this situation. Malicious node $M_1$ receives the $HELLO_{req}$ packet with $RREQ_{ID}$ as 11 and realizes that it
is a probe to determine if there is any delay in response from the node $N_{17}$. $M_1$ sends a $HELLO_{rep}$ packet to $N_1$ so that it reaches $N_1$ within the required time interval. It may be noted that as this $HELLO_{rep}$ is not coming from $N_{17}$, rather it is generated by $M_1$ artificially, so it can timed in a way that it reaches $N_1$ within the required time interval. $M_1$ tunnels the request packet ($RREQ$ ID as 11) to the attacker node $M_4$, which then sends the request packet to the destination victim node $N_{17}$. The last attacker node of the wormhole tunnel $M_4$ drops all the $HELLO_{rep}$ packets coming from the node $N_{17}$ (and all other nodes).

### 6.4 Proposed IDS to detect camouflaging wormhole attack

In this section, an IDS is proposed to detect the camouflaging wormhole attack, discussed in the last section. In the proposed IDS, the entire region is divided in clusters and for each cluster there is a Sensor Monitor (SM). All the nodes in a cluster falls within the range of the SM. SM keeps track of all the HELLO messages (including $HELLO_{req}$ and $HELLO_{rep}$) and TC messages received and transmitted from the nodes in that cluster. The monitor checks for successful $HELLO_{req}$ and $HELLO_{rep}$ sequence corresponding to a pair of nodes ($X$ and $Y$, say) that is completed within the given required time interval. In such a case, the SM ($SM_1$ say) needs to verify whether the $HELLO_{rep}$ arrives from the genuine node ($Y$) or is generated by a malicious node of the wormhole tunnel. To do this, the SM ($SM_1$) sends query to all the other SMs asking if they have the pair $X$ and $Y$ as their one hop neighbour. If there is another SM ($SM_2$ say) with a positive reply then it is checked if $SM_1$ and $SM_2$ are in one hop. If the two SMs are not in one hop then definitely there is wormhole attack between $X$ and $Y$.

The basic idea of the working of the attack detection scheme is described in two parts. In first part of the scheme, $HELLO_{req}$ table ($Tab.HELLO_{req}$) is created. This is described by Algorithm 14. When a SM receives a $HELLO_{req}$ packet, it checks if the table, $Tab.HELLO_{req}$ has overflowed or not. If $Tab.HELLO_{req}$ is full then the monitor deletes the
oldest record and stores unique identifier of the HELLO message (HELLO\textit{req}\_ID), originator IP address of the HELLO message (Origin\_HELLO\textit{req}), destination IP address of the HELLO message (Dst\_HELLO\textit{req}), receive time of the HELLO packet (Rcv\_Time\_HELLO\textit{req}) in the table, Tab\_HELLO\textit{req}. As already mentioned, HELLO\textit{req} and HELLO\textit{rep} are special type of control packets used in OLSR.

![Diagram](image)

**Figure 6.2:** Example scenario of camouflaging wormhole attack with monitor

After receiving a HELLO\textit{req} message, the neighbors must respond with a HELLO\textit{rep} message. The second part of the scheme is accomplished by Algorithm 15. HELLO\textit{rep} packet and Tab\_HELLO\textit{req} are considered as inputs by Algorithm 15. When the SM receives HELLO\textit{rep} packet, it finds the corresponding HELLO\textit{req} packet from the table Tab\_HELLO\textit{req} by matching the source IP (Origin\_HELLO\textit{rep}) and the destination IP (Dst\_HELLO\textit{rep}) of HELLO\textit{rep} packet with the source IP (Origin\_HELLO\textit{req}) and the destination IP (Dst\_HELLO\textit{req}) of Tab\_HELLO\textit{req}. Then SM compares the values of the receiving time of HELLO\textit{req} packet
6.4. Proposed IDS to detect camouflaging wormhole attack

(say, \(t_1\)) and HELLO\_rep packet (say, \(t_2\)). Then the difference between the values of \(t_2\) and \(t_1\) is assigned to the variable \(t\). SM sets a threshold \(\tau\) for the scheduled time out interval.

**Algorithm 14:** Create \(\text{Tab}_\text{HELLO}_\text{req}\) for sensor monitor

**Input:** HELLO\_req packet  
**Output:** \(\text{Tab}_\text{HELLO}_\text{req}\)

1. Receive a HELLO\_req packet  
2. IF \(\text{Tab}_\text{HELLO}_\text{req}\) is full  
   DELETE oldest record.  
   ELSE
   ADD HELLO\_req\_ID, Origin\_HELLO\_req, Dst\_HELLO\_req and Rcv\_Time\_HELLO\_req in \(\text{Tab}_\text{HELLO}_\text{req}\).  
3. HALT

If the value of \(t\) is less or equal to \(\tau\), it implies that the HELLO\_rep has come within the time limit. Now, it has to verify whether there exits any wormhole tunnel between the nodes \(\text{Origin}_\text{HELLO}_\text{req}\) and \(\text{Dst}_\text{HELLO}_\text{req}\). This SM (say, \(\text{SM}_1\)) checks with the other SMs, whether there exist other \((X, Y)\) pairs where, \(X = \text{Origin}_\text{HELLO}_\text{req}\) and \(Y = \text{Dst}_\text{HELLO}_\text{req}\). If it is found that another SM (say, \(\text{SM}_2\)) contains same pair of nodes within its range, then attack is suspected. The wormhole tunnel between the nodes \(\text{Origin}_\text{HELLO}_\text{req}\) and \(\text{Dst}_\text{HELLO}_\text{req}\) is confirmed when \(\text{SM}_1\) and \(\text{SM}_2\) are not in 1-hop range with each other.

To explain the detection mechanism, we use an example of a simple WSN scenario, which is illustrated in Figure 6.2. In Figure 6.2, \(\text{SM}_1\), \(\text{SM}_2\), \(\text{SM}_3\), \(\text{SM}_4\), \(\text{SM}_5\) are the SMs and one SM covers the sensor nodes within its range. \(\text{N}_2\), \(\text{N}_3\), \(\text{N}_4\), \(\text{N}_5\), \(\text{N}_6\) and \(\text{M}_1\) are the 1-hop neighbors of node \(\text{N}_1\) and \(\text{N}_{12}\), \(\text{N}_{16}\), \(\text{N}_{15}\), \(\text{N}_{14}\), \(\text{N}_{13}\) and \(\text{M}_4\) are the 1-hop neighbors of node \(\text{N}_{17}\). Here \(\text{M}_1\) and \(\text{M}_4\) are the two malicious nodes. \(\text{M}_1\) creates a private tunnel to \(\text{M}_4\) with the help of another two attacker nodes \(\text{M}_2\) and \(\text{M}_3\). But the attacker node \(\text{M}_1\) declares itself as \(\text{N}_{17}\) and \(\text{M}_4\) advertises itself as \(\text{N}_1\). When a HELLO\_req packet is
transmitted from the originator node $N_1$ to $N_{17}$, the sensor monitor $SM_5$ stores the values of $HELLO_{req\_ID}$, $Origin_{HELLO_{req}}$, $Dst_{HELLO_{req}}$, $Rcv\_Time_{HELLO_{req}}$, i.e., 11, $N_1$, $N_{17}$, 1.15 mins, respectively in $HELLO_{req}$ table of the monitor $SM_5$ (i.e., Table 6.1).

**Algorithm 15:** Camouflaging wormhole attack detection mechanism

**Input:** $HELLO_{rep}$ packet, $Tab_{HELLO_{req}}$

**Output:** Detection of camouflaging wormhole attack

1. Sensor Monitor (say, $SM_1$) receives $HELLO_{rep}$ message.

2. IF $Origin_{HELLO_{req}} = Origin_{HELLO_{rep}}$ and $Dst_{HELLO_{req}} = Dst_{HELLO_{rep}}$

   Calculate, $t = t_2 - t_1$

   IF $t \leq \tau$

   Check the other SMs whether there exist other $(X, Y)$ pairs where $X = Src_{HELLO_{req}}$ and $Y = Dst_{HELLO_{req}}$

   IF found (say, $SM_2$), and IF $SM_1$ and $SM_2$ are not 1-hop neighbours

   Wormhole attack is detected between $(X, Y)$ pair.

3. STOP

The monitor $SM_5$ sets a fixed threshold for time out. Suppose in this example the value of this threshold $\tau$ is set as 5 sec. The node $N_1$ receives the $HELLO_{rep}$ packet and its receiving time is 1.2 mins. Now, $t_2 = 1.2$ mins = 72 seconds, $t_1 = 1.15$ mins = 69 seconds. Therefore, $t = t_2 - t_1 = (72 - 69)$ seconds = 3 seconds. Since, $\tau = 5$ seconds and $t = 3$ seconds, $t < \tau$. $SM_5$ observes that $HELLO_{rep}$ packet comes to node $N_1$ within the timeout.

**Table 6.1:** $Tab_{HELLO_{req}}$: $HELLO_{req}$ table for sensor monitor

<table>
<thead>
<tr>
<th>No.</th>
<th>$HELLO_{req_ID}$</th>
<th>$Origin_{HELLO_{req}}$</th>
<th>$Dst_{HELLO_{req}}$</th>
<th>$Rcv_Time_{HELLO_{req}}$ (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>$N_1$</td>
<td>$N_{17}$</td>
<td>1.15</td>
</tr>
<tr>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Now, the sensor monitor $SM_5$ tries to find out whether the link between the nodes $N_1$ and $N_{17}$ is authorized or not. Thus, the monitor $SM_5$ queries other monitors, if they
have the pair \( N_1 \) and \( N_{17} \). SM_3 replies that nodes \( N_1 \) and \( N_{17} \) are its 1-hop neighbors. SM_5 suspect that there exists an wormhole attack between nodes \( N_1 \) and \( N_{17} \). When it is confirmed that SM_5 and SM_3 are not 1-hop neighbors, SM_5 notifies about the wormhole tunnel between nodes \( N_1 \) and \( N_{17} \).

### 6.5 Performance evaluation

Performance of protocols, IDSs, etc. in WSNs are generally measured in terms of the following parameters.

- **Packet Delivery Ratio (PDR):** PDR is the ratio of the number of packets delivered to the number of packets generated by the source.

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simulator</td>
<td>NS-2(Version 2.34)</td>
</tr>
<tr>
<td>2</td>
<td>Operating System</td>
<td>Linux(Redhat 5)</td>
</tr>
<tr>
<td>3</td>
<td>Channel Type</td>
<td>Channel/Wireless Channel</td>
</tr>
<tr>
<td>4</td>
<td>Traffic Model</td>
<td>Constant Bit Rate (CBR)</td>
</tr>
<tr>
<td>5</td>
<td>Source type</td>
<td>UDP</td>
</tr>
<tr>
<td>6</td>
<td>Area (m* m)</td>
<td>1000 * 1000 (initially)</td>
</tr>
<tr>
<td>7</td>
<td>Number of sensor node</td>
<td>5-100</td>
</tr>
<tr>
<td>8</td>
<td>Simulation Time</td>
<td>500 s</td>
</tr>
<tr>
<td>9</td>
<td>Routing Protocol</td>
<td>AODV</td>
</tr>
<tr>
<td>10</td>
<td>Antenna</td>
<td>Omni-directional</td>
</tr>
</tbody>
</table>

- **End-to-End Delay:** End-to-End delay is defined as the average transit time of a packet, i.e., the time taken for a packet to reach the destination from the source.

- **Throughput:** Throughput is computed as the amount of data transferred divided by the data transfer time.
It may be noted that, in any wormhole attack the adversary nodes form a tunnel and just sniffs the traffic. So there is no significant impact on any of these three performance parameters.

![Figure 6.3: Sensor monitors SM1 and SM2 within 1-hop and observe nodes X and Y: normal situation](image_url)

However, wormhole attack is severe, as it can cause all packets to be delivered through malicious nodes, resulting in compromise of confidentiality and integrity.

![Figure 6.4: Sensor monitors SM1 and SM2 not within 1-hop but observe nodes X and Y: wormhole attack](image_url)

Thus, accuracy and detection rate of the IDS are the most important parameters in evaluating the quality of the IDS. The accuracy obtained is always 100%, explained as follows. If the IDS determines a situation to be wormhole attack, it is really so. As
discussed in Section 6.4, the IDS runs in all SMs and observes successful $HELLO_{req} - HELLO_{rep}$ packet sequence(s) with a pair of nodes (X and Y, say) that are present in the corresponding cluster.

![Diagram showing sensor monitors SM1 and SM2 within 1-hop and observing nodes X and Y: wormhole attack](image)

**Figure 6.5:** Sensor monitors SM1 and SM2 within 1-hop and observe nodes X and Y: wormhole attack

Further, the time duration between the $HELLO_{req} - HELLO_{rep}$ sequence must be within the given the required time interval. In such a case, the SM verifies whether the $HELLO_{rep}$ arrives from the genuine node (Y) or is generated by a malicious node of the wormhole.
tunnel. To do this, the SM sends query to all the other SMs asking if they have the pair X and Y as their one hop neighbour. If there is another SM with a positive reply, then it is checked if both the SMs are in one hop. If the two SMs are not in one hop, then definitely there is wormhole attack between X and Y, because, as the SMs that are not in one hop, they can never see the same pair of sensor nodes (say, X and Y).

![Comparison of Packet Delivery Ratio](image.png)

**Figure 6.7:** Comparison of PDR under normal condition of OLSR and wormhole attack

This concept is explained using Figure 6.3 and Figure 6.4. In Figure 6.3, it may be observed that sensor nodes X and Y are within the clusters of both SM\(_1\) and SM\(_2\), hence both these SMs observe X and Y. So even if SM\(_1\) and SM\(_2\) receive the HELLO\(_{req}\) – HELLO\(_{rep}\) packet sequence from X and Y, as SM\(_1\) and SM\(_2\) are within 1-hop, IDS does not raise an alarm. On the other hand, in Figure 6.4, SM\(_1\) and SM\(_2\) are not in 1-hop and hence SM\(_1\) observes packets from X and SM\(_2\) observes packets from Y. However, camouflage wormhole attacker node M\(_1\) mimics as Y for SM\(_1\) and attacker node M\(_4\) mimics as X for SM\(_2\). So SM\(_1\) and SM\(_2\) both observe the nodes X and Y within their cluster, but SM\(_1\) and SM\(_2\) are not in 1-hop which is surely an attack as indicated by the IDS.
The detection rate is observed to be around 99% on the average. The major reason for slight drop in detection rate is explained using Figure 6.5. In Figure 6.5, it may be observed that sensor node X in within the range of SM$_1$ and Y is within the cluster of SM$_2$. In this situation, X and Y are not in 1-hop, but SM$_1$ and SM$_2$ are in 1-hop. So, SM$_1$ observes packets from X and SM$_2$ observes packets from Y. However, camouflage wormhole attacker node $M_1$ mimics as Y for SM$_1$ and attacker node $M_3$ mimics as X for SM$_2$. So SM$_1$ and SM$_2$ both observe the nodes X and Y within their cluster, which is an anomalous scenario. But as SM$_1$ and SM$_2$ are in 1-hop, IDS does not raise an alarm, resulting in drop in detection rate. It may be noted that, such situations (based on location of sensor nodes) are not very frequent, because nodes X and Y must not be in 1-hop but the corresponding SMs (SM$_1$ and SM$_2$) must be in 1-hop.

![Comparison of Packet Delivery Ratio](image)

**Figure 6.8:** Comparison of PDR under normal condition of OLSR and proposed camouflaging wormhole attack

Also, wormhole attack is generally launched between two nodes that are reasonably apart. Figure 6.6 illustrates the detection rate versus the number of nodes (parameters shown in Table 6.2). Nevertheless, as wormhole attack causes many connections to maliciously flow through the tunnel, there may be congestion leading to slight compromise
in the quality parameters mentioned above.

![Comparison of Packet Delivery Ratio](image)

**Figure 6.9:** Comparison of PDR under camouflaging wormhole attack with and without traditional IDS

![Comparison of Packet Delivery Ratio](image)

**Figure 6.10:** Comparison of PDR under simple wormhole attack with and without traditional IDS

So we will also present NS-2 simulation results pertaining to PDR, end-to-end delay and throughput because of the wormhole attack and IDS. The parameters used in NS-2 simulations are shown in Table 6.2. As shown in the table, for simulation we consider
an area of 1000m *1000 m and 5-100 nodes randomly deployed in this area. The initial battery power of each node is set to 20 Joules.

![Comparison of Packet Delivery Ratio](image)

**Figure 6.11:** Comparison of PDR under camouflaging wormhole attack with and without the proposed IDS

![Comparison of End to End Delay](image)

**Figure 6.12:** Comparison of delay under normal condition of OLSR and wormhole attack

First, we report our experimental results for PDR. Figure 6.7 illustrates the PDR of OLSR versus the number of nodes under normal condition and traditional wormhole...
attack.

![Comparison of End to End Delay](image1)

**Figure 6.13:** Comparison of delay under normal condition of OLSR and proposed camouflaging wormhole attack

![Comparison of End to End Delay](image2)

**Figure 6.14:** Comparison of delay under camouflaging wormhole attack with and without traditional IDS

PDR is *slightly* compromised by wormhole attack, because many connections pass through the tunnels resulting is congestion and packet loss.
Also, the drop in PDR increases slightly with the increase in number of nodes because congestion rises with number of nodes. It may be noted that, unlike other cases e.g., power aware attacks, blackhole based attacks, sleep deprivation based attacks (discussed in previous chapters), the reduction in PDR for wormhole attack is quite insignificant.
Figure 6.17: Comparison of throughput under normal condition of OLSR and wormhole attack

Figure 6.18: Comparison of throughput under normal condition of OLSR and proposed camouflaging wormhole attack

Figure 6.9 illustrates the PDR of OLSR under camouflaging wormhole attack, with and without the traditional IDS [134] in execution. Results show that, when the traditional IDS is executed, drop in PDR due to the camouflaging wormhole attack cannot be handled.
6.5. Performance evaluation

The major reason is that, unlike most of the attacks in WSNs, which drop packets causing denial of service, in wormhole attack the packets are just snifed, which cause loss of integrity and confidentiality but not packet loss. The only impact of wormhole attack on PDR is due to the fact that, many connections are made to pass through the wormhole attack.
tunnel, resulting in a situation where much more connections pass through only a few links compared to a normal situation. So there is congestion and packet loss resulting in slightly lower PDR.

![Comparison of Aggregate Throughput](image)

**Figure 6.21:** Comparison of throughput under camouflaging wormhole attack with and without the proposed IDS

Figure 6.8 illustrates the PDR of OLSR versus the number of nodes under normal condition and proposed camouflaging wormhole attack. It may be noted that, drop in PDR in case of the proposed camouflaging wormhole attack is slightly more compared to the traditional wormhole attack. The reason is that, both these forms of wormhole attack impact the PDR in a similar nature however congestion in camouflaging wormhole attack is slightly higher. This is because of the requirement of extra control packets and processing (tables and attack launching algorithms) involved with the attacker nodes in the camouflaging wormhole attack. Following that, we study the performance of the proposed IDS based on PDR. As already discussed, traditional IDS cannot detect the camouflaging wormhole attack. To elaborate, the detection rate of camouflaging wormhole attack using the traditional IDS [139] is almost 0%. Such IDSs detect the attack on the principle–”delay of the packet sequence for checking source-destination pairs i.e.,
HELLO$_{req}$ − HELLO$_{rep}$ is higher than a threshold” and the camouflaging wormhole attack suppresses this delay using malicious packets.

However, it may be noted that simple wormhole attack on OLSR can be easily detected by tradition IDSs [139]; Figure 6.10 illustrates this fact. Figure 6.11 illustrates the PDR of OLSR under camouflaging wormhole attack, with and without the proposed IDS in execution. Results show that, when the proposed IDS is executed, the drop in PDR due to camouflaging wormhole attack is nullified.

Similar trends were found for the other two parameters i.e., end-to-end delay and throughput. Figures 6.12 through 6.21 illustrate this fact.

**6.6 Conclusion**

In this chapter we demonstrated that design and deployment of appropriate IDSs for traditional routing protocols make them secure to simple existing attacks. OLSR was considered as the routing protocol, wormhole was taken as the attack and the IDS proposed in [139] were used in the study. The IDS [139] is based on measuring delay of routes in between a source-destination pair using special packets. If the delay is higher, then wormhole attack is ascertained using cryptographic techniques. Following that an intelligent version of the basic wormhole attack called camouflaging wormhole was proposed. It was then demonstrated analytically and using NS-2 simulations that the IDS [139] could not secure OLSR from the camouflaging wormhole attack. To summarise, enhanced attacks cannot be handled by existing IDSs for proactive routing protocols like OLSR and for such cases new IDSs are required.

So, by the study made in this chapter and Chapter 5, it may be concluded that, if a protocol is in existence for a certain duration, then appropriate IDSs are available of the shelf, which can withstand simple traditional attacks. However, enhanced or intelligent versions of existing attacks cannot be handled by traditional IDSs, in case of both reactive
and proactive routing protocols. So, new IDSs need to designed for such attacks. Further, in Chapter 3 and Chapter 4, it was concluded that, enhancements to both reactive and proactive routing protocols improve quality, however, they become vulnerable to even simple attacks. In the next chapter, we will consider the extreme case i.e., impact of an intelligent attack on an enhanced protocol.