A Cross Layer Framework to Mitigate a Joint MAC and Routing Attack in Multihop Wireless Networks

Soufiene Djahel[∓], Farid Naït-Abdesselam[∓] and Ashfaq Khokhar[±]

[∓]LIFL – UMR CNRS USTL 8022 – IRCICA
University of Lille , France
{soufiene.djahel, farid.nait-abdesselam }@lifl.fr

[±]Electrical and Computer Engineering Department
University of Illinois at Chicago
ashfaq@ece.uic.edu

Abstract— **It is well known that security threats, in wireless ad hoc networks, are becoming a serious problem which may lead to harmful consequences on network performance. Despite that, many routing protocols still not resilient to such threats or their countermeasures are not efficient. Moreover, the vulnerability of MAC layer protocols to some attacks exacerbates the damage caused by the threats at higher layers. Therefore, cooperation between layers is compulsory to face such devastating threats. In this paper, we address a cross-layer attack targeting proactive routing protocols, which is launched at the routing level and reinforced at the MAC layer in order to amplify the resulted damage. We demonstrate that this attack can severely compromise the routing protocols and lead to large data packets loss. We particularly analyze it under the Optimized Link State Routing (OLSR) protocol in detail and propose a lightweight solution to cope with it. The simulation results confirm the efficiency of this solution.**

Keywords – Cross-layer Attack, Virtual-link Attack, False Validation Attack, Ad Hoc Networks, OLSR.

I. Introduction

The increase in computation power, the compactness of size, incorporation of mobility and ease of connectivity from anywhere are amongst the major factors that resulted in tremendous growth of handheld devices in recent years. From cordless phones to cellular networks and from WiFi to sensors, the wireless medium has become the preferred backbone of today's deployed networks. The newest model being introduced is the Mobile Ad hoc Networks (MANET), in which mobile nodes, within the transmission range of each others, can communicate directly over the wireless link, while those that are far apart use other nodes as relays. The properties of MANET, such as shared wireless medium, open network architecture, stringent resource constraints and rapidly changing topology make this type of network vulnerable to a bunch of attacks at different layers, especially at MAC layer in which attacks are launched easily. Therefore, the task of securing such network remains hard and necessitates careful investigation.

To address the routing problems with the intrinsic features of MANET, numerous protocols have been devised and further standardized by the IETF (Internet Engineering Task Force), among which the Optimized Link State Routing (OLSR) protocol [3]. The major drawback of OLSR, as all other ad hoc protocols, is that it has not been designed and thought with respect to security issues. Hence it is exposed to many types of misuse leading to a dramatic drop of the network performance and services. Additionally, any node can misbehave and try to disrupt the routing process by injecting tampered or even fake information in the network. Notice that the lack of security considerations in the design of these routing protocols have penalized, especially, the neighbor discovery process since it becomes easy to spoof any identity/link of/with nodes and disseminate false topology information within the whole network.

In this work, we address one of the attacks targeting neighbor discovery phase in OLSR. This attack is launched at routing level by implementing a virtual link attack (VLINK) leading to establishment of false symmetric link between the target nodes connected via an asymmetric link. So, an incorrect MPR (Multi-Point Relay) set may be elected by the target nodes as well as their neighbors leading to selection of broken routes to forward data packets. Subsequently, a false validation attack is initiated at MAC level by another colluding node in order to reinforce the VLINK attack and make it more destructive. To counter this attack, we propose a cross-layer solution in which the routing layer needs to get a confirmation from MAC layer regarding the status of a specific link before advertising it to the network. In order to check the symmetry of a link, the RTS (Request to Send) and CTS (Clear to Send) frames format is modified to prevent the malicious node acting at MAC layer from falsely validating the well reception of the RTS and DATA frames being transmitted by one of the target nodes.

The rest of the paper is organized as follows. The next section gives a brief description of some works dealing with security threats in OLSR. Next, we address the joint virtual link and false validation attack on OLSR in section III. In section IV, we present our proposed solution and analyze its robustness against the possible security threats. In section V, we report our simulation and discuss the obtained results. Finally, section VI concludes the paper.

II. Related work

In recent years, many schemes have been proposed to secure the routing protocols against different attacks launched by malicious or compromised nodes. In the sequel, we briefly describe some of these solutions and emphasize their strengths and limitations.

In [2], packet leashes have been used to protect routing protocols against wormhole attacks. In this scheme, a leash is defined as any information attached to a packet in order to limit its maximum transmission distance. Two types of leashes have been proposed: *geographical leashes* and *temporal leashes*. In the geographical leash, the sender appends its location and sending time to the packet. Based on this information, the receiving node computes an upper bound on the distance separating it from the sender. One of the disadvantages of this solution is that it requires a coarse synchronization of all nodes in the network which is not always feasible in MANET. In the temporal leash, the sender attaches the sending time into the packet, which allows the receiver to computes the traveling distance of that packet. This distance is calculated based on the assumption that propagation of wireless signal is equal to the speed of light. This latter also suffers from the same disadvantage of the former one as clocks synchronization is pre-requirements for its efficiency.

SOLSR (Secure OLSR) [6] aims to prevent any threats targeting the integrity of the exchanged control traffic in the network. To this end, it proposes to piggyback a packet's signature to the transmitted packet, while using hash chains to secure the Time To Live and Hop Count mutable fields. Furthermore, SOLSR provides also a countermeasure to cope with the Wormhole attack which targets the neighbor discovery phase. This solution can be summarized as follows: the node sends probe packets to measure their travel time, from which it can infer the travel distance and then compares it with its transmission range, if the traveled distance is greater than the transmission range then this message may have been tunneled through the wormhole. This solution is efficient however it is still exposed to some threats such as the Byzantine behavior of legitimate nodes and misuses at MAC layer which may significantly affect its robustness.

A new attack targeting OLSR was introduced in [7], in which two malicious nodes N_1 and N_2 collude each other to disturb the protocol's functioning. In this attack, the first attacker N_1 inserts in its Hello message all nodes 2-hops neighbors (link spoofing attack) of the victim node to force its election as the only MPR of this node. Afterwards, the second attacker N_2, which is chosen by N_1 as its only MPR node, drops all routing packets passing through it. Consequently, the victim node is isolated from the network. To defend against this attack, the authors propose to extend the list of 1-hop neighbors nodes advertised in Hello message to include also the list of 2-hops neighbors of the sender. Based on this information, a node can detect whether one of its neighbors advertises a forged links in its hello message or not. The detection is carried out as follows; if node N_1 is 1-hop neighbor of node X then N_1 should be advertised as 2-hops neighbor in all X's 1-hop neighbor Hello message. This rule cannot be satisfied if N_1 misbehaves and consequently it can be detected easily. This scheme may fail to take a correct decision when the nodes' mobility increases. This due to the fact that false alarms may be triggered frequently when links between nodes break.

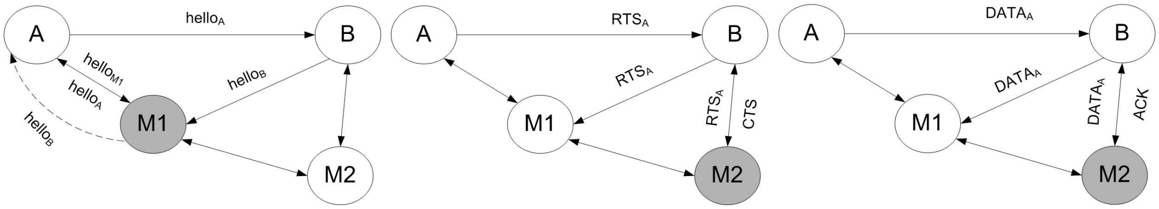
The solution proposed in [11] is based on three hops acknowledgment to cope with the cooperative black hole attack in OLSR. It adds two extra packets to OLSR, Hello-rep packet which is a slight modification to Hello message and a small acknowledgment packet. Each MPR node M acquires the list of its 3-hops neighbors reached through a distinct pair of consecutive MPR nodes (M1, M2), where M2 is the MPR node of M1. Afterwards, the node M selects one node from this set to which it requests an authenticated acknowledgment. This acknowledgment aims to confirm the reception of the TC message generated/forwarded by M. Notice that the authentication process is carried out using a pre-established secret key between node M and the requested node during network initialization. If the number of missed acknowledgements overtakes a predefined threshold the MPR nodes on the suspicious path are considered as malicious and consequently will never be selected as MPR. Moreover, no further packet will be forwarded for these detected nodes.

Another attack against OLSR, called Node Isolation attack, is described in [8]. In this attack, an MPR node denies to generate its TC (Topology Control) message to prevent its MPR selector's nodes from communicating with other nodes in the network. Notice that the attacker node is selected as the only MPR node by the victims by using the same technique described in [7]. To defend against this attack, the authors propose a solution that consists of two phases: detection phase and avoidance phase. In the detection phase each node uses the promiscuous mode to verify whether its MPR node generates correctly its TC message or not. In the second phase, a slight modification to Hello message format is carried by adding a new field named Request-value. Further on, any MPR node receiving a Hello message in which the field Request-value is set to 1 it has to advertise the sender identity as an MPR selector in its TC message. Therefore, even if the attacker doesn't generate the TC message the victim nodes ensure the communication with the rest of the network.

III. Joint Virtual link attack and false validation attack

Many devices with different computation and communication capabilities establish temporary links to form an ad hoc network. As opposed to homogenous environment, where symmetric links are the more general observed fact, routing in a heterogeneous MANET is dominated by many asymmetric links. There are several reasons for the appearance of such links, some of which are stated as follows:

- Due to the varying transmission ranges the devices with stronger communication capabilities may reach the weaker ones but the opposite is not possible.
- In order to achieve power-aware communication, the wireless devices adjust their transmitting power according to their residual power such that their lifetimes are extended. In such communication circumstances, some of the symmetric links may become asymmetric when



(a) Step 1: attack carried at routing level by node M1 (b) Step 2: attack carried at MAC level by node M2 (c) Step 3: attack carried at MAC level by node M2

Figure 1: The attack description

the communication capability of a node degrades due to decrease in the residual power.
- The transmission range of some devices having the same communication capabilities may vary due to fading [10] and random transient phenomenon.

Malicious nodes may get benefits from asymmetric links to launch attacks as depicted in Figure III. In this figure, we consider nodes A, B and M1 running OLSR and having different transmission capabilities. Node A is beyond the transmission range of node B, and similarly B is unable to receive messages sent by node M1. During the neighbor discovery phase, the malicious node M1 relays B's Hello message towards A in order to establish a fake symmetric link (virtual link (VLINK) as dubbed in [9]). At the end of this phase, both nodes A and B believe that they share a symmetric link between them. Therefore a serious degradation of OLSR performance can be resulted as shown later in section V.
To summarize, we present the following sequence of Hello messages being exchanged to set up the fake symmetric link.

- **Scenario 1**
  1) $A \longrightarrow * : Hello, \{\emptyset\}$.
  2) $B \longrightarrow * : Hello, \{A, ASYM\}$.
  3) $M1 \longrightarrow A : Hello, \{A, ASYM\}$.
  4) $A \longrightarrow * : Hello, \{B, SYM\}$.
  5) $B \longrightarrow * : Hello, \{A, SYM\}$.

- **Scenario2**
  1) $B \longrightarrow * : Hello, \{\emptyset\}$.
  2) $M1 \longrightarrow A : Hello, \{\emptyset\}$.
  3) $A \longrightarrow * : Hello, \{B, ASYM\}$.
  4) $B \longrightarrow * : Hello, \{A, SYM\}$.
  5) $M1 \longrightarrow A : Hello, \{A, SYM\}$.
  6) $A \longrightarrow * : Hello, \{B, SYM\}$.

where $*$ denotes the dissemination of a message and {Id, link} refers to the content of Hello message where id is the neighbor identity and link is the status of the link connecting the sender of the message and the node id.

Notice that we distinguish two scenarios which lead to establish the fake symmetric link as illustrated above. In the first scenario the malicious node M1 has to relay the B's Hello message only once to launch the attack, however in the second scenario it has to relay this message twice. Therefore it spends more energy in this latter than the former case.

Since the default value of the interval separating two consecutive transmissions of Hello message is set to 2 seconds, then whenever the victim node B, transmitting packets towards the node A, detects a link break at MAC layer it launches a new shortest path search from the routing table. Notice that a link is lost if the number of missed CTS or ACK frames has overtaken a specified threshold. So after finding a new path the node B transmits its data packets successfully to the intended destination until the next Hello message from A is received again, and the same scenario will be repeated.

In order to prevent such situation, the malicious node M2 replies to all RTS and DATA packets sent by node B by sending back the corresponding validation frames CTS and ACK respectively as depicted in Figures 1(b) and 1(c). This misbehavior is called false validation attack. Therefore, the victim node B keeps constantly transmitting its packets through the compromised link and consequently none of them reaches its destination

For both traffic flows TCP and UDP this attack leads to data packets loss since no link break advertisement is sent to the higher layer to replace the broken link. For TCP flows the sender node reduces its sliding window size gradually each time the expected end-to-end acknowledgment is missed until it reaches zero and the flow is interrupted accordingly, however in UDP traffic the sender continues transmitting its packets until its completion and hence it consumes more energy uselessly. For the security point of view, this attack leads the sender node to falsely accuse the intended receiver as misbehaving or decreases its reputation and trust level if any monitoring system is set at routing layer such as watchdog or other schemes that require an explicit authenticated acknowledgment to verify that its next hop forwards the packets correctly. Therefore, false alarms may be triggered in the network and consequently longer paths and network partition may result.

IV. The proposed solution

In this section we give an overview of the detailed functioning of our solution and its assumptions along with the analysis of the possible scenarios which may be conducted by the colluding nodes M1 and M2 in order to break the solution.

In order to cope with the attack described in the previous section, we have developed a cross-layer solution based on cooperation between routing and MAC layers. In this solution, we assume that each pair of nodes shares a secret which is undisclosed to any other node and that each node holds a collision free hash function such as SHA-1 (Secure Hash Algorithm 1) and MD5 (Message Digest 5). Notice that the preliminary distribution of keys or secrets between the nodes in MANET can be carried out using some well known schemes proposed in the literature such as [4] and [5].

Whenever a node receives a Hello message advertising its identity as an ASYM neighbor then it schedules a transmission of RTS+ frame towards the sender of this message. The frame RTS+, as depicted in Figure 2, is a special RTS frame in which we add a new field of 16 bits dubbed RTS sequence number (RSN) and replace the @R (destination address) field with the hash value of the shared secret (SS) between the sender and the intended receiver combined with the RSN value. The purpose of adding the field RSN is to prevent reply attacks. Moreover, we use the value hash (RSN \\ shared secret) as a destination address to prevent any malicious node from replying an old RTS+ frame in order to deplete the node A's energy. Note that the symbol \\ represents the concatenation operation of RSN and the SS.

The value of RSN is increased by 1 at each transmission or retransmission of RTS+ frame as well as upon reception of a CTS+ frame. Note that the duration field in RTS+ is calculated as follows:

$$Duration = T_{CTS+} + SIFS \tag{1}$$

because no DATA frame transmission will follow the reception of CTS+ frame. Note that T_{CTS+} refers to the transmission time of the CTS+ frame.

Each node receiving the RTS+ frame computes the hash value of the RSN value combined with its shared secret with the source node, if it is equal to the value received in RTS+ then the node sends back the corresponding CTS+ frame with duration field set to 0[1] and the @R field sets to the hash value of the shared secret combined with the value (RSN+1). The format of CTS+ is shown in Figure 3 where its size is 10 bytes larger than the standard CTS frame.

If the sender of RTS+ didn't receive the corresponding CTS+ within the timeout period for several times then this is a confirmation that the intended receiver is under attack launched by a third node and consequently no symmetric link with this victim node will be advertised in the next Hello message.

Remark: Since the nodes in MANET are equipped with limited battery power and modest computation capabilities, we have opted for hash function rather than public/symmetric key cryptography as it is characterized by its low cost and fast operations. Notice that the operation speed is a strict requirement since the delay separating the reception time of the last bit of RTS+ and the transmission of the first bit of CTS+ should not overtake the SIFS duration.

SECURITY ANALYSIS

Let us now analyze the possible scenarios by which the malicious nodes M1 and M2 try to compromise the proposed solution.

Despite the fact that the destination address of RTS+ frame is hidden the node M1 may relay all the heard RTS+ frames or a randomly chosen subset of them towards the victim node. In this case, the victim node will certainly receive one RTS+ in which it is the intended receiver, however due to the incurred delay (d), as a consequence of the retransmission of the RTS+ by the node M1, the CTS+ will be received after the expiration of the timeout value TO_{CTS+}. Hence, the link with the destination node is deleted. Moreover, even though the node M1 is equipped with a set of directional antennas it is unable to receive a frame using one antenna and transmits by another antenna simultaneously. Therefore, the incurred delay for forwarding the RTS+ frame remains important. The values of TO_{CTS+} and delay are calculated as shown in the equations below.

$$TO_{CTS+} = T_{RTS+} + SIFS + T_{CTS+} \tag{2}$$

$$d = TO_{CTS+} + T_{CTS+} \tag{3}$$

where TO_{CTS+} is the expected duration for receiving the CTS+ at the sender node, T_{RTS+} and T_{CTS+} are the transmission time of RTS+ and CTS+, respectively, whereas the signal propagation delay is ignored.

For more sophisticated scenario, we suppose that the malicious node M1 records the CTS+ at time t and sends it to node M2 via an encrypted packet. Then, the node M2 replays it later at time t+ Δ_t (because due to nodes mobility the links status change frequently and the nodes have to check the symmetry of every new established link) in order to falsely validate the subsequent RTS+ towards the same destination. The CTS+ frame replied by node M2 will not be considered as a valid CTS+ since the expected hash value would be calculated using a RSN larger than the one used for the old CTS+ kept by M2. Moreover, the node M2 is unable to compute the expected hash value as the shared secret is unknown.

The operation of the proposed solution is summarized in the flowchart given in Figure 4. This flowchart describes the treatment carried out by any node in the network upon reception of a Hello message. As we can see from this flowchart, after sending the RTS+ the node waits for CTS+ reception. If it is received after timeout expiration or not received at all and timeout is expired then it is ignored and the missed CTS+ counter is increased. Otherwise, if it is received before timeout expiration then its validity and authentication should be checked as well in order to prevent any forged or old CTS+ replied by a malicious node.

[1] The duration field is set to 0 because no further DATA packet will be exchanged.

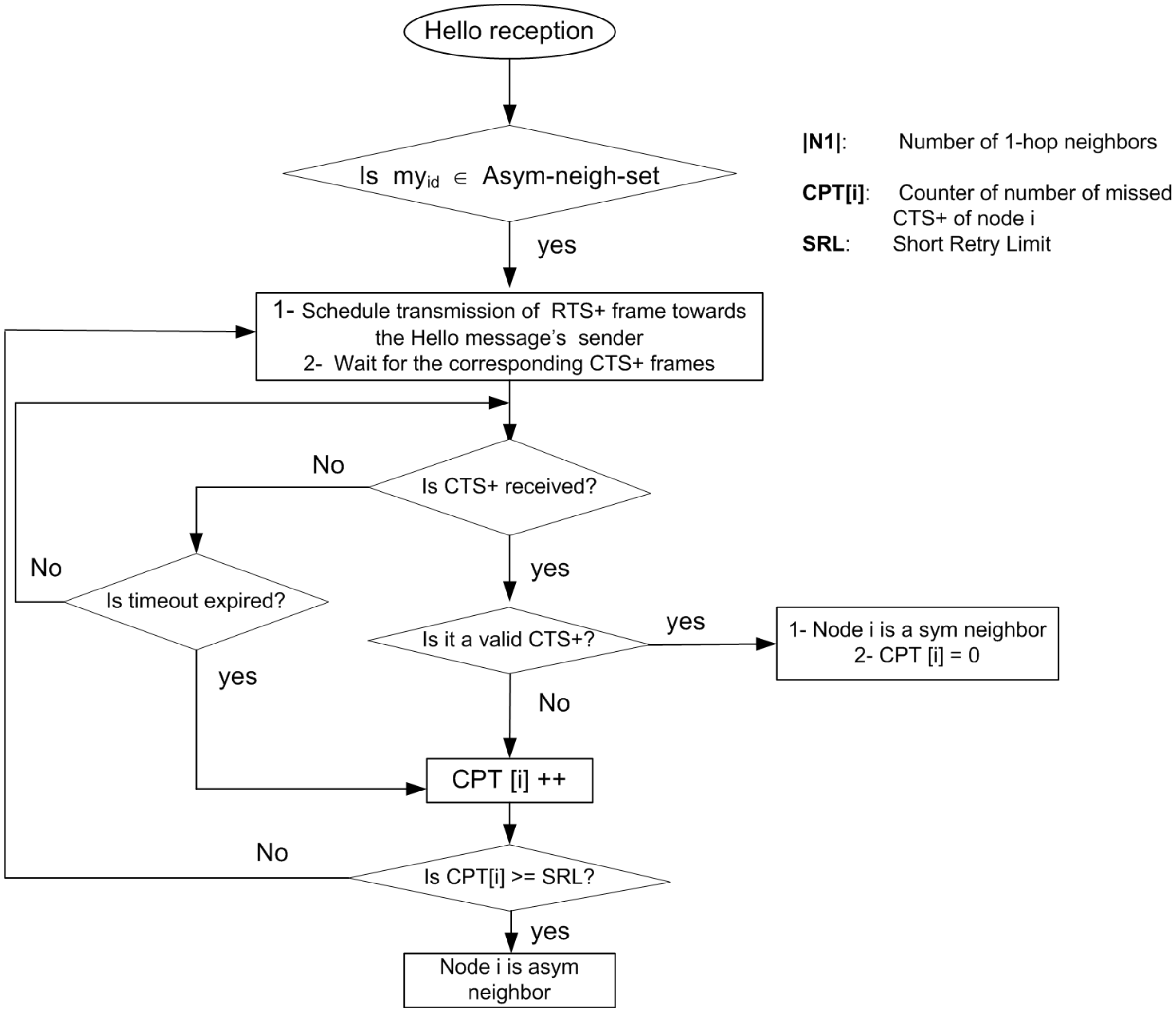

Hello reception
Is $my_{id} \in$ Asym-neigh-set
yes
1- Schedule transmission of RTS+ frame towards the Hello message's sender
2- Wait for the corresponding CTS+ frames
No
Is CTS+ received?
No
Is timeout expired?
yes
yes
Is it a valid CTS+?
yes
1- Node i is a sym neighbor
2- CPT [i] = 0
No
CPT [i] ++
No
Is CPT[i] >= SRL?
yes
Node i is asym neighbor
|N1|: Number of 1-hop neighbors
CPT[i]: Counter of number of missed CTS+ of node i
SRL: Short Retry Limit

Figure 4: Flowchart describing the functioning of our solution

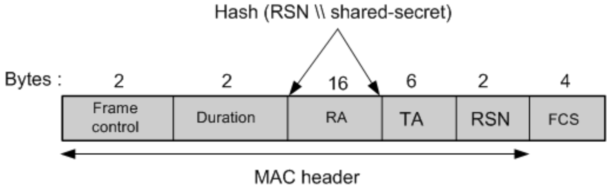


Figure 2: The format of RTS+ frame (32 bytes)

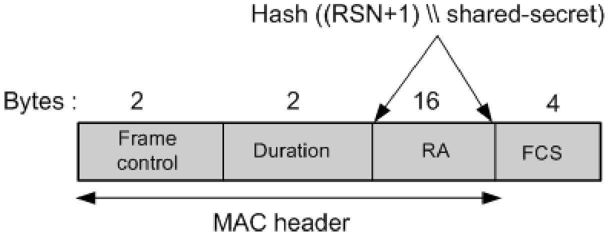


Figure 3: The format of CTS+ frame (24 bytes)

V. Simulation

This section reports the simulation results obtained by implementing the attack described in the section III in OPNET 14.0 network simulator [12]. The simulation settings are summarized in table I. We consider a MANET consisting of 14 wireless nodes having different transmission ranges. These nodes are distributed within the area as shown on the topology depicted in Figure 5.

In order to highlight the impact of this attack a CBR traffic flow *f* (500 bytes/packet and 50 packets/s) is initiated from the node N_2 towards the node N_{10}. Notice that the transmission of data packets is started 20 seconds after the beginning of the simulation in order to allow each node to construct routes towards the rest of the network.

On the other hand, the nodes M1 and M2 colludes to launch a cross layer attack against the nodes A and B, by applying the same steps described in section III. As a result of this attack, the MPR sets of nodes B and A are changed due to

Simulation parameters	Parameter value
Area	1500m *1000m
MAC protocol	IEEE 802.11b
Transmission range	250 m M1: 150m M2: 200m A : 300m B : 250m
Traffic type	CBR
Data rate	11mbps
CBR packets size	512 bytes
Buffer size	62 packets
Short Retry Limit (SRL)	7
Long Retry Limit (LRL)	4
Mobility model	Random way point
Hash function	MD5 (128 bits)
Simulation time	600s
# simulation epochs	5

Table I: Simulation settings

the new established fake symmetric link as illustrated in Table II and consequently the routes towards far away nodes are also changed accordingly. Therefore the new shortest paths shown in Table IV replace the earlier paths depicted in Table III. Hence, the link (B, A) is becoming a black hole which absorbs all the packets routed through it.

The Figures 6 and 7 graph the data packet delivery ratio of the flow f in the case where only VLINK attack is carried out and the case where both attacks are launched together, respectively. As we can see from these figures, when only VLINK attack is launched the delivery ratio decreases dramatically however a number of transmitted data packets still able to reach their destinations. This is due to the fact that when link break is detected at MAC layer as a consequence of missed CTS packets for SRL (short retry limit) times, node B proceeds for selection of a new path towards the node N_{10}. This new path is maintained until it is replaced again by the compromised route upon reception of the subsequent Hello message forwarded by the malicious node. In the other hand, when both of the attacks are launched (cross layer attack) the delivery ratio drops sharply because, in this case, no link break is detected as each data packet transmitted by node B is validated by node M2 (M2 sends back the corresponding CTS and ACK frames) which forces the node B to keep transmitting/forwarding data packets over this path.

Moreover, as depicted in Figure 8, when both of the attacks are launched together the node B will consume more energy in forwarding the data packets passing through it as compared to the case where VLINK attack is launched solely. As a consequence the energy of node B will deplete quickly which decreases its life time. As a simple comparison, the node B transmits 520 bytes (500 bytes corresponds to one data packet's size and 20 bytes corresponds to the RTS frame's size) rather than 140 bytes (7 transmissions of RTS frame which is equal to SRL) in case where VLINK attack launched solely.

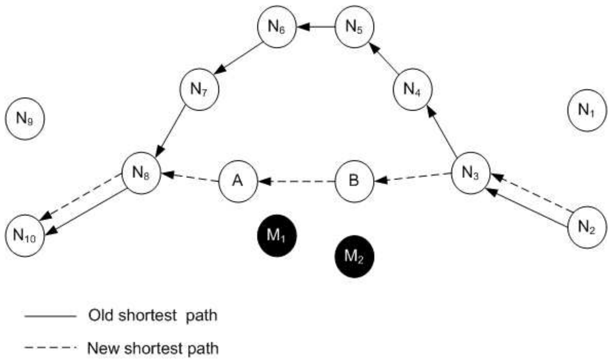


Figure 5: Network topology illustrating an example of the studied cross layer attack

Node	MPR set before attack	MPR set after attack
A	N8	N8, B
B	N3	N3, A

Table II: The MPR sets of nodes A and B

Dest addr	Next hop	Number of hops
A	N3	7
N8	N3	6
N10	N3	7

Table III: Routing table of node N2 before the attack

Dest addr	Next hop	Number of hops
A	N3	3
N8	N3	4
N10	N3	5

Table IV: Routing table of node N2 after the attack

Solution efficiency and overhead evaluation

In the sequel we set up a network consisting of 30 nodes with different transmission ranges, among these nodes 4 are attackers which collude each other to launch cross layer attacks against the other nodes. The nodes are randomly placed within the area and 8 CBR traffic flows are generated in the network (500bytes/packet, 50 packets/s).

Figure 9 shows that our solution performs well when the speed of nodes is low (0 m/s and 5 m/s) because the lower mobility of nodes allows a faster verification of links symmetry using the proposed technique, hence almost the same packets delivery ratio is maintained as compared to the case of network without attackers. When the nodes move faster the link verification phase may take a longer delay and therefore some data packets may be dropped due to the lack of an established path to the destination or data buffer overflow at MAC layer. Despite that, our solution keeps ensuring around 84% of the packets delivery ratio reached in the case where no attack is launched.

To assess the overhead generated by our scheme in terms of the number of the extra bytes sent by each node we vary the mobility speed of nodes as well as their pause time. Note

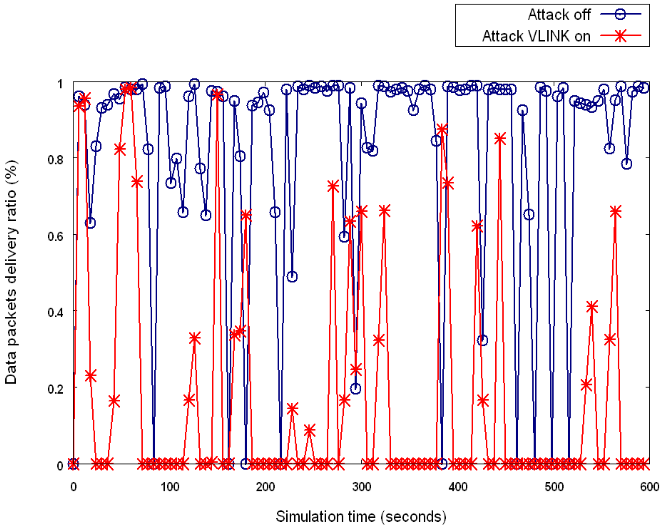


Figure 6: Data packets delivery ratio under VLINK attack solely

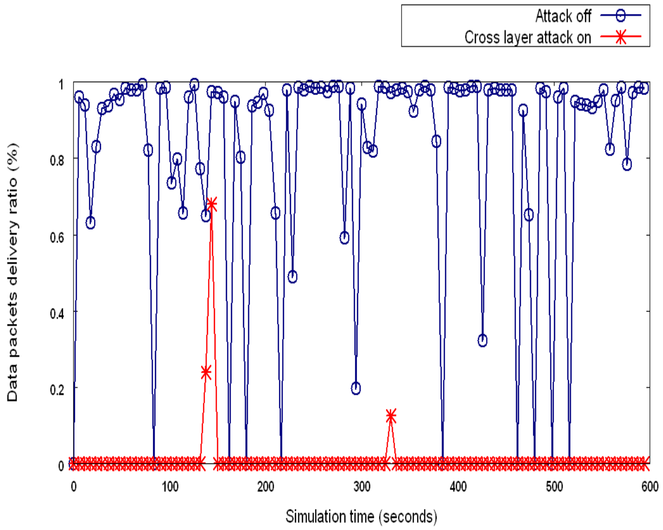


Figure 7: Data packets delivery ratio under the cross layer attack

that in the case of static network (nodes speed = 0m/s) the value of pause times is insignificant. According to the results shown in Figures 10(a) and 10(b) we can see that the extra bytes transmitted by the nodes to prevent the cross layer attack are very small compared to the transmitted bytes representing Hello messages, for a static network. This difference decreases gradually with the increase of nodes speed until the overhead induced by RTS+ and CTS+ surpasses the one induced by Hello messages when the nodes speed reaches 20m/s and their pause times is 0 and 10 seconds. This increase is justified by the rapidly change of the one hop neighbors set due to the high mobility of nodes, therefore links are appeared and disappeared quickly which increase the number of the transmitted RTS+ and their corresponding CTS+ frames in

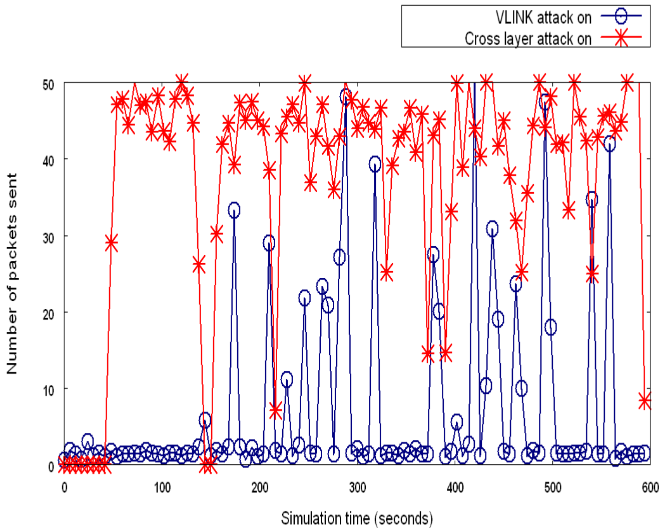


Figure 8: Data packets forwarded by node B

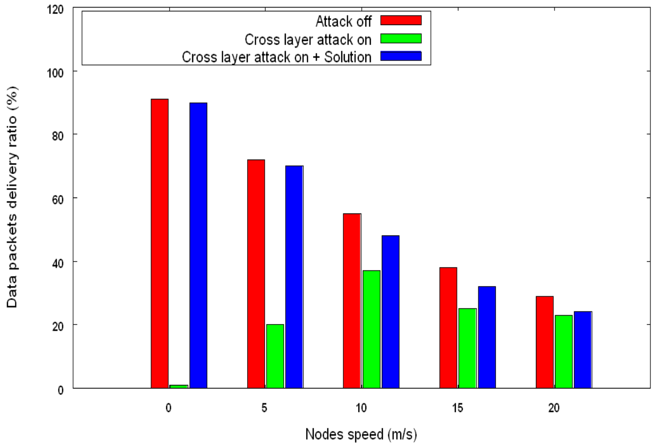


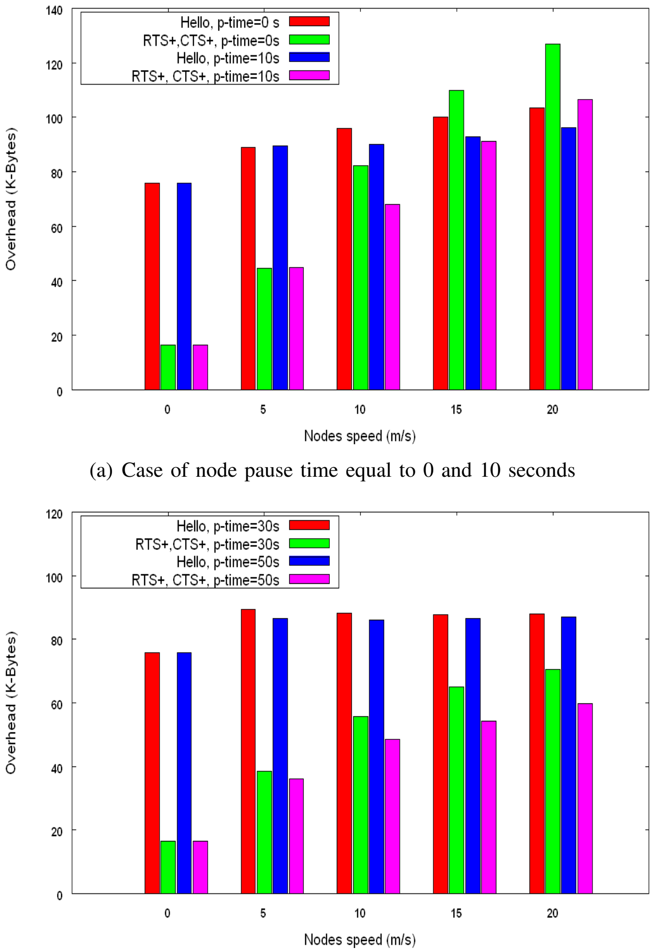
Figure 9: The proposed solution efficiency in terms of data packets delivery ratio under various nodes speed

order to verify the symmetry of these new links. Consequently, we see that generally the extra overhead induced by our solution increases when the speed of nodes turns to larger and their pause times gets smaller.

As a conclusion, the cross layer attack being studied is more harmful for static networks and is less damaging in highly mobile networks. In the other hand, our solution works perfectly with static networks and maintains good results when the nodes start moving.

VI. Conclusion

In this paper, the joint virtual link and false validation attack is analyzed in detail in wireless ad hoc networks running OLSR. This attack can target any routing protocol in MANET however its damage differs from one protocol to another. This attack is launched by two colluding malicious nodes where the first attacker acts at routing level and the second one at MAC layer. A cross-layer solution is proposed in order to avoid the



(a) Case of node pause time equal to 0 and 10 seconds

(b) Case of node pause time equal to 30 and 50 seconds

Figure 10: Variation of the overhead added by RTS+ and CTS+ frames versus nodes speed and pause time

harm caused by this attack. In this solution, a node has to check the symmetry of each new link by sending a special MAC frame dubbed RTS+ and waiting for the corresponding CTS+ frame. The reception of the frame CTS+ within the timeout period and with a valid authentication value confirms the symmetry of the checked link. The simulation results confirm that our solution can efficiently prevent the above attack.

References

[1] IEEE Standard for Wireless LAN Medium Access Control and Physical Layer Specification, P802.11, IEEE, 1999.

[2] Y. Hu, A. Perrig and D.B. Johnson, ”Packet leashes: a defense against wormhole attacks in wireless ad hoc networks”, *in IEEE infocom*, San Francisco, USA, Mar. 2003.

[3] T. Clausen and P. Jacquet, ”Optimized Link State Routing Protocol (OLSR)”, *IETF RFC 3626 (Experimental)*, Oct. 2003.

[4] S. Zhu, S. Xu, S. Setia and S. Jajodia,” Establishing Pairwise keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach”, *in the* 11_{th} *IEEE International Conference on Network Protocols(ICNP’03)*, Atlanta, Georgia, USA, Nov. 2003.

[5] S. Capkun, L. Buttyan and J. P. Hubaux, ”Self-Organized Public-Key Management for Mobile Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, Vol. 2, No. 1, pp. 5264, 2003.

[6] F. Hong, L. Hong, C. Fu, ”Secure OLSR”, *in 19th International Conference on Advanced Information Networking and Applications (AINA’05)*, Taipei, Taiwan, Mar. 2005.

[7] B. Kannhavong, H. Nakamaya and A. Jamalipour, ”A Collusion Attack Against OLSR-based Mobile Ad Hoc Networks”, *in Proc. of Global Telecommunications Conference (GLOBECOM ’06)*, San Francisco, California, USA, Nov. 2006.

[8] B. Kannhavong, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, ”A study of a routing attack in OLSR-based mobile ad hoc networks”, *International Journal of Communication Systems*, Mar. 2007.

[9] S. Djahel and F. Naït-Abdesselam, ”Avoiding Virtual Link Attacks in Wireless Ad Hoc Networks”. *In Proc. ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-2008)*, Doha, Qatar, Apr. 2008.

[10] M. K. Awad, K. T. Wong and Z. Li, ”An Integrated Overview of the Open Literatures Empirical Data on the Indoor Radiowave Channels Delay Properties”. *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, VOL. 56, NO. 5, MAY 2008.*

[11] S. Djahel, F. Naït-Abdesselam and A. Khokhar, ”An Acknowledgment-Based Scheme to Defend against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol”. *In Proc. International Conference of Communication (ICC 2008)*, beijing, China, may 19-23, 2008.

[12] OPNET Technologies, *OPNET Modeler.* http://www.opnet.com/.