International Journal of Research in Advanced Engineering and Technology
ISSN: 2455-0876; Impact Factor: RJIF 5.44
www.engineeringresearchjournal.com
Volume 2; Issue 2; March 2016; Page No. 72-76

Improving security using schematic technology for wireless networks

[1] Dr. Naveen Verma, [2] Sandeep Kumar

[1] Assistant Professor, Department of Computer Science, B. R. A. Govt. College, Kaithal (Haryana).

[2] Research Scholar, Department of Computer Science & Applications, Singhania University, Pacheri Bari, Jhunjhunu (Rajasthan).

Abstract
Security is an indispensable requirement for most network applications, though security is regarded as a standalone component of the architectures of many systems, in case of wireless networks. It must get adequate attention. The types of services expected in wireless networks often require the network security architecture. However, for ensuring a good level of security for such types of networks is not a trivial task. In case of wireless communications, the threats and attacks against wireless networks them are more diverse and often large-scale, and it's not possible to deal with all types of security threats with a single mechanism. Rather, a combination of different security schemes for a single network could be the solution. For example, an attack at the physical layer such as jamming cannot be handled by any key management scheme. Hence, several mechanisms at different layers can be employed simultaneously, to provide holistic security for wireless networks and in addition the level of security in the data transmission and communication phase can be increased via efficient key management schemes.

Keywords: PREQ, AODV, Jamming, MANETs

1. Introduction

1.1 Implementation of Jellyfish Attack

This section presents simulation and implementation of JFDV attack on AODV. To show this, simulation is carried out by making few nodes to act as attackers and analysis is done on network performance. Jellyfish node causes unnecessary delay in delivering data packets affecting throughput of the network. For an attacker to insert itself in an active route between source and destination, it needs to send control packets without any delay. On-demand routing protocols like AODV are vulnerable to this attack, because whenever source node floods the RREQ in the network, malicious node receiving the RREQ packet broadcasts the packet without any hop count updation and delay to its neighbourhood. Whenever the legitimate nodes receive the original source request packets, they are dropped because legitimate nodes have already received packet from the attacker and treat the currently received packets as duplicate packets. Thus, malicious node is included in an active route and disturbs the data forwarding phase. Thereafter, jellyfish attacker nodes delay the forwarding of data packets. We have considered 10 sec delay in our implementation.

Figure 1.1 gives pseudo code of the JFDV attack. When the packet is received by a routing agent, it will call packet reception function, Receive_Request() if it is AODV packet, else packet transmission routine Packet_Forward(). Receive_Request will send it to Forward_Request(). Packet_forward() will schedule transmission of packets accordingly JFDV attacker or not. Thereafter, Route_Find() will check if a node is jellyfish attacker or not. If yes, it will delay the forwarding of packets by 10 sec otherwise forward data packets with zero delay.

```
Recieve_Request (Packet p, Delay d)

{//If a node has no active route to destination. It will forward
Request packet.
        if (JFDV Attack)
            {Forward_request (p, 0);}
        else
            {hop_count = hop_count + 1
                Forward_request(p, d);}}
Packet_forward (Packet p, Delay d)
{if (ttl=0){drop(p); return;}
        if (p is addressed to this node)
        Packet is processed normally
        if (p is a AODV broadcast packet and JFDV attack)
        {schedule Transmission (p, 0.01)}
        else {schedule Transmission (p, 0)}}
Route_Find (Packet p, Delay d)
{if (Route is up)
        {if (JFDV Attack) {if (p addressed to this node)
        {Forward (p, 0)} else {//Scheduling the packets p
        with delay 10sec Forward (p, 10)}
        else {Forward (p, 0)}}}
```

Fig 1.1: Pseudo code for implementing Jellyfish attack in AODV

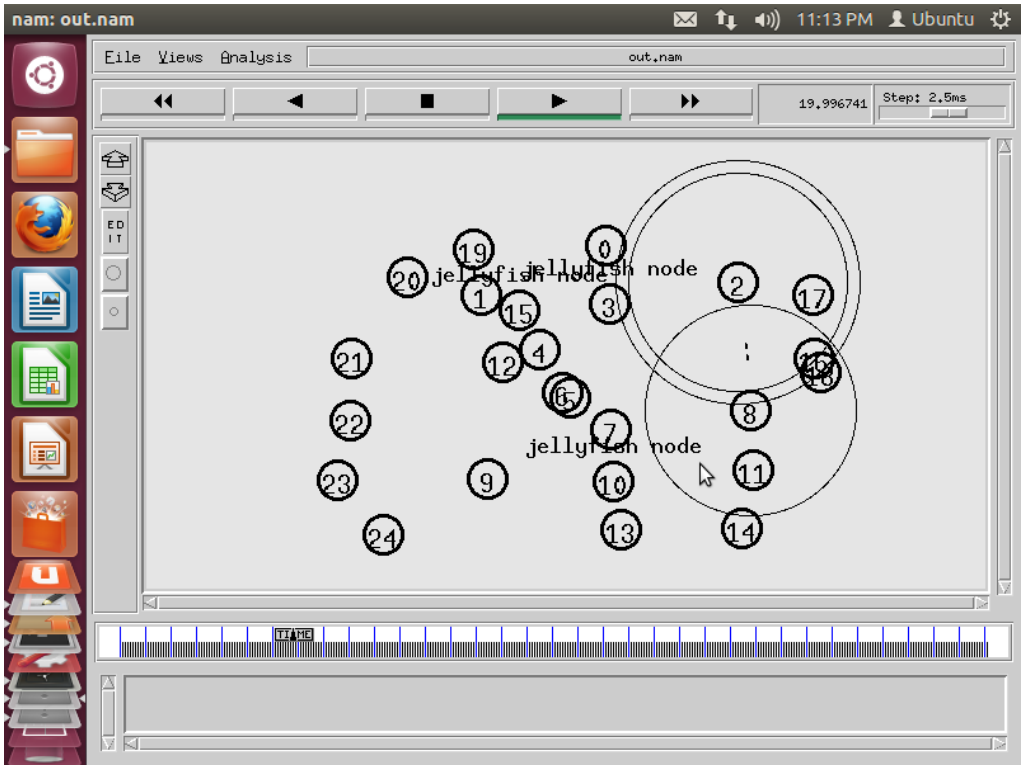

Fig 1.2: NAM output window with Nodes 3, 10, 15 as jellyfish attacker nodes

2. Proposed Algorithm for JFDV Detection and Prevention

JF attacker on becoming the part of routing mesh tends to delay the packets it receives before forwarding them. When ACK is delayed and sender does not receive ACK in a specified time or error for destination unreachable is received by source node then it assumes that packet is lost and begins to retransmit the packet, this result in increased congestion and reduced throughput. A scheme is proposed to revamp the declined performance of network by detecting and preventing JFDV attack.

In MANET, nodes communicate on hop-by-hop basis. Each node can act as a router as well as host. The packet delivery time is summation of processing delay at each router, queuing delay introduced by intermediate nodes, packet transmission time and propagation delay. In our proposed algorithm, every node broadcasts packet to its neighbour node with TTL=1 and neighbouring node IP address as destination IP address after a fixed time interval (1 sec) and a timer is set to keep track of delayed packets. A counter is used to avoid false positives and each node is twice given a chance to not be marked as attacker incorrectly. Timer is set such that it takes threshold delay value. Threshold delay value selected depends on this packet delivery time and it is taken as 5s. If node that sent broadcast packet, receives back a packet from some other neighbour node then timer is checked and if timer is found to be expired then node is suspected to be a JF node and value of counter is decremented and if value of counter falls below 0 then node is considered as a JF attacker node. Once attacker node is detected, its address is saved in malicious list and re-routing is initiated to prevent the attacker node from disturbing the performance of network and in this process attacker node is precluded from the route and transmission of packets is done via route consisting of legitimate nodes only. Eradication of false positives is necessary so as to preserve benign nodes from being marked as attackers when they are not. The nodes flush their malicious list so as to allow attacker node to behave properly in the future. The algorithm is given in Figure 4.3.

```
P: Broadcast packet
Count=2
T: Timer
For each node
{Create a packet P Broadcast the packet to its neighbouring nodes}
For each node
{If (P received) {If (T expired) {Jellyfish attacker suspected
                        Count= Count -1
                        If (Count < 0) {Node is a jellyfish node}}}}
For each node
{While (route discovery)
        {If (RREP from jellyfish attacker)
                {Reject RREP}}
```

Fig 1.3: Pseudo code of Proposed Algorithm

3. Result Analytical feature

JF attack effects adversely throughput of network because of congestion caused due to retransmissions and PDR decreases due to increased RTO led by delay introduced by attacker. It increases average end-to-end delay of the network. The effect is more devastating as number of jelly fish node increases. The performance degradation depends on the delay introduced by the jellyfish nodes. We have used incorporated a delay of 10 sec by a jelly fish node. Manifestation of results is in the form of graphs is given below.

The Figure 2.1 below shows the increase in E2E delay as number of jellyfish nodes increase from 1, 3, 6 to 9 which is

4%, 12%, 24% and 36% of the network. The results show that our proposed algorithm detects multiple jellyfish nodes efficiently and reduces average end-to-end delay of the network in presence of jelly fish nodes. The proposed algorithm incurs consistent delay making it suitable for larger networks with multiple malicious nodes.

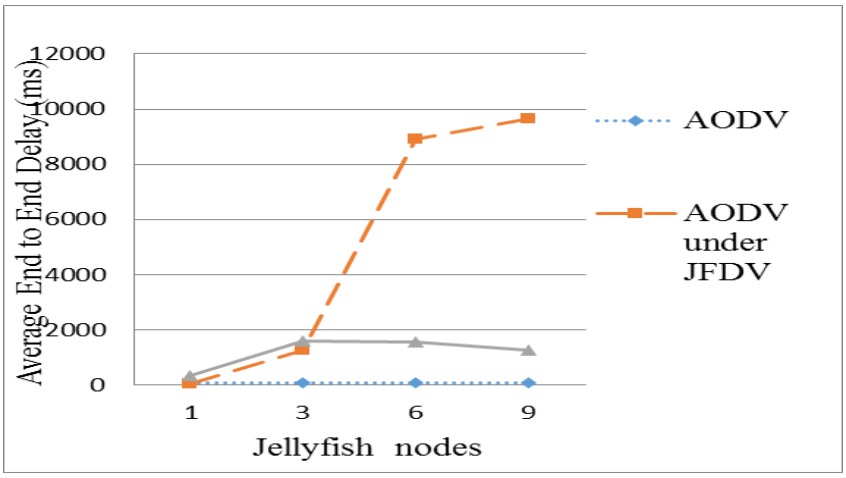

Fig 2.1: Average end-to-end delay Vs. Number of jellyfish node

Table 2.2 Comparison of Average E2E delay Vs. Number of jellyfish nodes

No. of malicious nodes ↓ / Protocols →	AODV	AODV under JFDV	Proposed Algorithm
1	87.11	40.85	348.07
3	87.11	1254.11	1533.07
6	87.11	8924.76	1544.28
9	87.11	9635.6	1261.25

As shown in Table 2.2 above, the delay increases from 40.85 ms to 9635.6 ms as number of jellyfish nodes increase from 1 to 9. Our proposed algorithm reduces this average delay to 348.07 ms, 1533.07 ms, 1544.28 ms to 1261.25 ms for 1, 3, 6, 9 malicious nodes, respectively.

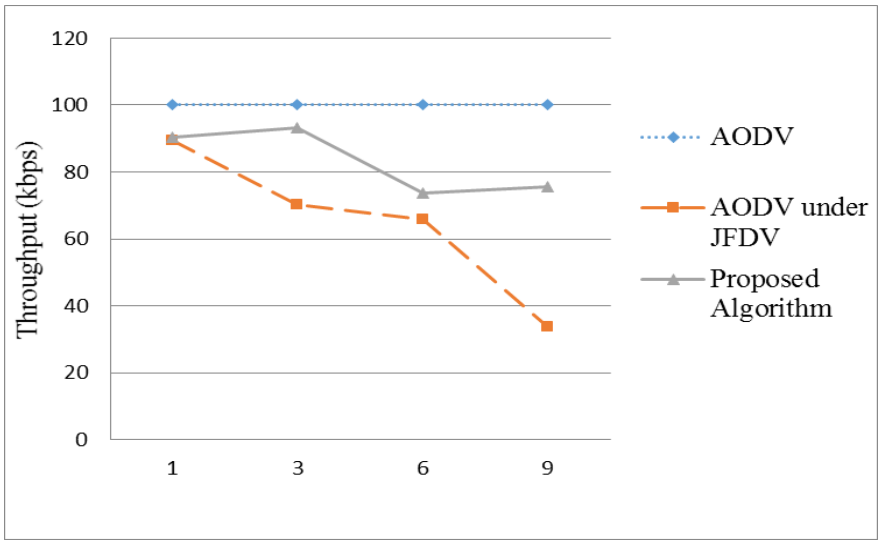

Fig 2.3: Throughput vs Number of jellyfish nodes

As shown in Table 2.3 below, the throughput increases from 33.62 kbps to 75.52 kbps with 9 malicious nodes. There is a decrease in throughput (89.43 kbps to 33.62 kbps) of the network as number of jellyfish nodes increases from 1 to 9. Our proposed algorithm maintains the throughput of the network with an average value of 83.20 kbps as jellyfish nodes varies from 1 to 9.

Table 2.3 Comparison of Throughput Vs Number of jellyfish nodes

No. of malicious nodes ↓ / Protocols →	AODV	AODV under JFDV	Proposed Algorithm
1	100.29	89.43	90.33
3	100.29	70.34	93.19
6	100.29	65.72	73.77
9	100.29	33.62	75.52

Figure 2.3 shows throughput obtained during simulation run of 100 sec and it can be seen that throughput is depreciated when attacker node is introduced in network and on application of proposed algorithm, improvement is made. The decrease in throughput is less with single attacker node i.e. 89.43 kbps. In normal AODV, it is 100.29 kbps and with proposed scheme it is 90.33 kbps. With 3 attacker nodes, throughput of our proposed algorithm reaches to 93.19 kbps. The throughput increases gradually with simulation time as more number of packets is received successfully at destination node as time increases.

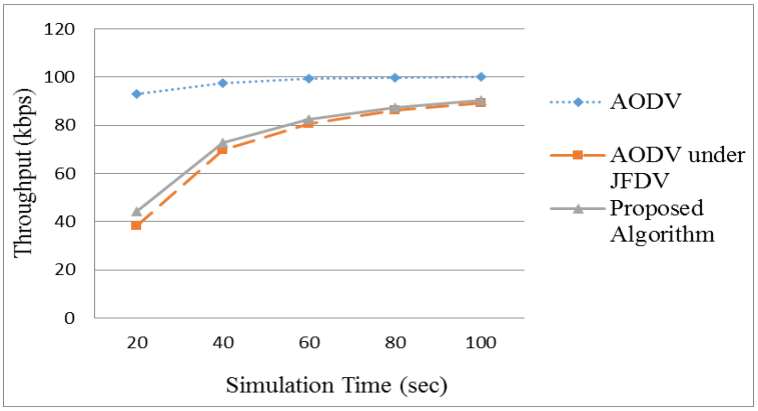

Fig 3.3: Throughput Vs Simulation Time when there is one attacker

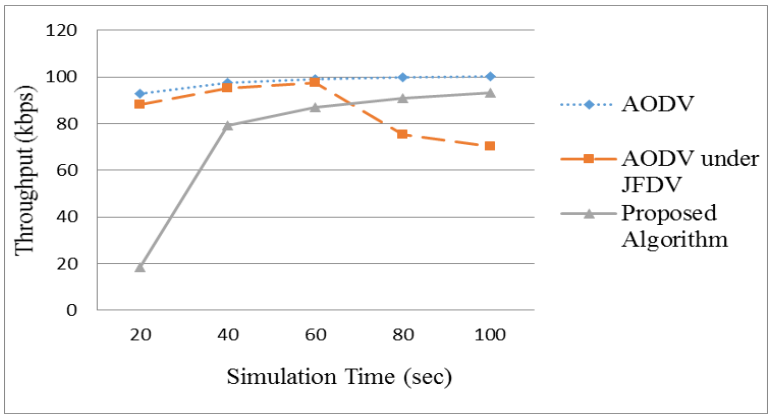

Fig 3.4: Throughput Vs Simulation Time when there are 3 attackers

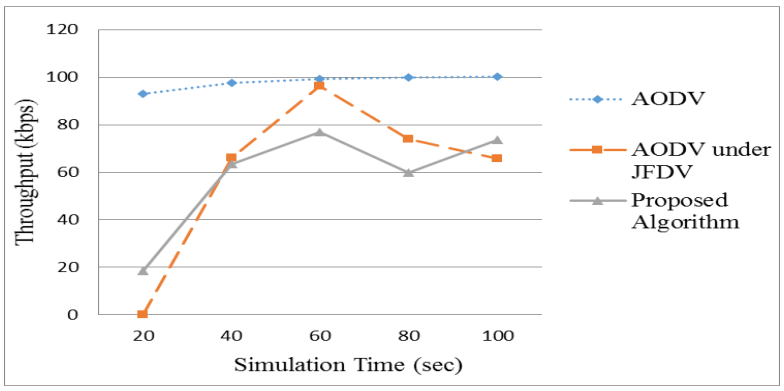

Fig 3.5: Throughput Vs Simulation Time when there are 6 attackers

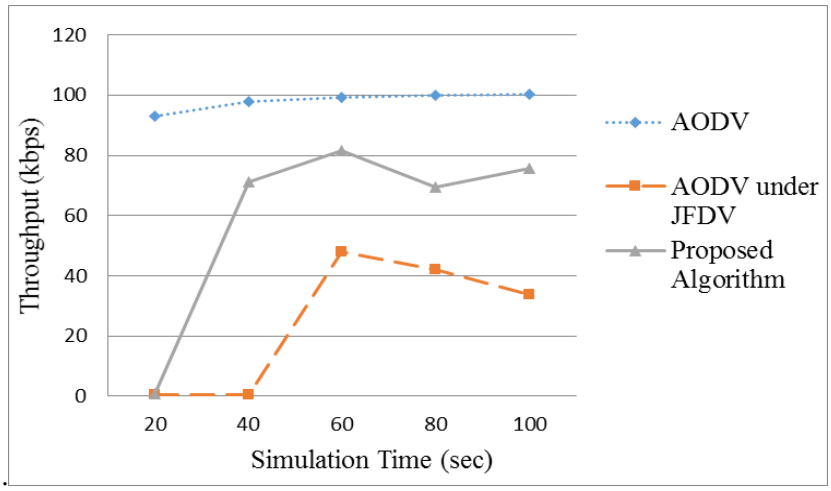

Fig 3.6: Throughput Vs Simulation Time when there are 9 attackers

Since proposed algorithm is reactive, it will take some time to detect an attack and isolate attacker form the network so initially throughput is low and gradually as simulation proceeds, proposed algorithm is successful in detecting jellyfish node, the throughput increases and reaches to 73.77 kbps with 6 attacker nodes and 75.52 kbps with 9 attacker nodes.

4. Conclusion

This work introduces a new metric in MANET, to provide more security using the DMPD algorithm which represented the following parameters (i) number of packets generated and forwarded by the neighboring nodes (ii) the past activity of the node (iii) signal strength (iv) delay of forwarding packets and (v) throughput. It increased the efficiency and security compared to the other existing algorithms. And the indirect monitoring method is the greatest boon to this project. The Future enhancement of this work is dealing with the weightage of parameters in the node which is not concentrated in this work.

5. References

1. Aad JP Hubaux, Knightly EW. Impact of Denial of Service Attacks on Ad Hoc Networks, IEEE/ACM Transactions on Netwroking, Aug. 2008; 16:791-802.
2. Dhiman Deepika, Nayyar Anand. Complete Scenario of Routing Protocols, Security Leaks and Attacks in MANETs, in Journal Proceedings of the IJARCSEE, October, 2013; 3(10).
3. Hetal P Patel, Minubhai B Chaudhari. Survey: Impact of Jellyfish on Wireless Ad-Hoc Network, in proceeding of INJCR'10, 2010; 10(5):2, 5-9.
4. Hepikumar R Khirasariya. Simulation Study of Jellyfish Attack in MANET (mobile ad hoc network) using AODV Routing Protocol, in proceeding of AISec'10, 2010, 1-3.
5. Kaur Manjot, Nayyar Anand. A Comprehensive Review of Mobile Adhoc Networks (MANETS) in International Journal of Emerging Trends & Technology in Computer Science (ISSN2278-6856), November-December. 2013; 2(6).
6. Perkins CE, Royer EM, Das SR. Ad hoc on-demand distance vector (AODV) routing, IETF internet draft. MANET Working Group; Jan 2004.
7. Perkins C. AODV routing implementation for scalable wireless ad-hoc network simulation (SWANS). http://jist.ece.cornell.edu/docs/040421-swans-ao
8. I Aad, Hubaux JP, Knightly EW, Impact of Denial of Service Attacks on Ad Hoc Networks, IEEE/ACM Transactions on Networking, Aug 2008; 16:791-802.