IJCSC International Journal of Computer Science & Communication Vol. 1, No. 2, July-December 2010, pp. 299-302

A Priority Based Protocol for Mitigating Different Attacks in Mobile Ad hoc Networks

Himadri Nath Saha[1], Debika Bhattacharyya[2] & P. K. Banerjee[3]

[1,2]CSE,Institute of Engineering and Management, Kolkata
[3]ETCE, Jadavpur University, Kolkata
Email: [1]him_shree_2004@yahoo.com, [2]bdebika@yahoo.com

ABSTRACT

To combat different attacks such as black hole attack or gray hole attack or worm hole attack imposed by malicious nodes we propose a priority based mechanism that detects the existence of malicious nodes without relying on any central authority (CA), an approach significantly different from the existing ones. According to this scheme a node will maintain a list of priorities of its neighbours and when a neighbour's priority becomes less than a certain value (threshold value) then that node is disconnected from its neighbourhood. Our analytical results as well as simulations under realistic conditions demonstrate that the proposed mechanism works effectively even when a large number of malicious nodes are present.

Keywords: Gray Hole, Black Hole, Worm Hole, Local Monitoring, AODV

1. INTRODUCTION

There is significant interest in the research and development of Mobile Ad Hoc wireless networks for a variety of emerging applications. As they are infrastructure less networks, these multi hop wireless networks are especially suited for scenarios where it is infeasible or expensive to deploy significant networking infrastructure. However, the open nature of the wireless communication channels, it's infrastructure less property and the hostile environments where they may be deployed, make them vulnerable to a wide range of security attacks. These attacks could involve eaves dropping, message tampering, or identity spoofing, which have been addressed by customized cryptographic primitives. Many attacks are targeted directly at the data traffic by dropping all data packets (Black hole attack), selectively dropping data packets (Gray hole attack), Two colluding malicious nodes may launch a wormhole attack to involve themselves in a route by simply giving the false illusion that the route through them is the shortest (Worm hole attack) and performing statistical analysis on the data packets to obtain critical information, such as the location of primary entities in the network. For an attacker to be able to launch damaging data attacks, one option is to have a large number of powerful adversary nodes distributed over the network and possess cryptographic keys. Alternately, the attacker can impose such attacks by having a few powerful adversary nodes that need not authenticate themselves to the network (i.e., external nodes). The attacker can achieve this by targeting specific control traffic in the network. Typical examples of control traffic are routing, monitoring aliveness of a node, topology discovery, and distributed location determination. Section II describes our proposed protocol, Section III gives the simulation results and section IV concludes the paper.

2. ATTACK MITIGATION BY PRIORITY PROTOCOL SCHEME

The proposed scheme consists of the following steps:

Step 1: Whenever a node enters in a Mobile Ad Hoc network IP allocation is the first step in which the node will get it's IP along with initial priority and we have adopted the technique of Prime DHCP [1].

Step 2: Neighbour Discovery is the second step of the proposed scheme. New node will send the HELLO packets to its neighbours and discover the identity of the neighbours along with their priority.

Step 3: Authentication is the next step of the scheme in which it will broadcast information about its existence and exchange keys with the neighbours according to the scheme HEAP [3] which is a hop-by-hop authentication protocol. HEAP authenticates packets at every hop by using a modified HMAC-based algorithm along with two keys and drops any packets that originate from outsides. Specifically, with the initial bootstrapping phase, every node shares a pair wise secret hash key, called o-key, with each of its neighbours and generates one common secret hash key called i-key, and securely distributes it to all of its one hop neighbours. If any node in the network is detected as malicious then the detector should broadcast it to all of its neighbours. Then it will generate a new MAC for every individual neighbour using its pair wise key and will generate an i-key. Only that node and its neighbours have access to i-key. All the neighbours will be able to authenticate that the packet is originated from that node using the o-key. HEAP helps to mitigate Denial of Service (DoS).

Step 4: Next the node will update its priority entry by using either priority up gradation function (PUF) or priority Reduction function (PRF). For that we have designed a Priority Up Gradation Function (PUF) and Priority Reduction Function (PRF) as follows.

Priority Up Gradation Function (PUF): Each node in MANET will upgrade the priority of its neighbours which have given its acknowledgement message (ACK) and others will be deleted from the list of neighbours. We have to design the PUF as increasing function, keeping in mind that the rate of increment is very high. So we have chosen PUF as exponential function shown in Fig.1 and the derivation is shown below. When the priority of a node becomes high then the node becomes more reliable.

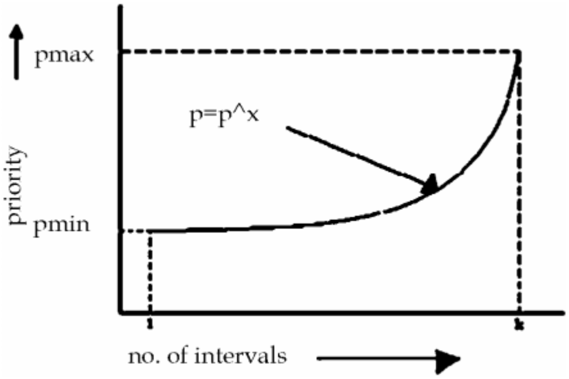

Fig.1: Priority Up Gradation Function (PUF)

After every interval each node will update its priority but it can not go on increasing it's priority without a limit. So, there is a limit up to which a node can increase it's priority and then saturates beyond which it can not go. This value is called the saturation limit and for our convenience say this value is $P_{\max}$.

For our calculation needs, let us assume that after *k* intervals a node attends its maximum priority $P_{\max}$. As PUF has been chosen as an increasing function exponential function so we can write

$p = p^x$ where p is the priority of a node and x is the exponent (1)

Let us assume that a node enters the network with priority $P_{\min}$. Then

After the 1st interval, the priority $p = p^x_{\min}$· After 2nd interval, $p = (P^x_{\min})^x = P^{x^2}_{\min}$. So in general after *k* intervals, $p = P^{x^k}_{\min}$ (2)

Therefore, $P_{\max} = P^{x^k}_{\min}$

i.e. $x = (\log(P_{\max}) / \log(P_{\min}))^{\frac{1}{k}}$ (3)

i.e. $\log(P_{\max}) / \log(P_{\min}) > 1$ i.e. $x > 1$

i.e. the PUF is increasing which is the required condition.

Priority Reduction Function (PRF)

Each node in MANET will degrade the priority of its neighbours whenever the neighbours are detected as malicious by other node s(neighbours) in Manet.

To find the nature of the PRF the following points must be considered.

(a) The amount of priority reduction increases with successive no. of detections.

(b) The amount of reduction will be greater if the priority of the verifier node is high and vice versa.

(c) No node will be disconnected from the network only after a few detections.

Thus PRF function depends on the no. of detections. and the priority of the verifier node.

Here, a node can detect other node for malicious activity only once.

We are designing the PRF function as shown in Fig. 2. We have shown the nature of PRF for different values of number of detection (say n1, n2, n3 and n4). The nature of decrement is exponential.

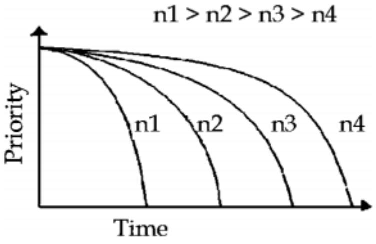

Fig.2: Priority Reduction Function (PRF)

The PRF function can be designed in the following manner:

Let, y = priority of the malicious node

Z = priority of the verifier node

n = number of detections as malicious node

Therefore, $y = y - z^n / 2 \times P_{\max}$ (4)

Where $P_{\max}$ is the maximum priority i.e., the saturation limit in the network and $z^n/2 \times P_{\max}$ is the amount of priority reduction by a single node.

If a node having maximum priority detects the node having minimum priority then the minimum priority node will not be deleted at first detection. For this reason we introduced 2 × $P_{\max}$ factor in denominator.

Now, we calculate the value of $P_{\max}$ from the above function. In a Mobile Ad Hoc Network for a particular time instant the number of node say $N = n + 1$. So, for a particular node the number of neighbours at a time can have a maximum value n. Then the worst case arises when that particular node has the maximum priority$P_{\max}$ and all the neighbour nodes have lowest priority $P_{\min}$. So, we need to design $P_{\max}$ in such a way that after all the n nodes have detected it then it must be deleted from the network. We know that

$$y = y - z^n / 2 \times P_{\max}$$

where $z^n / 2 \times P_{\max}$ is the amount of priority reduction by single node.

The total amount of priority reduction due to the detection of all the n nodes is

$T = P_{\min}^1 / 2 \times P_{\max} + ... + P_{\min}^n / 2 \times P_{\max}$ Where

$P_{\min}^1 / 2 \times P_{\max}$ is the amount of priority reduction by the first node.

Lastly $P_{\min}^n / 2 \times P_{\max}$ is the amount of priority reduction by the nth node.

$$\text{i.e.}\ T = [P_{\min}^1 + ... + P_{\min}^n] / 2 \times P_{\max} \tag{5}$$

$$\text{So}\ T = A / 2 \times P_{\max} \tag{6}$$

where $A = P_{\min}^1 + ... + P_{\min}^n$. Now we should have $P_{\max} <= T$ because otherwise n nodes having $P_{\min}$ will not be able to delete the node having $P_{\max}$.

i.e. $P_{\max}^2 \leq A / 2$

After certain number of detections if the priority of the malicious node becomes lower than $P_{\min}$ then the node gets disconnected from the network.

Data Transmission and Malicious Node Detection: Each node is transmitting data using Ad Hoc on demand Distance Routing Protocol and local monitoring is used for malicious node detection.

(a) *Ad Hoc On Demand Distance Vector Routing(AODV):* AODV is a source initiated on demand routing protocol used to deny the potential threat from Black hole attack. Every mobile node maintains a routing table that maintains the next hop node information for a route to the destination node. It uses the specified route if a fresh enough route to the destination node is available in its routing table.

(b) *Local Monitoring:* Local Monitoring is a collaborative detection strategy used where a node monitors the control traffic going in and out of its neighbours. This strategy was introduced in [4][5] for static sensor networks

3. SIMULATION RESULTS

We have implemented PUF and PRF using GCC compiler in Linux platform. We have tested PUF and PRF for different values of $P_{\min}$. Test results are given below. In our simulation program we have assumed $P_{\min} = 10$ and $n = 10$. So, we will now determine the upper limit of $P_{\max}$ which is determined as follows:

$$P_{\max}^2 \leq [10^1 + \ldots + 10^n] / 2$$

$$\Rightarrow P_{\max}^2 \leq [10 + 100 + 1000 + \ldots + 10^{10}] / 2$$

$$\Rightarrow P_{\max}^2 \leq 11111111111 / 2$$

$$\Rightarrow P_{\max}^2 \leq 5555555555.5$$

$$\Rightarrow P_{\max} \leq 74535.59925$$

$$\Rightarrow P_{\max} \approx 74535$$

In our simulation program we have assumed the value of k = 10 i.e. the number of steps after which the node reaches its maximum value is taken to be 15. Then the Priority Up gradation function i.e PUF is determined as follows: From Equation (1) $p = p^x$ where p is the priority and x is the exponent. We have already calculated $P_{\max} = 74535$ and we have assumed that $P_{\min} = 10$. So from equation (3) $x = (\log(P_{\max}) / \log(P_{\min}))^{\frac{1}{k}}$ where k is the number of steps after which the node reaches its maximum value. Here $k = 15$ i.e. x = 1.111346005 (putting the values of $P_{\max}$, $P_{\min}$ and k)

i.e. $p = p^{1.111346005}$. Now we can calculate the priority of different nodes. Let us first determine the priority of node 6.

After 1st step, node 6 will have $P_6 = 10^{1.111346005}$.

i.e. $P_6 = 12.92248406$ where P_6 is the priority of node 6.

After 2nd step, $P_6 = 12.92248406^{1.111346005}$

i.e. $P_6 = 17.18264205$ after 2nd step.

After 3rd step, $P_6 = 17.18264205^{1.111346005}$ .i.e. P_6=23.583717 after 3rd step.

So node 6 will attain a priority = 23.583717 after 3 steps. This is shown in Fig. 3 where all nodes have been shown marked with their respective priority value. In the same way we can calculate the priority value of node 1 which attains it's maximum priority value=74535 after 15 steps.

Similarly other nodes upgrade its priority after certain number of steps (k) and the priority values are shown in figure–3. In this paper we have shown only one test case with node 6 but we have also tested all the cases with different values of priority. Our scheme is giving excellent performance even the number of nodes are high in the network and test results will depict that the scheme istime efficient and robust. In following figure-3 we have assumed that all the nodes are mobile and the network has been set up in Ad hoc basis.

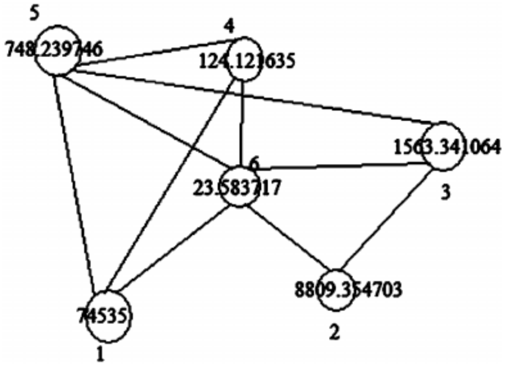

Fig. 3: A Network is Shown having Five Nodes Marked with their Respective Priority Value

Let node 6 is detected by node 1, node 2, node 5 and node 3 consecutively. Then the priority of node 6 reduces as follows:

(1) $P_6 = 23.583710$ after detected by node 1.

(2) $P_6 = 23.421227$ after detected by node 2

(3) $P_6 = 19.66552352$ after detected by node 5

(4) $P_6 \leq P_{\min}$ so the node 6 is deleted from the network, because in this step it crosses the threshold value $P_{\min}$ after detected by node 3.

4. CONCLUSIONS

In this paper we have presented an algorithm to mitigate different types of attack like worm hole, black hole, Gray hole i.e. any type of collaborative attack in an efficient manner. Our protocol does not need the help of CA (central authority). We derived the necessary formula for mitigating attacks and we have shown analytically that our proposed protocol successfully detect the malicious node and delete it from the network after being detected as malicious node. Since our protocol is independent of CA so any node can act as peer node by upgrading its priority value under normal condition after certain interval. Also any node will degrade its priority value when it has shown any malicious activity.This scheme can be implemented in distributed manner which will provide less network dependency.

REFERENCES

[1] Yuan-Ying Hsu and Chien-Chao Tseng, "Prime DHCP: A Prime Numbering Address Allocation Mechanism for MANETs", *IEEE Communications Letters*, **9**, No. 8, August 2005.

[2] RFC 4861 Neighboor Discovery for IPV6, T. Narten et al, September 2007.

[3] Rehan Akbani, Turgay Korkmaz, G.V.S. Raju, "HEAP: A Packet Authentication Scheme for Mobile Ad Hocnetworks:HEAP", *ScienceDirect,Adhoc Network*, **6**, Issue 7, Septmember 2008.

[4] Issa Khalil, Saurabh Bagchi, Ness B. Shroff, "MOBIWORP: Mitigation of the Wormhole Attack in Mobile Multihop Wireless Networks Ad hoc Network", **6**, Issue 3, May 2008, 344–362.

[5] Khalil, S. Bagchi, N.B. Shroff, "LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks", *International Conference onDependable, ScienceDirect, Computer Network rk*, **51**, Issue 13,12, Septmember 2007, 3750–3772.

[6] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das, "Ad Hoc on Demand Distance Vector (AODV) Routing", *IETF Internet Draft*, Draft-ietf-manet-aodv-12.txt, November 2002.

[7] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad hoc Wireless Networking", in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996.

[8] B.J. Culpepper, H.C. Tseng, "Sinkhole Intrusion Indicators in DSR MANETs", *in: Proceedings of BroadNets '04*, October 2004, pp. 681– 688.

[9] Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures, Chris Karlof, David Wagner,Ad Hoc Networks, **1**, Issues 2-3, September 2003, Pages 293-315.

[10] Defending Against Cache Consistency Attacks in Wireless Ad hoc Networks Wensheng Zhang, Guohong Cao,Ad Hoc Networks, **6**, Issue 3, May 2008, Pages 363-37.

[11] Edith C.H. Ngai a, Jiangchuan Liu b, Michael R. Lyu, "An Efficient Intruder Detection Algorithm against Sinkhole Attacks in Wireless Sensor Networks", *Computer Communications,* **30** (2007), 2353–2364.

[12] Khattab, M. El-Hadidi, H. Mourad, "Analysis of Wireless CSMA/CA Network using Single Station Superposition (SSS)", *International Journal of Electronics and Communications (AE),* **56,** 2002, 71–81.

[13] Y.C. Hu, A. Perrig, D.B. Johnson, Packet leashes: a Defense Hoc Routing, in: Proceedings of the 1st ACM International against Wormhole Attacks in Wireless Networks, in: IEEE Workshop on Performance Evaluation of Wireless Ad hoc, INFOCOM, 2003, pp. 1976–1986.

[14] L. Hu, D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks", in: *Network and Distributed System Security Symposium*, 2004, pp. 131–141.

[15] I. Khalil, S. Bagchi, N.B. Shroff, "LITEWORP: a Lightweight Countermeasure for the Wormhole Attack in Multihop Wire-less Networks", *in: International Conference on Dependable Systems and Networks (DSN)*, 2005, pp. 612–621.

[16] Y.C. Hu, A. Perrig, D. Johnson, "Rushing Attacks and Defense in Wireless Ad hoc Network Routing Protocols", *in: References ACM Workshop on Wireless Security (WiSe'03)*, 2003, pp. 30–40.