A Cryptographic Hashing Solution for mitigating Persistent Packet Reordering Attack in Wireless Ad Hoc Networks

P Raghavendra Raju
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
Mangalore, India
raghavendra_nec@yahoo.co.in

Dr.K.C.Shet
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
Mangalore, India
kcshet@nitk.ac.in

***Abstract*— In this paper, we propose a new scheme for mitigating the persistent packet reordering attack. This attack mainly makes use of the end to end congestion control mechanism of TCP. The proposed scheme uses symmetric key cryptography and modified SHA-1 (mSHA-1) hash function for verifying packet reordering. This is implemented with AOMDV routing protocol using NS-2 network simulator. The proposed solution blocks the attack after its detection. The efficiency of the proposed scheme is evaluated on different performance metrics like congestion window and TCP Goodput as the attack mainly targets the congestion control mechanism of TCP.**

Keywords- Jellyfish attack, AOMDV, wireless adhoc networks, packet reordering, and keyed hash function.

I. INTRODUCTION

Ad-hoc means "for a particular purpose without consideration of wider application". A Wireless Ad-hoc Network is a group of nodes without any existing infrastructure and forms a temporary network. These networks are used in emergency search, disaster management, electronic class rooms, military operations, conferences etc. An ad-hoc network does not contain any centralized administration. In ad-hoc networks, nodes can either enter/leave the network as they wish. A packet can reach to a destination from source through multiple hops. Each node in the network acts as both router and also as a host. As a router, the node forwards the packet it receives to other nodes. Ad-hoc networks can handle topology changes and malfunction in nodes. For example, if a node leaves the network and causes link breakages, then affected nodes can request new routes. Even though it increases delay, the network is still operational.

The unique characteristics of an adhoc network such as dynamic topology, resource constraints, lack of network infrastructure or centralized administration, make it more susceptible to a number of attacks and thus the vulnerability in this networks will be more. As this technology is increasing day by day and will be widely used in the years to come, providing the security to this type of networks is a major issue. Denial of service (DOS) attack is one of the major threats to the Adhoc networks, in which Protocol-compliant DOS attacks are the most difficult to defend against. In [2], Aad et al. refer to such attacks as Jellyfish attacks.

The rest of this paper is as follows. In section II, we describe packet re-ordering attack in ad hoc networks. In section III, we discuss about related works on the attack. The description of proposed scheme is given in section IV. The results and analysis of this scheme are shown in section V. Finally, section VI concludes the paper with discussing the future work as well.

II. JELLYFISH RE-ORDERING ATTACK

TCP is the most widely used transport layer protocol in internet today. TCP has several issues and challenges while using in adhoc networks [3].

- Limited Bandwidth degrades the throughput of the network.
- Power consumption is another factor that reduces the throughput.
- As the adhoc networks have dynamic topology, there will be packet loss and the nodes initiate route discovery procedure frequently.

Ad hoc networks are prone to different Denial-of-Service (DoS) attacks because of its dynamic topology, remote location and services. The different types of DoS attacks in Adhoc networks are jamming, exhaustion and integration, selective forwarding, tampering, misdirection, sinkholes, Sybil, wormholes, and flooding [1]. A very common attack in wireless networks is Jellyfish attack. It targets TCP's congestion control mechanism. The main goal of the Jellyfish nodes is to reduce the goodput of all the flows to near-zero by either reordering the packets or dropping a small fraction of packets. These forwarding mechanisms are variants of Jellyfish attack [2].

Many of the attacks disobey the protocol rules, but the Jellyfish attack obeys all the protocol rules. The main strength of this attack is that it is compliance with all the data plane and control plane protocols, so that the detection and diagnosis of the attack becomes difficult and time consuming. This attack mainly targets closed-loop flows as such flows respond to network conditions like packet loss and packet delay. These attacks are passive and are difficult to detect. There are three variants of Jellyfish attacks: Jellyfish Re order attack, Jellyfish periodic-dropping attack, Jellyfish delay variance attack. The first JF attack is the Reordering attack. In networks due to route changes and multi path routing, TCP has vulnerability towards packet reordering. To improve robustness against this packet reordering, a number of modifications to TCP have been proposed. No TCP variant is robust to malicious and persistent reordering as employed by the JF reordering attack [2].

In this attack the malicious node delivers all the packets

978-0-7695-4817-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCS.2012.83

it receives, but instead of forwarding them in a FIFO order, it forwards in a random order from the queue or the malicious node places the packets in a random buffer instead of a FIFO buffer (queue) i.e. the attacker node reorder the packets persistently which results in near zero good put, even though all packets transmitted by the source arrive at the destination.

Fig. 1 shows an example of Jellyfish Reorder Attack, where the JF Node reorders the buffer that contains the packets and sends those packets from the buffer. The packets arrive in out-of-order at the destination and the destination sends duplicate ACKs (DUPACK) to the sender. If three such DUPACKs arrive at the source, it retransmits the packets without waiting for retransmission timeout. Even though the packets arrive at the destination, the source retransmits the packets again, assuming that the packets have been lost. This persistent packet reordering causes false retransmit.

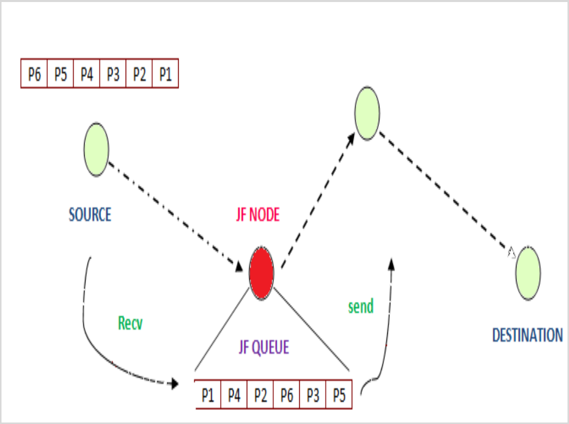

Figure 1. Jellyfish Reorder Attack

Suppose that, if the sender sends the packets from 1 to 10 and the receiver receives in such a way that the packet 1arrives at last, then the reordering length will be 9. If the re-ordering length is greater than or equal to the threshold of DUPACK, then the source assumes that the packets are lost and this cause false fast retransmits, which makes the TCP sender to reduce its congestion window unnecessarily and transmit the segments again. If TCP enters congestion control state then the throughput of TCP degrades significantly. This affects the TCP performance.

In this paper, we propose a new scheme for mitigating the Jellyfish reorder attack which uses SHA-1 hash function modified with the help of regularly distributed pseudo random function to distribute a secret between source and destination. After the distribution of the secret key, the sender performs hashing on the sequence number of packets it sends and the obtained hashed value is sent along with those packets. The same hash function is used by the destination to verify whether the packets received are in out-of-order.

Fig. 2 shows the behavior of standard TCP congestion window of the sender with the attack and without the attack. The dotted line in the figure shows the congestion window behavior without attack and the solid line represents the congestion window behavior under the attack. The graph is plotted by considering simulation time (in sec) on X-axis and window size on Y-axis.

From fig. 2, we can observe that the congestion window of TCP sender with and without the attack. Without the attack the congestion window increases linearly but under the attack, the congestion window reduces consistently.

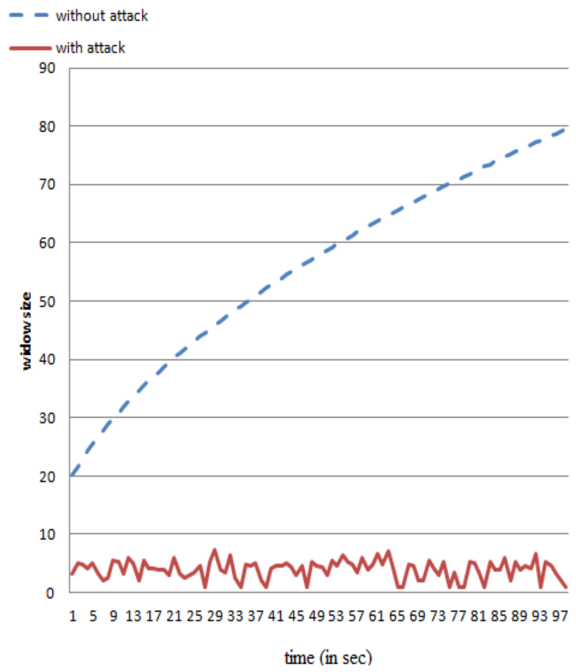

Figure 2. TCP congestion window under JF Reorder attack

III. RELATED WORKS

Imran Raza et al.[4] proposed a solution by introducing two new states R_F and R_{W+F+0} in TCP Reno State transition diagram and uses TCP timestamp options to avoid fast retransmit and timeout problems. The proposed solution prevents TCP Reno to reduce its congestion window size unnecessarily when retransmissions are due to persistent packet reordering attack rather than packet loss; but this cannot consider the malicious node or the malicious route which causes the reordering.

Das, A et al. [5] proposed a novel security scheme for wireless ad-hoc network based on shared information. They proposed to keep redundancy in the number of shares to withstand loss of some shares due to transmission loss as well as due the presence of network layer security threats, but this scheme does not fully mitigate the Jellyfish reorder attack and it does not identify the malicious route.

Tarun Banka et al. [6] proposed a new metric, reorder density function (RD), to represent the reordering of packets in a stream.

There are several derived metrics to monitor packet reordering in network. There are some existing metrics for determining the reordering such as Percentage of Late Packets, Mean Displacement of Packets and the Reorder entropy [7][8].

IV. PROPOSED SCHEME

In this section, we describe in detail the proposed solution for Jellyfish Reorder attack. The proposed solution uses the symmetric key cryptography for mitigating the attack. This cryptography shares a secret key between the source and the destination. This will be a difficult problem to share a secret key in adhoc networks as they don't have any Centralized authority for distributing the key with secure communication links.

A. *Key Distribution*

The secret key between the source and destination is distributed in such a way that the secret key is sent in a separate packet after the hashed value of the secret key reaches the destination. We know that, when source has to transfer data, it first initiates connection establishment to the destination. To do this, the source broadcasts RREQ packets until it reaches to the destination.

So, in this scheme, before the source broadcasts RREQ packets, it randomly generates a key. With this key, the source computes the hashed value using the SHA-1 hash function and appends this to the RREQ (Route REQuest) packet, by adding an extra field: hash value. After receiving the packet, the receiver stores the hashed value in a buffer until its key arrives. After a specified delay, the source node sends the corresponding secret key by distributing it to the destination node in a key distribution packet [9].

The format of the Key Distribution packet is similar to the RREQ packet, as shown in Fig. 3.

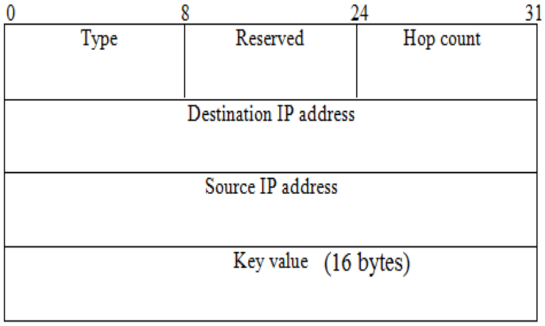

Figure 3. Key Distribution packet

As this packet is used for distributing the Secret key, it contains a field for Key value. The Type field is set to 5 as a new packet type added to the protocol. The Hop Count field indicates the number of hops from the source to the destination. The Destination IP Address field indicates the IP address of the destination, and the Source IP address field indicates the IP address of the source. The intermediate nodes broadcast the packet until it reaches the destination, similar to that of RREQ packet.

When the destination receives the key distribution packet, it performs hashing on the key value obtained from the packet and verifies it with the hashed value that is received along with the RREQ packet. If both the values are matched then the destination verifies that the key value is from the authorized source.

B. *Modified SHA-1*

As we mentioned that we used Modified Sha-1 for verification of packet reordering, the procedure below is the detailed explanation of the algorithm. We made modification to SHA-1(mSHA-1) hash function [10] using a pseudo-random function.

In the modified hashing function, instead of using the logical functions of SHA-1, we changed these logical functions to pseudo-random function. Due to non-repeating period and randomness of the pseudo-random function, the hash values generated will be unique [11].

According to this, the pseudo-random function used is: $F(w_i) = w_i*\sqrt{3}$, where w_i is the message divided into blocks. As $\sqrt{3}$ results in a non repeating period, the hash values generated will be unique. We also added the secret generated by the source and the function becomes

$$F(w_i) = w_i*\sqrt{3}*K \tag{1}$$

where K is the secret key. After using this pseudo random function the output value of hash function depends only on secret key and the input message. It means that only the holders of secret key can compute appropriate hash value for the message.

C. *Transmitting hash value*

Now, when the source transmits the TCP packets, it performs the keyed-hashing on the sequence number of packets that it sends at regular intervals. The hashed value is sent along with the packets by using a dummy TCP Helper packet that contains the hashed value.

The Helper packet contains a new header field for storing the hashed value.

The destination node receives the packets along with the Helper packet and it also calculates the hashed value on the sequence number of packets received and verifies it with the received hashed value.

If the hashed value doesn't match, then the route is treated as malicious and if misbehavior continues for at least $T_{reorder}$ seconds, a JF-reorder attack is detected and that route is blocked.

V. RESULTS AND ANALYSIS

A. *Simulation Environment*

We used NS-2 [12] network simulator to implement the attack and the proposed scheme. We implemented in three different scenarios. First scenario considers the normal case (idle), where no modifications are done, the second with the attack scenario, and the third with the proposed solution. We used AOMDV (Ad Hoc On-demand Multipath Distance Vector) routing protocol [13] as the test bed of our approach. The simulation parameters are given in table.1.

TABLE I. SIMULATION PARAMETERS

PARAMETER	VALUE
Number of nodes	20
Simulation time	15sec

Routing Protocol	AOMDV
Queue Type	Drop Tail
Packet Size	1500 bytes
Transport protocol	TCP
MAC Layer Protocol	802.11
Queue size	50

We adopted the following main performance metrics to evaluate the proposed scheme.

B. Performance Metrics

1) TCP Goodput

Good put is defined as the number of unique packets delivered to an end host in a given amount of time.

Fig. 4 shows the good put of the network.

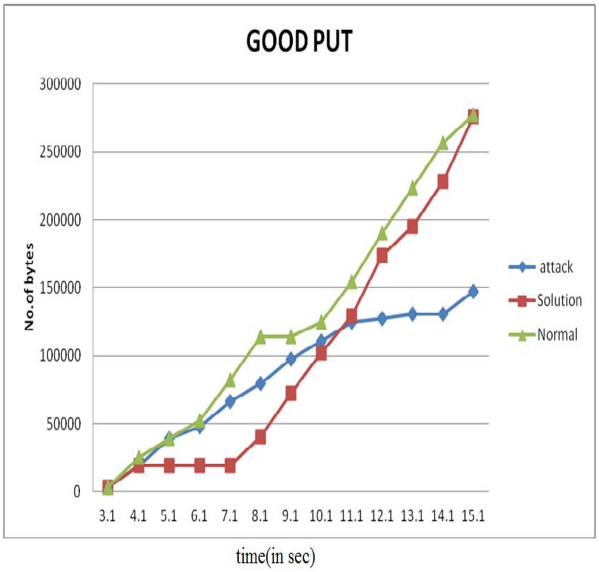

Figure 4. TCP GOOD PUT

The good put is calculated by removing the retransmitted packets from the total packets sent by the source to get unique packets. It reaches to the normal scenario with the proposed solution. Under the attack, the good put decreases as more false retransmitted packets are sent by the source at a particular time, but not the original packets that are to be delivered to the destination.

2) Congestion window

As this attack mainly targets the end-to-end congestion control mechanism of TCP, we considered Congestion window as one of the metrics for our evaluation.

Fig. 5 shows the congestion window of the sender.

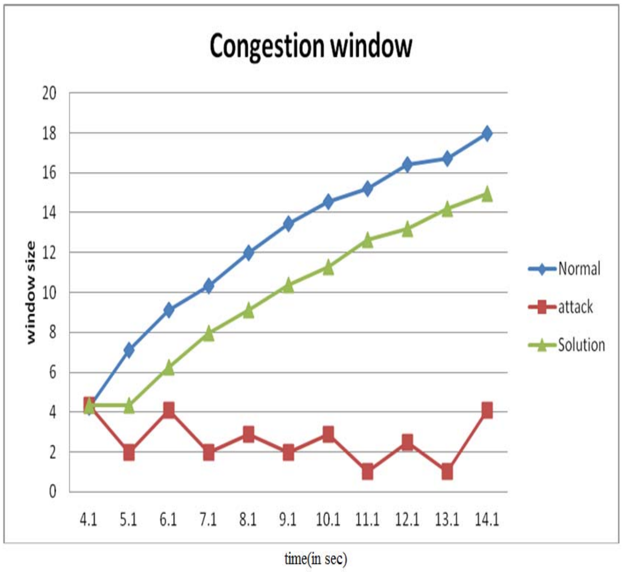

Figure 5. Congestion window

Under the attack, the congestion window decreases due to the occurrence of false retransmissions. Under the solution scenario, the congestion window decreases until the attack is detected and after that the window increases exponentially.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have discussed the persistent packet reordering attack and also intend to mitigate the attack in wireless adhoc networks. This attack obeys all the protocol rules and is difficult to detect. The proposed solution uses the cryptographic hashing solution and sequence numbers of packets sent by the sender to perform keyed hashing. By using the hashed value, the destination comes to know whether there exists a malicious node in the network and an appropriate action is taken on that malicious route, to mitigate the attack. The simulation results show that the performance of the network increases using the proposed solution and is effective in mitigating the attack. The future work includes extending this solution to the other variants of Jelly Fish attack in wireless adhoc networks.

REFERENCES

[1] Kannhavong, B.; Nakayama, H.; Nemoto, Y.; Kato, N.; Jamalipour, A.; "A survey of routing attacks in mobile ad hoc networks", IEEE transactions on wireless communications,vol.14,pp.85-91,Oct.2007.

[2] I. Aad and J.P. Hubaux, E.W. Knightly, "Impact of Denial of Service Attacks on Ad Hoc Networks", IEEE/ACM Transactions on Netwroking,vol.16,pp.791-802,Aug.2008.

[3] V. Anantharaman and R. Sivakumar, "A Microscopic Analysis of TCP performance Over Wireless Ad hoc Networks", Proceedings of the 2002 ACM SGMETRICS, Marina Del Rey, California, 2002.

[4] *Imran Raza, S. A. Hussain, Amjad Ali , Muhammad Hassan Raza, "Persistent Packet Reordering Attack in TCP Based Ad Hoc Wireless Networks", International Conference on Information and Emerging Technologies (ICIET), Karachi ,pp. 1-6, June 2010.*

[5] Das, A.;Basu,S.S.;Chaudhuri, A.; "A novel security scheme for wirel ess ad-hoc network", 2nd International Conference on Wireless Communication, Vehicular Technology, Information

Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), Chennai,pp.1-4, March 2011.

[6] T. Banka A. A. Bare A. P. Jayasumana, “Reorder Density (RD): A Formal, Comprehensive Metric for Packet Reordering”, Lecture Notes in Networking Technologies, Services, and Protocols, Springer, 2005.

[7] B. Ye, A. P. Jayasumana and N. Piratla, "On Monitoring of End-to-End Packet Reordering over the Internet," Proc.Int. Conference on Networking and Services (ICNS'06), Santa Clara, CA, July 2006.

[8] Banka, T., Bare, A. and Jayasumana, A., “Metrics for Degree of Reordering in Packet Sequences,” Proc. IEEE 27 Local Computer Networks Conf, Nov. 2001, pp. 333-342.

[9] Joe, I.; , "Secure routing with time-space cryptography for mobile ad-hoc networks," Military Communications Conference, 2005. MILCOM 2005. IEEE , vol., no., pp.1869-1874 Vol. 3, 17-20 Oct. 2005

[10] Abror Abduvaliyev, Sungyoung Lee and Yong-Koo Lee, “Modified SHA-1 hash function (mSHA-1)”, Proceeedings of ITC-CSCC2009, Jeju, Korea, pp. 1320-1324 , July 2009.

[11] Abduvaliev, A.; Sungyoung Lee; Young-Koo Lee; “Simple Hash Based Message Authentication Scheme for Wireless Sensor Networks”, 9th International Symposium on Communications and Information Technology,(ISCIT), Icheon, pp. 982-986, Sept. 2009.

[12] UC Berkeley and USC ISI, “The network simulator ns-2”, Part of the VINT project, Available from http://www.isi.edu/nsnam/ns, 1998

[13] Marina, M.K, and Das,S. R, “On-demand Multipath Distance Vector Routing for Ad Hoc Networks”, Proc. of 9th IEEE Int. Conf. On Network Protocols, pp.14-23 (2001).