Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 5, May 2016, pg.656 – 661

PACKET DROP DETECTION IN WIRELESS AD HOC NETWORKS

Shilpa Shetty A, Sannidhan M S

[1]Computer Science and Engineering, NMAMIT, Nitte, VTU, India
[2]Assistant Professor, Computer Science and Engineering, NMAMIT, Nitte, VTU, India
[1] shilpashettya@gmail.com; [2] sannidhan@nitte.edu.in

Abstract— Link error and malicious nodes are responsible for packet drop in wireless ad hoc networks. When we observe a sequence of packet losses, may be due to link errors only, or by both link errors and malicious drop, we should determine the malicious node. In the insider-attack case, whereby malicious nodes that are part of the route exploit their knowledge of the interactions to drop a small amount of packets critical to the network performance. To improve the detection accuracy, the correlations between lost packets is identified. Homomorphic linear authenticator (HLA) based public auditor is developed that allows the detector to verify the packet loss.

Keywords— Public Auditor, homomorphic linear authenticator, bit vector, nodes, encryption

I. INTRODUCTION

Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped, but also identify whether the drop is intentional or unintentional. Specifically, due to the open nature of wireless medium, a packet drop in the network could be caused by harsh channel conditions e.g., fading, noise, and interference, link errors, or by the insider attacker. In an open wireless environment, link errors are quite significant, and may not be significantly smaller than the packet dropping rate of the insider attacker. So, the insider attacker can camouflage under the background of harsh channel conditions. In this case, just by observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss.

Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped, but also identify whether the drop is intentional or unintentional. Due to the open nature of wireless medium, a packet drop in the network could

be caused by channel conditions link errors, or by the insider attacker. In an open wireless environment, link errors are quite significant, and may not be significantly smaller than the packet dropping rate of the insider attacker. So, the insider attacker can be because of the background of harsh channel conditions. In this case, just by observing the packet loss rate is not enough to accurately identify the exact cause of a packet loss.

II. RELATED WORK

The privacy preserving [1] discusses on the insider attack case, where nodes that are part of the route exploit their knowledge of the communication context to selectively drop a small amount of packets critical to network performance. The technique used here is privacy preserving: the public auditor should not be able to discern the content of a packet delivered on the route through the auditing information submitted by individual hops, no matter how many reports are sent to the public auditor.

The Sprite system [2] provides incentive for mobile nodes to cooperate and report actions honestly. The system does not require any tamper proof hardware at the node. Each node monitors the transmission of a neighbour to make sure that the neighbour forward other's traffic. If the neighbour does not forward others traffic then it is considered uncooperative and its uncooperative reputation is propagated throughout the network.

A Secure and Objective Reputation – based Incentive (SORI) [3] encourages packet forwarding and discipline selfish behaviour. The reputation of a node is quantified by objective measures; the propagation of reputation is efficiently secured by a one-way-hash-chain-based authentication scheme; and secure routing is in its place. The mobile nodes are typically constrained by power and computing resources, a selfish node may not be willing tin use its computing and energy resources to forward packets that are not directly beneficial to it, even though it expects others to forward packets on its behalf. This system also has a punishment scheme to penalize selfish nodes because the presence of selfish nodes can degrade the overall performance of the non-cooperative ad-hoc network.

The Audit-based misbehaviour detection [4] discusses on providing security by generating a group key and distributing it among all nodes. Gray hole is a malicious attack that makes the node to refuse forwarding certain packets and to drop the packets. The attacker selectively drops the packet originating from a single IP address or a range of IP addresses and forwards the remaining packets. It is detected and rectified using the sequenced queue based routing algorithm.

Privacy-preserving, public auditing for cloud [5] discusses on making users to use cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party public auditor (TPA) to check the integrity of data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerable user data privacy, and introduce no additional online burden to user. The paper proposes secure cloud storage supporting privacy-preserving public-auditing.

The detecting malicious packet dropping by observing traffic patterns [6] proposes an operationally viable approach to finding out where the loss occurred. Traffic can be severely disrupted by routers refusing to serve their advertised routes, announcing non-existent routes, or simply failing to withdraw failed routes, as a result of either malfunction or malice. The key idea behind detecting malicious packet is finding where the packet loss has occurred in the network using a protocol and maintaining log.

The attackers may disrupt packet forwarding i.e., the data plane of the network by drooping packets routed to it by neighbours. Authentication of routing protocol messages is not sufficient to prevent the disruption of routing.

In detection of malicious dropping [7], it is whether the packet losses are due to link errors only, or due to the combined effect of link errors and malicious packet drop. This paper proposes to exploit the correlations between the lost packets. A homo-morphic linear authenticator based public auditing is developed that allows the detector to verify the truthfulness of the packet loss information reported by the nodes.

The dynamic source routing protocol [8] is a simple and efficient routing protocol designed specifically for the use in multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring requiring no existing infrastructure or administration. Network nodes cooperate to forward packets for each other to allow communication over multiple "hops" between nodes not directly within wireless transmission range of one another. As the nodes in the network move about or join or leave the network, and as wireless transmission conditions such as sources of interference change, all routing is automatically determined and maintained by the DSR routing protocol.

Secure data collection in wireless sensor networks using randomized dispersive routes [9] discusses on the possible threats in wireless sensor networks like node failure, data security etc. This paper proposes the techniques to overcome the "black holes" that are formed due to compromised node and denial of service by using some routing mechanisms. A structure has been developed that generates randomized routes. Therefore, even though adversary or defender comes to know about the routing algorithm still he cannot pin point the routes in where each packet is traversed randomly.

III. SYSTEM ARCHITECTURE

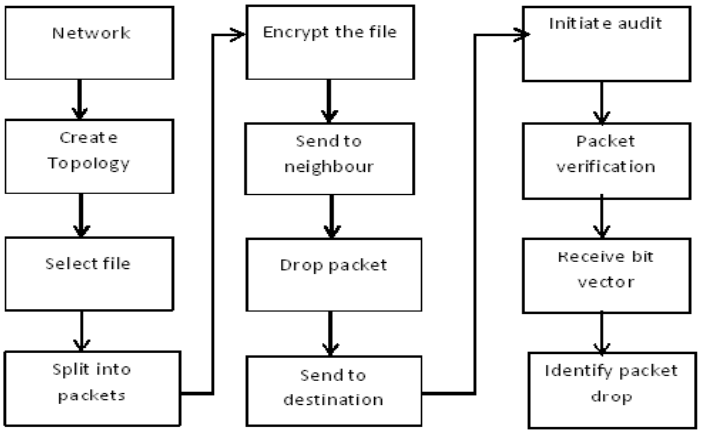

Fig 1 Architectural Diagram

We are using four modules

.First, in the network module we are first selecting the number of nodes we want to include. Then we link nodes to create a path for sending packets from source node to node used as destination. After the path is established, the user is asked to select a file

to transmit. The file is then split into required number of packets. After this the file is encrypted by using the cryptographic algorithm. The file can then be transmitted to the next node in the path that is established from source to destination.

The independent auditor works as follows. Once the packet reaches the destination, if source thinks that all packets have not reached destination then it can initiate the audit process by sending a request message to the auditor. On request the auditor sends a challenge to all the nodes and requests a vector from all the nodes in the path traversed by the remaining packets. This is a bit vector that contains information about the packets that have been received by all nodes. If a node has received a packet then the bit for that sequence number packet will be stored as one. If the node has not received the packet then it stores a zero for that packet. By collecting all such vectors from all the nodes the auditor starts the drop detection process. It now compares all the bit vectors with the vector that it has and then it is able to determine whether the packet has been dropped.

The set up stage executes after the path is established. Here initially the user is asked to select a file to transmit. The file is then split into required number of packets. After this the file is encrypted by using the cryptographic algorithm. The file can then be transmitted to the next node in the path that is established from source to destination. After the auditing process, if the file is verified as received without any error, then the file can be decrypted.

The last stage is packet loss detection where, on request by the source, the auditor sends a challenge to all the nodes and requests a vector from all the nodes in the path traversed by the remaining packets. This is a bit vector that contains information about the packets that have been received by all nodes. If a node has received a packet then the bit for that sequence number packet will be stored as one. If the node has not received the packet then it stores a zero for that packet. By collecting all such vectors from all the nodes the auditor starts the drop detection process. It now compares all the bit vectors with the vector that it has and then it is able to determine whether the packet has been dropped and which is that node that is responsible for the loss of packets.

IV. IMPLEMENTATION

Java is one of the high-level languages used for programming which was brought into picture during the third generation in the development of languages for developing programs. The basic concepts used in C and C ++ language are also used in Java.

Hardware Requirements

- Any Processor above 500 MHz.
- 128Mb RAM
- 15 Gb Hard disk
- Standard Mouse and keyboard
- VGA and HR Monitor

Software Requirements

- Windows OS
- JDK 1.5 or greater
- SQLyog

V. RESULTS

The application is installed on a lap top, the application will be visible as shown in Fig1.Once the path is established, the user is asked to select a file to transmit. After splitting into packets, the packets are encrypted and then transmitted to the next node in the path that is established from source to destination. Meanwhile, a malicious user node may drop the packet purposefully. So

the task is to determine node that has dropped the packet .Fig 2 shows how a file can be selected for transmission. Fig 3 shows the packet drop detection.

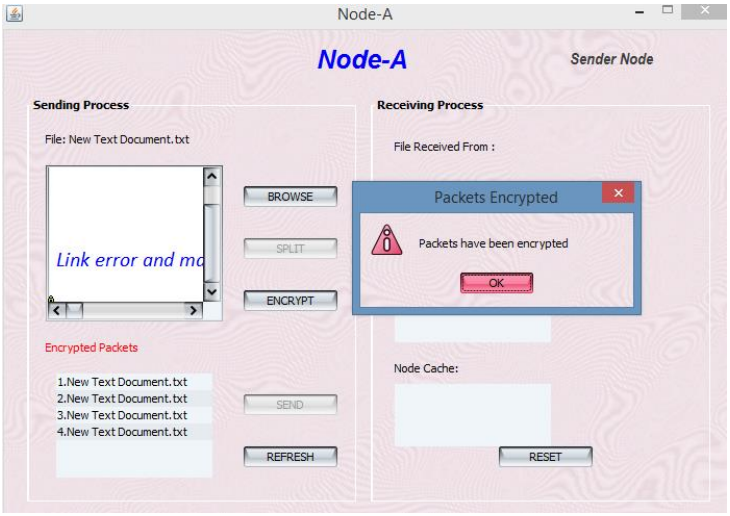

Fig 2:Selecting file

Fig 3:Packet loss detection

VI. CONCLUSION

This paper shows that it is possible to determine the packet loss that happens because of a malicious node in the network. Especially when the number of packets dropped is very few compared to the one that happens because of errors in the link through which the packets are being transmitted. So it is helpful to identify such nodes and then exclude these nodes from being part of the network configuration. And this technique has less computation at nodes.

REFERENCES

[1] T. Shu, S. Liu, and M. Krunz, "Privacy-preserving and truthful detection of packet dropping attacks in wireless ad hoc networks", in IEEE Transactions on mobile computing., 2015, pp. 813-827

[2] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks," in Proc. IEEE INFOCOM Conf., 2003, pp. 1987

[3] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation- based incentive scheme for ad hoc networks," in Proc. IEEE Wireless Commun. Netw. Conf., 2004, pp. 825–830.

[4] Y. Zhang, L. Lazos, and W. Kozma, "AMD: Audit-based misbehaviour detection in wireless ad hoc networks," IEEE Trans. Mobile Comput., PrePrint, Vol. 99, published online on 6 Sept. 2013.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. IEEE INFOCOM Conf., Mar. 2010, pp. 1–9.

[6] Julian Benadit P, Sharmila Baskara, Ramya Taimanessamy, 'Detecting malicious packet dropping using statistical traffic patterns', IJCSI International Journal of Computer Science Issues, No 2, May 2011

[7] R. Rao and G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," in Proc. IEEE GLOBECOM Conf., 2003, pp. 2957–2961.

[8] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks," in Ad Hoc Networking. Reading, MA, USA: Addison-Wesley, 2001, ch. 5, pp. 139–172.

[9]] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," IEEE Trans. Mobile Comput., vol. 9, no. 7, pp. 941–954, Jul. 2010.

