International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 9, November- 2012

Detection of Colluding Adversaries in a Packet Drop Attack on MANET

Shirina Samreen
Associate Professor, Dept. of Computer Science Engg.,
Guru Nanak Institutions Technical Campus,
Ibrahimpatnam, Ranga Reddy, A.P., India

Dr. G. Narasimha
Associate Professor, Dept. of Computer Science Engg.,
JNTUH College of Engineering,
Kukatpally, A.P., India

***Abstract*—A mobile ad hoc network is a spontaneous self-organized infrastructure-less network wherein the networking activities like routing and data transmission are carried on by the nodes themselves in a collaborative manner. However, since nodes are resource-constrained with limited battery power, few nodes may be selfish which expect services from other neighbouring nodes but refuse to provide any service to it's neighbours. More specifically, the selfish nodes drop the packets belonging to some other node instead of forwarding them to the next hop on the route. A number of mechanisms have been proposed to defend against packet drop attacks carried out by an individual malicious node. Such mechanisms are random audit based which cannot detect collaborative packet drop attack wherein the attack is carried out cooperatively by colluding adversaries for which the defense mechanism becomes still complicated. We propose a mechanism to detect colluding adversaries which collectively carry out packet drop attack.**

***Keywords*—Mobile Ad hoc Networks (MANETs), Colluding adversaries, Packet Drop Attack, Audit based detection.**

I. INTRODUCTION

Evolution of wireless networking and mobile computing hardware have resulted in wide spread usage of mobile ad hoc networks in many distributed applications. The infrastructure less property and the easy deployment along with the self-organizing nature makes them useful for many applications like military applications and fast response to disasters. Despite it's applicability to multiple applications, the MANET cannot be considered as an alternative to a wired network and it demands a lot of research on security issues. In a MANET, communication can be established among nodes equipped with wireless transceivers without the usage of any routers. In other words, nodes themselves act as routers as well as source and they depend on each other for forwarding packets from a source to a destination. The main problem of communication in a MANET results from the inconsistency of the nodes to transmit the packet to some destination. This inconsistency results from a number of factors: Firstly, each node's transmission range is limited and nodes are mobile. Hence the dynamic nature of the network may cause a node which forwarded the data packets for some source/destination pair at some point of time, not being able to do so at a later point of time due to mobility which may effect it's transmission range. Secondly, the limited battery power of the nodes may effect it's packet forwarding behaviour.

Apart from these factors, the inherent characteristics of a MANET may cause the security of communication to be compromised easily. A node's capability of promiscuous overhearing of neighbourhood nodes within it's transmission range may raise issues for the confidentiality of data packets. Unlike wired networks, there is no clear line of defense in a MANET like a firewall or gateway and every node is vulnerable to an attack. The overall performance of the network depends upon every node since nodes have to collaborate for all network activities. The malicious adversaries usually exploit this feature of cooperative participation of nodes in the routing activity to launch attacks.

Hence we need to design security primitives for routing and also for detecting any adversaries in the network which launch various attacks. A packet drop attack is one of the attacks wherein the adversary simply drops the packets without forwarding. This may be due to it's selfishness to preserve battery power or it might have been compromised by an external attacker. In this paper, we propose to investigate the collaborative packet drop attack which is a serious threat to the communication in MANET. Since MANETs are being used in a wide variety of applications involving data transmission, secure and robust data delivery to the destination has to be accomplished. A resource efficient and reactive approach to detect a packet drop attack is based on random audits on nodes for the behavioural proofs. It is resource efficient in the sense that it does not involve communication and computation overhead since it is triggered only when the destination senses a significant drop in the packet delivery ratio.

We propose to develop a new mechanism for detecting colluding adversaries which together carry out a packet drop attack. The REAct system is a reactive and resource

efficient approach for detecting a misbehaving node which carries out a packet drop attack individually. This approach fails in the presence of colluding adversaries as has been shown in [1]. The authors in [1] illustrate a colluding adversarial model under which REAct approach fails for which another approach based on hash calculation on the received packets for node behavioural proofs has been proposed. But this approach requires the source node to share a secret key with each intermediate node. We consider two adversarial models involving colluding adversaries for which we have proposed detection mechanisms. The first adversarial model is the one wherein the colluding adversaries are two non-consecutive nodes separated by innocent intermediate nodes. The second one involves colluding adversaries which are a set of consecutive nodes on the path from source to destination. Our approach is based on bloom filters used by REAct system as node behavioural proofs and does not require any secret to be shared between the source and the intermediate nodes.

The remainder of the paper is organized as follows: Section II presents the related work which discusses the various approaches to detect the packet drop attack, then we specifically consider the collaborative packet drop attack and discuss the various defense mechanisms. Section III presents the proposed approach under two adversarial models and the respective detection mechanisms along with the corresponding pseudo code. Finally section V presents the conclusion wherein we discuss the efficiency of the approach.

II. RELATED WORK

A. Detection of Packet drop Attack

A MANET environment consists of self-organized wireless nodes which form a multi-hop network and nodes have to collaborate to perform all network activities including the routing, forwarding of data packets which belong to other nodes. Since nodes are resource-constrained, they may not be motivated to expend their energy to help other nodes in data transmission which results in many packet drop attacks. A lot of research has been done for defense against such types of attacks. These mechanisms can be categorized into three as follows :

- Credit-based techniques
- Monitoring based techniques
- Acknowledgement based techniques.

The credit based techniques by Buttyan and Hubaux [2], [3] are based upon the usage of credits called nuggets that will be awarded for a node for packet forwarding. Two models have been proposed known as Packet Purse Model and Packet Trade Model. In both these models, each intermediate node receives nuggets for packet forwarding activity which it requires for transmitting it's own data packets. Hence every node intends to increase it's nugget count for which it performs packet forwarding for other nodes. Another approach known as Sprite proposed by Zhong et al [4] uses a central server reachable through internet called Credit Clearance service which either charges or credits the nodes for packet forwarding activity depending on whether they have provided the service to others or utilized the service from others. The drawback of these techniques is that, they need tamper-resistant hardware to prevent the nodes from modifying the credit-related information

Monitoring based techniques are based upon the promiscuous listening of neighbourhood by the wireless nodes which use the omni-propagation of wireless signals to keep track of the behaviour of their neighbours. Marti et al [5] proposed a mechanism that can be used with Dynamic source routing (DSR) protocol which includes two components namely watchdog and pathrater. The watchdog in each node monitors the behaviour of it's neighbours to see if they forward the packets to their next-hop neighbours. The information gathered by watchdog is used by the pathrater to rate the paths and the path which best avoids misbehaving nodes is chosen. Another approach called CONFIDANT [6] was proposed by Buchegger and Boudec which involves a monitor on each node keeping track of forwarding activity of neighbours and propagation of any suspicious behaviour to reputation system which rates the suspicion based on some factors. This information may further be passed on to path manager based on rating of suspicion which modifies the route cache. Finally, trust manager propagates alarm messages to all the nodes about the suspected node. Michiardi et al [8] proposed another mechanism called CORE which is a reputation based mechanism wherein reputation metrics are assigned to the nodes based upon observations made by neighbours, positive reports and task specific behaviour. The drawback of both these approaches is that, they are based upon promiscuous overhearing which is energy consuming and may raise false alarms in the presence of receiver collisions and ambiguous collisions. It may be difficult to use in multi-channel networks which use directional antennas since nodes may be engaged in parallel transmissions in orthogonal channels.

Acknowledgement based techniques require the nodes forwarding the data packets to send acknowledgements to their multi-hop upstream neighbours in the reverse direction of data traffic. An example of this scheme is 2ACK technique proposed by kejun Liu [9] wherein the misbehaviour is detected based upon number of packets which missed the acknowledgments. Padmanabham et al [10] proposed a technique based on traceroute wherein the source probes the route by sending pilot packets that are indistinguishable from data packets. The drawback of these techniques is that they are proactive in nature which leads to

lot of network traffic created in the form of acknowledgement packets.

B. *Detection of Collaborative Packet Drop Attack*

In the presence of colluding adversaries, there exists a continous threat of collaborative attacks on MANETs and a number of mechanisms have been designed for the defense against these attacks. Collusion attacks are possible upon routing as well as key management. In [11], a group key management model to protect against collusive attack has been developed to distribute the keys in such a way that probability of entire network being compromised is minimum. In [12], the optimized link state routing protocol has been analyzed against a collusive attack model wherein the proposed technique detects the attack by utilizing the information from downstream neighbours present at two hops.

Collaborative intrusion detection systems have been designed in [13] which assume a clique or a cluster network structure. Another approach involves certain ideas borrowed from immune systems for the collaborative detection of adversaries [14]. Intrusion detection system called as honesty based IDS which makes collaborative decisions based upon multiple threshold values including rewards and penalties for packet forwarding has been proposed in [15].

A mechanism to detect Byzantine behaviours during packet forwarding has been proposed in [16]. The destination sends the feedback to the source whenever significant drop in packet delivery ratio is found. The source then performs binary search based query procedure to locate the faulty link in the path. This method provides protection against individual as well as collusive Byzantine behaviours.

Our approach is based on the REAct system which can be used to locate individual misbehaving nodes that perform packet drop attack. The working of REAct system is as follows: Assume that data transmission is going on between source node S and destination node D through a path (S, n_1,n_2,…..n_i,…D). Whenever the destination D senses a significant packet drop, it sends a feedback to the source S. The source then detects the misbehaving node in the path from S to D and eliminates it from the routing path. The REAct system assumes that there exists atleast two node disjoint paths for every pair of nodes in the network . Also, the source knows the identity of every intermediate node on the path from S to D and a pair wise key is used to protect the communication. The source chooses a random intermediate node n_i in the path and checks to see if it receives all the packets from it's upstream neighbour . For this, S sends an audit request packet to n_i through a path which is other than (S, n_1, n_2,..n_i) which specifies the packet sequence numbers of those packets based on which behavioural proof has to be generated by n_i. The node n_i constructs the bloom filter based on the contents of these packets which acts as a behavioural proof. The main idea of REAct systems behind the usage of bloom filters is that, it occupies much lesser storage when compared with the total length of selected packets and hence the communication overhead on the audited node is reduced. After the bloom filter is generated, n_i sends it to S. The source S will construct it's own bloom filter and compares it with the one received from n_i. If they do not match, S understands that node n_i is unable to receive all packets from it's previous hop and packets are being dropped before they reach node n_i. Hence the misbehaving node is present in the path segment from S to n_i. If they match, then S understands that node n_i received all the packets from it's previous hop and hence the misbehaving node is in the path segment from n_i to D. The auditing continues in the next step wherein the node for auditing is chosen from a smaller suspicious path segment (either S to n_i or n_i to D) obtained from the previous step. This process of using binary search approach to reduce the length of suspicious path segment in every step is repeated until the path segment consists of only two suspicious nodes. The corresponding link is then removed from the path a new route is discovered.

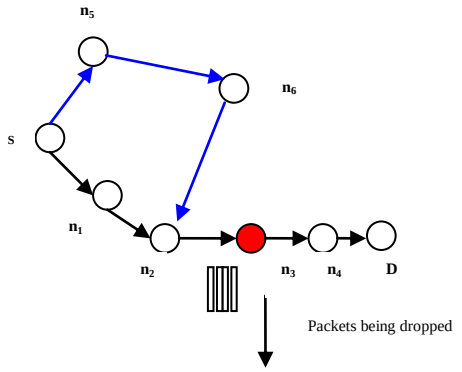
The main drawback of REAct system is that, it can detect individual misbehaving nodes which drop packets but when this attack is carried on by colluding adversaries, the technique fails. The main reason behind it's failure is the assumption that a node can successfully generate behavioural proofs only when it receives all packets.

In the figure below we illustrate an example of the REAct approach. The source node S selects a random node on the path from S to D for auditing (say n_2). The node n_2 will generate the behavioural proof in the form of a bloom filter which is sent to S. Since n_2 received all it's packets from it's upstream neighbour n_1, it's bloom filter matches to that of S. Hence S concludes that the misbehaving node is in the path segment from n_2 to D. The same technique of selecting a random node for auditing from the suspicious path segment is repeated and the length of the suspicious path segment keeps reducing in each step until the length reduces to just two nodes. At this point of time, the link n_3-n_4 becomes the suspicious link and at this point of time, based on the bloom filter of n_3 it can be concluded that n_3 receives all packets but drops them without forwarding it to n_4. Hence node n_3 is concluded as the misbehaving node.

In all the figures below, we use the following colouring representation:

- Blue coloured path indicates audit path.
- Black coloured path indicates the routing path used for data transmission from S to D.

- Red colored nodes indicate the misbehaving nodes and red colored path indicates communication among malicious nodes through side channel



An approach to defend against collaborative packet drop attack was proposed in [17] but this approach protects against only one type of adversarial model wherein two colluding adversaries are non-consecutive nodes in the path from S to D separated by intermediate innocent nodes. Also the approach requires the source to share a secret with every intermediate node on the path from S to D. The approach also does not protect against a second type of adversarial model which is a step ahead compared to the former adversarial model. In this second type of adversarial model, all intermediate nodes between colluding adversaries are also compromised and hence we have a set of consecutive nodes on the path which act as colluding adversaries.

Our approach provides a mechanism which does not require the source to share a secret with every intermediate node. It also addresses the second adversarial model wherein a set of consecutive nodes on the path act as colluding adversaries. To address the second adversarial model, our approach depends upon the promiscuous overhearing of transmissions at a node by the neighbours.

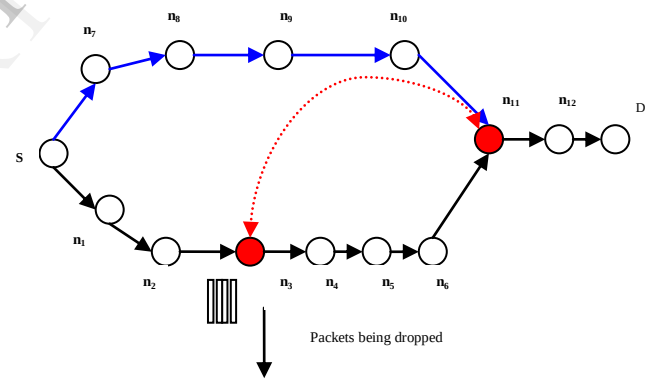
III. PROPOSED APPROACH

In this section, we describe the working of our approach under the two different adversarial models. Our approach makes the following assumptions. We assume that every pair of nodes has atleast two node disjoint paths. The source node knows the identity of every intermediate node on the path from the source to destination which can be used by a source routing protocol such as dynamic source routing (DSR). To address the second adversarial model, our approach assumes that the source maintains the list of neighbours for each intermediate node on the path and each node is supposed to maintain information about the packet forwarding behaviour of it's neighbours in the form of number of packets overheard along with the time stamp. Whenever there is a significant drop in the packet delivery ratio, the destination sends a feedback to the source which triggers an audit by the source.

Multiple malicious nodes exist in our adversarial models and these nodes can communicate through a side channel. They share all their secret keys and act as colluding adversaries to carry out a packet drop attack. The nodes can impersonate each other and collaborate such that one of them drops the packets and the remaining nodes help it to avoid detection.

A. Adversarial Model 1:

Two non-consecutive nodes n_i and n_k on the path from source S to destination D are colluding adversaries which are separated by non-adversarial nodes. The node n_i receives all packets from it's predecessor on the path but it drops all packets without forwarding it to it's successor on the path and hence no nodes after n_i receive any packet. If the node n_k is chosen for auditing, it will communicate with the node n_i the audit request packet specifying the sequence numbers of the packets. The node n_i generates the bloom filter and forwards it to node n_k. The node n_k sends back the bloom filter to the source S along with it's signature. If this bloom filter matches with that of source S, then S assumes that the misbehaving node is in the path segment from n_k to D. An example of the above adversarial model is as follows:



The above figure illustrates the colluding packet drop attack. In the path from S to D, there are two colluding adversaries n_3 and n_{11} which together carry out the attack by communicating through a side channel. The node n_3 drops all packets without forwarding it to it's next hop and hence no node after n_3 in the path from S to D receives any packets. If node n_{11} is chosen by S for auditing, it sends the audit request packet to n_3 which generates the bloom filter and sends it back to n_{11}. The node n_{11} sends it to S after signing it resulting in S assuming that node n_{11} has received all packets. Hence S will choose the wrong path segment for auditing . The situation becomes even more complicated if S audits n_3, n_5 and n_{11}, the behavioural

proofs will be conflicting since n_5 is a non-malicious node and it's bloom filter does not match with that of S where as n_{11}'s bloom filter matches even though it does not receive any packets. Hence it becomes difficult to identify the adversary based on conflicting results.

The above adversarial model can be countered through the modules COLL ATTCK DEFNS and FIND COLL ADV.
The module COLL ATTCK DEFNS works as follows: Let n_k be one of the colluding adversary and the random node chosen for auditing, then it first takes the bloom filter of n_k and compares with the bloom filter of S. Then it checks the bloom filter of predecessor n_{k-1} with that of S. If it does not match, then it implies that node n_{k-1} has not received all the packets from it's upstream neighbours but node n_k claims to receive them which is not possible without n_{k-1} forwarding it. Hence we can conclude that a collaborative packet drop attack is happening through the help of some upstream malicious node n_i. Hence we need to locate that node in the path segment from S to n_{k-2} for which we use the FIND COLL ADV module to locate that node whose bloom filter matches to that of S and such a node is the adversary.

The module FIND COLL ADV works as follows: In the path segment (n_i, n_{i+1}, n_{i+2},…., n_{k-1}, n_k) , n_i and n_k are colluding adversaries. After finding that node n_k is misbehaving and working in collaboration with another malicious node to perform the packet drop attack, we need to locate the other adversary n_i. The path segment S to n_{k-1} is considered and a random node n_x is chosen for auditing. If the bloom filter of n_x matches then, we check the bloom filter of it's successor n_{x+1}. If that also matches, it implies that the adversary is downstream to n_x, the path segment n_{x+1} to n_{k-1} is considered. If the bloom filter of n_x matches but the bloom filter of it's successor n_{x+1} does not match then we arrive at the conclusion that n_x is the colluding adversary. If the bloom filter of n_x does not match, then the adversary is upstream to n_x and the path segment S to n_{x-1} is considered.

COLL ATTCK DEFNS (Source S, Destination D)

S sends random audit packet to node n_i
Node n_i creates a bloom filter B_i and sends it to S
S sends the same audit packet to the predecessor node n_{i-1}
Node n_{i-1} creates a bloom filter B_{i-1} and sends it to S.
S checks for match with B_i and B_{i-1}

If B_i matches and B_{i-1} matches then
    Suspicious path segment reduced to n_i-D
    COLL ATTCK DEFNS (n_i, D)
EndIf
If B_i matches but B_{i-1} does not match then
    Colluding adversary present in path segment S- n_{i-2}
    FIND COLL ADV(S, n_{i-1})
EndIf
If B_i does not match but B_{i-1} match then
    Blacklist n_{i-1} as it is carrying out packet drop attack
EndIf

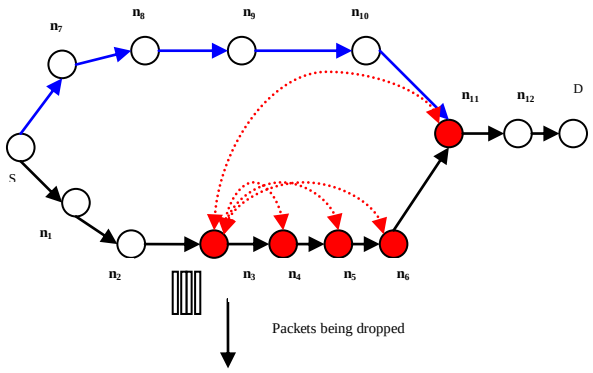
If B_i does not match and B_{i-1} does not match then
    Suspicious path segment reduced to S- n_{i-1}
    COLL ATTCK DEFNS (S, n_{i-1})
EndIf

FIND COLL ADV (Node A, Node B)

S sends random audit packet to node n_x
Node n_x creates a bloom filter B_x and sends it to S
S checks for match with it's own bloom filter
If B_x does not match bloom filter of S then
    Colluding adversary present upstream to n_x
    Suspicious path segment reduced to A-n_{x-1}
    FIND COLL ADV (A, n_{x-1})
EndIf
If B_x matches the bloom filter of S then
    Check the bloom filter B_{x+1} of the successor n_{x+1}
    If B_{x+1} also matches the bloom filter of S then
        Colluding adversary present downstream to n_x
        Suspicious path segment reduced to n_{x+1}-B
        FIND COLL ADV (n_{x+1} , B)
    EndIf
    If B_{x+1} does not match the bloom filter of S then
        Blacklist node n_x as the colluding adversary
    EndIf
EndIf

B. Adversarial Model 2:

A set of consecutive nodes n_i, n_{i+1}, n_{i+2},…. n_k on the path from source S to destination D are acting as colluding adversaries. In this scenario, the first node in the set receives the packets from it's predecessor node n_{i-1} in the path but it drops them without forwarding them to it's successor n_{i+1} in the path. The node n_i buffers all these packets and whenever source S sends the audit request packet to any node n_x in the set (n_{i+1}, n_{i+2}, …. n_k) on the path , the node simply communicates with node n_i the audit request packet specifying the packet sequence numbers obtained from source S. The node n_i then constructs the bloom filter and sends it to node n_x which sends it back to the source S along with it's signature. The source S verifies the received bloom filter with it's own bloom filter. If a match occurs, then S assumes that n_x has received all packets. In this way, any node n_x in the set (n_{i+1}, n_{i+2}, …. n_k) being audited will obtain the bloom filter from n_i and S assumes that misbehaving node is in the path segment from n_{x+1} to D. But the fact is, no node in the path after node n_i receives the packets since n_i drops all the packets. An example of the above adversarial model is as follows:



In the path from S to D, the nodes n_3, n_4, n_5, n_6 and n_{11} have been compromised and act as colluding adversaries. All these nodes work in cooperation to allow n_3 to perform packet dropping and also escape from being detected. No node after n_3 in the path from S to D receives any packets and if the random node chosen for auditing is any node from the set of colluding adversaries, they simply obtain the bloom filter from node n_3 and send it back to S which results in S considering the wrong path segment as suspicious.

In this case, the bloom filters of n_k as well as n_{k-1} will match S. We go for using the promiscuous listening mode which requires that each node maintains the details of the forwarding behaviour of it's neighbouring nodes. Each node maintains the information about the packets which it hears from it's neighbour by incorporating the id of the neighbour node and also the timestamp at which the packet was overheard. Whenever the random audit request packet is sent from source S, it also includes along with packet sequence numbers, the time period specified in the form of start timestamp and end timestamp during which the packets might have been received from upstream neighbours and forwarded to downstream neighbours on the path from S to D. Whenever audit request packet is sent to a node, it first generates the bloom filter which is sent to source S. If it matches, there exists a possibility of colluding attack involving consecutive neighbouring nodes. So S uses the information from neighbouring nodes about the packet overhearing statistics. If the neighbour of the node being audited reports that, no transmission is overheard within the time period as specified by audit request packet but the bloom filter matches, then that node is malicious which is getting the bloom filter from one of it's upstream neighbour which is the colluding adversary.

The modules COLL ATTCK DEFNS MODL2 and PROCESS PATHSEG are collectively used to counter this adversarial model. The working of COLL ATTCK DEFNS MODL2 is as follows: It first chooses a random node n_i in the path segment from S to D for auditing. Then it checks the bloom filters of n_i and predecessor n_{i-1}. When both of them match with the bloom filter of S, it checks for the packet overhearing statistics from the neighbourhood. If no packet overheard at n_i and n_{i-1}, then set of consecutive colluding adversaries are present upstream n_i which have to be located. This is done the module PROCESS PATHSEG. If no packet overheard at n_i but packet overheard at n_{i-1}, then nodes n_i and n_{i-1} are colluding adversaries and they are blacklisted. If packet overheard at n_i and also at n_{i-1}, then colluding adversaries are present downstream n_i and suspicious path segment is reduced to n_i -D.

The module PROCESS PATHSEG (A, B) works as follows: A random node n_i is chosen from the path segment A-B and the bloom filter is checked with that of S. If it matches and according to neighbours of n_i, if no packet overheard at n_i, then we blacklist all nodes from n_i to B as consecutive colluding adversaries. Also, there are more colluding adversaries upstream n_i and we further process the path segment from A- n_i. If bloom filter of n_i matches and also packet is overheard at node n_i, then n_i is starting node in the set of consecutive colluding adversaries.

COLL ATTCK DEFNS MODL2 (Source S, Destination D)

S sends random audit packet to node n_i
Node n_i creates a bloom filter B_i and sends it to S
S sends the same audit packet to the predecessor node n_{i-1}
Node n_{i-1} creates a bloom filter B_{i-1} and sends it to S
S checks for match with B_i and B_{i-1}

If B_i matches and B_{i-1} matches then
    Check for packet overhearing statistics from neighbour of n_i and neighbour of n_{i-1}
    If no packet overheard at n_i and no packet overheard at n_{i-1} then
        Colluding adversaries present as consecutive nodes in the path segment S- n_i
        Blacklist node n_i and n_{i-1}
        PROCESS PATHSEG (S, n_{i-2})
    EndIf
    If no packet overheard at n_i and packet overheard at n_{i-1} then
        Colluding adversaries are n_i and n_{i-1}
        Blacklist nodes n_i and n_{i-1}
    EndIf
    If packet overheard at n_i and packet overheard at n_{i-1} then
        No adversaries in the path segment S- n_i
        Suspicious path segment reduced to n_i -D
        COLL ATTCK DEFNS MODL2 (n_i, D)
    EndIf
EndIf

PROCESS PATHSEG (Node A, Node B)

Choose a random node n_i and send the audit request packet
Collect the packet overhearing statistics from n_i's neighbour
Node n_i generates the bloom filter B_i

If B_i matches and no packet overheard at n_i then
Blacklist node n_i and all nodes in the path segment from n_i -B
Consecutive Colluding adversaries existing upstream to n_i
PROCESS PATHSEG (A, n_i)
EndIf
If B_i matches and packet overheard at n_i then
n_i marks the starting node in the set of consecutive colluding adversaries
Blacklist node n_i
EndIf

IV. CONCLUSION

Our proposed mechanism efficiently detects the colluding adversaries without the need of having the source node share a secret with every intermediate node unlike the approach proposed in [17]. Apart from this, it detects the colluding adversaries under two adversarial models one of which involves a set of consecutive nodes acting as colluding adversaries. For the second adversarial model, it depends upon promiscuous overhearing of neighbourhood which has it's own shortcomings in the presence of collisions. We plan to simulate our proposed approach under the above mentioned adversarial models employing the ns-2 network simulator. We also plan to address the shortcoming in the approach used for the second adversarial model which results due to promiscuous overhearing in our future work.

REFERENCES

[1] W. Kozma, and L. Lazos, "REAct: resource-efficient accountability for node misbehavior in ad hoc networks based on random audits," in Proceedings of the Second ACM Conference on Wireless Network Security (WiSec), pp. 103-110, 2009.
[2] L. Buttyán, and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," Mobile Networks and Applications, 8(5), pp. 579-592, 2003.
[3] M. Jakobsson, J.-P. Hubaux, and L. Buttyan, "A micropayment scheme encouraging collaboration in multi-hop cellular networks," in Financial Crypto, 2003.
[4] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple cheat-proof, credit-based system for mobile ad-hoc networks," in IEEE INFOCOM, pp. 1987-1997, 2003.
[5] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings of the 6th Annual international Conference on Mobile Computing and Networking (MobiCom), pp. 255-265, 2000.
[6] S. Buchegger and J.-Y. L. Boudec, "Self-policing mobile ad-hoc networks by reputation systems," IEEE communications Magazine, pp. 101-107, 2005.
[8] P. Michiardi, and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in Proceedings of IFIP Joint Working Conference on Communications and Multimedia Security, pp.107-121, 2002.
[9] K. Liu, J. Deng, P. Varshney, K. Balakrishnan, "An Acknowledgment Based Approach for the Detection of Routing Misbehavior in MANETs," IEEE Transactions on Mobile Computing, 6(5), pp. 536550, 2007.
[10] V. Padmanabhan, D. Simon, "Secure traceroute to detect faulty or malicious routing," ACM SIGCOMM Computer Communication Review, 33(1), pp. 77-82, 2003.
[11] M. Younis, K. Ghumman, M. Eltoweissy, "Key management in wireless ad hoc networks: collusion analysis and prevention," in IEEE Performance, Computing, and Communications Conference (IPCCC), pp. 199- 203, 2005.
[12] B. Kannhavong, H. Nakayama, A. Jamalipour, "A Collusion Attack Against OLSR-based Mobile Ad Hoc Networks," in IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-5, 2006.
[13] N. Marchang, and R. Datta, "Collaborative techniques for intrusion detection in mobile ad-hoc networks," Ad Hoc Netw. 6(4), pp. 508-523, 2008.
[14] K. Yeom and J. Park, "An immune system inspired approach of collaborative intrusion detection system using mobile agents in wireless ad hoc networks", in International conference of Computational intelligence and security, 2005.
[15] P. Sen, N. Chaki, R. Chaki, "HIDS: Honesty-Rate Based Collaborative Intrusion Detection System for Mobile Ad-Hoc Networks," Computer Information Systems and Industrial Management Applications (CISIM), pp.121-126, 2008.
[16] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," ACM Trans. Inf. Syst. Secur. 10(4), 1-35, 2008.
[17] Weichao Wang Bharat Bhargava Mark Linderman "Defending against collaborative packet drop attack"