Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Secure Protocol for Leader Election and Intrusion Detection in MANET

Kalaivani.R
Computer Science and Engg,
Adhiyamaan college of Engg,India

RamyaDorai.D
Computer Science and Engg,
Adhiyamaan college of EnggIndia

Abstract— ***As the nodes in MANET are battery limited, one node per cluster is elected as leader to run IDS for all the nodes in the cluster. Leader is elected based on the residual energy of each node that is sufficient to run IDS. Too much of resources are wasted for the implementation of intrusion detection scheme for every node. Hence nodes are grouped into cluster and cluster head is elect to serve other node in network, where as selfish node with maximum resources are not nominated for cluster head selection, because of self interest to save its own power. Nodes are provided incentives in the leader election process by VCG mechanism for preventing the nodes from exhibiting the selfish behaviour.1) To ensure security and to detect the intrusion in Mobile Ad hoc networks select a leader from the 1hop cluster as cluster head contains most resource.2)To avoid the issues arise due to optimal collection of leader and performance overhead, a solution is Mechanism based design theory.3)The solution provides nodes with incentives in the form of reputations to encourage nodes in honestly participating in the election process.***

Keywords— ***leader election, intrusion detection system, mechanism design, MANET security.***

I. INTRODUCTION

Mobile ad hoc networks usually consist of mobile battery operated computing devices that communicate over the wireless medium. While the processing capacity and the memory space of computing devices increase at a very fast speed, the battery technique lags far behind. Therefore, it is critical to derive energy conservation scheme to increase their device and network operation time. This is very inefficient in terms of resource consumption since mobile nodes are energy limited. To overcome this problem, a common approach is to divide the MANET into a set of one hop clusters where each node belongs to at least one cluster. The nodes in each cluster elect a leader node (cluster head) to serve as the IDS for the entire cluster. The leader-IDS election process can be either random or based on the connectivity. Both approaches aim to reduce the overall resource consumption of IDSs in the network. However, we notice that nodes usually have different remaining resources at any given time, which should be taken into account by an election scheme.

The connectivity index-based approach elects a node with a high degree of connectivity even though the node may have little resources left. With both election schemes, some nodes will die faster than others, leading to a loss in connectivity and potentially the partition of network. Although it is clearly desirable to balance the resource consumption of IDSs among nodes, this objective is difficult to achieve since the resource level is the private information of a node. Unless sufficient incentives are provided, nodes might misbehave by acting selfishly and lying about their resources level to not consume their resources for serving others while receiving others services. Moreover, even when all nodes can truthfully reveal their resource levels, it remains a challenging issue to elect an optimal collection of leaders to balance the overall resource consumption without flooding the network. If the transmission range of the node increases, then it must have the highest transmission power in order to retain the signal level to the larger distance. It is enough for each node to have transmission range in order to reach its extreme member in the cluster. By doing so required transmission power gets reduced and energy efficiency also improved which in turn increases the lifetime of the node in the network.

II. LEADER ELECTION ALGORITHM FOR MANET

Leader election algorithm can accommodate arbitrary (possibly concurrent) topological changes and therefore well-suited for use in mobile ad hoc networks. The algorithm is based on finding an extreme and uses diffusing computations for this purpose. The algorithm is weakly self-stabilizing and terminating, and present a proof of correctness using linear-time temporal logic Leader election algorithms find many applications in both wired and wireless distributed systems.

A mobile ad hoc network is a collection of wireless, mobile devices that make communication possible by routing packets to one another. Each node has a limited transmission radius and can communicate directly with the "neighboring" nodes that fall within this radius. Communication with other nodes is made possible by routing packets through its neighboring nodes. Since nodes are mobile, the network topology can change as nodes move in and out of transmission range of one another. Recently, there has been considerable interest in using leader election algorithms in wireless environments for key distribution, routing coordination, sensor coordination, and general control. Here, node mobility may result in frequent leader election, making the process a critical component of system operation.

Designing distributed algorithms for such dynamically changing networks is a very challenging task. The classical definition of the leader election problem is to eventually elect a unique leader from among the nodes in a network.

[1) First modification arises from the fact that in many situations, it may be desirable to elect a leader with some system-related characteristic rather than simply electing a "random" leader. Leader election based on such an ordering among nodes fits well with the class of leader election algorithms that are known as "extreme-finding" leader-election algorithms.

2) Second modification is motivated by the need to accommodate frequent topology changes - changes that can occur during the leader election process itself. Network partitions can form due to node movement; multiple partitions can also merge into a single connected component.

It is important to realize that it is impossible to guarantee a unique leader at all times. When a network becomes partitioned, a component will be without a leader until the leader-election process terminates. Similarly, when components merge together there will temporarily be two leaders in the merged component. Thus, the modified problem definition requires that eventually every connected component has a unique leader. Informally, the algorithm operates as follows. Nodes periodically poll their leader. When a node is disconnected from its leader, the node detecting event starts a fresh diffusing computation to determine the new leader.

Leader Election Algorithm

The algorithm uses three messages.

1) Election:

Election messages are used to "grow" the spanning tree. Upon detecting leader departure, the source node, s, will start a diffusing computation by sending an Election message to all its immediate neighbors, denoted by the set. Each node, i, other than the source, will designate the neighbor from which it first receives an Election message as its parent in the spanning tree. The parent of node i is denoted by the variable. Upon setting its parent pointer, node i will propagate the received Election message to all its neighboring nodes (children) except its parent, i.e., the set of nodes and may receive Election messages from multiple neighbors, but will have only one parent.

2) ACK:

When node receives an Election message from a neighbor that is not its parent, it immediately responds with an Ack message. Node will not return immediately an ACK message to its parent. Instead, node will maintain a "pending ACK" for its parent, which it will send only after it has received an ACK from all of its children. As we will see shortly, the ACK message sent by to its parent will contain leader-election information based on the ACK messages have received from its children. Once the spanning tree is completely grown via propagated Election messages, the spanning tree starts "shrinking" back towards the source. Once the source node for a computation has received ACKs from all of its children, it then broadcasts a Leader message to all nodes announcing the identity of the leader.

3) Leader:

Once the source node for a computation has received ACKs from all of its children, it then broadcasts a Leader message to all nodes announcing the identity of the leader [11].

III. INTRUSION DETECTION SYSTEM (IDS)

Many historical events have shown that intrusion prevention techniques alone, such as encryption and authentication are not sufficient. As the system become more complex, there are also more weaknesses, which lead to more security problems. Intrusion detection can be used as a second wall of defense to protect the network from such problems.

If the intrusion is detected, a response can be initiated to prevent or minimize damage to the system. Some assumptions are made in order for intrusion detection systems to work. The assumption is that user and program activities are observable. The second assumption, which is more important, is that normal and intrusive activities must have distinct behaviors, as intrusion detection must capture and analyze system activity to determine if the system is under attack. Intrusion detection can be classified based on audit data as either host- based or network-based.

A network-based IDS captures and analyzes packets from network traffic while a host-based IDS uses operating system or application logs in its analysis. Based on detection techniques, IDS can also be classified into three categories. Anomaly detection systems: The normal profiles (or normal behaviors) of users are kept in the system. Misuse detection systems: The system keeps patterns (or signatures) of known attacks and uses them to compare with the captured data.

Any matched pattern is treated as an intrusion. Like a virus detection system, it cannot detect new kinds of attacks. Specification-based detection: The system defines a set of constraints that describe the correct operation of a program or protocol. Then it monitors the execution of the program with respect to the defined constraints.

Architectures for IDS in MANETs

The network infrastructures that MANETs can be configured to are either °at or multi-layer, depending on the applications. Therefore, the optimal IDS architecture for a MANET may depend on the network infrastructure itself. In a network infrastructure, all nodes are considered equal, thus it may be suitable for applications such as virtual classrooms or conferences. On the contrary, some nodes are considered different in the multi-layered network infrastructure. Nodes

may be partitioned into clusters with one cluster head for each cluster. To communicate within the cluster, nodes can communicate directly.

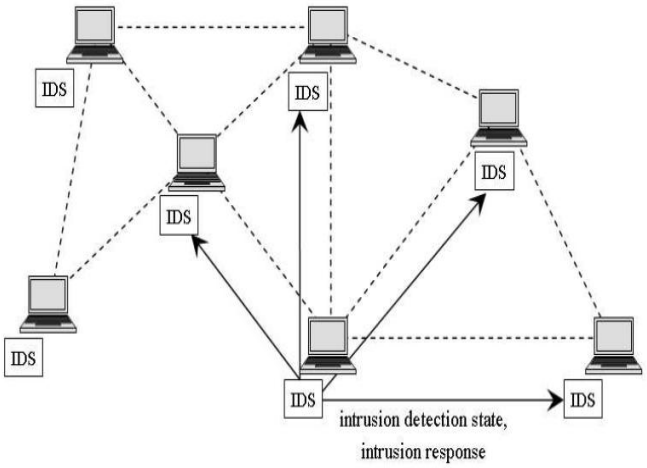

Fig 1 .Architectures for IDS in MANETs

Mechanism Design

The mechanism design problem is to define a game (i.e., its rules and payoff functions) in such a way that the outcome of the game played by independent agents according to the rules set by the mechanism designer will be the desired outcome, which is called the social optimum. In other words, the game should be designed in such a way that choosing a strategy that result in the social optimum is a dominant strategy for each player, where dominant means that no player has an incentive to unilaterally deviate from the strategy. Generally, any game will result in all players playing dominant strategies and the resulting state is called dominant-strategy equilibrium. The goal of a mechanism designer is to define rules such that the social optimum is dominant-strategy equilibrium [12].

IV. SELFISH NODE DETECTION

Selfish nodes:

Nodes that do not forward other nodes packet, thus maximizes their benefit at the expense of all others. They are assumed to always behave rationally, so they cheat only if it gives them an advantage.

Characteristics of selfish nodes as follows:

1) **Do not participate in routing process:** A selfish node drops routing messages or it may modify the Route Request and Reply packets by changing TTL value to smallest possible value.
2) **Do not reply or send hello messages**: A selfish node may not respond to hello messages, hence other nodes may not be able to detect its presence when they need it. International Journal of Wireless & Mobile Networks
3) **Intentionally delay the RREQ packet**: A selfish node may delay the RREQ packet up to the maximum upper limit time. It will certainly avoid itself from routing paths.
4) **Dropping of data packet**: A selfish nodes may participate in routing messages but may not relay data packets

Credit Based Technique

The basic idea of Credit based schemes provides incentives for nodes to faithfully perform networking functions. In order to achieve this goal, virtual (electronic) currency or similar payment system may be set up. Nodes get paid for providing services to other nodes. When request other nodes to help them for packet forwarding they use the same payment system for such services. Credit based schemes can be implemented using two models: The Packet Purse Model (PPM) and the Packet Trade Model (PTM).

Detection and Prevention

Watchdog: In this approach a node sends a packet to its neighbor and then overhears the neighbor forwarding it one hop further along the route. Thus a misbehaving node dropping or manipulating packets is immediately identified and routes using this node can be avoided. Unfortunately this mechanism is too simple and has two major drawbacks.

First: it is error-prone; a packet collision between AB causes a false negative detection and a collision between BC cause a false positive detection (A acknowledges the retransmission even though it failed). The model also relies on all clients to have equal sending ranges – this conflicts with modern Wi-Fi-controllers using energy control.

Second: When a node recognizes its neighbor as non-participating it does not spread this information, but is only supposed to find a new route around the problem, thus even rewarding the non-participating node (now it does not have to forward other node's data anymore)[8].

V. PROPOSED WORK

Reputation System Model:

The reputation system model can be used to:

(1) Motivate nodes to behave normally.

(2) Punish the misbehaving nodes.

Moreover, it can be used to determine whom to trust. To motivate the nodes in behaving normally in every election round, we relate the cluster's services to nodes' reputation. This will create a competition environment that motivates the nodes to behave normally by saying the truth. To enforce our mechanism, a punishment system is needed to prevent nodes from behaving selfishly after the election. Misbehaving nodes are punished by decreasing their reputation and consequently are excluded from the cluster services if the reputation is less than a predefined threshold.

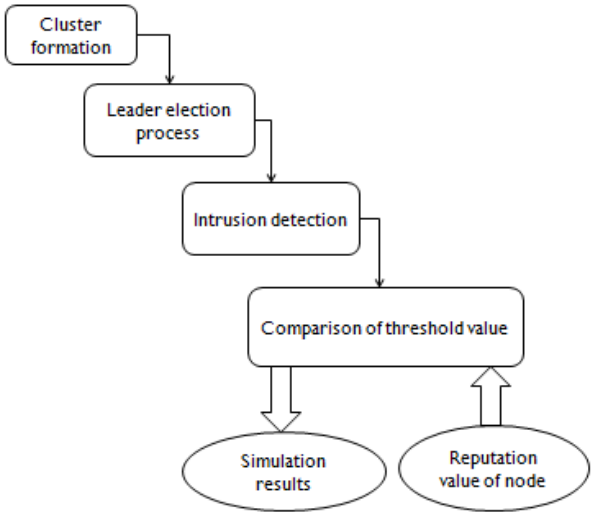

Fig 2. System model

To enforce the mechanism, a punishment system is needed to prevent nodes from behaving selfishly after the election. Misbehaving nodes are punished by decreasing their reputation and consequently are excluded from the cluster services if the reputation is less than a predefined threshold.

Monitor:

It is used to monitor the behavior of the elected leader. To reduce the overall resource consumption, we randomly elect a set of nodes, known as checkers, to perform the monitoring process.

Information Exchange:

It includes two types of information sharing: (1) exchange of reputation with other nodes in other clusters (i.e., for services purposes). (2) To reduce the false positive rate, the checkers will exchange information about the behavior of the leader to make decision about the leader's behavior.

Reputation System:

It is defined in the form of a table that contains the ID of other nodes and their respective reputation R.

Threshold Check:

It has two main purposes:

(1) To verify whether nodes reputation is greater than a predefined threshold.

(2)To verify whether a leader behaviors exceeds a predefined misbehaving threshold.

Service System:

To motivate the nodes to participate in every election round, the amount of detection service provided to each node is based on the node's reputation.

Punishment System:

To improve the performance and reduce the false-positive rate of checkers in catching and punishing a misbehaving leader

VI. CONCLUSION

The proposed work specifies that intrusion detection based on clustering and leader election technique considerably reduces the resource consumption and detects the intrusion. An unbalanced resource consumption of IDSs in MANET and the presence of selfish nodes have motivated to an integrated solution for prolonging the lifetime of mobile nodes and for preventing the emergence of selfish nodes. The solution motivated nodes to truthfully elect the most cost efficient nodes that handle the detection duty on behalf of others. Moreover, the sum of the elected leaders is globally optimal. To achieve this goal, incentives are given in the form of reputations to motivate nodes in revealing truthfully their costs of analysis. Reputations are computed using the well known VCG mechanism by which truth-telling is the dominant strategy. We analyzed the performance of the mechanisms in the presence of selfish and malicious nodes. To decrease

the percentage of leaders, single node clusters, maximum cluster size and increase average cluster size. These properties allow improving the detection service through distributing the sampling budget over less number of nodes and reduce single nodes to launch their IDS.

ACKNOWLEDGMENT

Our thanks to the experts who have contributed for the development of a **Secure Protocol for Leader Election and Intrusion Detection in MANET** and its simulated solution.

References

[1] Animesh Patcha and Jung-Min Park" "A Game-Theoretic Intrusion Detection Model For Mobile Ad Hoc Networks" Fifth Annual IEEE Information Assurance Workshop,United States Military Academy, West Point, New York, June 2004.

[2] Ayman Bassam Nassuora and Abdel-Rahman H. Hussein"CBPMD: A New Weighted Distributed Clustering Algorithm for Mobile Ad hoc Networks (MANETs)" American Journal of Scientific Research ISSN 1450-223X Issue 22(2011), pp.43-56.

[3] Kai Chen and Klara Nahrstedt"Ipass: An Incentive Compatible Auction Scheme To Enable Packet Forwarding Service In Manet"

[4] Luzi Anderegg and Stephan Eidenbenz"Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad hoc Networks with Selfish Agents" Los Alamos National Laboratory Publication No.LA-UR:03-1404.

[5] Martin Schütte"Detecting Selfish And Malicious Nodes In Manets" seminar: sicherheit in selbstorganisierenden netzen,hpi/universität potsdam, sommersemester 2006

[6] Noman Mohammed,Hadi Otrok,Lingyu Wang ,Mourad Debbai And Prabir Bhattacharya "Mechanism Design-Based Secure Leader Election Model For Intrusion Detection In Manets" IEEE Transaction ondependable and secure clustering,2011

[7] Paul Brutch and Calvin Ko "Challenges In Intrusion Detection For Wireless Ad-Hoc Networks" Network Associates Laboratories.

[8] Shailender Gupta,C.K.Nagpal and Charusingla"Impact Of Selfish Node Concentration In Manets" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 2, April 2011

[9] Sivaranjani.v and Rajalakshm"Secure Cluster Head Election For Intrusion Detection In Manet" Journal of Computer ApplicationsISSN:0974 – 1925,Volume-5,Issue EICA2012-4, February 10, 2012.

[10] Sonja Buchegger and JeanYves Le Boudec"Performance Analysis Of The Confidant Protocol (Cooperation Of Nodes: Fairness In Dynamic Adhoc Networks)" In Proceedings of DARPA Information Survivability Conference and Exposition, 2000.

[11] Sudarshan Vasudevan, Neil Immerman, Jim Kurose, Don Towsley UMass"A Leader Election Algorithm For Mobile Ad Hoc Networks" Computer Science Techincal Report 03-01 January 13, 2003

[12] Y. Xiao, X. Shen, and D.-Z"A Survey On Intrusion Detection In Mobile Ad Hoc Networks" Wireless/Mobile Network Security Du (Eds.) pp. 170 – 196 2006 Springer

