Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Intrusion Detection System in MANET with Secure Leader Election Model

E. Swetha,
M. Tech Student,
Dept of CSE,AITS,
Tirupathi, A.P., India.

K. Sangeethasupriya,
Asst. Professor,
Dept of CSE,AITS,
Tirupathi, A.P., India.

Abstract— Mobile Ad hoc Networks (MANET) have been highly vulnerable to attacks due to the dynamic nature of its network infrastructure. Among these attacks, routing attacks have received considerable attention since it could cause the most devastating damage to MANET. Even though there exist several intrusion response techniques to mitigate such critical attacks, existing solutions typically attempt to isolate malicious nodes based on binary or naı¨ve fuzzy response decisions. However, binary responses may result in the unexpected network partition, causing additional damages to the network infrastructure, and naı¨ve fuzzy responses could lead to uncertainty in countering routing attacks in MANET. we propose a series of local election algorithms that can lead to globally optimal election results with a low cost. We address these issues in two possible application settings, namely, Cluster Dependent Leader Election (CDLE) and Cluster Independent Leader Election (CILE). The former assumes given clusters of nodes, whereas the latter does not require any pre-clustering. Finally, we justify the effectiveness of the proposed schemes through extensive experiments.

Key words: Mobile ad hoc networks, intrusion response, risk aware, VCG mechanism.

1. Introduction

MOBILE Ad hoc Networks (MANET) are utilized to set up wireless communication in improvised environments without a predefined infrastructure or centralized adminis-tration. Therefore, MANET has been normally deployed in adverse and hostile environments where central authority point is not necessary. Another unique characteristic of MANET is the dynamic nature of its network topology which would be frequently changed due to the unpredict-able mobility of nodes. Furthermore, each mobile node in MANET plays a router role while transmitting data over the network. Hence, any compromised nodes under an adver-sary's control could cause significant damage to the functionality and security of its network since the impact would propagate in performing routing tasks. Several work addressed the intrusion response actions in MANET by isolating uncooperative nodes based on the node reputation derived from their behaviors. Such a simple response against malicious nodes often neglects possible negative side effects involved with the response actions. In MANET scenario, improper countermeasures may cause the unexpected network partition, bringing additional damages to the network infrastructure. To address the above-mentioned critical issues, more flexible and adaptive response should be investigated. The notion of risk can be adopted to support more adaptive responses to routing attacks in MANET. However, risk assessment is still a nontrivial, challenging problem due to its involvements of subjective knowledge, objective evidence, and logical reasoning. Subjective knowl-edge could be retrieved from previous experience and objective evidence could be obtained from observation while logical reasoning requires a formal foundation.

A. MOTIVATING EXAMPLE

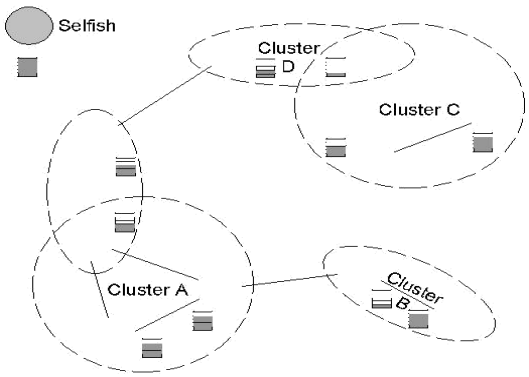


Fig 1: Example Scenario of Leader Election MANET

Figure 1 illustrates a MANET composed of ten nodes labeled from N_1 to N_{10}. These nodes are located in 5 one-hop clusters where nodes N_5 and N_9 belong to more than one cluster and have limited resources level. We assume that each node has different energy level, which is considered as private information. At this point, electing nodes N_5 and N_9 as leaders is clearly not desirable since losing them will cause a partition in the network and nodes will not be able to communicate with each other. However, with the random election model, nodes N_5 and N_9 will have equal probability, compared to others, in being elected as leaders. The nodes N_5 and N_9 will definitely be elected under the connectivity index-based approach due to their connectivity indices. Moreover, a naive approach for electing nodes with the most remaining resources will also fail since nodes' energy level is considered as private information and nodes might reveal fake information if that increases their own benefits. Finally, if the nodes N_2, N_5 and N_9 are selfish and elected as leaders using the above models, they will refuse to run their IDS for serving others. The consequences of such a refusal will lead normal nodes to launch their IDS and thus die faster.

B. Our Proposed Solution

In this paper,we propose a solution for balancing the resource consumption of IDS among all nodes while preventing nodes from behaving selfishly. To address the selfish behavior, we design incentives in the form of reputation to encourage nodes to honestly participate in the election scheme by revealing their cost of analysis. The cost of analysis is designed to protect nodes' sensitive information (resources level) and ensure the contribution of every node on the election process (fairness). To motivate nodes in behaving normally in every election round, we relate the amount of detection service that each node is entitled to the nodes' reputation value. Besides, this reputation value can also be used to give routing priority and to build a trust environment. The design of incentives is based on a classical mechanism design model, namely, Vickrey, Clarke, and Groves (VCG) [2]. The model guarantees that truth-telling is always the dominant strategy for every node during each election phase. On the other hand, to find the globally optimal cost-efficient leaders, a leader election algorithm is devised to handle the election process, taking into consideration the possibility of cheating and security flaws, such as replay attack. The algorithm decreases the percentage of leaders, single node clusters, maximum cluster size and increases average cluster size. Last but not least, we address these issues in two possible settings, namely, Cluster Independent Leader Election (CILE) and Cluster Dependent Leader Election (CDLE). In the former, the leaders are elected according to the received votes from the neighbor nodes. The latter scheme elects leaders after the network is formulated into multiple clusters. In both schemes, the leaders are elected in an optimal way in the sense that the resource consumption for serving as IDSs will be balanced among all nodes overtime. Finally, we justify the correctness of proposed methods through analysis and simulation. Empirical results indicate that our scheme can effectively improve the overall lifetime of a MANET. The main contribution of this paper is a unified model that is able to: (1) Balance the IDS resource consumptions among all nodes by electing the most cost-efficient leaders. (2) Motivate selfish nodes to reveal their truthful resources level.

C.Possible Applications of Leader Election Scheme

The problem of selfishness and energy balancing exists in many other applications to which our solution are also applicable. Like in IDS scheme, leader election is needed for routing and key distribution in MANET. In key management, a central key distributer is needed to update the keys of nodes. In routing, the nodes are grouped into small clusters and each cluster elects a cluster head (leader) to forward the packets of other nodes. Thus, one node can stay alive while others can be in the energy-saving mode. The election of leader a node is done randomly, based on connectivity (nodes' degree) or based on a node's weight (here the weight refers to the remaining energy of a node [8]. We have already pointed out the problems of random model and connectivity model. We believe that a weight-based leader election should be the proper method for election. Unfortunately, the information regarding the remaining energy is private to a node and thus not verifiable. Since nodes might behave selfishly, they might lie about their resource level to avoid being the leader if there is no mechanism to motivate them. Our method can effectively address this issue.

D. Paper Outline

The rest of this paper is organized as follows: Section II formulates the problem. Section III describes our leader election mechanism where the cost of analysis function, reputation model and payment design are given. Section IV analyzes our mechanisms against selfish and malicious nodes. Section V devises the election algorithm needed to handle the election process. Section VI provides the proof of correctness and security properties of the algorithm. Section VII presents empirical results. Section VIII reviews related work. Finally, Section IX concludes the paper and discusses our future work.

2. Problem Statement

We consider a MANET where each node has an IDS and a unique identity. To achieve the goal of electing the most cost efficient nodes as leaders in the presence of selfish and malicious nodes, the following challenges arise: First, the resource level that reflects the cost of analysis is considered as a private information. As a result, the nodes can reveal fake information about their resources if that could increase their own benefits. Second, the nodes might behave normally during the election but then deviate from normal behavior by not offering the IDS service to their voted nodes In our model, we consider MANET as an undirected graph G = (N, L) where N is the set of nodes and L is the set of bidirectional links. We denote the cost of analysis vector as $C = \{c_1, c_2, \ldots, c_n\}$ where n is the number of nodes in N . We denote the election process as a function vt_k (C, i) where vt_k (C,I)=1 if a node I votes for a node k: vt_k (C, i) = 0,

otherwise. We assume that each elected leader allocates the same budget B (in the number of packets) for each node that has voted for it. Knowing that, the total budget will be distributed among all the voting nodes according to their reputation. This will motivate the nodes to cooperate in every election round that will be held on every time T_{ELECT} . Thus, the model will be repeatable. For example, if B = 25 packet/sec and the leader gets 3 votes, then the leader's sampling budget is 75 packet/sec. This value is divided among the 3 nodes based on their reputation value.

The objective of minimizing the global cost of analysis while serving all the nodes can be expressed by the following Social Choice Function (SCF):

$$SCF=S(C)=Min\sum_{k\epsilon N} Ck \sum_{i\epsilon N} vtK(C,I).B)$$

Clearly, in order to minimize this SCF, the following must be achieved. First, we need to design incentives for encouraging each node in revealing its true cost of analysis value c, which will be addressed in Section III. Second, we need to design an election algorithm that can provably minimize the above SCF while not incurring too much of performance overhead.

2.1 Leader Election Mechanisam

In this section, we present our leader election mechanism for truthfully electing the leader nodes. To make the paper self-contained, the background on mechanism design is given in Subsection III-A. Subsection III-B formulates our model using the standard mechanism design notations.

A.Mechanism Design Background

Mechanism design is a sub-field of microeconomics and game theory [9]. Mechanism design uses game theory [10] tools to achieve the desired goals. The main difference between game theory and mechanism design is that the former can be used to study what could happen when independent players act selfishly. On the other hand, mechanism design allows a game designer to define rules in terms of the Social Choice Function (SCF) such that players will play according to these rules. The balance of IDS resource consumption problem can be modeled using mechanism design theory with an objective function that depends on the private information of the players. In our case, the private information of the player is the cost of analysis which depends on the player's energy level. Here, the rational players select to deliver the untruthful or incomplete information about their preferences if that leads to individually better outcomes. The main goal of using mechanism design is to address this problem by: 1) Designing incentives for players (nodes) to provide truthful information about their preferences over different outcomes. 2) Computing the optimal system-wide solution, which is defined according to Equation 1?

A Mechanism design model consists of n agents where each agent I € {1,…..,n}has a private information, p $\boldsymbol{\theta_i \in \Theta_i}$ Known as the agent type.Morever, it defines a set of stratagies Ai for the agent i.The agent can choose αi € Ai to input in the mechanisam.The preference of the agent from the output is calculated by a valuation function vi(θi,0) .This is aqunatifaction in term of real number ot evaluate the output for an agent i, thus the utility of the node is calculated as **Ui = Pi – Vi($\boldsymbol{\theta i - o}$).** This means, the utility is the combination of output measured by valuation function and the payment it received the mechanisam.

$$\boldsymbol{Pi - Vi(\theta i, o) = \ Ui \geq U = Pi - Vi(\theta i, o)}$$

B. Cost Function.

During the design of the cost of analysis function, the following two problems arise: First, the energy level is considered as private and sensitive information and should not be disclosed publicly. Such a disclosure of information can be used maliciously for attacking the node with the least resources level. Second, if the cost of analysis function is designed only in terms of nodes' energy level, then the nodes with the low energy level will not be able to contribute and increase their reputation values.

Percentage of sample calculated by cost function

PS(Percentage of sampling)	$Class_4$	$Class_3$	$Class_2$	$Class_1$
After 200 sec	55%	20%	15%	10%
After 600 sec	45%	24%	18%	13%
After 1000 sec	40%	26%	20%	14%

Cl1 if PF < $\rho 1$

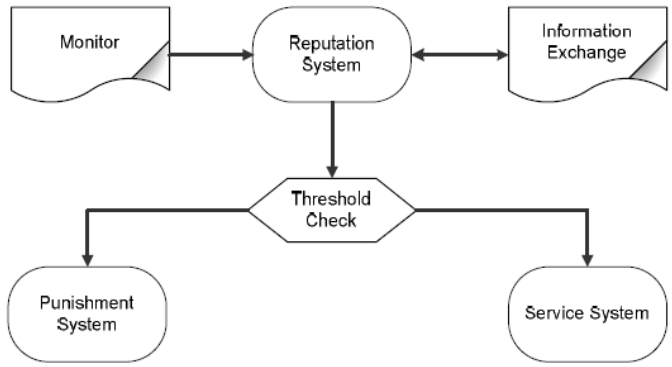
CL = cli $If\ \rho i - 1\ \leq PF < \rho i; i\ \epsilon\ [2, l-1]$

CL1 $if\ PF\ \geq\ \rho l - 1$

However, we considered energy level, expected lifetime and the present P S of node to calculate the cost of analysis. We can extend the cost of analysis function to more realistic settings by considering the computational level and cost of collecting and analyzing traffic.

C Reputation System Model

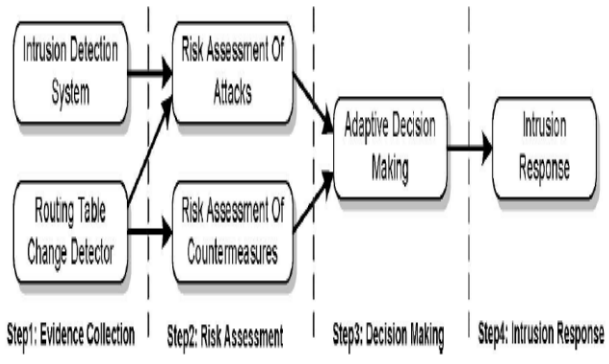
Before we design the payment, we need to show how the payment in the form of reputation can be used to:
(1) Motivate nodes to behave normally and
(2) Punish the misbehaving nodes. Moreover, it can be used to determine whom to trust.
To motivate the nodes in behaving normally in every election round, 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. As an extension to our model, we can extend our reputation system to include different sources of information such as routing and key distribution with different assigned weights. Figure 2 shows the abstract model of our reputation system where each node has the following components



Reputation System Model

3. **Risk Aware Mechanisam**

In this section, we articulate an adaptive risk-aware response mechanism based on quantitative risk estimation and risk tolerance. Instead of applying simple binary isolation of malicious nodes, our approach adopts an isolation mechanism in a temporal manner based on the risk value. We perform risk assessment with the extended D-S evidence theory introduced in Section 3 for both attacks and corresponding countermeasures to make more accurate response decisions illustrated in the below Diagram



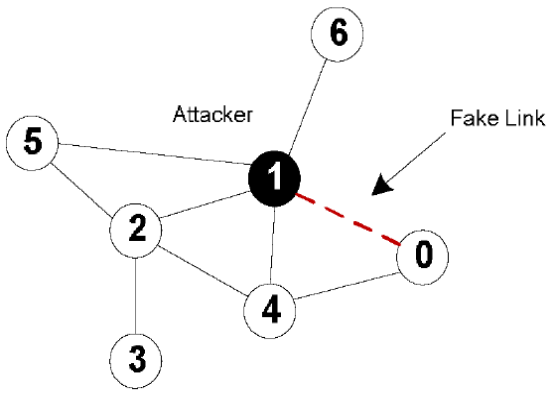
3.1 Overview

Because of the infrastructure-less architecture of MANET,our risk-aware response system is distributed, which means each node in this system makes its own response decisions based on the evidences and its own individual benefits. Therefore, some nodes in MANET may isolate the malicious node, but others may still keep in cooperation with due to high dependency relationships. Our riskaware response mechanism is divided into the following Four steps shown in above

Evidence collection. In this step, Intrusion Detection System (IDS) gives an attack alert with a confidence value, and then Routing Table Change Detector (RTCD) runs to figure out how many changes on routing table are caused by the attack.

Risk assessment. Alert confidence from IDS and the routing table changing information would be further considered as independent evidences for risk calculation and combined with the extended D-S theory. Risk of countermeasures is calculated as well during a risk assessment phase. Based on the risk of attacks and the risk of countermeasures, the entire risk of an attack could be figured out.

Decision making. The adaptive decision module provides a flexible response decision-making mechanism, which takes risk estimation and risk tolerance into account. To adjust temporary isolation level, a user can set different thresholds to fulfill her goal.



Example scenario.

Intrusion response. With the output from risk assessment and decision-making module, the corresponding response actions, including routing table recovery and node isolation, are carried out to mitigate attack damages in a distributed manner.

3.2 Response to routing Attacks

In our approach, we use two different responses to deal with different attack methods: routing table recovery and node isolation. Routing table recovery includes local routing table recovery and global routing recovery. Local routing recovery is performed by victim nodes that detect the attack and automatically recover its own routing table. Global routing recovery involves with sending recovered routing messages by victim nodes and updating their routing table based on corrected routing information in real time by other nodes in MANET. Routing table recovery is an indispensable response and should serve as the first response method after successful detection of attacks. In proactive routing protocols like OLSR, routing table recovery does not bring any additional overhead since it periodically goes with routing control messages. Also, as long as the detection of attack is positive, this response causes no negative impacts on existing routing operations. Node isolation may be the most intuitive way to prevent further attacks from being launched by malicious nodes in MANET. To perform a node isolation response, the neighbors of the malicious node ignore the malicious node by neither forwarding packets through it nor accepting any packets from it. On the other hand, a binary node isolation response may result in negative impacts to the routing operations, even bringing more routing damages than the attack itself.

3.3 Selection of Evidences

Basic probability assignments of Evidences 2 to 5 are based on. Equations are piecewise linear functions, where a, b, c, and d are constants and determined by experts. d is the minimum value of the belief that implies the status of MANET is insecure. On the other hand, 1-d is the maximum value of the belief that means the status of MANET is secure. a, b, and c are the thresholds for minimum belief or maximum belief for each respective mass function

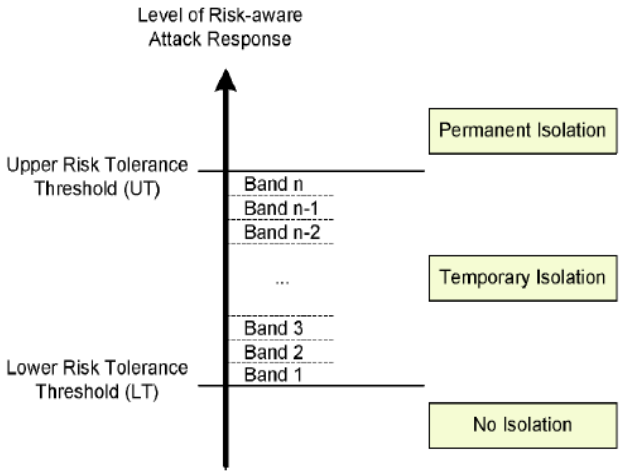
$$m(Insecure) = \begin{cases} d & x \in [0, a] \\ (\frac{1-2d}{c-a})(x-a) & x \in (a, c] \\ 1-d & x \in (c, 1], \end{cases}$$

$$m(Secure) = \begin{cases} 1-d+(\frac{2d-1}{b})x & x \in [0, b] \\ d & x \in (b, 1], \end{cases}$$

$$m(Secure, Insecure) = \begin{cases} \frac{1-2d}{b}x & x \in [0, a] \\ d - \frac{2d-1}{b}x - & \\ \quad (\frac{1-2d}{c-a})(x-a) & x \in (a, b] \\ 1-b- & \\ \quad (\frac{1-2d}{c-a})(x-a) & x \in (b, c] \\ 0 & x \in (c, 1]. \end{cases}$$

3.4 Adaptive Decision Making

Our adaptive decision-making module is based on quantitative risk estimation and risk tolerance, the response level is additionally divided into multiple bands. Each band is associated with an isolation degree, which presents a different time period of the isolation action. The response action and band boundaries are all determined in accordance with risk tolerance and can be changed when risk tolerance threshold changes. The upper risk tolerance threshold (UT) would be associated with permanent isolation response. The lower risk tolerance threshold (LT) would remain each node intact. The band between the upper tolerance threshold and lower tolerance threshold is associated with the temporary isolation response, in which the isolation time (T) changes dynamically based on the different response level given by (18) and (19), where n is the number of bands and i is the corresponding isolation band



4. Proposed Algorithm

4.1 LEADER ELECTION ALGORITHM

To design the leader election algorithm, the following requirements are needed:

(1) To protect all the nodes in a network, every node should be monitored by a leader.

(2) To balance the resource consumption of IDS service, the overall cost of analysis for protecting the whole network is minimized.

In other words, every node has to be affiliated with the most cost efficient leader among its neighbors. Our algorithm is executed in each node taking into consideration the following assumptions about the nodes and the network architecture:

• Every node knows its (2-hop) neighbors, which is reasonable since nodes usually maintain a table about their neighbors for routing purposes.
• Loosely synchronized clocks are available between nodes.
• Each node has a key (public, private) pair for establishing a secure communication between nodes.
• Each node is aware of the presence of a new node or removal of a node.

Leader Election

To start a new election, the election algorithm uses four types of messages. *Hello*, used by every node to initiate the election process; *Begin-Election*, used to announce the cost of a node; *Vote*, sent by every node to elect a leader;*Acknowledge*, sent by the leader to broadcast its payment, and also as a confirmation of its leadership. For describing the algorithm, we use the following notation:

• ***service-table (k)*:** The list of all ordinary nodes, those voted for the leader node k.
• ***reputation-table (k)*:** The reputation table of node k. Each node keeps the record of reputation of all other nodes.
• ***neighbors (k)*:** The set of node k's neighbors.
• ***leadernode (k)*:** The ID of node k's leader. If node k is running its own IDS then the variable contains k.
• ***leader(k)*:** A boolean variable that sets to TRUE if node k is a leader and FALSE otherwise.

Initially, each node k starts the election procedure by broadcasting a *Hello* message to all the nodes that are one hop from node k and starts a timer T1. This message contains the hash value of the node's cost of analysis and its unique identifier (ID). This message is needed to avoid cheating where further analysis

Algorithm 1 (Executed by every node)

/* On receiving *Hello*, all nodes reply with their cost */
1. if (received *Hello* from all neighbors) then
2. Send *Begin-Election* (IDk, costk);
3. else if(neighbors(k)=Ø) then
4. Launch IDS.
5. end if

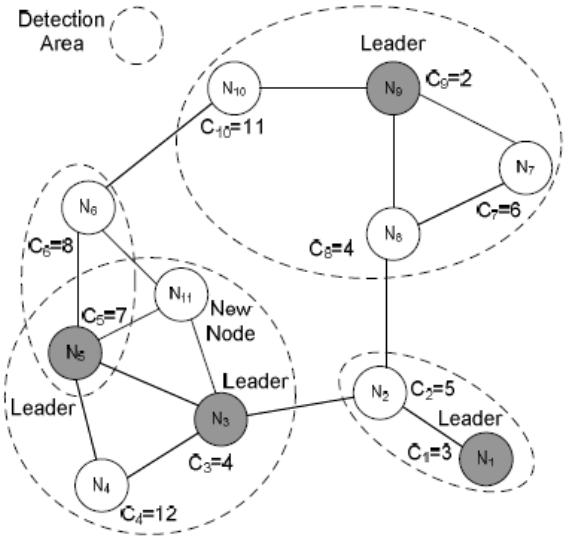
On expiration of T1, each node k checks whether it has received all the hash values from its neighbors. Nodes from whom the *Hello* message have not received are excluded from the election. On receiving the *Hello* from all neighbors, each node sends *Begin-Election* as in Algorithm 1, which contains the cost of analysis of the node and then starts timer T2. If node k is the only node in the network or it does not have any neighbors then it launches its own IDS.

Algorithm 2 (Executed by every node)
/* Each node votes for one node among the neighbors */
1. if ($\forall$ n _ neighbor(k), $\exists$ i _ n : ci $\leq$ cn) then
2. send V ote(IDk, IDi , costj_=i);
3. *leadernode(k)*:= i;
5. end if

On expiration of T2, the node k compares the hash value of *Hello* to the value received by the *Begin-Election* to verify the cost of analysis for all the nodes. Then node k calculates the least-cost value among its neighbors and sends *Vote* for node i as in Algorithm 2. The *Vote* message contains the IDk of the source node, the IDi of the proposed leader and second least cost among the neighbors of the source node costj_=i. Then node k sets node i as its leader in order to update later on its reputation. Note that the second least cost of analysis is needed by the leader node to calculate the payment. If node k has the least cost among all its neighbors then it votes for
itself and starts timer T3.

Algorithm 3 (Executed by Elected leader node)
/* Send Acknowledge message to the neighbor nodes */
1. Leader(i) := TRUE;
2. Compute Payment, Pi;
3. updateservice−table(i);
4. updatereputation−table(i);
5. Acknowledge = Pi + all the votes;
6. Send Acknowledge(i);
7. Launch IDS.
On expiration of T3, the elected node i calculates its payment using equation 5 and sends an *Acknowledge* message
to all the serving nodes as in Algorithm 3. The *Acknowledge* message contains the payment and all the votes the leader received. The leader then launches its IDS.

4.2 Adding A Node
When a new node is added to the network, it either launches its own IDS or becomes an ordinary node of any leader node.To include a new node to the IDS service, four messages are needed: *Hello*, *Status*, *Join* and *Acknowledge*. *Hello* is sent by a new node n to announce its presence in the network. This Hello message is similar to the one presented in the previous section. Upon receiving the *Hello*, all the neighbors of the new



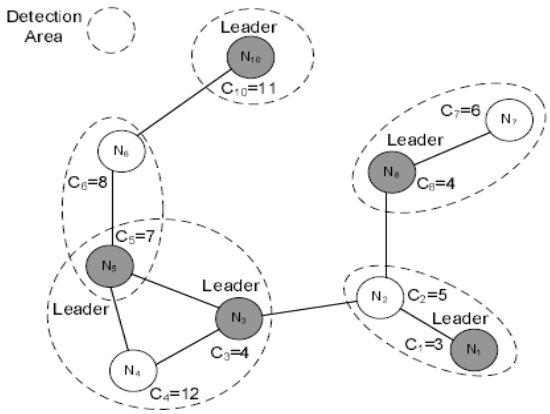
MANET after adding a new node

node, reply with a *Status* message. If the neighbor node k is a leader node, then the *Status* message contains its cost. On the other hand, if node k is an ordinary node, the *Status* message contains the ID of its leader node as in Algorithm 4.

Algorithm 4 (Executed by neighboring nodes)
/* The neighboring nodes send 'Status' to new node */
1. if (leader(k) = TRUE) then
2. *Status* := Costk;
3. else
4. *Status* := *leadernode(k)*;
5. end if;
6. send Status(k, n);

4.3. Removing a node

When a node is disconnected from the network due to many reasons; such as, mobility or battery depletion, then the neighbor nodes have to reconfigure the network. We assume that whenever a node dies, its neighbors are aware of it. At first a *Dead(n)* message is circulated to all neighbors to confirm the removal of node n. On receiving the *Dead(n)* message, the neighbor node k checks whether node n is its leader node or not. If node n is the leader node of node k, then node k announce a new election and updates its reputation table. On the other hand, if node n is an ordinary node then its leader node update its serving list.



A MANET after adjustment

Algorithm 5 (Executed by neighboring nodes)
/* The neighboring nodes reconfigure the network and */
/* declare new election if necessary*/
1. if (leadernode(k) = n) then
2. leadernode(k):= NULL;
3. updatereputation(k);
4. send Begin − Election as in Algorithm 1;
5. end if;
6. if (leader(k) = TRUE) then
7. if (n _ service(k)) then
8. updateservice();
9. end if;
10. end if;

5. PERFORMANCE OF RESULTS

A.Performance Metrics

The main objective of our simulation results is to study the effect of node selection for IDS on the life of all nodes. To show the negative impact of selfish node, we conducted two experiments: *Time taken for the first node to die and percentage of packet analysis.* Besides, we use the following metrics to evaluate our algorithm against others: *Percentageof alive nodes, energy level of nodes, and percentage of leader node, average cluster size, maximum cluster size and number of single node clusters.* Our experiments have been conducted in both static and dynamic networks. For a static network,we compare our algorithm with both random and connectivity models, while for dynamic network, we only compare with connectivity model since we believe that the random model will perform almost the same as in static one. Our experimental results have a 95% confidence and a 5% precision.

B.Experimental Results

Nodes can behave selfishly before and after the election.A node shows selfishness before election by refusing to be a leader. On the other hand, selfishness after election is considered when nodes misbehave by not carrying out the

detection service after being a leader. Both kinds of selfishness have a serious impact on the normal nodes. To show the seriousness and impact of selfishness before election on resource consumption, depicts the impact of selfish nodes on the life of normal nodes. The result indicates that the normal nodes will carry out more duty of intrusion detection and die faster when there are more selfish nodes. the impact of selfishness after election on security. We consider the presence of 20% of selfish nodes out of 10 nodes. As selfish nodes do not exhaust energy to run the IDS service, it will live longer than the normal nodes.Thus, the more the time goes, the more the chances that the selfish node will be the leader node. Hence, the percentage of packet analysis decreases with time, which is shown in This is a severe security concern since fewer packets are analyzed.

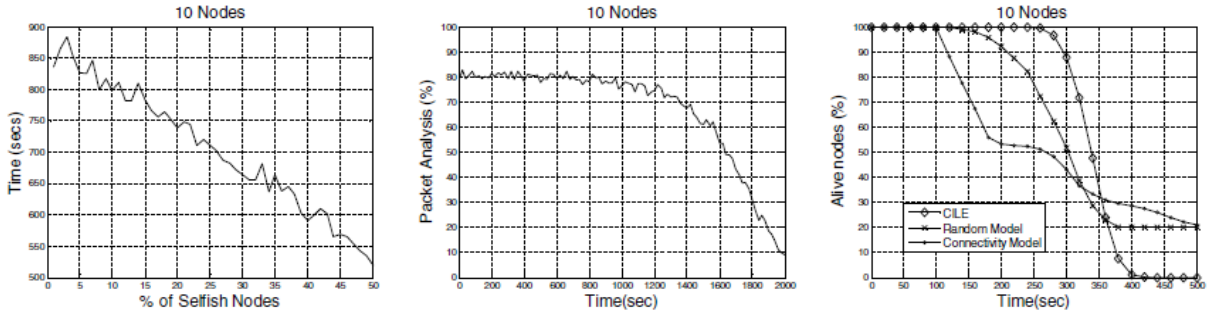


Fig. 6. (a) Time for Normal Node to Die (b) Percentage of Packet Analysis (c) Percentage of Alive Nodes

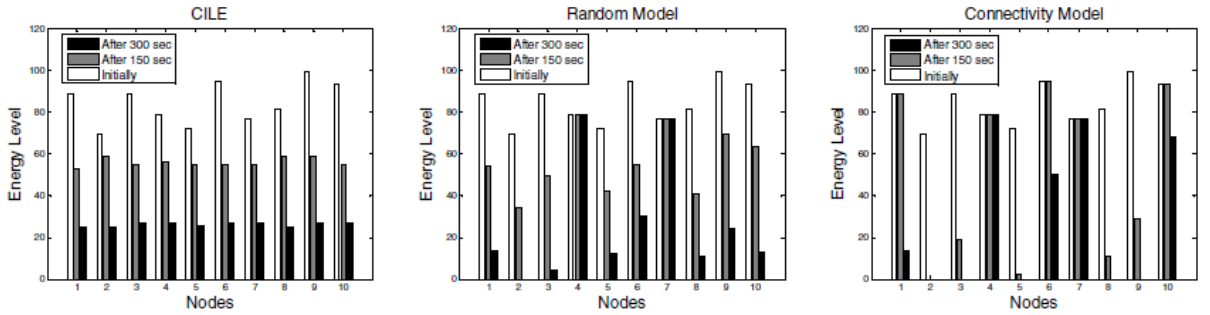


Fig. 7. (a) Energy Level of Our Model (b) Energy Level of Random Model (c) Energy Level of Connectivity Model

6. Conclusion

The unbalanced resource consumption of IDSs in MANET and the presence of selfish nodes have motivated us to propose 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 costefficient 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 also analyzed the performance of the mechanisms in the presence of selfish and malicious nodes. To implement our mechanism, we devised an election algorithm with reasonable performance overheads. We also provided the algorithmic correctness and security properties of our algorithm. We addressed these issues into two applications:CILE and CDLE. The former does not require any preclustering whereas CDLE requires nodes to be clustered before running the election mechanism. Simulation results showed that our model is able to prolong the lifetime and balancethe overall resource consumptions among all the nodes in the network. Moreover, we are able to decrease the percentage of leaders, single node clusters, maximum cluster size and increase average cluster size. These properties allow us to improve the detection service through distributing the sampling budget over less number of nodes and reduce single nodes to launch their IDS.

Reference

[1] T. Anantvalee and J. Wu. *A survey on intrusion detection in mobile ad hoc networks*. Wireless/Mobile Network Security, 2006.

[2] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: *A truthful and costefficient routing protocol for mobile ad hoc networks with selfish agents*.In proc. of the ACM International Conference on Mobile Computing and Networking (MobiCom), 2003.

[3] F. Anjum and P. Mouchtaris. *Security for Wireless Ad Hoc Networks*.John Wiley & Sons. Inc., USA, 2007.

[4] S. Basagni. *Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks*. In proc. of the IEEE InternationalVehicular Technology Conference (VTC), 1999.

[5] S. Basagni. *Distributed clustering for ad hoc networks*. In proc. of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN), 1999.

[6] M. Bechler, H. Hof, D. Kraft, F. Pahlke, and L. Wolf. *Clusterbased security architecture for ad hoc networks*. In proc. of the IEEE INFOCOM, 2004.

[7] P. Brutch and C. Ko. *Challenges in intrusion detection for wireless adhoc networks*. In proc. of the IEEE Symposium on Applications and the Internet (SAINT) Workshop, 2003.

[8] S. Buchegger and J. L. Boudec. *Performance analysis of the CONFIDANT protocol (cooperation of nodes - fairness in dynamic adhoc networks)*. In proc. of the ACM MOBIHOC, 2002.

[9] A. Mas-Colell, M. Whinston, and J. Green, *Microeconomic Theory.* Oxford Univ. Press, 1995.

[10] P. Morris, *Introduction to Game Theory,* first ed. Springer, 1994.

