TRIUMF: Trust-Based Routing Protocol with controlled degree of Selfishness for Securing MANET against Packet Dropping Attack

Ahmed M. Abd El-Haleem[1] and Ihab A. Ali[2]

[1] Assistant Lecture, Communication Department, Faculty of Engineering, Helwan University Helwan, Egypt

[2]Associate Professor, Communication Department, Faculty of Engineering, Helwan University Helwan, Egypt

Abstract

In a mobile ad-hoc network, nodes cannot rely on any fixed infrastructure for routing purposes. Rather, they have to cooperate to achieve this objective. However, performing network functions consumes energy and other resources. Therefore, some network nodes may decide against cooperating with others; selfish nodes. Also security issues are more paramount in such networks even more so than in wired networks. In particular these networks are extremely under threat to insider; malicious nodes; especially through packet dropping attacks. Selfish and malicious nodes are termed as misbehaving nodes. Giving the selfish nodes the incentive to cooperate, while isolating the malicious nodes have been an active research area recently.

In this paper, we design a novel secure reactive routing protocol for Mobile ad hoc networks (MANETs), called TRIUMF (Trust-Based Routing Protocol with controlled degree of Selfishness for Securing MANET against Packet Dropping Attack). In the proposed protocol trust among nodes is represented by trust value, which consists of cooperation score, direct trust and indirect trust. The proposed trust routing allows controlled degree of selfishness to give an incentive to the selfish nodes to declare its selfishness behavior to its neighbor nodes, which reduce the searching time of misbehaving nodes to search for the malicious nodes only. In the proposed routing protocol two node-disjoint routes between the source and destination nodes are selected based on their path trust values, one marked as primary and the other as secondary. We use both DLL-ACK and end- to-end TCP-ACK as monitoring tools to monitor the behavior of routing path nodes: if the data packet successfully transmitted, then the path nodes trust value are updated positively; otherwise, if a malicious behavior is detected then the path searching tool starts to identify the malicious nodes and isolate them from the routing path and the network. Finally our scheme reduces the searching time of malicious nodes, and the routing protocol avoids the isolated misbehaving node from sharing in all future routes, which improves the overall network throughput.

Keyword: *Ad Hoc Network, Trust-Based routing, Secure Routing Protocol, network security.*

1. Introduction

Mobile Ad Hoc Network (MANET) is an infrastructure-less network, consisting of a set of mobile nodes without any support of base stations or access points. The mobile nodes are free to change their position with any speed and at any time, and they play the role of terminals and routers allowing hop by hop communication among nodes outside wireless transmission range. For lack of network infrastructure, the nodes have to communicate cooperatively. Cooperation at the network layer means routing and forwarding packets. Some nodes may deviate from the protocol for selfish or malicious reasons, these nodes are called misbehaving nodes.

Selfish nodes wish to use system services while taking an advantage of saving their resources by deviating from regular routing and forwarding. *Malicious* nodes wish to mount an attack to either a specific node or the network as whole. Both selfish and malicious nodes disrupt the routing protocol operation and reduce the network throughput. This brings up the need for secure routing protocols, where the Routing protocols must cope with such selfish and malicious behavior.

Several routing protocols have been proposed in the literature (see, e.g., [1], [2], [3]). These focus mainly on efficiency issues such as scalability with respect to network size, traffic load, mobility, and on the adaptability to network conditions such as link quality and power requirements. Some of the proposed routing algorithms also address security issues by using cryptographic tools to secure the routing protocol messages (e.g., [4], [5], [6], for a survey, see [7], [8]). Recently, a new class of routing protocol has been proposed, namely trust based routing. Trust based routing protocols consist of two parts: a routing part and a trust model, for a survey see [9]. Routing decisions are made according to the trust model. Trust and reputation have been used in many settings to cope with uncertainty in interactions. Trust is used to assess the risk associated with cooperating with others; it is an

estimate of how likely another is to fulfill its commitments. The trust routing protocols have to be able to identify trustworthy nodes and find a reliable and trustworthy route from sender to destination node. This has to be realized within a few seconds or better tenths of seconds, depending on the mobility of the nodes and the number of hops in the route. Most of the existing trust based routing protocols have given a trust based model without specifying an accurate threshold to distinguish between legitimate nodes and malicious nodes, and uses continuous promiscuous monitoring of the neighbors; which violate the TCP protocol rules.
This paper focuses on Packet Dropping Attack, and presents a novel routing protocol resistant to various packet dropping scenarios. Here, the malicious node tends to threaten network throughput through the use of packet dropping attack. This kind of attack could be even worse when supported by the malicious node sending link–layer acknowledgements to neighbor nodes to delay the detection of the attack and hence further decrease the throughput. In this paper, four packet dropping scenarios are considered. In Inclusive Packet Dropping, the malicious node simply drops all received data link layer (DLL) PDU's while positively acknowledging them. This attack is also called Black Hole attack [10], [11], [5]. Periodic Packet Dropping is used by malicious nodes to drop a small fraction of incoming DLL PDU's once per retransmission time out, a variant of JellyFish (JF) attack reported in [10], [11], [12]. In Frequent Packet Dropping, the malicious node may possibly drop a fraction of incoming DLL PDU's on a random basis. In Selective Packet Dropping, the malicious node drops only these PDU's coming from specific source(s), going to specific destination(s), or following a specific route. The last two attacks are called Gray Hole attack. In all packet dropping attack scenarios, the overall network throughput is expected to deteriorate [10].
In our proposed routing protocol we establish two node-disjoint routes between the source and destination nodes, these routes have the highest path trust values; to route around misbehaving nodes; one is marked as primary and the other as secondary. Unlike all previous research efforts made to tolerate Packet Dropping Attacks, our work allow a controlled degree of node selfishness; to save their resources partially; and detect the malicious activity faster. We use both DLL-ACK and end-to-end TCP-ACK as monitoring tools; without continuous promiscuous monitoring of the neighbors; and when detecting a malicious activity a new path searching technique is used to identify the malicious or compromised nodes in the routing path and isolate them. Based on this claim, the routing protocol avoids the isolated misbehaving node from sharing in all future routes, resulting in an improved overall throughput performance for the network.

The rest of the paper is organized as follows. Section 2, describes the research that has already been done in this area. The network assumptions and the proposed protocol operation are presented in Section 3. Our trust model is described in Section 4. Section 5 presents a discussion about the performance and the security of TRIUMF. Finally we conclude our work and discuss our plan for future work in section 6.

2. Related work

In [13] Marti et al. proposed a mechanism called as watchdog and pathrater on DSR to detect the misbehaving nodes in MANETs. The approach introduces two extensions to DSR: *A watchdog* detects misbehaving nodes, by maintaining a buffer of transmitted packets and overhearing of other node forwarding's. It compares each overheard packet with the packets in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. *A pathrater* avoids routing packets through the detected malicious nodes. Each node estimates a link metric with respect to the reliability of links and knowledge about misbehaving nodes. A node assigns this metric to links to every other known node and periodically updates the metric.
The downside of their method is that they cannot distinguish the misbehaving nodes from node failures. An honest node can easily be rated malicious if the transmission breaks up.

CONFIDANT [14] is a protocol which also attempts to detect the malicious nodes in ad hoc networks. Monitor, Reputation System, Path Manager and Trust Manager are the main components of CONFIDANT protocol. For each packet a node forwards, the *monitor* on that node attempts to ensure that the next-hop node also forwarded the packet correctly (overhearing). When the monitor detects an anomaly, it triggers action by the *reputation system*, which maintains a local ratings list. These lists are potentially exchanged with other nodes; the *trust manager* handles input from other nodes. Finally, the *path manager* chooses paths from the node's route cache based on a blacklist and the local ratings list. CONFIDANT has scalability problems with the number of nodes. The tables maintained by the reputation system of each node may become huge. Also, in scenarios with very high mobility, the overhead can increase considerably.

Xiaoqi Li et al [15] propose TAODV (Trusted AODV) which extends the widely used AODV (Ad hoc On demand Distance Vector) routing protocol and employs the idea of a trust model to protect routing behaviors in

the network layer of MANETs. In the TAODV, trust among nodes is represented by opinion, which is an item derived from subjective logic. The opinions are dynamic and updated frequently: If one node performs normal communications, its opinion from other nodes' points of view can be increased; otherwise, if one node performs some malicious behaviors, it will be ultimately denied by the whole network. A trust recommendation mechanism is also designed to exchange trust information among nodes. In TAODV when a node *A* is not sure whether it should believe or disbelieve any other nodes, it will use the cryptographic schemes as proposed in SAODV [16] to perform routing discovery operations. Then after some successful or failed communications, node A will change its opinions about other nodes gradually using the trust updating algorithm. Once the trust relationship is established among most of the nodes in this ad hoc network, these nodes can use the trusted routing protocol which is based on trust model to perform routing operations.

In [17] Balakrishnan et al, propose a s cheme of TWOACK to prevent selfishness in mobile ad hoc networks. They proposed two network-layer acknowledgment-based schemes, termed the TWOACK and the S-TWOACK schemes, which can be simply added-on to any source routing protocol. When a node forwards a packet, the node's routing agent verifies that the packet is received successfully by the node that is two hops away on the source route. This is done through the use of a s pecial type of acknowledgment packets, termed TWOACK packets. TWOACK packets have a very similar functionality as the ACK packets. A node acknowledges the receipt of a data packet by sending back a two-hop TWOACK packet along the active source route. If the sender/forwarder of a d ata packet does not receive a TWOACK packet corresponding to a particular data packet that was sent out, the next-hop's forwarding link is claimed to be misbehaving and the forwarding route broken. Based on this claim, the routing protocol avoids the accused link in all future routes, resulting in an improved overall throughput performance for the network. The S-TWOACK (Selective-TWOACK) scheme is a d erivative of the basic TWOACK scheme, aimed at reducing the routing overhead caused by excessive number of TWOACK packets. The basic drawback of this scheme is that it can't determine exactly which node is the misbehaving node; it only marks the link interconnecting the two nodes as misbehaving link and tries to avoid using this link in the future.

Trusted Dynamic Source Routing (TDSR) [18] extends the widely used DSR routing protocol and employs the idea of Trust Network Connect (TNC) to protect routing behaviors. In the TDSR, trust among nodes is represented by trust score, which consists of direct trust and indirect trust. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. TDSR framework has two main modules: DSR routing protocol, and Trust model. The trust model has three components: Requestor, used by the node *A* to ask another *B* to execute a function. Decision Executer (DE), used by receiving node *B* to look up the blacklist, if node *A* is in the list, then it will discard the request, otherwise deliver the request to the DM. Decision Maker (DM), makes decisions whether to execute a function according to the node's trust score, and send a decision to DE or Requestor. The main drawbacks of TDSR are, it might not determine the value of direct trust of neighboring node in presence of collisions, and it is not mentioned clearly how nodes can calculate the threshold to add a node to the blacklist.

Muhammad Zeshan et al, [19] proposed a two folded approach, to detect and then to isolate a malicious node causing packet dropping attacks. First approach will detect the misbehavior of nodes and will identify the malicious activity in network. When a Source node forwards any packet to the Destination through a route, all intermediate nodes will send back an ACK packet to its source node. If the Source node doesn't receive the ACK from any intermediate node, it will send again its packet for Destination after a specific time but if again this activity was observed, Source node will broadcast a packet to declare the malicious activity in the network. Then upon identification of misbehaving nodes in network other approach will isolate the malicious node from network. All nodes which lie in the transmission range of active route and also the nodes which are on the active route become in promiscuous listening mode and count number of packet coming into and going out of the nodes of active route. Each node in this range maintains a list of sent and dropped packets and when number of dropped packets by a particular node exceeds a certain threshold, the monitoring node in that range declares that node as misbehaving node. The basic drawback of this scheme is, nodes cooperate together to obtain an objective opinion about another node's trustworthiness, which give the misbehaving node the chance to falsely report the value of trust score (False Misbehavior).

3. The proposed Protocol

In this section we describe our solution to address the Packet Dropping Attack in MANETs. The proposed protocol makes the first effort to distinguish between the malicious and selfish node, and allow a controlled degree of node selfishness. The proposed monitoring tool detects the malicious activity and then the path searching tool identifies the malicious or compromised nodes in the network and isolates them, and the proposed routing protocol routes around the misbehaving node.

3.1 Network Model and Assumptions

In this work, we assume that the MANET nodes are situated in a bounded 2-dimensional space, within which they are free to move, and a bi-directional communication symmetry on every link between the nodes. For simplicity we also assume that the destination-node is non-malicious, and any routing path contains at most one malicious node.
Mobile nodes in MANETs often communicate with one another through an error-prone, bandwidth-limited, and insecure wireless channel. We are not concerned with the security problem introduced by the instability of physical layer or link layer. We only assume that: (1) Each node in the network has the ability to discover all of its neighbors; (2) Each node in the network can broadcast some essential messages to its neighbors with high reliability; (3) Each node in the network uses its MAC address as a unique identifier (node ID); (4) Each node in the network have a *black list* containing the misbehaving nodes, a *trust table* containing the learned network nodes' trust value; which are broadcasted to the node's neighbors periodically; and a *Data Packet Information (DPI) cache* to store information about the received and processed data or TCP-ACK packets.
In the network layer, a new node model is designed as the basis of our trust model. Some new fields are added into a node's routing table to store its trust value about other nodes and to record the positive and negative ratings when it performs routing with others.

3.2 Operation of TRIUMF

In TRIUMF we use AOMDV [20], or multipath DSR [21] to establish a two node-disjoints paths between the source and destination nodes.
But here we modified the RREQ packet to contain a list of unwanted nodes, which the source node doesn't want them to be a members on the discovered route temporarily, also the destination node may have this list (as seen below) and it discard all routes which contains this unwanted nodes.
Also during the RREQ flooding process the intermediate nodes will insert the previous node trust ratings (α, β) (defined later on) in the RREQ packet if the previous node trust value T (defined later on) less than the trust value contained in the RREQ packet. If the two trust values are equal then the node updates the trust ratings if it has a greater certainty factor f (defined later on).
When the destination node receive RREQ packet from multiple nodes, it select a two node disjoint paths with the highest path trust value, and certainty factor and unicasts two RREPs (contain the path trust rating) back to the source along the selected two routing paths.

$$T_P = \min\left(T_{S,n1}, T_{n1,n2}, T_{n2,n3}, \ldots, T_{ni,D}\right) \quad (1)$$

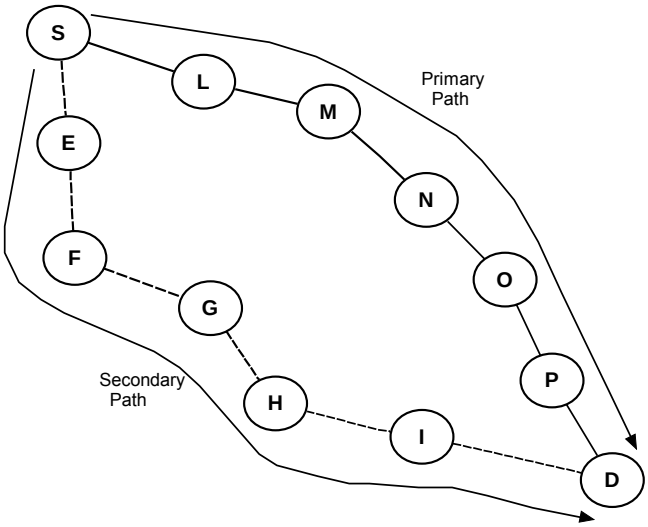


Fig. 1 two node disjoint paths between S and D

The source node mark the highest trusted route as primary used for data forwarding and the other as secondary used as a backup path, as shown in figure 1. The two node-disjoint routes are adopted to ensure reliable communication and search for malicious nodes.

3.2.1 Controlled selfishness behavior

The misbehaving node may be a selfish or malicious node, selfish node will hurt the network connectivity and it reported as malicious node in all reported trust based routing protocols. We use the observation of that, there is a different in needs of selfish and malicious nodes, where selfish node needs: (1) *to use network resources,* (2) *save its resources "drop any forwarded packet form other nodes and don't want to be a member in any new routes"*. But the malicious "dropper" node needs: (1) *to be a member in all new routes,* (2) *mount a denial of service attack by dropping the data packets it receives*.
Depend on the different needs of selfish and malicious nodes. We will allow some degree of selfishness for nodes to save their resources (e.g. battery power; where nodes behave differently based on their energy levels. When the energy lies between full energy E and a threshold E_s, the node behaves properly. For an energy level lower than the threshold E_s, it uses its energy for transmissions of its own packets).
A new field is inserted in the Hello packet containing the selfishness status. Each node use this filed in Hello packet to inform its direct neighbor nodes about its selfishness status, if it is in selfish mode all neighbor nodes will:

1- Remove it from the active routes, which it is an intermediate node on it, and send Route Error (RERR) packet to the sources to establish new routes.
2- Allow it to deny being a member in any new route, and dropping any Route Request (RREQ) packet came from it.

3- Forward to/from it the packets which contain it as destination/source address.

The selfish node neighbors will restrict its selfishness behavior by a time threshold, and a repetition threshold.

By allowing some degree of node selfishness the selfish node declare itself to its neighbor, and malicious node will not declare itself as selfish node because it inconsistent with its needs. So the selfish nodes are excluded from the responsibility of data forwarding. At the same time, this helps the identification of malicious nodes easier. Here we can differentiate between selfish and malicious nodes and save the misbehaving searching time (the time to find the misbehaving "selfish and malicious" node, and route around them) to only a searching time to find the malicious node only. We known that the misbehaving searching time need to be very small "i.e. find the misbehaving node very fast", because due to the node mobility the route life time is small.

3.2.2 Route monitoring tool

In our approach we use the DLL-ACK and the end to end TCP-ACK as a monitoring tool to monitor the behavior of the routing path, then use a path searching tool to search the misbehaving path to find the malicious node, and then put the malicious node ID in the black list to isolate it.

During the data transmission the source node send its data packet over the primary path only and each node in the path store the received data packet information in its Data Packet Information (DPI) cache, then forward it to its downstream neighbor, and wait for a data link layer acknowledgment (DLL-ACK) from the neighbor node, if it did not receive data link layer acknowledgement; it concludes that this neighbor node should be down. In such case, the neighbor is excluded from the node's routing table until it becomes up. However the neighbor's trust rating doesn't change.

On the other hand the source node waits to receive the end to end TCP-ACK from the destination node via the primary and secondary paths:

Case I : if there is no malicious node in the primary and secondary paths, then the source node receive the TCP-ACK over the two routes (primary and secondary) as shown in figure 2, or receive the TCP-ACK only over the secondary route.

Then the source node sends a biggy back Positive Trust Update Message (PTUM) upon transmitting the next packet. If the node which sent this message received an acknowledgment from a neighbor node in the data link layer and through this neighbor in the transport layer, and receive PTUM message from the source node. Then each node in the primary path will update the trust value of its upstream and downstream neighbors, and remove the information about the confirmed data packet from its DPI cache. Also destination node will send a biggy back PTUM message when transmitting the next TCP-ACK packet to the source node, to update the trust value of nodes in the secondary path.

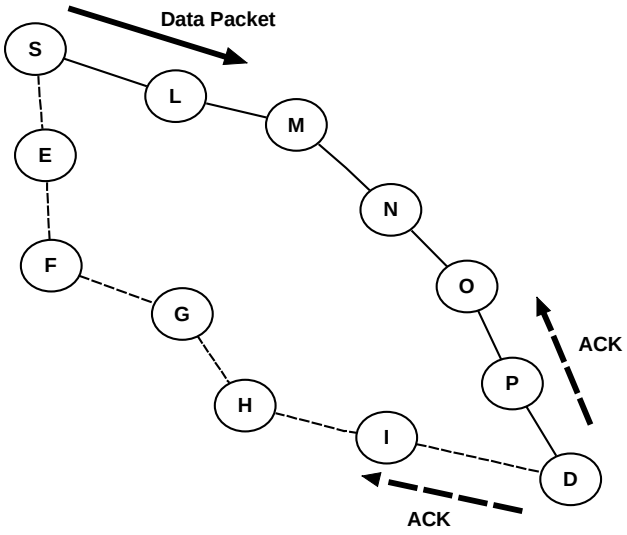


Fig. 2 The source node sends the data on the primary path only, and the destination replay with ACK on both primary and secondary paths

Case II: if the source node received an acknowledgment from a neighbor in the data link layer and receive an acknowledgment in the transport layer over the primary path only, even after retransmitting this message (TCP rules); it concludes that the neighbor node or one of its following nodes in the primary routing path may be malicious node trying to make blocking attack and send a faked TCP-ACK or there is a malicious node in the secondary routing path drop the TCP-ACK packet.

Case III: if the source node received an acknowledgment from a neighbor in the data link layer and did not receive an acknowledgment in the transport layer over the primary or secondary route paths, even after retransmitting this message (TCP rules). Then the source node knows that the data packet doesn't reach its destination, i.e. there is a malicious node in the primary path trying to make blocking attack.

In last two cases II and III the source node run the malicious search mechanism, to find the malicious node.

3.2.3 Route searching mechanism

If the source node concludes that there is a malicious node in the primary or secondary routs it will run the route searching mechanism by sending a Malicious Search Packet (MSP); which contains information about the lost data packet; via the primary route towered the destination node.

The MSP packet is a high priority packet, and every node receive this packet compare its information with the data packet information's stored in its DPI cache, if it found a match (the node received this data packet and forward it to the next node) it will forward the MSP packet to the next node with overhearing to assure that

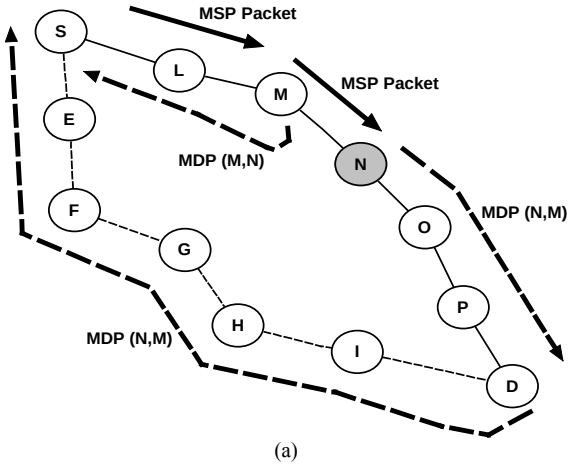
the neighbor node will forward it. The node which found a mismatch will stop forwarding of MSP packet and generate a Malicious Detection Packet "MDP (detecting node ID, detected node ID)"; it is a high priority packet forwarded with overhearing. Also the node which found that its downstream node doesn't forward the MSP packet generates the MDP packet. The node generating the MDP packet forwards it in the opposite direction to the detected malicious node, toward the source or destination node.
We make MSP and MDP high priority packets to speed up the detection process, and forwarded with overhearing to avoid the malicious node to drop that packets and break the searching and detection process.

Case I: if the primary path contains a malicious node, let node N is the malicious node. The source node sends the MSP packet to node L and overhears to be sure that node L will forward that packet. After comparison, node L forwards the MSP packet to M and overhears, then node M co mpares and forwards it to node N and overhears. The malicious node N has two choices:

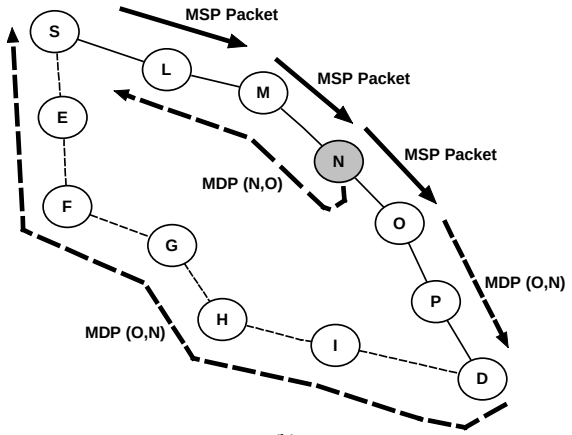
(1) It either, stops forwarding the MSP packet and report the destination node using MDP packet (N, M), that node M is the malicious node; node N deny the receiving of this data packet from node M. At the same time node M s ure that it forward this data packet to node N and receive a D LL-ACK from node N, where it didn't overhear node N forward MSP packet, then node M report the source node that node N is the malicious node; using MDP packet (M, N) as shown in figure 3-a.

(2) Or, it will forward the MSP packet to node O, then node O didn't find a match in its DPI cache, so it will send the MDP packet (O, N) to the destination node. At the same time the malicious node N can inform the source node that node O don't forward the MSP packet; send a M DP packet (N, O) as shown in figure 3-b.

In both cases when the source and destination nodes; and all nodes in the routing path; receive the MDP packet, they will update the trust value of both the detecting and detected node negatively and the trust value of other nodes in the routing path positively. Because the honest node (node M or node O) will suffer from the misbehaviors of malicious node N, so it will insert the malicious node N ID in its black list regardless of its trust score to prevent any future cooperation with it a nd isolate it from the network.



(a)



(b)

Fig. 3 Malicious node N (a) don't forward the MSP packet, (b) forward the MSP packet.

When the destination node receives the MDP packet it forwards it to the source via the secondary route, on the other hand if the source node doesn't receive a M DP from the destination node via the secondary route it will send the received MDP from the primary route to it until it can discard the suspect nodes from any future selected routes to that source. Finally the source node mark the secondary route as primary and start a r oute discovery phase to find a secondary node-disjoint route not containing both the detecting and detected node on it.

Case II: if the secondary path contains a malicious node, let node H is the malicious node. The source node sends the MSP packet to node L, and node L forward it to node M → N → O → P → to the destination. When the destination node receives the MSP packet, it will be sure that there is no malicious node in the primary route, and then the destination node will modify the MSP packet to contain the information of TCP-ACK packet and forward it to node I via the secondary path. Node I forward it a fter comparison to node H (the malicious node) the malicious node H has the same two choices as in case I as shown in figure 4.

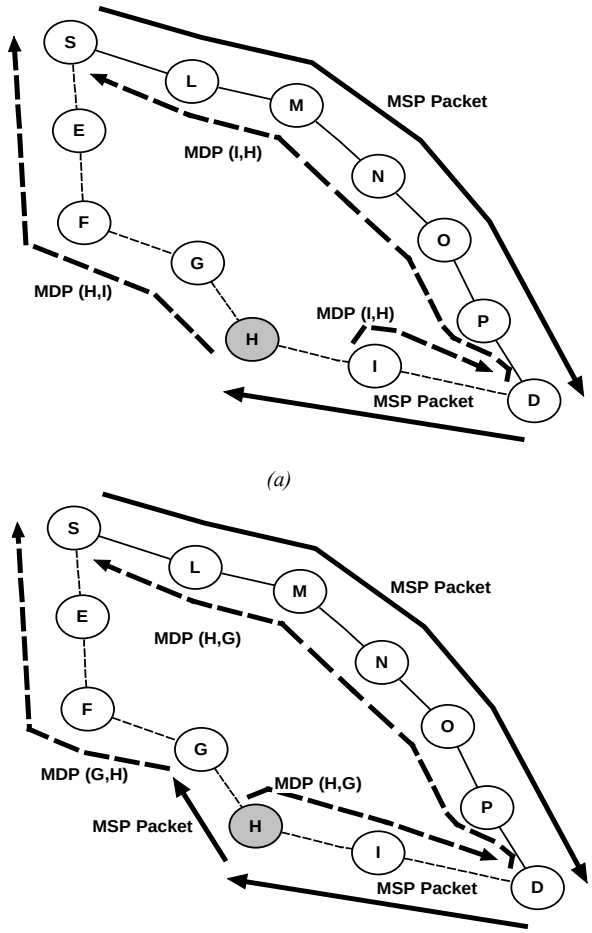


Fig. 4 Malicious node H (a) don't forward the MSP packet, (b) forward the MSP packet.

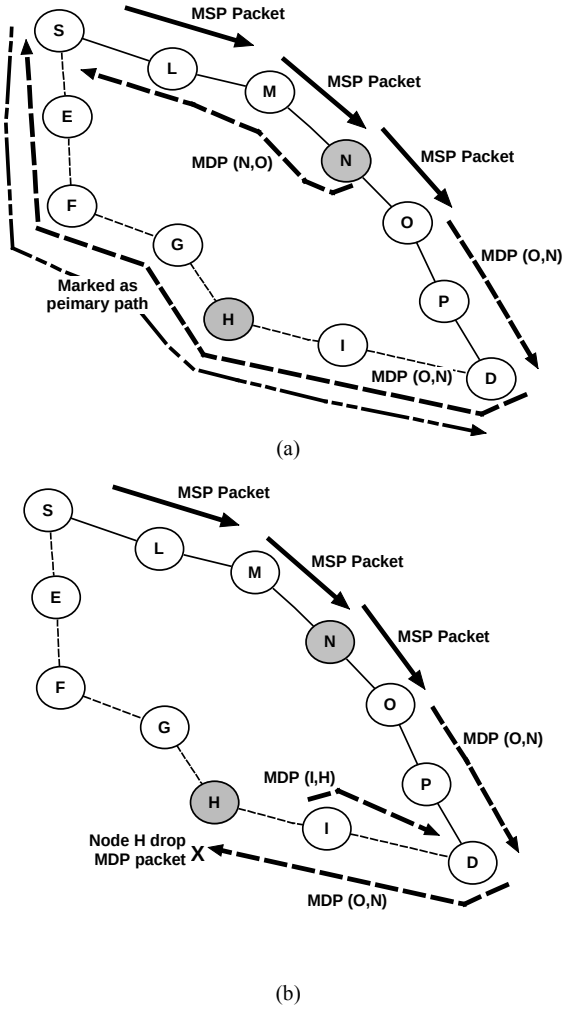


Fig. 5 The source node S (a) receives the MDP packet (b) don't receive the MDP packet.

When any node receives the MDP packet, it will update the trust values of both the detecting and detected node negatively, and the trust value of other nodes in the routing path positively. Also the detecting node (node I or node G), will insert node N ID in its black list; regardless of its trust value; to reject any future cooperation between them.

When the destination node receives the MDP packet it forwards it to the source via the primary route, on the other hand if the source node doesn't receive a MDP from the destination node via the primary route it will send the received MDP from the secondary route to it until it can discard the suspect nodes from any future selected routes to that source. Finally the source node starts a route discovery phase to find a secondary node disjoint route not containing both the detecting and detected node on it.

Case III: if both the primary and secondary routes contain malicious nodes. The source node sends the MSP packet via the primary path and waits the MDP packet, if:

(1) The primary path malicious node send the MDP packet (N, O) to the source node, on the other hand the destination node receive an MDP packet (O, N) from node O, then it will forward it to the source node via the secondary path. If the secondary path malicious node forward the MDP packet (O, N), then the source node receive the MDP packet (O, N) and mark the secondary path as primary and search for new secondary path don't contain both the detecting and detected nodes (M, N), as shown in figure 5-a. Because the new primary route contain also a malicious node, then the source node don't receive TCP-ACK packet from the destination, so it start a new malicious search procedure to find the malicious node. When finding the new malicious node the source node marks the new secondary route as primary and search a new secondary route, and so on.

(2) The primary path detecting node send the MDP packet (O, N) to the destination node, then the destination node send the MDP packet (O, N) to the source node via the secondary path. Figure

5-b shown that there is a malicious node in the secondary path drop the MDP packet (O, N), and don't reach the source node. When node I found that node H drops the MDP packet (O, N), it will send a new MDP packet (I, H) to the destination node. When the source node doesn't receive the MDP packet from both the primary and secondary paths, it will try to run the malicious search again, and if it doesn't receive the MDP again it conclude that there are a malicious nodes in the primary and secondary paths. So the source node flooding a RREQ toward the destination node, and when the destination node receive the RREQs it delete the paths contains nodes (O, N, I, H) and select the two highest trusted paths.

3.2.4 Malicious node isolation

When a neighbor of a malicious node detect its malicious activity it will send the MDP packet and put the malicious node ID on its black list to isolate it. Also when the trust value of a given node reduced below a given threshold δ it will be marked as misbehaving node and its ID inserted in the black list.
After small number of transaction all malicious node's neighbors will put its ID on their black lists, so the malicious node will be fully isolated from MANET. The misbehaving node can rejoin the network only if it moves from its location and have new neighbors (whose ask the old neighbors about the node reputation), and if its reported trust value is above the trust threshold δ.

4. Trust Model

In this section, we briefly introduce the proposed trust model adopted in the secure routing, which is based on the beta probability density function. The beta probability density function has already been used in trust and reputation systems to calculate trust value as in [22], [23], [24]. Will also use it to calculate a new value called the cooperation score of selfish node, and to find a threshold to decide if a node is trustworthy or untrustworthy.
In our model, a node i (deciding node) maintains three kinds of ratings about every neighboring node j (suspect node). In the scheme, the node calculates the cooperation score $C_{i,j}$ and direct trust value $T_{i,j}^{D}$ according to a distribution (called the "prior") which is updated as soon as new observations are made, and a reported rating from neighboring nodes (indirect trust value $T_{i,j}^{I}$). This three kinds of rating are used to calculate the overall trust value $T_{i,j}$.
Computing direct trust value and cooperation score are based on the statistical updating (i.e. positive rating; α, or negative rating; β) of beta probability density function [25]. The probability density function of the overall trust value T on the interval (0, 1), can be represented to have a Beta distribution as in [25]:

$$\mathbf{f}(T \mid \alpha, \beta) = \frac{\Gamma(f)}{\Gamma(\alpha)\Gamma(\beta)} T^{\alpha-1} (1-T)^{\beta-1} \qquad (2)$$

Where
α: is the positive trust rating ($\alpha > 0$), with initial value =1
β: is the negative trust rating ($\beta > 0$), with initial value =1
} Observation
T: is the trust value $0 \le T \le 1$
$f = \alpha + \beta$, is the certainty factor where the larger the sum $(\alpha + \beta)$, the more certainty about the trust value [26], [27].

Moreover, the advantage of using the Beta function is that it only needs to update two parameters as soon as observations are made or reported.

To estimate the trust value by using Bayes rule which minimize the posterior expected value of loss function, loss can be represented as squared-error loss [26], [27], [28], where the Minimum Mean Square Estimate (MMSE) determine the estimated value of trust which minimize the estimation error is,

$$T = E[f(T \mid \alpha, \beta)] = \frac{\alpha}{\alpha + \beta} \qquad (3)$$

Where $T = 1$ means blind trust and $T = 0$ means fully distrust.

4.1 Cooperation score

The cooperation score used to measure the unselfishness behavior of a suspect node j by a neighbor deciding node i during the periodical time t, is given by:

$$C_{i,j} = E\left[f\left(C_{i,j} \mid S_{i,j}^{+}, S_{i,j}^{-}\right)\right] = \frac{S_{i,j}^{+}}{S_{i,j}^{+} + S_{i,j}^{-}} \qquad (4)$$

Where:

$\boldsymbol{S_{i,j}^{+}}$: is the positive cooperation rating and equals the number of periodical time t the suspect node j is observed unselfish by the deciding node i. where if the timer t expired with unselfishness behavior from the suspect node j, then the deciding node i update the value of positive cooperation rating as:

$$S_{i,j}^{+}|_{new} = u\, S_{i,j}^{+}|_{old} + 1 \qquad , S_{i,j}^{+}|_{0} = 1 \qquad (5)$$

Where the weight u is a discount factor or forgetting factor [25], [29], [30], it aims to allow a well behaving node to improve its cooperation score and trust value after it has been observed to be selfish due to low level of battery power or drop packets due to mobility or collisions [30], and to overcome the permanent exclusion of an innocent neighbor that may later discover the true misbehaving node, and $0 \le u \le 1$ [25].

$\boldsymbol{S_{i,j}^{-}}$: is the negative cooperation rating and equals to the count of the suspect node *j* selfishness behavior observed by the deciding node *i*, and it is updated using:

$$S_{i,j}^{-}|_{new} = u\, S_{i,j}^{-}|_{old} + 1 \qquad , S_{i,j}^{-}|_0 = 1 \tag{6}$$

Where the negative cooperation rating is updated if the suspect node *j* observed in selfishness behavior over the expected selfishness time period t_s; which is a node dependant timer depend on the battery power and node type; or the selfishness rate R_s exceed a given threshold $R_{s,max}$

$$R_s = \frac{no.\ of\ selfishness\ times}{observation\ time/t_s} < R_{s,max} \qquad , R_s \geq 0 \tag{7}$$

Here we can control the degree of acceptable selfishness behavior by controlling the values of t_s and $R_{s,max}$, to avoid normal nodes with sufficient energy to intentionally report low power level to avoid forwarding packets for other nodes.

4.2 Direct trust value

During the data forwarding phase, if the transaction on the rout is successful the route member node *i* will update the neighbor nodes direct trust positive rating,

$$\alpha_{i,j}^{D}|_{new} = u\, \alpha_{i,j}^{D}|_{old} + 1 \qquad , \alpha_{i,j}^{D}|_0 = 1 \tag{8}$$

and if it receive a MDP packet, it will update the direct trust negative rating of the detected and detecting nodes,

$$\beta_{i,j}^{D}|_{new} = u\, \beta_{i,j}^{D}|_{old} + 1 \qquad , \beta_{i,j}^{D}|_0 = 1 \tag{9}$$

In addition, whenever the periodic time t expires, we let $\alpha_{i,j}^{D}|_{new} = u\, \alpha_{i,j}^{D}|_{old}$ and $\beta_{i,j}^{D}|_{new} = u\, \beta_{i,j}^{D}|_{old}$ to decay the values of $\alpha_{i,j}^{D}$, $\beta_{i,j}^{D}$. This is to allow for redemption even in the absence of observations, either due retaliatory exclusion or simply lack of interaction.

So the direct trust value of a suspect node *j*, as observed during the periodic time *t* is,

$$T_{i,j}^{D} = E\left[f(T_{i,j}^{D} \mid \alpha_{i,j}^{D}, \beta_{i,j}^{D})\right] = \frac{\alpha_{i,j}^{D}}{\alpha_{i,j}^{D} + \beta_{i,j}^{D}} \tag{10}$$

4.3 Indirect trust value

Since a deciding node *i* have not inevitably a neighborhood relationship with all nodes on a route, it is sometimes needed to derive indirect trust values using recommendations, where the deciding node *i* can decide over the trust value of suspect node *j* by taking recommendations from other *K* nodes which may be neighbors of *j*, the reported trust rating $f\left(T_{k,j}^{R} \mid \alpha_{k,j}^{R}, \beta_{k,j}^{R}\right)$ weighted by the overall trust value from nod *i* to node *k* "$T_{i,k}$ " ,used to calculate the indirect trust value:

$$T_{i,j}^{I} = \frac{\sum_{k=1}^{K} T_{i,k}\, T_{k,j}^{R}}{K} = \frac{\sum_{k=1}^{K} \frac{T_{i,j}\, \alpha_{k,j}^{R}}{\alpha_{k,j}^{R} + \beta_{k,j}^{R}}}{K} \tag{11}$$

We included this functionality to reduce the chances of a malicious node misreporting its peers. Since a malicious node itself has low trust, its recommendation (which could be intentionally crafted to degrade the system) will not be given much importance by the node receiving the respective recommendations.

4.4 Overall trust calculation

The overall trust value of the suspect node *j* as calculated by the deciding node *i* " $T_{i,j}$ " represents the node *i*'s trust score during the periodical time *t*, which ranges from 0 t o 1, $T_{i,j} < \delta$ denote the node is untrustworthy, $T_{i,j} \geq \delta$ denote the node is trustworthy. Where δ is the threshold that is used to make decisions about other nodes.

$$T_{i,j} = E\left[f(T_{i,j}^{s} \mid \alpha_{i,j}^{s}, \beta_{i,j}^{s})\right] + T_{i,j}^{I} = \frac{\alpha_{i,j}^{s}}{\alpha_{i,j}^{s} + \beta_{i,j}^{s}} + T_{i,j}^{I} \tag{12}$$

Where $T_{i,j}^{s}$is the trust value including both cooperation score and direct trust value, the positive rating $\alpha_{i,j}^{s} = \alpha_{i,j}^{D} + w\, S_{i,j}^{+}$, the negative rating $\beta_{i,j}^{s} = \beta_{i,j}^{D} + w\, S_{i,j}^{-}$, and *w* is a weight determining the importance of selfishness.

4.5 Decision making

Using of the overall trusted value the deciding node must decide if the suspect node is trustworthy or untrustworthy node by using the trust value threshold δ. By using the simple binary hypothesis test to calculate the trust value threshold δ.

With $\mathbf{r_N} = (\boldsymbol{\alpha_N}, \boldsymbol{\beta_N})$ are the observations, N=1,2,3,4,….. , and **R** is the observation vector. The likelihood ratio test [28]

$$\frac{P(R|T^{+})}{P(R|T^{-})} \begin{matrix} trusted \\ > \\ < \\ untrusted \end{matrix} \frac{P(T^{-})(C_{10} - C_{00})}{P(T^{+})(C_{01} - C_{11})} \tag{13}$$

Where C_{00}: is the cost of correct decision of untrustworthy node.
C_{11}: is the cost of correct decision of trustworthy node.
C_{01}: is the cost of incorrect decision of trustworthy node.
C_{10}: is the cost of incorrect decision of untrustworthy node.
T^{+}: Trustworthy
T^{-}: Untrustworthy

Let $C_{10} - C_{00} = C_{01} - C_{11}$, then

$$\frac{P(R|T^+)}{P(R|T^-)} \underset{untrusted}{\overset{trusted}{\gtrless}} \frac{P(T^-)}{P(T^+)} \quad (14)$$

By using Bayes rule

$$P(R|T^+) = \frac{P(T^+|R)P(R)}{P(T^+)} \quad (15)$$

From (15) into (14) the likelihood ratio test become

$$\frac{P(T^+|R)P(R)}{P(T^+)} \Big/ \frac{P(T^-|R)P(R)}{P(T^-)} \underset{untrusted}{\overset{trusted}{\gtrless}} \frac{P(T^-)}{P(T^+)} \quad (16)$$

$$\frac{P(T^+|R)}{P(T^-|R)} \underset{untrusted}{\overset{trusted}{\gtrless}} 1 \quad (17)$$

Because

$$P\ (T \mid \alpha, \beta\ ) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\ T^{\alpha-1}\ (1-T)^{\beta-1} \quad (18)$$

By inserting (18) into (17)

$$\frac{(T^+)^{\alpha-1}\ (1-T^+)^{\beta-1}}{(T^-)^{\alpha-1}\ (1-T^-)^{\beta-1}} \underset{untrusted}{\overset{trusted}{\gtrless}} 1 \quad (19)$$

Because $T^+ = 1 - T^-$, $0 \le T^+ \le 1$

$$\frac{T^+}{1-T^+} \underset{untrusted}{\overset{trusted}{\gtrless}} 1 \quad (20)$$

Finally we have

$$T^+ \underset{untrusted}{\overset{trusted}{\gtrless}} 0.5 \quad (21)$$

Then δ = 0.5

Where from equation (4) $T^+ = \frac{\alpha}{\alpha+\beta}$, so:

$$\frac{\alpha}{\alpha+\beta} \underset{untrusted}{\overset{trusted}{\gtrless}} 0.5 \quad (22)$$

The larger the certainty factor $f = (\alpha + \beta)$, the more certainty about the trust value (i.e. if there are two trust values ${T_1}^+ = {T_2}^+ > 0.5$ with number of observations $(\alpha_1 + \beta_1) \ll (\alpha_2 + \beta_2)$, so the two nodes will decide to be legitimate nodes, but the observation made for node 1 is not enough for this decision (we not sour about the malicious activity of this node). In other hand the observations made for node 2 i s enough for this decision) [26], [27]. So the trust value threshold δ must be flexible directly proportional to *f*, and $0 \le \delta \le 0.5$. We use the negative exponential to describe this relation

$$\delta = 0.5\left(1 - e^{-(\alpha+\beta)}\right) = 0.5(1 - e^{-f}) \quad (23)$$

5. Discussion

From performance point of view, our trusted routing protocol allows the malicious node neighbors to isolate it after a few numbers of transactions. And allow a controlled amount selfishness behavior for nodes to give an incentive to the selfish nodes to declare its selfishness behavior to its neighbor nodes, which reduce the searching time of misbehaving nodes (malicious and selfish nodes) to search for malicious nodes only. Then we can find an isolate the malicious node; denied access to the network; in small amount of time, which resulting in an improved overall throughput performance for MANET.

From security point of view, our design will detect nodes' misbehavior finally and reduce the harms to the minimum extent, and select the highest trusted route. When a good node is compromised and becomes a Black Hole, its misbehavior will be detected by its neighbors. Then with the help of forgetting factor used in our trust model, the trust value from the other nodes to this node will be updated shortly, and this node will be isolated. Also, we allow a w ill behaving node to improving its trust value after it has been observed to be selfish or drop packets due to mobility or collisions and to overcome the permanent exclusion of an innocent neighbor that may later discover the true misbehaving node.

From flexibility point of view, our trust model gives each node a flexibility to define the trust value threshold δ. The default trust value threshold is 0.5, which can be varied by a node; $0 \le \delta \le 0.5$; according to the value of certainty factor *f*.

6. Conclusion and Future work

In this paper we proposed a general solution to packet dropping misbehavior in mobile ad hoc networks. The solution allows monitoring, detecting, and isolating of the droppers in short time without using promiscuous listening, and can differentiate between selfish and malicious nodes. In our trust routing protocol, nodes can cooperate together to perform trusted routing behaviors according to the trust relationship among them; to route around the misbehaving node. With the trust value threshold, nodes can flexibly decide whether its neighbor is a malicious node or not according to the value of certainty factor.

In the future we will simulate the proposed trust routing protocol to show the results and effectiveness of our solution, and compare it with existing trust based routing protocols like TAODV, TWOACK, and TDSR protocols. A detailed simulation evaluation will be conducted in terms of *Routing Packet Overhead*, *Security Analysis*, *Mean Time to detect dropper node*, *Overall Network Throughput*, and *Average Latency*. Also we will study the situation when there are more than one malicious node in the route from the source and destination, with asymmetry of communication link in both directions.

References

[1] D. Johnson and D. Maltz, "*Dynamic Source Routing protocol (DSR) for Mobile Ad Hoc Wireless Networks for IPv4*," RFC 4728, February 2007. [Online] Available: http://www.ietf.org/rfc/rfc4728.txt.

[2] Perkins CE, Belding-Royer E, Das SR. "*Ad hoc on-demand distance vector (AODV) routing*, " RFC 3561, July 2003. [Online] Available: http://www.ietf.org/rfc/rfc3561. txt.

[3] C.E. Perkins and P. Bhagwat, "*Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*," Proc. ACM SIGCOMM, pp. 234-244, 1994.

[4] B. Vaidya, S. S. Yeo , D.-Y. Choi , S. Jo Han, " *Robust and secure routing scheme for wireless multihop network,* " Personal and Ubiquitous Computing magazine , 4 April 2009. © Springer-Verlag London Limited 2009.

[5] Y-C. Hu, A. Perrig, and D. Johnson, "*Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*, " In Wireless Networks Journal 11, pp.21-38, 2005.

[6] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Neil Levine, C. Shields, and E. M. Belding-Royer, "*Authenticated Routing for Ad Hoc Networks,* " Proceedings of IEEE journal on selected areas in communications, Volume 23, No. 3, March 2005.

[7] Dongbin Wang, Mingzeng Hu, Hui Zhi, "*A Survey of Secure Routing in Ad Hoc Networks*," The Ninth International Conference on Web-Age Information Management (waim), pp. 482-486, 2008.

[8] Yih-Chun Hu, Adrian Perrig, "*A Survey of Secure Wireless Ad Hoc Routing*," IEEE Security and Privacy, vol. 2, no. 3, pp. 28-39, Mar. 2004.

[9] Marianne A. Azer, Sherif M. El-Kassas, Abdel Wahab F. Hassan, Magdy S. El-Soudani, "*A Survey on Trust and Reputation Schemes in Ad Hoc Networks*," Third International Conference on Availability, Reliability and Security 2008, pp. 881-886.

[10] I Aad., , J.-P. Hubaux, E.W. Knightly, "*Impact of Denial of Service Attacks on Ad Hoc Networks*, ", IEEE/ACM Transactions on Networking, Volume 16, Issue 4, pp. 791 – 802, Aug. 2008.

[11] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," in Proceedings of Mobicom, 2004.

[12] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants," in Proceedings of ACM SIGCOMM, 2003.

[13] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "*Mitigating routing misbehavior in mobile ad h oc networks,*" in Proceedings of the Sixth annual ACM/IEEE International Conference on M obile Computing and Networking, 2000, pp. 255–265.

[14] S. Buchegger and J.-Y. LeBoudec, "*Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks)*, " In Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking & Computing (MobiHoc 2002), pp. 226–236, Lausanne, Switzerland, June 2002.

[15] X. Li, M. R. Lyu, J. Liu, "*A Trust Model Based Routing Protocol for Secure Ad Hoc Networks*". In the Proceedings of IEEE Aerospace Conference (IEEEAC) 2004, pp. 1286-1295.

[16] M. G. Zapata and N. Asokan, "*Securing ad hoc routing protocols*," in Proceedings of ACM Workshop on Wireless Security (WiSe '02). Atlanta, USA: ACM Press, September 2002, pp. 1–10. [Online] Available: http://doi.acm.org/10.1145/570681.570682.

[17] K. Balakrishnan, J. Deng, and P.K. Varshney. "*Twoack: preventing selfishness in mobile ad h oc networks*,". In The IEEE Wireless Communication and Networking Conference(WCNC'05), pp. 2137-2142, New Orleans, LA,USA, March 2005

[18] Cheng Yong; Huang Chuanhe; Shi Wenming" *Trusted Dynamic Source Routing Protocol*," Wireless Communications, Networking and Mobile Computing (WiCom), Sept. 2007 pp. 1632 – 1636

[19] Muhammad Zeshan, Shoab A. Khan, Ahmad Raza Cheema, Attique Ahmed, "*Adding Security against Packet Dropping Attack in Mobile Ad Hoc Networks*," International Seminar on Future Information Technology and Management Engineering 2008, pp. 568 – 572.

[20] Mahesh K. Marina, and Samir R. Das "*Ad hoc on-demand multipath distance vector routing*," Wirel. Commun. Mob. Comput. 2006; pp. 969–988, Published online in Wiley InterScience (www.interscience.wiley.com).

[21] Nasipuri, A. Das, S.R. " *On-demand multipath routing for mobile ad hoc networks*," Computer Eight International Conference on C ommunications and Networks 1999, pp. 64–70.

[22] Dorothea Wagner, and Roger Wattenhofer, "*Algorithms for Sensor and Ad Hoc Networks*," Theoretical Computer Science and General Issues , Vol. 4621, 2007. © Springer-Verlag Berlin Heidelberg 2007

[23] Yufeng Wang, Yoshiaki Hori, Kouichi Sakurai " *On securing open networks through trust and r eputation - architecture, challenges and solutions*," IEICE Tech. Rep., vol. 106, no. 340, AI2006-12, pp. 1-6, Nov. 2006.

[24] Dimitri Melaye, Yves Demazeau " *Bayesian Dynamic Trust Model*," CEEMAS 2005, LNAI 3690, pp. 480–489, 2005. © Springer-Verlag Berlin Heidelberg 2005.

[25] A. Jøsang and R. Ismail "*The Beta Reputation System*," In Proc. of the 15th Bled Conference on E lectronic Commerce, Bled, Slovenia, June 2002, pp. 324-37.

[26] J. Mundinger and J.-Y. Le Boudec " *Analysis of a reputation system for Mobile Ad-Hoc Networks with liars,"* Performance Evaluation, Volume 65 , Issue 3-4, March 2008, Pages: 212-226 . © Elsevier Science Publishers B. V.

[27] Sonja Buchegger, Jean-Yves Le Boudec " *The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks*," in: Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2003 (WiOpt '03).

[28] Harry L. Van Trees "*Detection, Estimation, and Modulation Theory, Part I*," © Wiley-Interscience Publishers; 1 edition September , 2001.

[29] Yihui Zhang, Li Xu and Xiaoding Wang, " *A Cooperative Secure Routing Protocol based on Reputation System for Ad Hoc Networks,* " Journal of Communications, VOL. 3, NO. 6, pp. 43 -50, November 2008. © 2008 Academy Publisher.

[30] D. D jenouri, O. Mahmoudi, M. Bouamama, D. Llewellyn-Jones, M. M erabti, "*On Securing MANET Routing Protocol Against Control Packet Dropping* , " IEEE International Conference on Pervasive Services, July 2007, pp. 100-108.

Biography

Eng. Ahmed M. Abd El-Haleem is a Teacher Assistant in Communication department at Helwan University, was born in Egypt in 1979. He obtained his B.Sc., and M.Sc. degrees from Helwan University, Egypt in 2001 and 2006 respectively. His research interests include Computer Networks, and Secure Routing Protocols.

Dr. Ihab Ali, was born in Egypt in 1962. He obtained his B.Sc., M.Sc. and P h.D. from Helwan University, Egypt in 1985,1991 and 1997 respectively. He is a senior member of IEEE. His research interests include Computer Networks, Network Security and S ecure Routing Protocols.