EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

EigenTrust-based non-cooperative game model assisting ACO look-ahead secure routing against selfishness

Surendran Subbaraj[1*] and Prakash Savarimuthu[2]

Abstract

In mobile *ad hoc* networks (MANETs), selfish behavior is observed when nodes fail to forward data packets which are actually intended. This is generally assumed to be a kind misbehavior which might interrupt the network operations. Here, we propose a QoS-constrained EigenTrust-based non-cooperative game model for secure fault-tolerant ant look-ahead routing which attempts to identify trusted valid route and look-ahead route pairs which might help in choosing the alternate path in case of valid route failure. Simulation results illustrate that the proposed trust-based secure routing is able to accurately distinguish malicious nodes from good nodes with a limited overhead.

Keywords: Trust; ACO; Secure routing; Fuzzy; EigenTrust

Introduction

A mobile *ad hoc* network (MANET) is a decentralized, infrastructure-less network where wireless nodes move arbitrarily. They are being widely used for military applications, wearable devices, and rescue operations and in places where there is no pre-installed infrastructure. They are continuously changing and self-configuring networks. In a dynamic network, it is difficult to use multimedia and other advanced applications without quality-of-service (QoS) constraint. QoS shall be defined as the bundle of service primitives to be met while a network is in operation. In MANETs, designing a routing algorithm with given QoS constraint is NP-hard because of the unavailability of accurate path information and it is difficult to keep up-to-date information about the link owing to its dynamic nature and depletion of energy at node which causes link breakage.

Trust [1] is defined as a degree of belief about the behaviour of other entities. The nodes participating in data exchange should be safeguarded by trust and reputation mechanisms or else they could be attacked which might end up in unnecessary resource consumption of the entire mobile network. Attack might be direct or indirect, i.e., intruders might take charge of good nodes which result in non-cooperation leading to network destruction. Therefore, such nodes prone for compromise need to be identified *via* trust and reputation mechanisms in advance so that the network is safe for ever. The reason is that MANETs lack central administrative control due to wireless set up and that will serve as the prime concern since it is easy for attackers to eavesdrop the packets, and tamper or falsify them.

Mobility [1] is a critical factor in military applications as missions will start at a certain coordinate and will end up at another, and tracking the positions of soldiers is most compelling. Self-formation of units is the speciality in such applications. The participating nodes might be one among these: soldiers with wireless devices and unmanned vehicles or jets. Images, voice and video are the most frequent data exchanged in such network. Therefore, QoS is of prime concern. Any delay or false message delivered might lead to worst after-effects. Therefore, reliability of message transfers is of utmost importance before a packet reaches its destination. Since performance criteria are strictly related with time to be delivered, there are wide chances of attackers falsifying the message packets. Identifying such intruders and malicious nodes and isolating

* Correspondence: suren.subbaraj@gmail.com
[1]Department of Information Technology, Tagore Engineering College, Chennai 600127, India
Full list of author information is available at the end of the article

them from the network is a herculean task in a wireless setup, but which leads to reliable network operation.

Multipath routing protocols are widely used in such requirements. These are protocols which discover and store more than one route in their routing tables. Even if one route is broken, the other alternate is readily available for future use. Multipath routing protocols are employed to increase network reliability and fault tolerance. They also provide load balancing, which might help in reducing the congestion over specific routes. Since nodes always depend on neighbours in packet forwarding, establishing routes and tracing these established routes has to be done frequently.

In an adverse MANET setup, both route establishment and data transmission are vulnerable to a variety of attacks. Misbehaving nodes could disturb route discovery by impersonation or by responding with false route information. This may indirectly transfer the entire network control into the hands of intruders. Therefore, in order to provide complete security, protocols might be helpful. Reliable transport protocols are plainly insufficient to serve the above purpose. The attack and the aftermath is far beyond the limit of such protocols.

In this paper, we attempt to develop a fault tolerant and secure routing algorithm based on ant colony optimization algorithm with the following features:

- The fault-tolerant algorithm should have an effective route failure-handling mechanism to ensure the integrity of the network.
- Integrity should be maintained even in the events of congestion, bottlenecks or broken links which are prone to happen under a highly dynamic condition in MANET.
- Whenever a path breaks, the algorithm should try to use an alternate path, instead of initiating a new route discovery.
- The secure routing algorithm addresses the security issues by incorporating the concept of trust-based reputation mechanism to overcome the misbehaving entities.
- Trust evaluation using the EigenTrust fuzzy system helps to make routing decision for secure data transmission.

Due to dynamism of wireless setup, it is very difficult to validate all the route messages. The following may be the issues and potential solutions:

- Developing a fault-tolerant routing in case of route failures or node failures
- QoS metrics such as packet delivery ratio, throughput and delay should be considered to achieve a QoS-constrained routing
- Developing a secure routing by trust-based reputation mechanism which evaluates the trust worthiness of a node in order to continue the data forwarding along that node

Trust evaluation using the EigenTrust fuzzy system helps to make routing decision for secure data transmission.

Related work

Intelligent fault-tolerant routing

More intelligent approach to network routing [2-4] involves optimisation in route discovery and maintenance process. Swarm-based algorithms [5] and in particular, ant colony-based algorithms use the biotic techniques as meta-heuristic factors [6] in deciding the valid routes. Though these algorithms may result in approximately correct routes initially, the learning capability of these algorithms [7,8] makes them really adaptive to the MANET dynamism and is effective in a long-term basis. However, location-based ant colony optimization (ACO) routing algorithms are very much impractical and do not perform like other location-unaware routing protocols [9]. But routing decisions taken using ACO to learn the location would be better only with a moderate and consistent dynamism in the network topology.

Kwang et al. [10] proposed that ants are relatively small, and therefore can be piggybacked in data packets. It will not cost more to do frequent transmission of ants in order to provide updates of routing information for solving link failures. Hence, using ACO for routing in a dynamic network seems to be appropriate. Routing in ACO is achieved by transmitting ants rather than routing tables unlike fault-tolerant QoS-guaranteed routing algorithms. A fault-tolerant routing protocol [11] using a greedy ACO routing mechanism chooses only a single best path to destination. This routing achieves high packet delivery ratio and throughput ignoring the packet loss. Learning automata-based fault-tolerant routing algorithm [12] involves a learning automata perspective on handling the factors of fault tolerance. Failure in excluding fault-prone nodes may end up in a situation where the faulty nodes might act as routers and not participate in taking routing decisions with no packets forwarded to the network at last. Therefore, a faulty node has to exclude itself during the route discovery stage.

FTAR [11] introduces the notion of ‘worker ant-like control packets’, which are assigned the task of identifying the faulty routes from the existing set of valid routes. This protocol allows the option of sending the special control packets both reactively or proactively. The decision of reactive or proactive sending depends on the current load of the source node. In general, the proactive approach is chosen only when the source node has lesser than normal load with the aim of establishing correct routes to destination in future. Reactive approach involves frequent sending of control packets periodically which is determined by the handled traffic at the source node.

Misra et al. [13] used an ACO-based framework [14,15] for finding out the suitable path for routing packets. In their paper, they presented an algorithm FTAR which uses control packets called ants for acquiring routing information and are generated continuously by nodes in the network. These control packets deposit pheromone (control information) on each node, similar to pheromone deposited by real ants on the path they travel which is used for routing of packets.

ARA [16] algorithm and other ACO-based routing algorithms [16-19] use indirect information as a basis for finding valid routes. Like the pheromone traces of ants, some features of the MANET traffic *via* the nodes are used as environmental traces out of which the heuristics for the validity of the routes based on various QoS factors could well be studied. These algorithms use control packets called ants, which acquire routing information through sampling of paths. These ants are generated in a concurrent and completely independent manner. These are generated with the aim to test a path to an assigned destination. As an ant moves from the source node to the destination node, it collects information about the path, and uses this on its way back from the destination to the source. The ants also deposit pheromone to help future ants in the decision-making process. Each node contains a routing table which deposits the information collected by the ants during the forward and backward process.

DAR [20] is an alternate version of existing ant colony routing algorithms. It is designed with the goal to minimize computational complexity involved in creating routes from source to destinations. In other words, ACO routing algorithms take route decisions optimally over the full length of the paths to destination. HOPNET [21] and DAR [20] take hop-by-hop optimal decisions to forward the FANTS such that, attempts to finding a new route results in optimal routes to destination since every hop is examined for optimality. This results in a globally optimal route which is actually devised using local hop-by-hop information. Considering the node dynamism, this algorithm creates optimal routes but not in optimal time. However, missing of existing nodes or addition of new nodes on the path shall be reflected into the route tables more frequently than any other ACO routing protocols since every hop is examined for optimality. Diverse routing protocols [22] find two maximal shared-risk link group (SRLG)-disjoint routes to destination. However, though the routing overhead is minimal, the time and effort required to identify maximal disjoint routes is enormous, and as the network increases in size, this time spent for finding maximal disjoint routes also increases. In addition, to overcome the approximation and bandwidth problems of ACO routing algorithms, Khosrowshahi et al. [23] proposed a novel protocol where the node is switched between local and global zone while taking routing decisions.

Most ACO-based routing algorithms identify and apply all possible '*n*' paths, which degrade the performance of multipath routing algorithm [24,25]. As the number of possible routes increases, the relative performance of ACO multi-path algorithms also increases, but to a certain extent. Beyond this, due to numerous routes available and the route updates and routing tables getting populated and accumulated with more and more real-time information, the network performance degrades to a visible level. However, Misra et al. suggest some alternatives to make the process energy-aware [26]. Choosing only '*k*' path among the *n* available paths where $k < n$ would be a wise decision [27]. However, choosing the *k* factor during real-time routing requires the knowledge of *n* and then to determine a subset *k* from *n*. This is like wasting time and energy for acquiring real-time information of $k + 1$ to *n* paths and to drop them after deciding upon *k*. Another alternate approach is fixing the value of *k* by trial and error. This requires the network to route for a period of time and, based on the history, *k* shall be decided. However, this idea only uncovers the past mistakes and therefore, there is no guarantee for future surprises. Due to the dynamic topology of MANETs, it is very common that frequent route updates consume network bandwidth and node capacity. Therefore, algorithms which are practical in learning the QoS constraints as the network is in operation [28] are more essential.

Watchdog mechanism

In this method, a node will be chosen to supervise all its local neighbours. Without much overhead, this method would be able to detect malicious and selfish behaviour attacks. The watchdog mechanism proposed by Angelo et al. monitors its neighbour nodes by reading the received messages in order to ensure that the messages are forwarded without alteration [29]. With this observation, every node in the network shall be identified for its trustedness.

Misbehaviour detection and trust management

Since in MANET, all network operations purely rely on node cooperation, if some nodes tend to behave selfishly, then this would affect the entire network performance. There may be interruptions caused in providing network services if such things happen often. Nodes which do not cooperate to provide collective network performance are termed as selfish nodes [30]. Besides this, malicious nodes may also compromise internal nodes to behave selfishly [31]. Therefore, trust assignment and management [32,33] across the nodes of the network is mandatory for monitoring secure data exchange [34,35].

The trust management framework proposed by Ghorpade [36], does the above-mentioned activities by employing trust agent and recommendation agent. Trust agent is responsible for determining the trust of participating nodes based on the network events that happened. The recommendation agent shares the trust information given by the trust agents across the network. There is another central authority called combiner agent, which derives the final trust based on the information given by trust and recommender agents. Trust agent is deployed in every network node and the recommender as well as combiner agent is chosen to be a role which is served by the older and more stable nodes in the network.

Sanjay et al. [37] proposed FACES algorithm to establish security in MANETs. The algorithm has the 'share-your-friends' process through which the nodes share their friend list among one another. These nodes shared would be the trusted nodes and therefore, maximum voting obtained would indicate which node should be more trusted than other friends in the shared friend list. If malicious nodes are part of the network, no node or only very less nodes would opt to befriend it and therefore, malicious and selfish nodes shall be isolated from the network through this mechanism.

Weinjiai et al. [38] proposed a collaborative and trust-based outlier detection algorithm that factors in a node's reputation for MANETs. This algorithm works based on local as well as global views. The local view is updated based on the global view obtained *via* the trust of other nodes. This method is quite successful due to its very low communication overhead.

Reputation management

Patwardhan et al. [39] studied an approach in which the reputation of a node is determined by data validation. Here, a few set of randomly chosen nodes are termed as anchor nodes. These nodes are assumed to be pre-authenticated and hence the information provided by the anchor nodes is accepted as valid. The messages exchanged in the network is validated by the anchor nodes and if any such message is invalidated, then the node that had sent such messages would be the malicious node.

Ren et al. [40] proposed a node evaluation scheme in which each node evaluates the trustworthiness of its neighbours with the assistance of trustworthy neighbouring nodes. More specifically, the second-hand observations may be obtained from only a subset of the node's neighbours, and these selected neighbours are regarded as trustworthy sources with respect to the opinions toward other nodes.

Buchegger et al. [41] proposed the CONFIDANT scheme which enables monitoring and updation of route establishment which eventually avoids the non-cooperating nodes. The enhanced version of this scheme involves the Bayesian model. However, there is no hardware support for this mechanism.

Michiardi et al. [42] proposed the CORE scheme based on DSR. This mechanism monitors the cooperativeness of nodes periodically and thereby enforces node collaboration. This method is safe against attacks since it is impossible for a misbehaving node to maliciously decrease another node's reputation. However, this method does not identify whether a node is malfunctioning or really misbehaving.

Fuzzy-based trust management

Machine learning and computational intelligence have been well utilised with trust-based ATM networks [7,8,43,44]. Hallani et al. [45] uses a fuzzy-based approach to evaluate the trust of a node and then decide the trustworthiness of a node. Consideration has been given to four types of misbehaving nodes: (a) a node dropping packets randomly, (b) a node forwarding packets to wrong destination, (c) a node fabricating and transmitting false routing messages, and (d) a node launching replay attacks. Finally, the trust-based approach chooses the most trusted and reliable route from source to destination and it is achieved by choosing the route with the highest trust level out of all the discovered routes from source to destination. This route is assumed to be the most secure one.

Manickam et al. [46] proposed a fuzzy-based *ad hoc* on demand distance vector (FAODV) routing protocol. Here, fuzzy-based metrics are used for trust evaluation. Therefore, trust values could be rationally predicted and malicious behaviour shall be identified accurately when compared to other existing mechanisms.

The trust decision proposed by Rajaram et al. [47] is based on fuzzy logic. Here, a threshold trust value based on fuzzy logic is chosen initially. Any node possessing the trust greater than the threshold is assumed to be trustworthy. There are also grades assigned for nodes crossing the trust threshold. These assigned grades determine whether the node can participate in specific services (Table 1).

Methodology

Surendran et al. [54] described the fault-tolerant routing using the ant colony optimization approach. The algorithm uses ant-like agents called forward ants (FANT) and backward ants (BANT) to measure various parameters like next-hop availability (NHA), delay, and bandwidth as parameters for satisfying QoS constraints [54]. Using these parameters, path preference probability

Table 1 Comparison table for the existing methods

Title	Algorithm	Concept	Issues
Security through collaboration and trust in MANETs [38]	Gossip-based outlier detection algorithm	• Outlier detection uses local view formation, local view exchange, local view update, and global view formation	Longer time to converge to a global view if more nodes in MANET
Trust evaluation in wireless *ad hoc* networks using the fuzzy system [48]	Fuzzy trust algorithm	• Calculates the trust level of a route from source to destination	Suitable only for low mobile *ad hoc* networks
Friend-based *ad hoc* routing using challenges to establish security in MANET system [37]	FACES algorithm	• Sends challenges and shares friend lists to provide a list of trusted nodes to the source node through which data transmission finally takes place	More control overhead due to challenge request and challenge reply
Outlier detection using naïve Bayes in wireless *ad hoc* networks [49]	Naïve Bayes classifier	• Predicts the reliability of trust information provided by other adjacent nodes	High overhead
A reputation-based trust mechanism for *ad hoc* networks [50]	Reputation-based trust management algorithm	• Monitors the behavior of neighboring nodes and computing reputation based on monitoring	High computation overhead
Malicious node detection using fuzzy-based trust level in MANETs [47]	Fuzzy-based trust management	• Certificate authority employs fuzzy-based analyzer to distinguish between trusted and malicious behaviour of nodes by distributing the certificates only to the trusted nodes	More control overhead
A novel approach for misbehavior detection in *ad hoc* networks [51]	Fuzzy logic	• System learns the behaviour and applies the fuzzy logic concept for misbehaviour detection • Trust parameter is computed for each node which depends on the input parameters	More delay for control packet transmission due to more control packet overhead which effects the data packet transmission
A secure trusted auction-oriented clustering-based routing protocol for MANET [49]	Markov chain analysis of trust model, credit, and reputation scheme	• Effectively detects selfish nodes by credit and reputation scheme to enforce cooperation between nodes	High communication overhead
Malicious node detection in MANETs: a behavior analysis approach [52]	Trust management	• The approach proposes observing the behavior of mobile nodes depending on different factors • Each node in the network can recognize the malicious nodes and prevent them to participate in the communication	More control overhead
AOMDV-based TRIUMF implementation and performance evaluation [53]	Trust management	• The protocol uses an incentive mechanism for selfish node to declare its selfishness behaviour • It also uses two node-disjoint routes to reduce the malicious searching time	Unable to detect more than one malicious node in the route

is calculated. Path with higher path preference probability between source and destination is selected for transmitting data.

In this work, we describe how to enhance security in the routing phase by using a trust-based secure routing algorithm. It discovers a secure, trustworthy path from source to destination with minimal overhead based on evaluated trust value. ACO-based multiple node disjoint paths are identified for enhancing the security. Dynamic trust-based evaluation [55,56] helps to identify and exempt misbehaving nodes from using such disjoint paths. Certificate authority is also employed to distribute security certificates to trusted nodes.

Fault-tolerant and secure routing

The main goal of the ant-based look-ahead routing protocol is to find all available node-disjoint routes between a source-destination pair in advance with minimum routing overhead. To achieve this goal, the proposed protocol (Figure 1) works in three phases: (i) route discovery phase, (ii) route selection phase and (iii) route maintenance phase.

Route discovery phase

In the route discovery phase, if a source node has no existing routes, it starts a route discovery by initialization based on the pheromone values [26] of the paths. A zero (or very near to zero) pheromone value implies that the path is either not present or is very faulty and is not suitable for data transfer. If a source node has already existing routes, it selects a route based on path selection. The path selection is based on the probability of a path which indicates the goodness of a path. It depends on two factors, the pheromone content of the path and the time delay across the path. Time delay is a factor that can be used to choose between paths which have all the nodes functioning correctly.

The pheromone deposition is of two types. The ant updates the pheromone content of the path in the routing table of the source node and it also updates the individual pheromone content of each node that it traverses. Each worker ant, upon reaching its destination, retraces its path, and on the way, it updates the pheromone content of each node. Evaporation happens on all the nodes of each of the paths in the path set and on the nodes not currently in the path set. It decreases the pheromone content of the faulty paths that recently had a high confidence level. A path that works well has a good amount of pheromone on it. As the path becomes faulty, due to some malicious interrupts, the worker ants fail to deposit pheromone on that path. As a result, alternative route called secondary route is chosen and starts data transmission. Evaporation decreases the pheromone content of that failure path. But since it is still in the

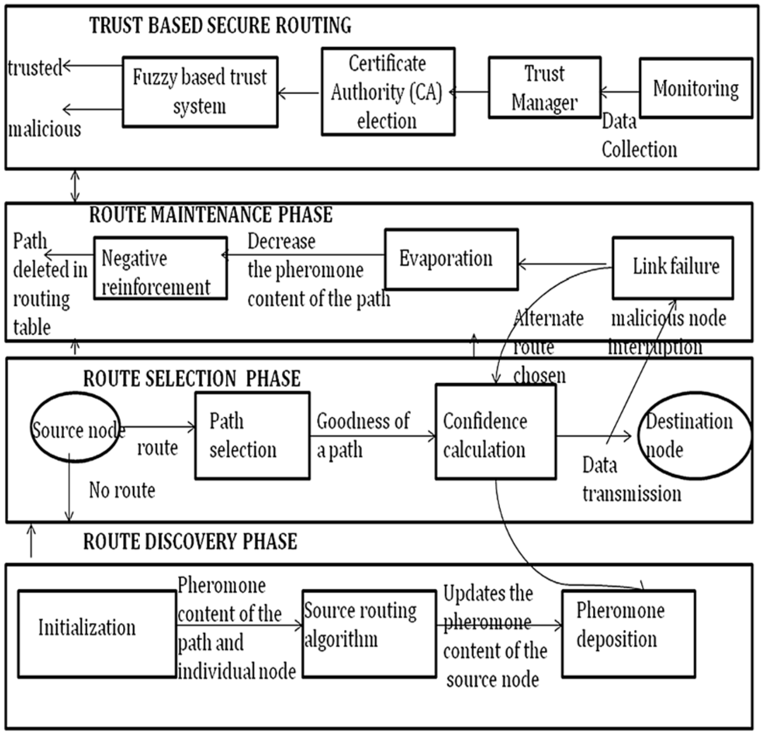


Figure 1 Block diagram for fault-tolerant and secure routing [54].

path set, routing through that path gives no gains and that particular path should be deleted through negative reinforcement.

Route selection and route maintenance phase

The main goal of the proposed ant-based look-ahead routing protocol [54] is to find all available node-disjoint routes between a source-destination pair with minimum routing overhead. Route discovery phase suggests the best and feasible paths for packet transmission. Due to the inherent learning nature of the ACO-based algorithms, the deposited pheromone values of chosen paths get strengthened during the actual transmission and eventually make it more appealing. Over a period of time, more mobile nodes might arrive in line of the chosen path which results in a more delayed, less available bandwidth and depletion of node's energy. To avoid this, the path preference probability of the chosen paths is checked periodically. If there are more new comers on the way, the path preference probability and goodness values are decreased automatically. The mobility of the nodes may also lead to link failures. If the goodness value of a node falls below the threshold value, then the node informs its precursor node by sending a message that the node is off. Then, the alternate routes are chosen for data transmission. These alternate routes are also periodically checked for their validity even though they are not currently used.

Trust-based secure routing

In MANETs, selfish or malicious nodes may want to maximize their utility by using resources from the network to send their own packets without forwarding others' packets [57]. Incentive-based trust models have been proposed in the literature [58,59]. In this work (Figure 2), we adapt a trust model based on a non-cooperative game that let the nodes quickly learn the appropriate cooperation behavior. In other words, our nodes rely only on private histories and overhead related to deriving values from public message exchanges are avoided.

The network environment is defined by the fraction of selfish/misbehaving nodes within the total population. The successful cooperation is measured as the fraction of packets originated by normal nodes that effectively reach their destinations. We assume the misbehaving action of nodes as selfish or malicious behavior. However, actual selfishness is harmless and maliciousness might be due to the impact of external entities. In this paper, we use the terms selfish and malicious interchangeably but is not the ideal case.

A very close non-cooperative trust model is adapted for best neighbor strategy-based routing [60,61] and maximum payoff strategy best neighbor routing [62]. We take as reference the non-cooperative game model of Mejia et al. [63]. In our trust model, we assume the interactions among nodes are based on the iterated prisoner's dilemma with random game pairs [64]. Each intermediate node decides whether it should retransmit or discard a packet that comes from a certain source node. This game-based forwarding strategy depends on two aspects: the past behavior of the network when the intermediate node acted as a source and the trust level that the intermediate node has in the source node.

The forwarding rate for any pair of nodes (A, B) is defined by

$$f_r(B, A : n) = \frac{n_A}{n} \tag{1}$$

where B is the sender, A is the forwarding node, n is the number of packets to be forwarded, and n_A is the number of packets actually forwarded.

Each node maintains a trust table based on the observed behavior of its neighbors. We do not record long observations for calculations of trust and trust table updates. Only the most recent m observed decisions are utilized to compute moving average of forwarding packet sequences. The value of m has to be large enough to obtain a fair evaluation of the forwarding rate but equally small enough to ensure that the forwarding rate actually corresponds to the current strategies [63]. The trust obtained above will be assumed as direct trust obtained as the first-hand information. Second-hand information is obtained from the recommendations from friend nodes and is derived by EigenTrust.

Initially, every node should try to start building a good reputation among neighbors [65]. Otherwise, after very few successful transmissions, it would appear as if the intermediate nodes are always cooperative regardless of the derived trust values. For unknown nodes, the reputation is assumed to be one initially. We assume that the location of selfish nodes are completely known; however, identifying the selfishness is a research in itself. Therefore, packets will be forwarded if there is at least one path with non-selfish forwarders.

A packet will reach the destination through an h-hop path if at least the first h nodes of the path are normal nodes. Since the nodes of the path are randomly selected, the probability of finding a consecutive sequence of h normal nodes is given by

$$P_r[B(h)] = \prod_{i=0}^{h-1} \frac{N_N - i}{N - i} \tag{2}$$

where N is the total number of nodes in the network, N_N is the total number of normal nodes among the N nodes of the population, P_h is the probability that a path has h hops, $P_{r/h}$ is the probability of finding r routes given that the path length is h hops, $B(h)$ is the event that there are no selfish nodes among h randomly

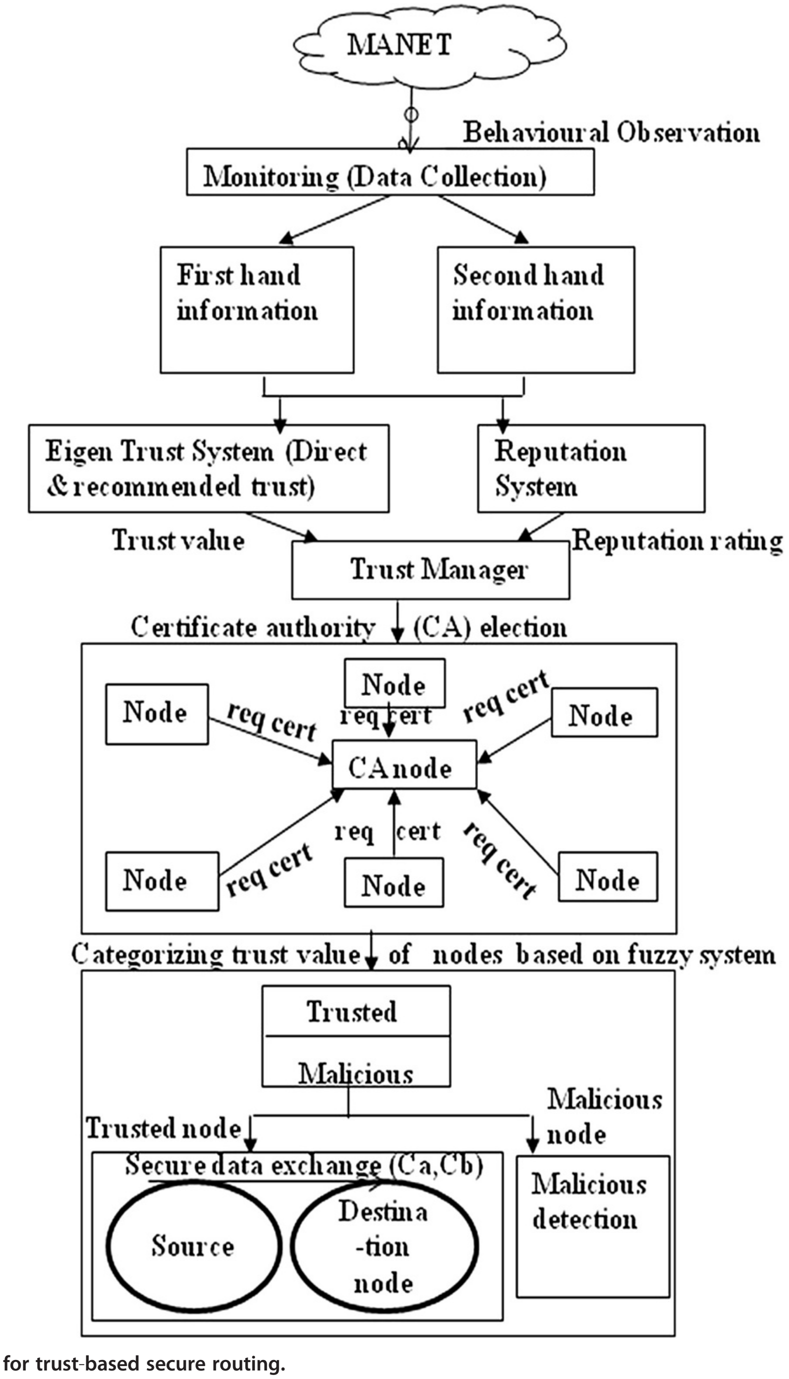


Figure 2 Block diagram for trust-based secure routing.

selected nodes, and A is the event that a packet finds at least one route made out exclusively of normal nodes.

The probability of having at least one selfish node in an h-hop path is $1 - P_r[B(h)]$. The probability that each one of r routes of h hops has at least one selfish node is $(1 - P_r[B(h)])^r$. The probability that at least one of the r routes is composed exclusively of normal nodes is $1 - ((1 - P_r[B(h)])^r)$. Since the probability of finding r routes of h hops is P_h. $P_{r/h}$, the probability that a packet finds at least one path composed exclusively of non-selfish nodes is given by

$$\begin{aligned} C_{\max} &= P_r[A] \\ &= \sum_h \sum_r P_h P_{r/h} (1 - ((1 - P_r[B(h)])^r)) \end{aligned} \tag{3}$$

From Figure 2, based on the observing behaviour, first-hand and second-hand trust information is collected.

First-hand trust information is obtained by direct observation whereas second-hand trust information is obtained from the friend nodes. We use EigenTrust and reputation [1,66] mechanism for trust and reputation updation. The node with maximum trust is chosen as certificate authority. This certificate authority is entitled for assigning certificates for data exchange to other nodes based on request. Upon such request, the Certificate Authority issues certificates only to trusted nodes and thus enables secure data transmission.

Local trust evaluation using EigenTrust

In Figure 3 [66], the local trust evaluated using EigenTrust value is normalized. In EigenTrust, each node will record the number of satisfactory transaction and number of unsatisfactory transaction. If node *i* and node *k* have no direct transaction, then Eigen trust will apply the concept of recommendation trust which obtained the trust value of *k* by asking his friends, and then weighted adding into the recommendation trust value of *k* in the eyes of node *i*. Local trust value is updated by merging the node's own local view and received view using the Dempster-Shafer theory. Finally, the updated trust value is used for global trust evaluation.

Certificate authority election

In Figure 4, the node with the maximum trust value [1] is elected as Certificate Authority by incorporating the Certificate Authority election algorithm.

Fuzzy-based trust system

In Figure 5, the fuzzy trust system [51] calculates fuzzy trust based on the number of message updates done by every node. During this process, each node maintains a table containing number of updates due to its neighbours. During trust computation, the number of RREQ’s, number of updates and number of RREP’s of the node are the input to the Fuzzy inference system which outputs the trust value depending on the values of the number of RREQ’s, number of updates and number of RREP’s of the respective node. Trust value of each node is updated if any change in the one of the following three inputs of that node namely RREQ’s, updates and RREP’s received. These computed trust values are then associated with the routing process during trust application. The defuzzification is the process of conversion of fuzzy output set into a single number. The method used for the defuzzification is ‘centroid method’. The input membership functions are number of RREQ’s, no. of updates and no. of RREP’s of the node. The output membership function is Trust.

Secure data transmission

Initially, the shared key of CA is known to all valid members within the network community. The source node sends a request message to CA node encrypting it with the shared key SKac (Figure 6). On receiving this request the CA node decrypts the message and first checks whether the source and destination nodes are valid. The CA node generates CERT A and CERT B, encrypts it with shared keys SKac and SKcb and forwards it to the source and destination nodes. The destination node decrypts and verifies CERT A, CERT B and generates Nonce N1 only if certificates are valid and sends to the source node. The source node decrypts and verifies CERT A, CERT B, N1 and generates Nonce N2 only if certificates are valid. If everything goes smooth, then data packet exchanges are initiated for secure transmission with CERT A and CERT B.

Simulation analysis and results

The simulations were performed in NS2. The following are the performance parameters analyzed (Table 2):

- Packet delivery ratio
  The ratio of the data packets delivered to the destinations to those generated by the constant bit rate (CBR) source.

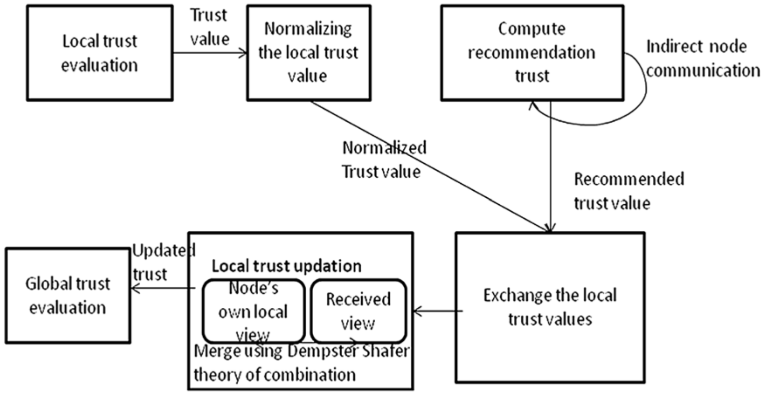
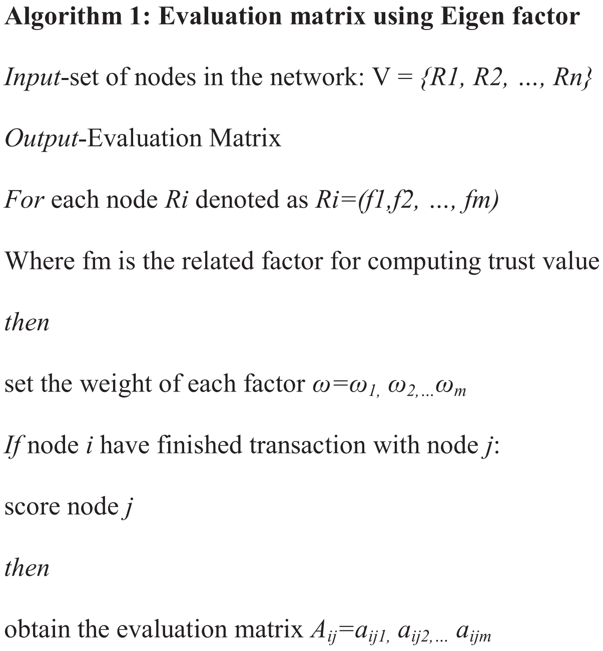
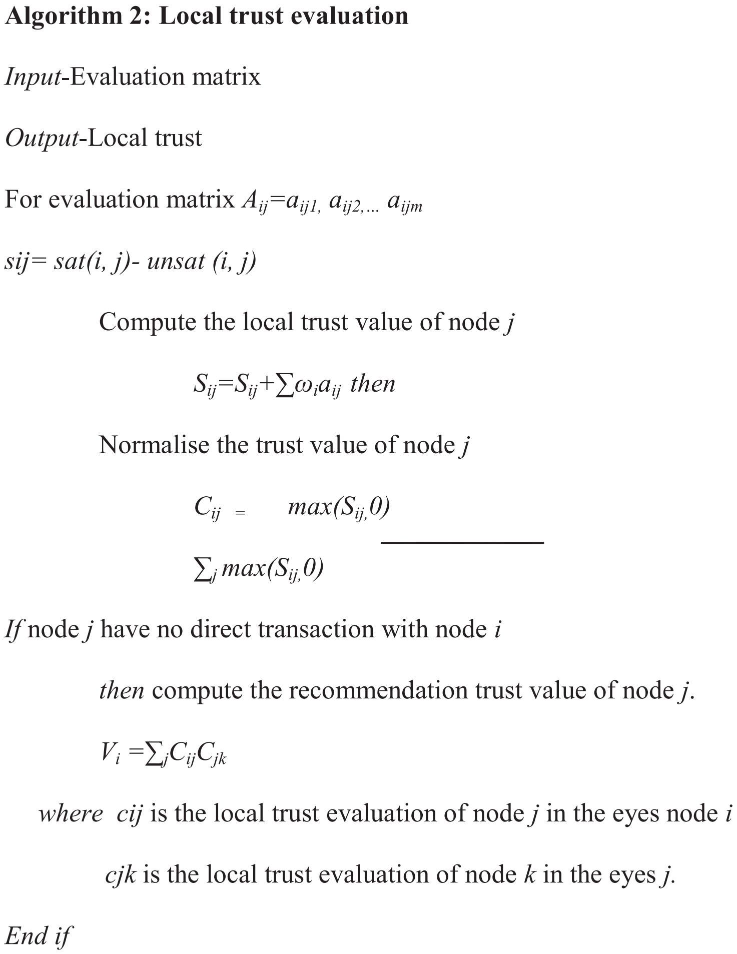

Local trust evaluation
Trust value
Normalizing the local trust value
Compute recommendation trust
Indirect node communication
Normalized Trust value
Recommended trust value
Global trust evaluation
Updated trust
Local trust updation
Node's own local view
Received view
Merge using Dempster Shafer theory of combination
Exchange the local trust values

Figure 3 Local trust evaluation using EigenTrust.



Algorithm 1: Evaluation matrix using Eigen factor

```
Input-set of nodes in the network: V = {R1, R2, …, Rn}
Output-Evaluation Matrix
For each node Ri denoted as Ri=(f1,f2, …, fm)
Where fm is the related factor for computing trust value
then
set the weight of each factor $\omega=\omega_{1,}\ \omega_{2,\ldots}\omega_m$
If node i have finished transaction with node j:
score node j
then
obtain the evaluation matrix $A_{ij}=a_{ij1,}\ a_{ij2,\ldots}\ a_{ijm}$
```



Algorithm 2: Local trust evaluation

```
Input-Evaluation matrix
Output-Local trust
For evaluation matrix $A_{ij}=a_{ij1,}\ a_{ij2,\ldots}\ a_{ijm}$
sij= sat(i, j)- unsat (i, j)
    Compute the local trust value of node j
        $S_{ij}=S_{ij}+\sum\omega_i a_{ij}$ then
    Normalise the trust value of node j
        $C_{ij} = \frac{max(S_{ij,}0)}{\sum_j max(S_{ij,}0)}$
If node j have no direct transaction with node i
    then compute the recommendation trust value of node j.
    $V_i = \sum_j C_{ij} C_{jk}$
  where cij is the local trust evaluation of node j in the eyes node i
        cjk is the local trust evaluation of node k in the eyes j.
End if
```

Algorithm 3: Local trust updation

```
$N_i$–i th node in network
$V_i$–local trust value of $N_i$
$V_i^{/}$-updated trust value of $N_i$
Input of $N_i$ : $V_i$
Output of $N_i$ :$V_i^{/}$
Upon reception of $Vk$ from node $nk$:
if $Vi$ != $Vk$ then
        merge $Vi$ and $Vk$ according to the following rules:
if node m is in both $Vi$ and $Vk$ then
        calculate the updated value $Ui$ of the corresponding columns for node $m$ in both $Vi$ and $Vk$ using the Dempster's rule of combination,
        Store $Ui$ to an intermediate list $TEMPi$ as an entry.
if node $m$ is in either $Vi$ OR $Vj$ , but not both, then add a virtual entry of node $m$ to the view that previously does not contain m, and set all the columns of this virtual entry as 0.
        Calculate the updated value $Ui$ of the corresponding columns for node m in both $Vi$ and $Vk$ using the Dempster's rule of combination
        Store $Ui$ to an intermediate list $TEMPi$ as an entry.
calculate the top $k$ outliers from $TEMPi$, and assign these $k$ top outliers to $V_i^{/}$ .
    broadcast $V_i^{/}$ to all of its immediate neighbors (i.e., number of hop = 1).
    else keep $Vi$ unchanged, and do not send any message out.
end if
```

Algorithm 4: Global trust evaluation

```
Input of ni: local trust value Vi
Output of ni: Global trust GV
For each node ni
Broadcast Vi to all of its immediate neighbours
Upon reception of Vk from its immediate neighbour nk:
Invoke Local Trust Update Algorithm
When no more message exchange occurs
V Vi = GV–
```

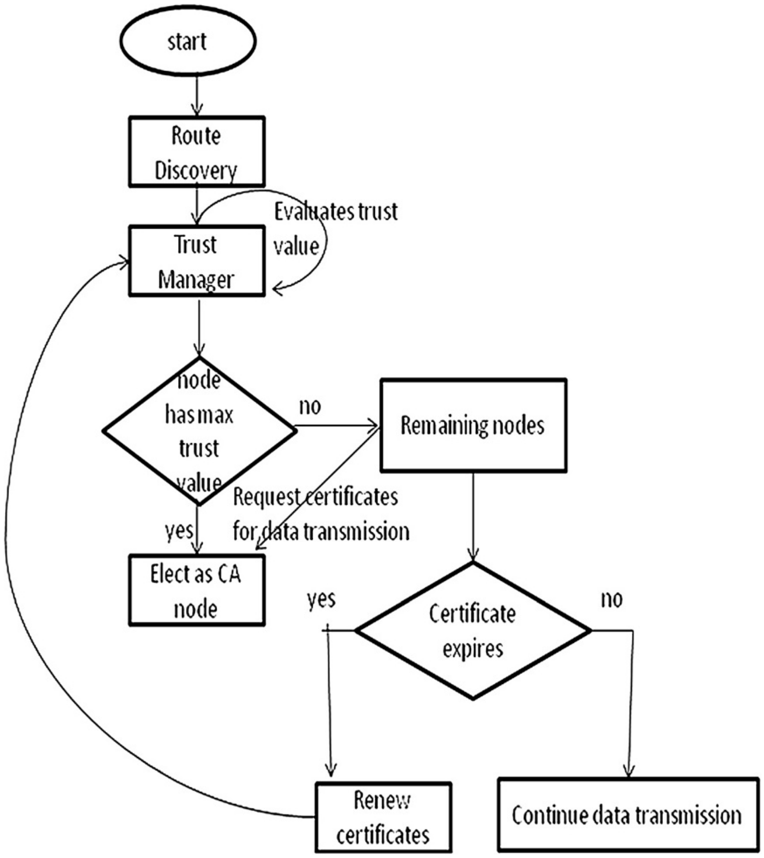


Figure 4 Certificate authority election.

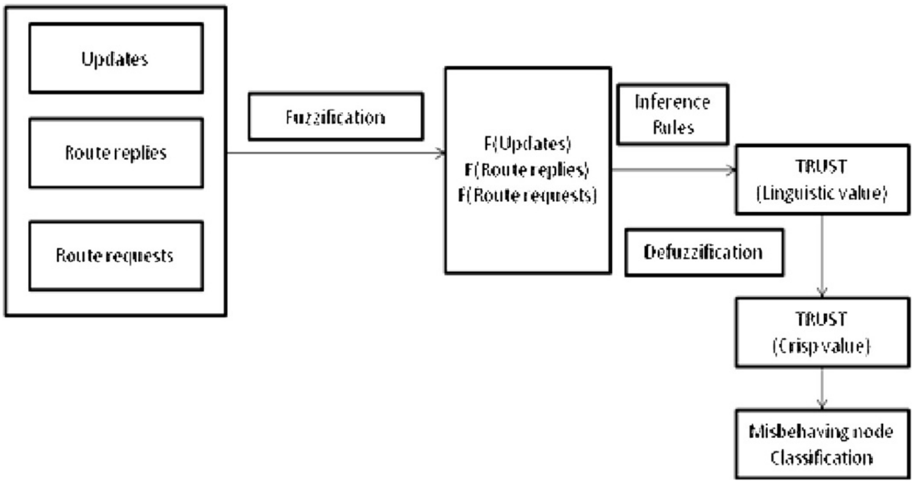


Figure 5 Fuzzy-based trust system.

- Throughput
  Throughput of the routing protocol means that in certain time, the total size of useful packets received at all the destination nodes.
- Routing overhead
  It is the ratio of routing packets to the total number of packets generated by the source.
- Malicious detection effectiveness
  This measures the performance of the algorithm. This is measured as total number of detected nodes divided by the total number of malicious nodes in the network.

Packet delivery ratio for varying misbehaving nodes with and without reputation

Scalability has been analyzed by comparing the packet delivery ratio, while increasing the number of misbehaving nodes as shown in Figures 7 and 8. The result was compared with ant colony-based fault-tolerant DSR approach among which our trust-based secure DSR gives a consistent packet delivery ratio when the number of misbehaving nodes increases. As the number of misbehaving nodes increases in the network, the number of active good nodes decreases. Thus, the availability of

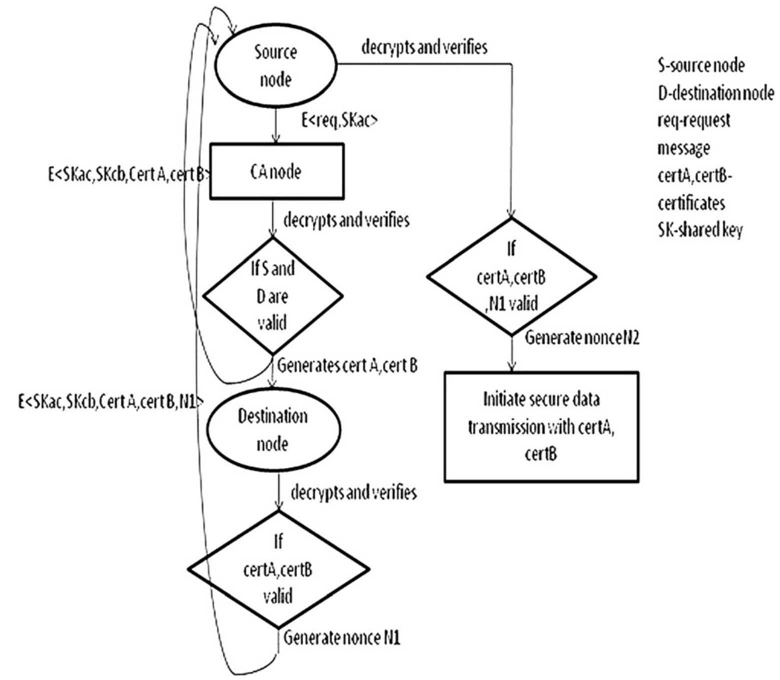


Figure 6 Secure data transmission.

Algorithm 5: Certificate exchange for secure data transmission

```
Input: CA node
Output: Exchange Of certificates
PUa,PUb = Public Key of node A and B.
PRa,PRb = Private Key of node A and B.
SKac = Shared Key of Source and CA.
SKbc = Shared Key of Destination and CA.
SID, DID = Source and Destination ID.
Generate Shared Key SKac
Source node request CA
E[CREQ(SID, DID, FTValue)SKac]
CA node decrypts CREQ looks for SID in ID repository.
if (SID==ID) then
CA node verifies for SID and checks for DID in its range.
Generate PUa,PRa, PUb,PRb, SKbc,
CERT A=SID,PRa,PUa,FTvalue,TS.
CERT B=DID,PRb,PUb,FTvalue,TS.
CA sends CREP as E[(CERT A)SKac] to source node A.
CA sends E[(CERT B)SKbc] to destination node B.
Else Display ("Transmission cannot be granted").
```

Table 2 Simulation parameters

Parameter	Values
Radio range of single node	250 m
MAC layer protocol	IEEE 802.11
Traffic pattern	CBR
Data packet size	512 bytes
Simulation area	3,000 m × 3,000 m
Number of nodes	100
Node mobility speed	0 to 30 m/s
Simulation time	100, 200, 300, 400 and 500 ms
Mobility model	Random way point model
Misbehaving nodes	10, 20, 30, 40 and 50

nodes in route establishment decreases. In worst cases, no path may be established between source and destination if the intermediate nodes are misbehaving and denied to forward other packets. It may also happen that misbehaving nodes are there on a route and drops the data packet. As a result, acknowledgments from the destination are missing; the source node of a TCP session may slow down or even stop sending packets. Packet delivery ratio is around 65% for without reputation in the presence of 50% misbehaving nodes. Without reputation mechanism, misbehaving nodes previously routed may

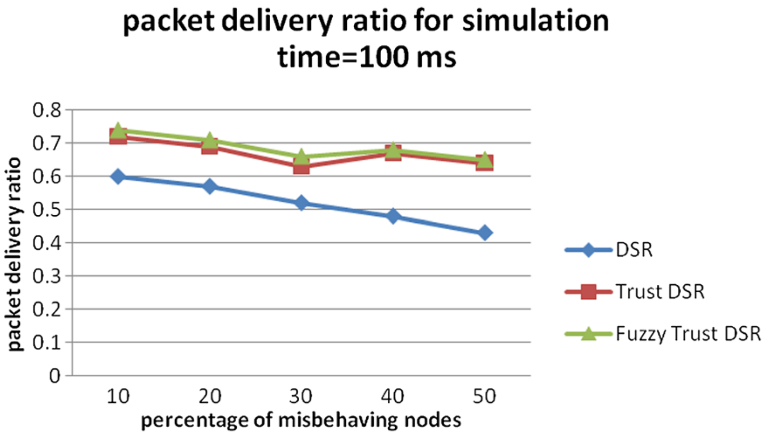


Figure 7 Packet delivery ratio for varying misbehaving nodes without reputation.

be treated as a genuine node for the next data transfer. So packet delivery ratio considerably reduced in case of 30% misbehaving nodes.

In case of fault-tolerant DSR, as the number of misbehaving nodes increases, packet delivery ratio decreases significantly around 40% when 50% of the nodes are misbehaving. This is because no misbehaving mitigation policy is incorporated with original DSR. In the proposed fuzzy trust DSR, cooperation is enforced to mitigate misbehaving. Thus, it can be seen that though the number of misbehaving nodes increases, the packet delivery ratio does not considerably lessen for both DSR and trust DSR. Packet delivery ratio is near about 80% for fuzzy trust DSR and 76% for trust DSR in the presence of 50% misbehaving nodes because nodes are forcefully compelled to take part into network functionality; otherwise, they would be detected as malicious and eventually isolated from the network.

Throughput for varying misbehaving nodes with and without reputation

From Figures 9 and 10, throughput is substantially reduced with increment of misbehaving nodes in the network for fault-tolerant DSR. Both DSR and fuzzy trust DSR may give good throughput when there is no misbehaving node. However, as the number of misbehaving nodes increases, the performance of fault-tolerant DSR degrades significantly. On the other hand, the proposed trust and fuzzy trust DSR gives steady throughput in the presence of misbehaving nodes. In the presence of 50% misbehaving nodes, the proposed fuzzy trust DSR gives throughput around 6.5 kbps whereas in fault-tolerant DSR, it becomes 5.2 kbps. Thus, it can be easily perceived that the proposed scheme effectively mitigates selfishness and enforce cooperation between nodes and increase node availability. On the other hand, throughput considerably reduced in case of 20% misbehaving nodes, because misbehaving nodes previously routed

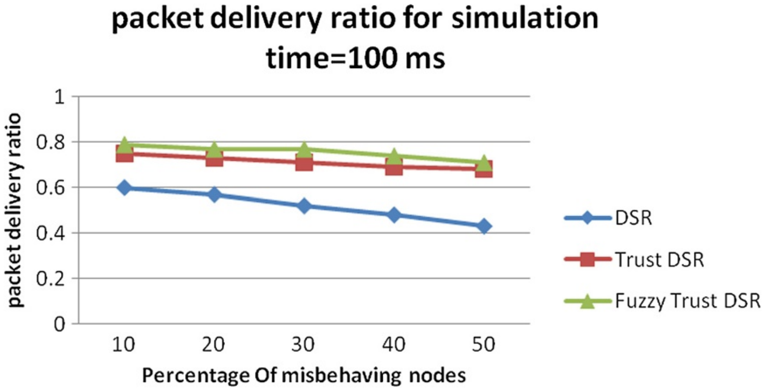


Figure 8 Packet delivery ratio for varying misbehaving nodes with reputation.

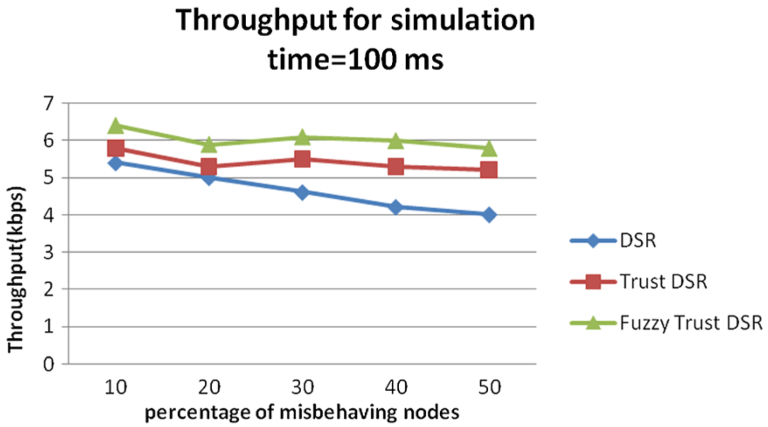


Figure 9 Throughput for varying misbehaving nodes without reputation.

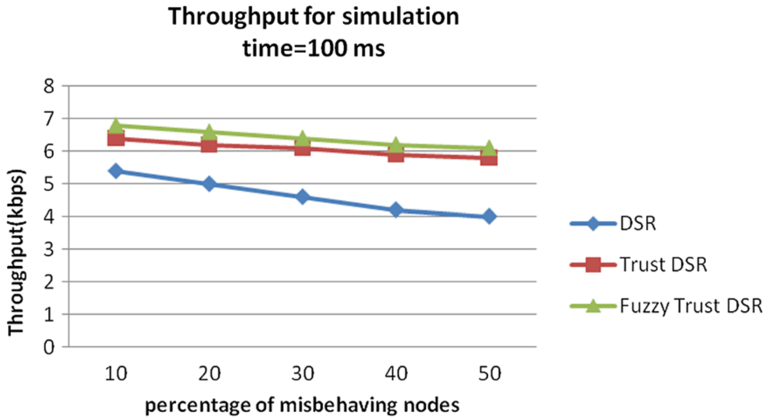


Figure 10 Throughput for varying misbehaving nodes with reputation.

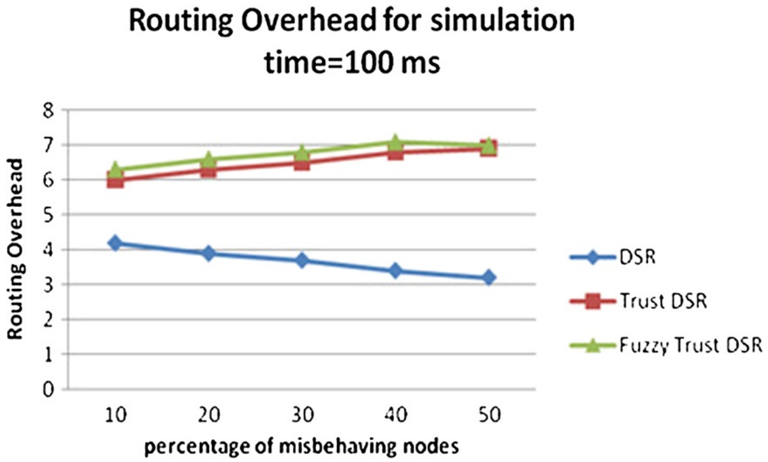


Figure 11 Routing overhead for varying misbehaving nodes without reputation.

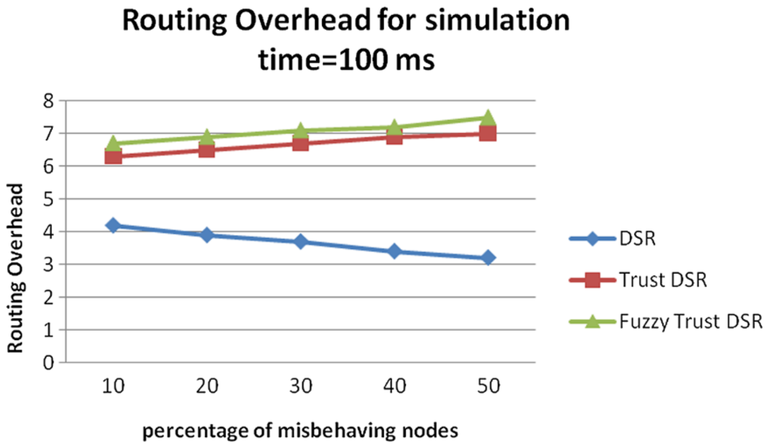


Figure 12 Routing overhead for varying misbehaving nodes with reputation.

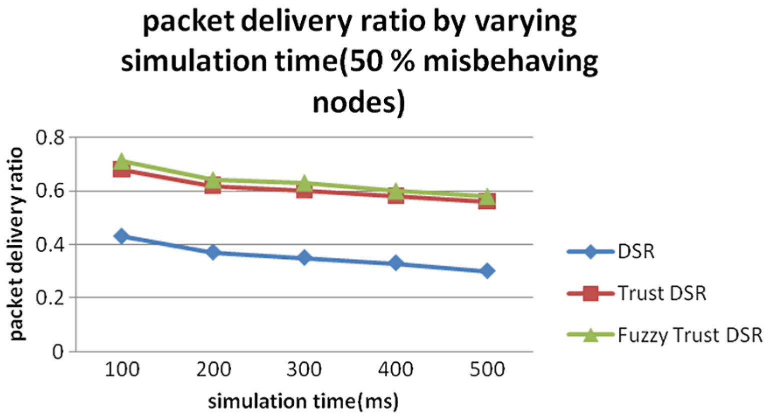


Figure 13 Packet delivery ratio for varying simulation time in the presence of 50% misbehaving nodes.

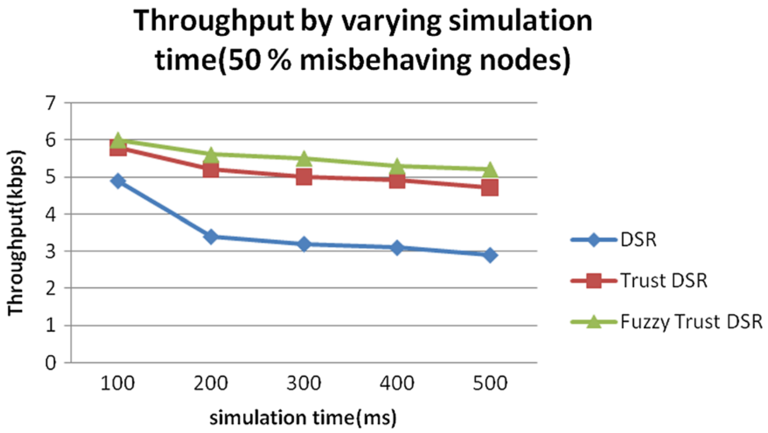


Figure 14 Throughput for varying simulation time in the presence of 50% misbehaving nodes.

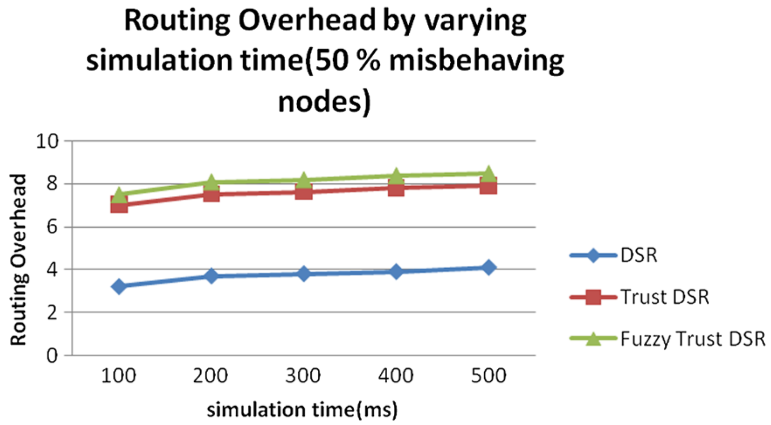


Figure 15 Routing overhead for varying simulation time in the presence of 50% misbehaving nodes.

may be treated as a good node for the next data transfer without reputation mechanism.

Routing overhead for varying misbehaving nodes with and without reputation

From Figures 11 and 12, we have analyzed that routing overhead decreases for fault-tolerant DSR as the number of selfish nodes increases because selfish nodes drop RREP. It varies a little from the proposed scheme since enforcement of cooperation and nodes are compelled to forward other packets. Routing overhead increases for proposed scheme due to the problem of missing routes and the overhead of searching for alternate routes.

Packet delivery ratio, throughput and routing overhead for varying simulation time with 50% misbehaving nodes

From Figures 13 and 14, packet delivery ratio and throughput decreases while increasing simulation time for fault-tolerant DSR. But both trust DSR and fuzzy trust DSR do not decrease considerably while increasing simulation time. If misbehaving nodes are there on a route and drops the data packet, acknowledgments from the destination will be missed and thereby the source node of a TCP session may slow down or even stop sending packets. Packet delivery ratio decreases up to 30% for fault-tolerant DSR, whereas fuzzy trust DSR maintains 60% packet delivery ratio for simulation time = 500 ms. On the other hand, from Figure 15, routing overhead increases for DSR and fuzzy trust DSR with increasing simulation time.

Accuracy for malicious node detection

Figure 16 shows the malicious detection effectiveness of the proposed algorithm. As the number of malicious nodes increases, then the detecting effectiveness reduces up to 64% for 50% misbehaving nodes. If there is less number of malicious nodes in the network, then the detection effectiveness may reach 90% for our proposed secure algorithm.

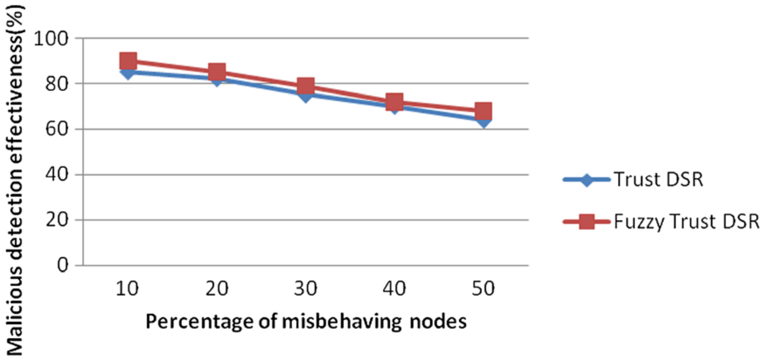


Figure 16 Accuracy for malicious node detection.

Conclusion

In this work, a quality of service-based EigenTrust non-cooperative game model for fault-tolerant and secure routing in mobile *ad hoc* networks (MANETs) is proposed based on the ant colony optimization (ACO) approach. The fault-tolerant approach uses bandwidth, delay and hop count to calculate multiple disjoint paths between source and destination to satisfy given QoS constraints. A trust-based secure routing is designed to distinguish the truly malicious nodes from the trusted nodes.

Fuzzy-based Certificate Authority is responsible of secure data exchange by allowing the trusted entities to participate in the network, isolating the malicious nodes. Integrated approach of trust and fuzzy logic-based Certificate Authority will secure the communication. Simulations prove that our proposed trust-based secure routing algorithm performs better with 15% to 20% improvement compared to the fault-tolerant algorithm.

The proposed algorithm prevents the system from packet dropping attacks; in future, it can be extended to all possible denial of service attacks. It can also be directed to prevent node congestion and to handle multimedia data exchanges. This work labels good and bad nodes separately and names bad nodes with selfish as well as malicious behavior as misbehaving nodes. In future, we have plans to extend this approach with a known mixture of selfish and malicious nodes and to further develop the trusted secured routing excluding the identified malicious nodes in the network setup.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Information Technology, Tagore Engineering College, Chennai 600127, India. [2]Department of Electronics and Communication Engineering, Jerusalem College of Engineering, Chennai 600100, India.

Received: 12 March 2014 Accepted: 28 April 2014
Published: 15 May 2014

References

1. V Manoj, A Mohammed, N Raghavendiran, R Vijayan, A novel security framework using trust and fuzzy logic in MANET. Intern J Distributed Parallel Systems **3**, 1 (2012)
2. G Di Caro, M Dorigo, AntNet, distributed stigmergetic control for communications networks. J Artificial Intelligence Res (JAIR) **9**, 317–365 (1998)
3. L Yuhua et al., Multi-layer clustering routing algorithm for wireless vehicular sensor networks. IET Communications **4**(7), 810–816 (2010)
4. H Cheng et al., Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks. Ad Hoc Networks **10**(5), 760–773 (2012)
5. C-C Shen, J Chaiporn, S Chavalit, H Zhuochuan, R Sundaram, *Ad hoc networking with swarm intelligence in Ant Colony Optimization and Swarm Intelligence* (Springer, Berlin Heidelberg, 2004), pp. 262–269
6. X Yuan, Heuristic algorithms for multi-constrained quality-of-service routing. IEEE/ACM Trans Networking **10**(2), 244–256 (2002)
7. F Antonios et al., The use of learning algorithms in ATM networks call admission control problem: a methodology. Computer Networks **34**(3), 341–353 (2000)
8. Y Ahmet Sekercioglu et al., Computational intelligence in management of ATM networks: a survey of current state of research. Soft Comput **5**(4), 257–263 (2001)
9. JJ Wang, M Song, Y Zhang, W J-y, Y Man, An ACO and position information based intelligent QoS routing mechanism in MANET. J China Universities Posts Telecommunications **17**, 6–16 (2010)
10. MS Kwang, HS Weng, Ant Colony Optimization for routing and load balancing: survey and new directions. IEEE Trans Syst Man Cybern **33**(5), 560–572 (2003)
11. S Misra, SK Dhurandher, MS Obaidat, P Gupta, K Verma, P Narula, An ant swarm-inspired energy-aware routing protocol for wireless ad-hoc networks. J Systems Software **83**(11), 2188–2199 (2010)
12. S Misra, P Venkata Krishna, A Bhiwal, A Singh Chawla, BE Wolfinger, C Lee, A learning automata-based fault-tolerant routing algorithm for mobile ad-hoc networks. J. Supercomput **62**(1), 4–23 (2011)
13. S Misra, SK Dhurandher, MS Obaidat, K Verma, P Gupta, Using ant-like agents for fault-tolerant routing in mobile ad-hoc networks, in *IEEE International Conference on Communications* (ICC'09, Dresden, 2009). 14–18 June 2009, pp. 1–5
14. A Colorni, D Marco, M Vittorio, Distributed optimization by ant colonies, in *Proceedings of the European Conference on Artificial Life*, vol. volume 142 (Elsevier, Amsterdam, 1991), pp. 134–142
15. M Dorigo, V Maniezzo, A Colorni, Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B Cybern **26**(1), 29–41 (1996)
16. M Gunes, S Udo, B Imed, ARA-the ant-colony based routing algorithm for MANETs, in *Proceedings of the International Conference on Parallel Processing Workshops* (IEEE, Piscataway, 2002), pp. 79–85
17. H Matsuo, K Mori, Accelerated ants routing in dynamic networks, in *Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking Parallel/Distributed Computing*, 2001, pp. 333–339
18. K Fujita, S Akira, M Toshihiro, M Hiroshi, An adaptive ant-based routing algorithm used routing history in dynamic networks, in *The 4th Asia-Pacific Conference Simulated Evolution Learning, Orchid Country Club, Singapore, 18-22 November 2002*. volume 1, pp 46–50
19. F Ducatelle, G Di Caro, LM Gambardella, Using ant agents to combine reactive and proactive strategies for routing in mobile ad-hoc networks. Int J Computational Intel App **5**(02), 169–184 (2005)
20. L Rosati, M Berioli, G Reali, On ant routing algorithms in ad hoc networks with critical connectivity. Ad Hoc Networks **6**(6), 827–859 (2008)
21. J Wang et al., HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network. Ad Hoc Networks **7**(4), 690–705 (2009)
22. J Mohammad, Z Azadeh Alsadat Emrani, H Seyed Mohamad, MSDP with ACO, A maximal SRLG disjoint routing algorithm based on ant colony optimization. J Network Comput App **35**(1), 394–402 (2012)
23. E Khosrowshahi-Asl, M Noorhosseini, AS Pirouz, A dynamic ant colony based routing algorithm for mobile ad-hoc networks. J Information Science Engineering **27**(5), 1581–1596 (2011)
24. F Neumann, C Witt, Ant Colony Optimization and the minimum spanning tree problem. Theoretical Comput Science **411**, 2406–2413 (2010)
25. V Ana Cristina Kochem, M Anelise, D Myriam Regattieri, V Aline Carneiro, Grant: inferring best forwarders from complex networks' dynamics through a greedy ant colony optimization. Comp Networks: The International Journal of Computer and Telecommunications Networking **56**(3), 997–1015 (2012)
26. S Misra, SK Dhurandher, MS Obaidat, K Verma, P Gupta, A low overhead fault-tolerant routing algorithm for mobile ad hoc networks: a scheme and its simulation analysis. Simulation Modelling Practice Theory **18**, 637–649 (2010)
27. M Chandra, R Baskaran, Review: a survey: Ant Colony Optimization based recent research and implementation on several engineering domains. Expert Syst with App **39**(4), 4618–4627 (2012)
28. CAO Huaihu, A QoS routing algorithm based on ant colony optimization and mobile agent. Procedia Engineering **29**, 1208–1212 (2012)
29. A Rossi, P Samuel, Collusion resistant reputation based intrusion detection system for MANETs. Intern J Comput Science Network Security (IJCNS) **9**, 11 (2009)
30. GS Mamatha, SC Sharma, A highly secured approach against attacks in MANETS. Intern J Comput Theory Engineering **2**, 5 (2010)
31. M Rajesh Babu, S Selvan, A lightweight and attack resistant authenticated routing protocol for mobile ad hoc networks. Intern J Wireless Mobile Networks (IJWMN) **2**, 2 (2010)

32. R Yonglin, B Azzedine, *An efficient trust based reputation protocol for wireless and mobile ad hoc networks* (IEEE "GLOBECOM" Proceedings, New Orleans, LA). 30 Nov - 4 Dec 2008
33. D He et al., ReTrust: attack-resistant and lightweight trust management for medical sensor networks. IEEE Trans Inf Technol Biomed **16**(4), 623–632 (2012)
34. W Li, J Parker, A Joshi, Security through collaboration in MANETs, in *Proceedings of the 4th International Conference on Collaborative Computing: Networking. Appl Work Sharing, Orlando, 13-16 Nov 2008, volume 10* (Springer LNICST, Berlin Heidelberg, 2008), pp. 696–714
35. A Attar et al., A survey of security challenges in cognitive radio networks: solutions and future research directions. Proceedings IEEE **100**(12), 3172–3186 (2012)
36. VR Ghorpade, Fuzzy logic based trust management framework for MANET. DSP J **8**, 8 (2008)
37. SK Dhurandher, MS Obaidat, K Verma, P Gupta, P Dhurandher, FACES: friend-based ad hoc routing using challenges to establish security in manets systems. IEEE Systems Journal **5**(2), 321–322 (2011)
38. W Li, A Joshi, T Finin, CAST: context aware security and trust framework for mobile ad hoc networks using policies. Distributed Parallel Databases **31**(2), 353–376 (2012)
39. A Patwardhan, A Joshi, T Finin, Y Yesha, A data intensive reputation management scheme for vehicular ad hoc networks, in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems - Workshops, (MobiQuitous), San Jose, CA, 17-21 July 2006, pp. 1–8*
40. Y Ren, A Boukerche, Performance analysis of trust-based node evaluation schemes in wireless and mobile ad hoc networks, in *the IEEE International Conference on Communications* (ICC'09, Dresden). 14-18 June 2009, pp.1–5
41. S Buchegger, JY Le Boudec, Performance analysis of the confidant protocol, in *MobiHoc'02: Proceedings of the 3rd ACM International Symposium on Mobile ad hoc Networking and Computing, Lausanne, Switzerland, 9-11 June 2002*, pp. 226–236
42. P Michiardi, R Molva, CORE, a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks, in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications Multimedia Security, Portorož, Slovenia, 26–27 September 2002*, 2002, pp. 107–121
43. AV Vasilakos, MP Saltouros, AF Atlassis, W Pedryz, Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques. IEEE Trans Syst Man Cyber Part C **33**(3), 297–312 (2003)
44. AV Vasilakos et al., Evolutionary-fuzzy prediction for strategic QoS routing in broadband networks, in *IEEE World Congress on Computational Intelligence, The 1998 International Conference on Fuzzy Systems Proceedings, Anchorage, AK, 4-9 May 1998 volume 2*, pp. 1488–1493
45. H Hallani, SA Shahrestani, Fuzzy trust approach for wireless ad-hoc networks. Communications of the IBIMA **1**, 212–218 (2008)
46. J Martin, L Manickam, S Shanmugavel, Fuzzy based trusted ad hoc on-demand distance vector routing protocol for MANET, in *the International Conference on Advanced Computing and Communications, 2007* (ADCOM 2007, Guwahati, Assam). 18-21 Dec 2007, pp. 414–421
47. A Rajaram, S Palaniswami, Detecting malicious node in MANET using trust based cross-layer security protocol. Intern J Comput Science Information Technologies **2**, 130–137 (2010)
48. S Pallavikhatr, *Trust evaluation in wireless ad hoc networks using fuzzy system.* Proceedings of the CUBE International Information Technology Conference, Pune, India, 03 - 06 September, 2012
49. C Pushphita, S Indranil, SK Ghosh, A secure trusted auction oriented clustering based routing protocol for MANET. Springer J On Cluster Comput **15**, 303–320 (2011)
50. R Yacine, E Vicente, V Mujica, S Dorgham, A reputation-based trust mechanism for ad hoc networks, in *Proceedings of the 10th IEEE Symposium on Computers and Communications, 2005, Cartagena, Spain, 27-30 June 2005*
51. V Sumalatha, PC Reddy, A novel approach for misbehaviour detection in ad hoc networks. Intern J Cryptography Security **2**, 1 (2009)
52. Y Khamayseh, R Al-Salah, MB Yassein, Malicious nodes detection in MANETs: behavioral analysis approach. J Networks **7**(1), 116–125 (2012)
53. M Ahmed, E-H Bd, A Ihab, I Ibrahim, AOMDV based TRIUMF implementation and performance evaluation. Intern J Comput Information Syst **3**(2), 60–69 (2011)
54. S Surendran, S Prakash, An ACO look ahead approach to QoS enabled fault-tolerant and secured routing in MANETs. PennSee Journal **75**(9), 407–417 (2013)
55. W Guo, Z Xiong, Dynamic trust evaluation based routing model for ad hoc networks, in *Proceedings of the 2005 International Conference on Wireless Communications, Networking Mobile Computing, Wuhan University, China, 23-26 Sept 2005 volume 2*, pp. 727–730
56. C Weinjiai, P James, J Anupam, Security through collaboration and trust in MANETs. Springer J Mobile Network Appl **17**, 342–352 (2012)
57. K Komathy, P Narayanasamy, Best neighbor strategy to enforce cooperation among selfish nodes in wireless ad hoc network. Computer Communications **30**(18), 3721–3735 (2007)
58. K Komathy, P Narayanasamy, Trust-based evolutionary game model assisting AODV routing against selfishness. J Network Comput Appl **31**(4), 446–471 (2008)
59. Y Wang et al., P2P soft security: on evolutionary dynamics of P2P incentive mechanism. Comput Communications **34**(3), 241–249 (2011)
60. R Axelrod, D Dion, The further evolution of cooperation. Science **242**(4884), 1385–1390 (1988)
61. M Mejia, N Peña, JL Muñoz, O Esparza, M Alzate, A game theoretic trust model for on-line distributed evolution of cooperation in MANETs. J Network Comput Appl **34**(1), 39–51 (2011)
62. H Ishibuchi, N Namikawa, Evolution of cooperative behavior in the iterated prisoner's dilemma under random pairing in game playing. IEEE Congress Evolutionary Computation **3**, 2637–2644 (2005)
63. SQ Huang et al., An MPS-BNS mixed strategy based on game theory for wireless mesh networks. Scientific World J **2013**, 2013 (2013)
64. M Mejia, N Peña, JL Muñoz, O Esparza, A review of trust modeling in ad hoc networks. Internet Res volume **19**(1), 88–104 (2009)
65. K Wrona, P Mähönen, Analytical model of cooperation in ad hoc networks. Telecommunication Syst **27**, 347–369 (2004)
66. S Rao, Y Wang, X Tao, The comprehensive trust model in P2P based on improved EigenTrust algorithm, in *the 2010 IEEE International Conference on Measuring Technology and Mechatronics Automation, Changsa City, 13-14 March 2010*

doi:10.1186/1687-1499-2014-78
Cite this article as: Subbaraj and Savarimuthu: **EigenTrust-based non-cooperative game model assisting ACO look-ahead secure routing against selfishness.** *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:78.