EFFICIENT QUALITY TRUST BASED ROUTING ALGORITHM TO ENHANCE SECURITY IN MANET

Mohammed Jaffer A.R¹ and Dr. Revathi S²

¹M.TECH, Software Engineering (CSE), B.S. AbdurRahman University, Chennai, India
²Associate Professor, Computer Science and Engineering, B.S. AbdurRahman University, Chennai, India

Abstract- A mobile ad-hoc network (MANET) is a peer-to-peer wireless network where nodes can communicate with each other without the use of infrastructure such as access points or base stations, these networks are self-configuring. In this paper, a new algorithm is proposed to calculate the trust and QoS metric estimation into establishing a trust-based QoS model. In this model, the estimate the trust degree among nodes from direct trust computation of direct observation and indirect trust computation by neighbors' recommendations are calculated. Due to the NP-completeness of the multi QoS constraints problem, account link delay only considered as the QoS constraint requirement. But proposed algorithm called EQTR (Efficient Quality Trust based Routing algorithm) designed from tradeoff between trust degree and link delay. The proposed algorithm is built in AODV (Ad hoc On-demand Distance Vector). The simulation results show that EQTR not only improve the security performance and also prevent attacks from malicious nodes. It improves the performance in terms of packet delivery ratio and end to end delay compared to classic AODV.

Keywords— Trust model, QoS, Efficient Quality Trust-based Routing, Trust degree and Link delay.

I. INTRODUCTION

A. MANET

Mobile Ad-hoc Networks (MANETs) comprises of a set of nodes connected by wireless links. The network is ad hoc because it does not rely on a preexisting infrastructure, such as routers in wired networks. Routing in MANETs is a challenging task due to dynamic topology and error prone shared environment. Data is sent between nodes in a MANET by hopping through intermediate nodes, which must make decisions about where and how to route the data. Although security issues in mobile ad hoc networks have been a major focus in the recent years, the development of fully secure schemes for these networks has not been entirely achieved till now. MANETs have a unique characteristics and constraints that make traditional approaches to security inadequate.

B. Challenges

The security of the ad hoc networks is considered from the attributes such as availability, confidentiality, integrity, authentication, non-repudiation, access control and usage control. Security approach used for the fixed networks are not feasible due to the salient characteristics of MANETs. So, it is necessary to design new security technique to adapt the special characteristics of MANETs and to improve the security of MANETs, one idea is to develop trust based method to allow a node to evaluate trust worthiness of other nodes.

To facilitate the implementation of the idea various trust models and trusted routing protocols are used to quantify trust relationships according to different security requirements. The basic problem with most of the routing protocols is that they trust all nodes of network and based on the assumption that nodes will behave or cooperate properly but there might be a situation where some nodes are not behaving properly. Most ad hoc network routing protocols becomes inefficient and shows dropped performance while dealing with large number of misbehaving nodes. Such misbehaving nodes support the flow of route discovery traffic but interrupt the data flow, causing the routing protocol to restart the route-discovery process or to select an alternative route if one is available.

C. Objective

To improve the security of MANETs, one natural idea is to develop trust establishment mechanisms
that allow a node to evaluate trust worthiness of other nodes. To facilitate the implementation of this idea, various trust models and trusted routing protocols, which quantify the trust relationships according to different security requirements, have been designed in the literatures. Therefore an algorithm is proposed that allows nodes to infer the trustworthiness of other nodes, especially of strangers is necessary. Recently Quality of Service (QoS) for MANETs has received increased attention. Traditional QoS routing requires not only finding a route from a source to a destination, but a route that satisfies end-to-end QoS requirements, often given in terms of throughput, bandwidth, jitter or delay.

The trust and QoS metric estimation in a trust-based QoS model is used to estimate the trust degree between nodes from direct trust computation of direct observation and indirect trust computation by neighbors’ recommendations. The trust issue with existing QoS routing and resist to malicious behaviors. Trust degree and QoS parameters estimates the classic routing and proposed approach is EQTR (Efficient Quality Trust based Routing algorithm) to enhance the security of networks.

D. Chapter Organisation

The main contributions are as follows: One, a novel model for calculate the trust among neighbor nodes and satisfying QoS requirement is proposed. Second, an approach to incorporate QoS requirements and trust degree into a routing algorithm (EQTR) is designed. Third, the EQTR is built into the classic AODV is provided.

The rest of the paper is organized as follows. Section 2 discusses related research work. Section 3 the design of the trust-based QoS model. Section 4 discusses about trust estimation. Section 5 demonstration of Route discovery using proposed an algorithm of EQTR. Section 6 evaluates the algorithm with extensive simulation results and discusses different scenarios. Finally, Section 7 concludes the paper.

II. RELATED RESEARCH WORK

Ismail et al. [1], have written survey of Wireless Sensor Network (WSN) is explained very clearly as Wireless Sensor Networking is one of the most promising technologies that have applications ranging from health care to tactical military. Security of such networks is a big concern, especially for the applications where confidentiality has prime importance. Therefore, in order to operate WSNs in a secure way, any kind of intrusions should be detected before attackers can harm the network. In this article, a survey of the state-of-the-art in Intrusion Detection Systems (IDSs) that are proposed for WSNs was presented. During first phase, detailed information about IDSs was provided. During second phase, a brief survey of IDSs proposed for Mobile Ad-Hoc Networks (MANETs) was presented and applicability of those systems to WSNs are discussed. Thirdly, IDSs proposed for WSNs were presented.

Hui et al. [2], focused on developing over a geographically limited area without well-established infrastructure. The on-demand trust-based routing protocol for MANETs, termed as Trust-based Source Routing protocol (TSR), was proposed with provides a flexible and feasible approach to choose the shortest route that meets the security requirement of data packets transmission.

Azzedine et al. [3], have written survey of requirements such as real-time or multi-cast communication in the last 15 years, the wireless networking community designed hundreds of new routing protocols targeting the various scenarios of this design space. The proposed work was to create taxonomy of the ad hoc routing protocols and to survey of compare representative examples for each class of protocols. Strive to uncover the requirements considered by the different protocols.

Ze Li et al. [4], focused the real-time transmission with stringent Quality of Service (QoS) requirements for wireless applications. At the same time, a wireless hybrid network that integrates a Mobile wireless Ad hoc Network (MANET) and a wireless infrastructure network has been proven to be a better alternative for the next generation wireless networks. By directly adopting resource reservation-based QoS routing for MANETs, hybrids networks inherit invalid reservation and race condition problems in MANETs. QoS-Oriented Distributed routing protocol (QOD) was proposed to enhance the QoS support capability of hybrid networks.

Erman and Faramarz. [5], developed Delay/Disruption Tolerant Networks (DTNs) have been identified as one of the key areas in the field of wireless communication, where in sparseness and delays are particularly high. Here emerging as a promising technology in vehicular, planetary/interplanetary, military/tactical, disaster response, underwater and satellite networks. DTNs are characterized by large end-to-end communication latency and the lack of end-to-end path from a source to its destination. These characteristics pose several challenges to the security of DTNs.
A trust-based QoS model essentially captures trust derivation, computation and application in a multi-QoS constraints environment. For the sake of simplicity, only link delay as QoS requirement is considered. The EQTR estimates the available link delay requirement by considering link quality, incorporating a trust-aware scheme into the route discovery procedure to enhance the security of network.

A. Definition of trust

A trust value is calculated from direct interactions are termed as direct trust. A trust value is built from recommendations by a trusted node or a chain of trusted nodes, which create a trust path, is called indirect trust. The use of recommendations can speed up the convergence of the trust evaluating process. The total trust relationship among nodes also contains the direct and indirect trust.

B. Trust model

Each node calculates the trust value for each of its neighbors (nodes that are within its transmission range). For the purpose of scalability, the trust degree value is calculated using local information (such as local topology information). Let $T_{ij}(t)$ denote the degree of trust of node i in its neighbor j at time t. The trust degree value is limited to a continuous range from 0 to 1. The trust degree 0 denotes complete distrust whereas the value 1 represents absolute trust. Define $T_{ij}(t)$ as the lighted average of two parts and is shown in equation 1.

$$T_{ij}(t) = w_1T_{ij}^d(t) + w_2T_{ij}^id(t).$$ (1)

$T_{ij}^d(t)$ represents the direct trust degree of node i in node j based on node i’s direct observation of node j’s packets forwarding behavior at time t. $T_{ij}^id(t)$ denotes the indirect trust degree that neighbors of node i have in node j by recommendation at time t. These neighbors of node i are also neighbors of node j. The weight factors $w_1$ and $w_2$ ($w_1+w_2 = 1, 0 \leq w_1 \leq 1$ and $0 \leq w_2 \leq 1$) are assigned to $T_{ij}^d(t)$ and $T_{ij}^id(t)$, respectively. Here, assume that node i has n neighbors (such as node k), then $T_{ij}^id(t)$ can be calculated as and is shown in equation 2.

$$T_{ij}^id(t) = \frac{1}{n}\sum_{k=1}^{n}T_{ik}^id(t).$$ (2)

$T_{ij}^id(t)$ is the average of the existing trust degrees of the neighbors at time t. The calculation of the indirect trust degree mainly depends on the neighbors’ recommendations. The neighbors’ recommendations improve the trust evaluation process for nodes that do not succeed in observing their neighbors due to resource constraints. Assessing the trust degree of each node using indirect recommendations brings several advantages.

IV. TRUST ESTIMATION

A. Computation of node trust

The trust degree values evaluated by monitoring the behavior of the neighbors form the basic blocks upon which the model is built. In this section, give
the solution that is used for estimating the node trust. For the sake of simplicity and also to minimize the overhead, use the forwarding ratio to calculate a node trust. The forwarding ratio is the number of packets forwarded correctly to the number of those supposed to be forwarded. Correct forwarding means node not only transmits a packet to its next hop but also forwards reliably. For instance, when a malicious neighbor forwards a data packet after tampering with data, this is not considered as correct forwarding. If a sender observes such a modification, the forwarding ratio of the neighbor will decrease. At time \( t \), \( T_j^i(t) \) is calculated using the forwarding ratio of node \( j \) as follows and is shown in equation 3.

\[
T_j^i(t) = \frac{F_j(t)}{T_j(t)}.
\]  

\( F_j(t) \) represents the number of packets forwarded correctly by node \( j \) at time \( t \). \( T_j(t) \) signifies the total number of packets from node \( i \) at time \( t \). Place all nodes in the promiscuous mode. When a node overhears a neighbor forwarding a packet, it should first check the forwarding behavior. Whenever it finds that its immediate neighbor nodes have received a packet to forward it increments \( R_j(t) \) by one. Whenever it finds that its immediate neighbor nodes have forwarded a packet it has to forward, it increases the \( F_j(t) \) by one. After each interaction, node \( i \) can monitor its neighbor nodes’ forwarding behavior by passive acknowledgment. If so, the trust degree between them increases. Otherwise, the trust degree decreases. The initial trust degree value is set to 0.5 (uncertain node). That is, adopt a neutral view on unknown nodes. A threshold \( \gamma \) is used to detect malicious nodes obviously \( \gamma > 0.5 \). Here, the different meanings of trust degree are listed in Table 1 and structure of the record in Table 2.

### Table 1 Different meaning of trust degree

<table>
<thead>
<tr>
<th>Level</th>
<th>Trust degree value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(( \gamma,1 ))</td>
<td>Trusted node</td>
</tr>
<tr>
<td>2</td>
<td>(0.5,( \gamma ))</td>
<td>Less trusted node</td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>Uncertain node</td>
</tr>
<tr>
<td>4</td>
<td>(0,0.5)</td>
<td>Suspect node</td>
</tr>
</tbody>
</table>

### Table 2 Structure of a trust record table

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Node trust</th>
<th>Direct trust degree</th>
<th>Indirect trust degree</th>
<th>Current time</th>
<th>Last updating time</th>
</tr>
</thead>
</table>

B. Trust-based QoS model

Each node monitors its neighbor nodes’ forwarding behavior to judge their trust degree using the simple trust model. Malicious nodes can be isolated from the network. The remaining nodes are trusted. Establishing an effective QoS evaluation model under the trusted network environment is a significant problem. In this section, give the solution to this challenge. To consider link quality, use the expected transmission count (ETX) as a metric, which was originally proposed in. It estimates the number of retransmissions required to send packets by measuring the loss rate of broadcast packets between pairs of neighboring nodes. Let \( X_j(t) \) denote the ETX measured by node \( j \) at time \( t \), which can be expressed as and is shown in equation 4.

\[
X_j(t) = \frac{1}{F_j(t) T_j(t)}.
\]  

\( F_j(t) \) is the forward delivery ratio at time \( t \), i.e., the ratio of the number of packets received by node \( j \) to the total packets it sends out; \( T_j(t) \) is the total delivery ratio at time \( t \), that is the probability that an ACK packet is successfully received by node \( j \). Then, \( F_j(t) T_j(t) \) is the probability that a probe packet is successfully sent to and acknowledged back by receiving node. Therefore, \( X_j(t) \) denotes the expected number of successful transmissions. To meet the QoS constraints, for example, link delay for delay-sensitive traffic, the delay cost consists of: transmission delay, propagation delay and waiting delay of a packet buffered in the interface queue. The transmission delay can be computed using the packet length and link bandwidth. Owing to the invariance of the packet length and link bandwidth, assume that the transmission delay equals \( \eta \). Besides, the waiting delay determined by the queue buffer size is random, so do not take it into account. Use \( \omega_j \) to denote the propagation delay a packet experiences when being delivered over a link starting from node \( j \). can compute the total link delay as follows and is shown in equation 5.

\[
\delta_j(t) = X_j(t) \cdot (\eta + \omega_j).
\]  

Considering both the trust degree of route and QoS requirements, a route that is less trusted and does not meet the desired QoS parameters must be adjusted in the network performance objective, thus a new routing cost metric \( C_r(t) \) can be designed as and is shown in equation 6.

\[
C_r(t) = \sum_{j \in R} \delta_j(t) \cdot (1 - T_j(t)).
\]  

It can be seen that a higher of \( C_r(t) \) means a higher cost metric for route \( r \), which is due to either the
lower quality, or the lower trust degree, or both. The solution to adjust the optimization problem is given as:
\[
\min \sum_{r \in R} C_r(t) \quad \text{s.t.} \quad T_{ij}(t) \geq \gamma, \quad i,j \in r, r \in R
\]

R is all available routes, y is also the threshold of trust degree of the network. For each node j on route r, the value of \( \gamma_j \), \( X_j(t) \), and \( T_{ij}(t) \) can be obtained. Therefore, node j can calculate an increase to the routing cost metric \( C_r(t) \) if its next hop is selected to join the route in order to have a minimum value of \( C_r(t) \).

V. THE PROCEDURE OF ROUTE DISCOVERY

A. The procedure of REQ delivery

The source node S wants to send a data packet to a destination node D, node S should check whether there is an available path to node D. The detailed process is given as follows: Step 1: The source node S looks up for a route entry to node D in its local routing table. The available route should meet the requirements: trust and QoS constraints. If such routes are found, go to Step 3; if there is no such a route, the source node S initiates a route discovery procedure. Step 2: The source node S checks its neighbor nodes’ trust degree by judging with the trust threshold \( \gamma \) from its local trust record table. Then, node S broadcasts route request packets REQ to its neighbor nodes with their trust degree is greater than \( \gamma \); Step 3: When the intermediate node j receives the REQ packet sent by the source node S, it first check whether it received the REQ packet: if so, it drops the REQ packet and the following procedure ends; otherwise, it collects the related information of RS\(_j\)(t), FS\(_j\)(t) and \( \delta_j(t) \) from the local routing table and trust record table at time \( t \), then it computes \( C'(t) \) as: \( C'(t) = \delta_j(t) \times (1 - T_{ij}(t)) \); in addition, node j fetches the field of C(t) from the REQ packet, updates the corresponding field with \( C''(t) = C(t) + C'(t) \), and then broadcasts the updated REQ packet to its neighbor nodes; Step 4: When the intermediate neighbor node k of node j has an available route to the destination node D, and the routing cost metric is least, that is, the trust degree of all nodes in the available route is greater than \( \gamma \).

The QoS parameter of total link delay overhead is minimum of C(t), so node k can generate a route reply packet REP to node S; Step 5: Otherwise, the node k performs the route discovery procedure similar to Step 3. Under the requirements of trust and QoS constraints, node k continues broadcasting the updated REQ packet to its next neighbor nodes, until the REQ packet arrives the destination node D; Step 6: When the destination node D receives several REQ packets, it fetches their fields of C(t), and decides to choose the optimal route which has the least C(t) by comparing the value of C(t). Meanwhile, the node D generates a route reply packet REP and sends back to the source node S along the reverse direction of optimal route. Up to this point, the procedure of REQ delivery is finished. The detailed formal description of the REQ delivery procedure is shown in algorithm 1 and 2. It gives the sub-procedures of SendingREQ and ReceivingREQ.

Algorithm 1: SendingREQ ()

1. //To the source code;
2. //Whether there is an optimal path to the destination
3. if (it exists) then //Already found that REQ
4. Sendingdata();
5. Else
6. Broadcasts the REQ packet;
7. End if
8. //To the intermediate node or the destination node
9. ReceivingREQ();

(a) SendingREQ sub-procedure.

Algorithm 2: ReceivingREQ ()

1. //When an intermediate node receives a REQ packet;
2. //Checks whether it is the destination of the route request;
3. if (it is the destination) then
4. Computes the least metric of C(t) from the existing several REQ packets;
5. //After the node chooses the feasible route destination starts
6. the REP delivery procedure
7. SendingREP();
8. Else
9. if (not duplicate REQ) then
10. Checks the freshness of the REQ packet
11. if (it is a fresh packet) then
12. Gets its value of T(t) and \( \delta(t) \) from the record tables in its caches;
13. if (T(t) = T' then
14. Computes the \( \delta(t) - \delta(t) \times T(t) \);
15. Fetches the C(t) field from the REQ packet as denoted by \( \delta(t) \) and updates the C(t) field by using \( C''(t) \) as:
16. \( C''(t) = C(t) + \delta(t) \);
17. Rebroadcasts the updated REQ packet;
18. else
19. Discards the updated REQ packet;
20. Wait for another REQ packet;
21. End if
22. else
23. Discards the REQ packet;
24. End if
25. else
26. Discards the REQ packet;
27. End if

(b) ReceivingREQ sub-procedure.

5.2. The correctness analysis of EQTR algorithm

As already mentioned, this system focuses on trust-based route selection. At anytime, a route’s trust is calculated by making use of the previously assigned trust
values of nodes. So, at time t, the trust of a route P is equal to the summation of node trust values in the route. As shown in Figure 5.1, at the time t, the Trust value of nodes B, D, C, E are 0.9, 1, 0.81, 0.7 respectively. The trust value of route P(A,B,D,F) equals 1.9. The trust value of route P(A,C,E,F) equals 1.51. The computation of route trust takes into account trust values of all intermediate nodes. Route trust denotes a joint probability at which packets will be forwarded if they are sent along the routing path.

**Figure 5.1 Route trust computation**

After the calculation, a set of path with their respective Route trust values. The route with the highest Route Trust Value is selected as the primary trustable path. Only in case of inevitable route failure, other alternative paths are considered during route rediscovery. In other words, the selection of trustable path is based on comparison of available paths based on their cost metrics.

**Table 4 Simulation parameters.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>Rectangular field</td>
<td>1000m×1000m</td>
</tr>
<tr>
<td>N</td>
<td>Number of nodes</td>
<td>50</td>
</tr>
<tr>
<td>R</td>
<td>Transmission radius of each node</td>
<td>250m</td>
</tr>
<tr>
<td>S</td>
<td>Maximum mobile speed</td>
<td>20m/s</td>
</tr>
<tr>
<td>P</td>
<td>Data payload size</td>
<td>512 bytes/packet</td>
</tr>
<tr>
<td>W1</td>
<td>Weighting factor of $T_{ij}(t)$</td>
<td>0.6</td>
</tr>
<tr>
<td>W2</td>
<td>Weighting factor of $T_{ij}(t)$</td>
<td>0.4</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Weighting factor of updating node trust</td>
<td>0.5</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time interval of trust node</td>
<td>0.05s</td>
</tr>
<tr>
<td>T</td>
<td>Simulation time</td>
<td>900s</td>
</tr>
<tr>
<td>M</td>
<td>Number of malicious nodes</td>
<td>1-20</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Threshold of trust degree value</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5 Simulation parameters in EQTR**

<table>
<thead>
<tr>
<th>Scene</th>
<th>Malicious nodes</th>
<th>Max node speed (m/s)</th>
<th>Trust update interval (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 1</td>
<td>10</td>
<td>0-30</td>
<td>0.05</td>
</tr>
<tr>
<td>Scene 2</td>
<td>0-20</td>
<td>15</td>
<td>0.05</td>
</tr>
<tr>
<td>Scene 3</td>
<td>5</td>
<td>10</td>
<td>0.02, 0.1</td>
</tr>
</tbody>
</table>

**VI. PERFORMANCE EVALUATION**

A. **Simulation environment**

The simulations are based on the IEEE 802.11b of MAC layer, which is included in the NS2. The nodes move from a random starting point to a random destination with a speed that is randomly chosen (the speed is uniformly distributed between 0 and 20s). As the destination is reached another random destination is targeted after a pause time. The transport protocol used for simulations is User Datagram Protocol (UDP). The related other parameters are listed in Table 4. Table 5 shows the simulation parameters in EQTR.

B. **Performance metrics**

Here implement the EQTR algorithm based on the classic routing protocol AODV. Use the following metrics to evaluate the performance of the routing algorithm with AODV, in which the first two metrics are the most important for best effort route and transmit protocols: (1) Packet delivery ratio: the fraction of the data packets delivered to destination nodes to those sent by source nodes. (2) Average end-to-end delay: the average time taken by the data packets from sources to destinations, including buffer delays during a route discovery, queuing delays at interface queues, retransmission delays at MAC layer and propagation time. The network topology is shown in Figure 6.1.

**Figure 6.1 Network topology during simulation time**

**C. Simulation results**

**Scene 1: Packet delivery ratio**

The effect of mobility of nodes on Delivery rate is analyzed. The mobility of nodes is determined by the pause time parameter setting in ns-2 simulator. The other parameters setting are done. Pause time is varied as 0, 100, 200, 300, 400 and 500 seconds. For each pause time, the Delivery rate is calculated, and has been found that
EQTR has better data delivery rate due to transmission along trustable routes is sown in Figure 6.2.

![Figure 6.2 Packet delivery ratio](image)

**Figure 6.2 Packet delivery ratio**

**Scene 2: Average end-to-end delay**

Evaluate the proposed protocols by varying number of nodes. When there are no inefficient nodes, the packet loss rate is very low. The reason is that, with the proportion of inefficient nodes increases, the probability packet loss tends to increase on routing routes.

The effect of load on packet loss rate is analyzed. The load in network is determined by the number of pairs of nodes in communication in ns-2 simulator. The other parameters setting are done. In Figure 6.3 it is evident that the packet loss is more in AODV than the packet loss in EQTR.

![Figure 6.3 Average end-to-end delay](image)

**Figure 6.3 Average end-to-end delay**

**VII. CONCLUSION**

A Proposed Efficient Quality Trust based Routing algorithm (EQTR) that enhances the security of network in the presence of malicious nodes. The proposed algorithm ensures the forwarding of packets through the trusted and least link delay routes only by monitoring the behavior of each other and meeting the QoS constraint accordingly. Once a malicious node is discovered, it is isolated from the network such that no packet is forwarded through or from it. The proposed Efficient Quality Trust based Routing algorithm called EQTR was implemented and integrated with NS2 based on AODV classic protocol. Its performance was evaluated through intensive simulations with AODV, Watchdog-DSR and QAODV in the presence of malicious nodes on the packet delivery ratio, average end-to-end delay, routing packet overhead and detection ratio. A comparison showed that EQTR shows better performance in most of the simulation scenarios.

In future work, will conduct extensive simulations and rigorous analysis to verify the performance of EQTR algorithm and compare it with existing protocols to evaluate the performance of applications such as intrusion detection, secure routing and key management.

**ACKNOWLEDGEMENT**

First offer our deepest gratitude to the almighty who has given us the strength and good health during complete this paper. I express profound gratitude to our respected dean Dr. K.M. MEHATA Ph.D., for his encouragement and guidance. I articulate our gratitude to our head of the department Dr. SHARMILA SANKAR M.E., Ph.D., who had taught me the way to do successful paper. I gratitude to my respected coordinator Dr. ANGELINA GEETHA M.C.A., M.E., Ph.D., who had the novel ideas for this paper. And I wish to express and sincere thanks to our internal guide respected Dr. S. REVATHI M.E., Ph.D., who had helped us lot and given valuable suggestions, which made us to finish this paper in very successful and neat way. Without whose constant encouragement and whole idea, the paper would not have been possible. Finally I want to thank my father Hazrath K.M. ABDUL RAHIM Sahib Areca nut Merchant for bestow lot of love and inspiration to grow up and make conquer in my life Thank you so much dad.

**REFERENCES**


No. 9, September 2012.


Mohammed Jaffer A.R is a M.TECH candidate of School of Computer Science and Engineering, B.S. AbdurRahman University. His main research interests include Software Engineering, Data Mining, trust computing, Mobile Ad hoc Network and wireless network routing protocol. jafermd@gmail.com.

Dr. Revathi S is an Associate Professor at the School of Computer Science and Engineering, B.S. AbdurRahman University. She received Ph.D. degrees from an Anna University at Chennai. Her research interests include wireless networking, Mobile Ad hoc Network, Security in Network and genetic Algorithm.