TRUST BASED CLUSTERING AND SECURE ROUTING SCHEME FOR MOBILE AD HOC NETWORKS

Pushpita Chatterjee
School of Information Technology
Indian Institute of Technology, Kharagpur, India
pushpitac@sit.iitkgp.ernet.in

ABSTRACT

In this paper we present a distributed self-organizing trust based clustering framework for securing ad hoc networks. The mobile nodes are vulnerable to security attacks, so ensuring the security of the network is essential. To enhance security, it is important to evaluate the trustworthiness of nodes without depending on central authorities. In our proposal the evidence of trustworthiness is captured in an efficient manner and from broader perspectives including direct interactions with neighbors, observing interactions of neighbors and through recommendations. Our prediction scheme uses a trust evaluation algorithm at each node to calculate the direct trust rating normalized as a fuzzy value between zero and one. The evidence theory of Dempster-Shafer [7], [8] used in order to combine the evidences collected by a clusterhead itself and the recommendations from other neighbor nodes. Moreover, in our scheme we do not restrict to a single gateway node for inter cluster routing.

Keywords: Ad hoc networks, Trust, Cluster, Security, Distributed Leader Election

1. INTRODUCTION

Ad-hoc networks are completely autonomous wireless temporary networks established using a group of mobile devices primarily for military, emergency and relief scenarios, where no infrastructure is available. It is a group of mobile nodes which do not require a centralized administration or a fixed network infrastructure. Also wireless links are susceptible to link attacks ranging from passive eavesdropping to active interfering. Unlike fixed hardwired networks with physical defense at firewalls and gateways, attacks on ad hoc networks can come from all directions and may target any node. Autonomous nodes have inadequate physical protection and can be captured, compromised, and hijacked easily. Attacks from a compromised node are more dangerous and much harder to detect. Damage includes leaking secret information, interfering message and impersonating nodes, thus violating the basic security requirements. All these mean that every node must be prepared to encounter with an adversary directly or indirectly.

Due to dynamic topology of the networks any security solution with static configuration would not be sufficient. Moreover, an authority responsible for distribution of keys for the whole network is vulnerable to single point failure. So we require a distributed architecture for this kind of network for its proper functionality. Any node must be prepared to operate in a mode that should not immediately trust on any peer. If the trust relationship among the network nodes is available for every node, it will be much easier to select proper security measure to establish the required protection. Moreover, it will be more sensible to reject or ignore hostile service requests. As the overall environment in ad hoc network is cooperative by default, these trust

Initial deployment of mobile nodes (say 15 nodes) and their transmission range and probable cluster formation shown in Fig.1. Small dark circles are nodes and concentric large circles are their transmission range.

The rest of the paper is organized as follows. Section 2 describes the related work on ad hoc network security solution and clustering algorithms along with trust management and leader election. In section 3, the proposed framework like clusterhead election, node registration has been described. Section 4 deals with secure intra and inter cluster routing. The trust measurement and combination of different evidences using the Dempster Shafer model and how this model can be used for formalizing the trust has been presented in section 5. Section 6 describes the analysis of proposal along with simulation experimentations.

1. RELATED WORK

The research works on cluster based ad hoc network security analysis can be broadly classified into three categories:

(a)Secure Clustering and Leader Election- Several algorithms like WCA [12], [6] are proposed for clustering the ad hoc networks. But none of them is able to completely handle the secure clustering. Moreover they do not specify the well defined clustering mechanism, new cluster head selection, and other important issues. Among several secure solutions based on clustering ad hoc networks, Varadharajan et al [2], uses NDTR architecture for clustering but does not deal with partitioning and merging of clusters. Researchers like Sudarshan et al [16], Vaidya et al [15] have proposed several algorithms for distributed leader election, in ad hoc network scenario but the malicious nature of the nodes is not considered.

(b)Secure Routing - Some clustering mechanism and routing mechanism are proposed in [13], [10], and [6]. A cluster based security architecture is proposed by Becheler et al. [4], which uses threshold cryptography scheme to distribute CA (Certification Authority). This approach is not realistic, because the warrantor does not have any information about the new node to be guaranteed. The network traffic generated by each new node is very high thereby causing wasting of both bandwidth and energy. Also, to renew the network key, the intervention of a trusted third party is needed so that it can subdivide the new key and distribute the fragment of the key over clusterheads. Rachedi et al [3] proposes a clustering algorithm based on trust and a DDMZ (Dynamic Demilitarized Zone) for protecting CAs for overcoming the drawbacks of [4] by hierarchical monitoring of nodes. But in that paper it is not clearly described how such a firewall like RA can be implemented in a self organized pure ad hoc network and protect against different kind of DoS attacks. Moreover, intra and inter cluster routing is not properly formulated.

(c)Handling Reputation and Trust Management Issues - The distributed trust model adopted by Abdul-Rahman and Hailes [11] is a decentralized approach for trust management. It uses a recommendation protocol to exchange trust-related information. It is applicable to any distributed system and not specifically targeted for ad hoc networks. Pretty Good Privacy [9] is an example of system proposed by using a web-of-trust authentication model; it uses the public key certificate. Hubaux et al. [5] proposed self-organized public key management system for fully self-organized ad-hoc network; the idea is that each user maintains a local certificate repository. This approach has two drawbacks: First, each user is required to build his local certificate repository before being able to use the system. Second, this approach assumes that trust is transitive, which is not always true. Virendra et al [14] proposes a technique for quantifying trust. In our model we are adapting a similar process to measure the trustworthiness of a neighbor node depending upon some metrics based on system requirements.

So existing literatures primarily concentrate on the clustering the networks but no one has taken care of trust level of nodes while electing the clusterhead (CH) or intra and inter cluster routing. There are few works on the secure leader election but they do not concentrate if any node advertises itself as leader maliciously then what measures should be taken. Again, a complete framework for cluster formation and trusted secure leader election and other important issues like inter cluster routing has not been addressed earlier.

1. PROTOCOL DETAILS

Our primary goal is to develop a distributed trust based framework for securing ad hoc networks and to devise a prediction scheme to evaluate degree of trust of each mobile node in the network. In this section we describe the assumptions, notations and metrics.

3.1 Assumptions

All nodes communicate via a shared bi-directional channel and operate in promiscuous mode. In other words, after each forwarding the node can hear if the intermediate node has forwarded the message to the destination or not. All nodes are identical in their physical characteristics, that is, if a node A is within the transmission range of B then B is also within the transmission range of A. It is also assumed that all nodes are equipped with a residual energy detection device and some energy consumption model.

Using the pair-wise key pre-distribution scheme, keys are distributed over the nodes of the network. After election, a network key is generated by the CHs. Any node wants to become a CH has to get access to the network key which is only sharable by the CHs. There are other keys also for secure communication, CH-group-key, the pair-wise secret key generated by pair of neighboring CHs to communicate to each other. Each mobile node maintains a Trust-Table of its one hop neighbors along with trusted pair-wise key for peer to peer communication without intervention of CH. Maximum allowable distance between any mobile node and CH will be one.

3.2 Notation

We will use the following notations to describe the methods of initialization, key generation, trust evaluation, node registration, and intra and inter cluster routing and node unjoin due to mobility for secure end-to end communication between mobile nodes.

$\mathbf{M_{id}}$	- Mobile Node Identity
$\mathbf{CH_{id}}$	- Clusterhead Identity
CH-k	- Private Key for CH
$\mathbf{K_{X\text{-}Y}}$	- Pair-Wise Encryption Key
$\mathbf{K_{CHX\text{-}CHY}}$	- Shared Secret Key between two clusterheads
CERT	- Trust Certificate
$\mathbf{K_{SYM\text{-}M}}$	- Symmetric key of Mobile Node

3.3 Metrics

- **Trust (T_V):** The overall description of calculation of Trust metric is described in Section 5.
- **Battery Power (B_P):** It is the estimation of time any node will be in active mode (can relay traffic and perform other basic functionalities) with its remaining battery capacity, this metric we refer as **Battery Powe**r.
- **Mobility (M_V):** If any node at any point A at time t_1 and at B at time t_2 then finding the Euclidian distance between A and B and **Mobility** is predicted by dividing the distance by the time interval (t_2-t_1).

3.4 Secure Distributed Leader Election Algorithm

After deployment each node sends "HELLO" beacon and try to find out how many nodes are deployed in its broadcasting range. Each node receiving this HELLO beacon replies with M_{id}

and public key. Each node getting this "REPLY" beacon increases the counter of its neighbor list, and stores the M_{id} and public key. Then an efficient secure distributed leader election algorithm SEC-LEAD is executed that can adapt arbitrary topological changes. To reduce the computation overhead the CH selection mechanism only resumes if the existing CH runs off its battery or the CH has to move from its previous position.

Algorithm Secure Distributed Leader Election (SEC-LEAD)

Step 1: A node (say M) wants to be CH, broadcasts "START-ELECTION" message with its mobility, battery power value to all its one hop neighbors.

Step 2: Getting this message each node within its broadcast range, calculates the global weight of that candidate node using a global function. $\mathbf{G_w= w_1*T_V+w_2*M_V+w_3*B_P}$, where w_1, w_2, w_3 are different weights such that ($w_1+w_2+w_3=1$)

Step 3: If G_w is greater than a predefined threshold, the node will vote for M by signing a Leader Certificate. Sends it to M.

Step 4: After a certain time interval (say T_{Elect}), the candidate node will count how many certificates it has already received**.**

Step 5: If this is greater than n/2 (where n is the number of neighbor nodes), it advertises itself as leader and broadcasts the leader message with the set of node-ids who has voted for it.

Step 6: If any node finds that its id is falsely included, it will generate a warning message to all its neighbors.

Step 7: After certain time say TCH, neighbor nodes will sign a TrustCert for Leader, sends it to M. (as self-organized Public Key Infrastructure) [see reference[5]]

Step 8: Thus M becomes a Leader and the elector nodes who has signed the certificate becomes its member**.**

3.5 Node Join or Registration

After deployment, CHs communicate between each other and find out their neighbor CH and generates shared key $K_{CHX-CHY}$ between them. When a new node wants to join in the network, the registration procedure of a new node is described in Algorithm 2.

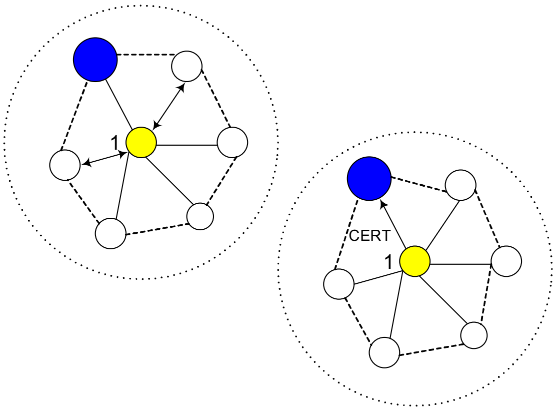

Fig 2: (a) Trust Information Collection (b) Trust Certificate Generation

Node under review shown in dark circle and the node marked as **1** is the corresponding CH and collection of recommendation trust from the one-hop neighbor nodes of the node and generation of Trust Certificate (CERT) are shown in Fig. 2.

Algorithm Node Registration (NOD-REG)

```
Step 1: Each CH starts to broadcast CH beacon and attracts some nodes to join its cluster.
Step 2: As the node M gets the CH beacon, it sends "JOIN" beacon to join the network with its
        public key.
Step 3: CH checks whether it is a duplicate message or not. If it is not a duplicate, CH stores
        the public key of M as its id and generates a pair wise shared key to communicate
        between CH and M. Also sends a secret key for secure intra cluster communication.
Step 4: Initially CH gives the node as Suspicious status and allows it to register subject to
        periodic review.
Step 5: Then CH sends a "WhocanSense" message along with the status of the newly joined
        node to its member nodes to review the status of the node.
Step 6: CH executes the Algorithm (described in algorithm CAL-TRUST) and calculates its
        direct trust about M. CH asks its one hop members of M to send their recommendation
        for M.
Step 7: CH executes Dempster-Shafer theory of combining evidences (described in Section
        5.2) to find the most probable belief of M.
Step 8: If Trust is higher than a threshold CH sends a trust certificate CERT. Thus M becomes
        a Trusted Member of the cluster.
```

For secure distribution of Trust Certificate, CERT is encrypted by the public key of the particular node. Thus, if any malicious node may able to sniff the CERT, it will not be able to get the CERT until it knows the symmetric key of the particular node for which this CERT is generated. So security is assured here. If any node with Suspicious status does not cooperate to the network, that is having a lower trust value, the CH sends a Warning message. In the next review if the CH finds the Warned node is still misbehaving, CH isolates it from the cluster and informs others that the node has been isolated.

3.6 Node Unjoin

Each node has to send an ALIVE beacon to CH at a certain time interval. If the CH cannot hear from a node at a certain time out, there will be two possible reasons: One is due to mobility the silent node may move to such place, from where it cannot sense the CH or the node is damaged. CH broadcasts a "WhocanSense" message and tries to sense the Silent node. If any node gives any reply to this message, the CH will try to establish a path to the silent node through that answering node and ask for its location and detail information about new CH. If the old CH gets the information about the new CH, the old CH sends the Trust CERT for that node to the new CH. Or if the CH is not getting any reply from any node about the Silent node the CH thinks either the node is damaged (that is no node can sense the particular node). Or it goes beyond the communication range of CH, CH just removes the information of the Silent node from its list.

1. SECURE ROUTING

In this section we will describe the intra and inter cluster routing. We consider that any node having status other than ``Trusted" will not be able to communicate outside the cluster. Thus the communication is secured. For intra cluster routing, any node is able to communicate to other within the cluster either encrypting the message by the pair wise secret key if two communicating nodes are in broadcast radio range, else either by other intermediate node to reduce the work load of the clusterhead or through clusterhead if no intermediate path is found. Each node is compelled to forward others' packet otherwise the trust level of the misbehaved node will be decreased and later it will be removed. Trust metrics are chosen in such a way that misbehavior of any node can be determined very easily. The next algorithm (Algorithm INTRA-ROUT) describes the intra cluster communication.

4.1 Intra Cluster Routing

Assume that mobile nodes M1, M2 are in same cluster and under same CH. M1 wants to communicate with M2. For this intra cluster communication we propose an algorithm described in Algorithm INTRA-ROUT.

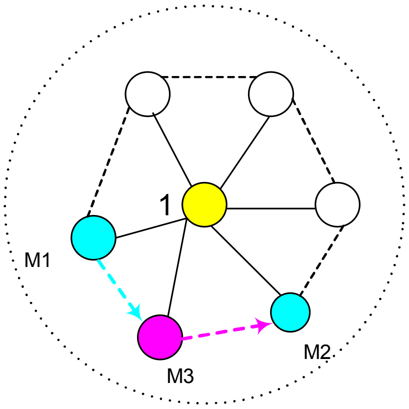

Fig 3: Intra Cluster Routing

Algorithm Intra Cluster Routing (INTRA -ROUT)

Step 1: M1 will check its neighbor list. If M2 is in the one hop neighbor list then it just encrypts the message by the pair-wise key K_{M1-M2} generated by M1 and M2 and sends it to M2.

Step 2: If it is not, M1 will ask the CH1 for the public key of M2. And asks its one hop neighbors that any node has any path to M2. Say, any node M3 (other than CH) responds that it has a path to M2.

Step 3: If no node responds to M1, M1 asks for the public key of M2 to CH1.

Step4: M1 generates a key (Ks) and encrypts (Ks) with the public key of M2 and encrypts the message with Ks, and sends it via M3 or CH1.

Step 5: Getting the message M2 sends an acknowledgement via the reverse path.

The pictorial representation of this procedure is given in Fig.3, where (M1 communicates to M2 via M3) in a same cluster and the CH represents as 1. As the message is encrypted with the session key and also the session key is encrypted by public key of M2, only M2 can have the access of it. So passive eavesdropping is not possible.

4.2 Inter Cluster Routing

Assume that mobile nodes M1, M2 are in different clusters C1 and C2 under Clusterheads CH1 and CH2. M1 wants to communicate with M2. For this inter cluster communication we devise the Algorithm INTER-ROUTE through trusted member node or through Clusterheads. Inter cluster communication using CHs as an intermediate node is shown in Fig.4. The proposed inter cluster routing can prevent Man-in-the-middle attack and passive eavesdropping.

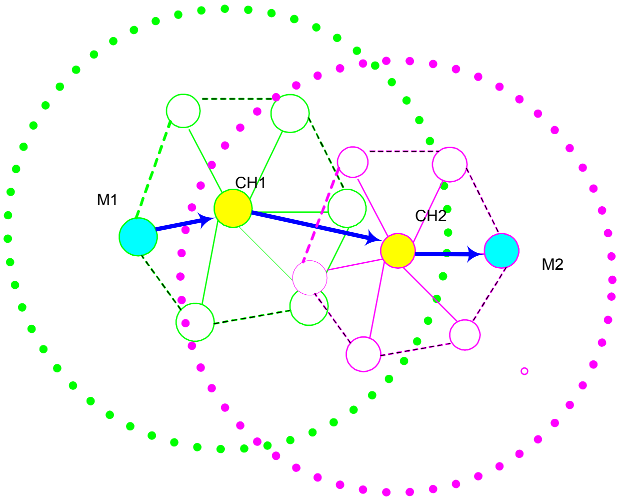

Fig. 4: Inter Cluster Routing

Algorithm Inter Cluster Routing (INTER-ROUTE)

Step 1: M1 sends the route request to CH1.

Step 2: CH1 checks the status of the M1; if M1 is not **Trusted**, CH1 just drops the request and generates a message. If M1 is trusted, CH1 generates a OK message.

Step 3: M1 starts Route discovery to get M2. Do Step 4 to Step 6, if M3 under CH1 responds in positive. If no member node replies do Step 7 to Step 9.

Step 4: M1 sends the route to CH1. CH1 checks the status of M3; if it is trusted, the CH1 generates a session key, Ks for inter cluster communication for M3.

Step 5: M3 gives reply to M1 with the public key of M2. M1 encrypts the Ks with the public key of M2 and encrypts the message with Ks and sends via M3.

Step 6: M3 gets the message and generates a session key with M2 and encrypts the total message with that session key and sends the message to M2.

Step 7: M1 sends the message to CH1. CH1 multicasts "Whocansense" query to all it neighbor CHs for having a communication to CH2.

Step 8: If CH2 replies or any other CH replies that it can sense CH2, CH1 initiates a route discovery request and asks for the public key of M2.

Step 9: M3 getting the public key CH1 encrypts the Ks with the public key of M2 and encrypts the message with Ks and sends over the discovered route.

Step 10: After a successful receipt M2 sends an acknowledgement via the same path.

1. TRUST EVALUATION

Initially, CH sets the status of a newly joined node as "Suspicious" and tries to evaluate the trustworthiness of the node. There are a lot of parameters for determining the trust level of a node. As we want to devise a mechanism for evaluating the trust of a node according to its contribution towards proper functioning of the network and minimizing the number of bad nodes from the network, we are dealing with the metrics given below:

5.1 Trust Metric

$f =$ No. of packets forwarded
$d =$ No. of packets dropped
$m =$ No. of packets misrouted
$i =$ No. of packets falsely injected
$R_p =$ Total no. of packets received by B sent from A
$S_p =$ Total no. of packets sent by B to A

```
Algorithm Trust Evaluation (CAL-TRUST)
Step 1: Collect data for Rp, Sp, f, d, m, i.
Step 2: Find the threshold values associated to each behavior $f_n$, $d_n$, $m_n$.
Step 3: Calculate ratio $f_s$, $d_s$, $m_s$, $i_s$ of each behavior and $R_p$, $S_p$ total sent or received packet
        accordingly.
Step 4: Calculate the deviation $f_d$, $d_d$, $m_d$, $i_d$ from the corresponding threshold.

        $f_s = f / R_p$      and   $f_d = fn / fs$
        $d_s = d / R_p$      and   $d_d = d_n / d_s$
        $m_s = m / R_p$      and   $m_d = m_n / m_s$
        $i_s = i / S_p$      and   $i_d = i_n / i_s$

Step 5: Calculate the corresponding direct trust value using the formula
        $Trust (t) = (w_1*f_d) - (w_2*d_d) + (w_3*m_d) + (w_4*i_d)$
```

After collecting the information about B, A will run the Algorithm CAL-TRUST and calculates its Direct trust about B. Whenever CH asks A's opinion about B, it will send the trust value. And w_1, w_2, w_3, w_4 are predefined weights {0, 1} for cooperative and non-cooperative behaviors. Using the similar algorithm CH will calculate its direct trust about B.

5.2 Dempster Shafer Theory of combining Evidences and its application to Trust Prediction

The Dempster-Shafer (DS) theory for uncertainty was first developed by Arthur Dempster [7] and extended by Glenn Shafer [8]. The theory provides necessary tools to combine various evidences and gives them various weightings, according to their importance in the final decision making, their quality and relevance. We justify the use of the DS theory by the uncertain nature of the trust prediction problem and the need to combine the different criteria (evidences).

We suppose that we are concerned with the value of some quantity u, and the set of its possible values is U. The set U is called frame of discernment. In our prediction scheme, the frame of discernment U is a trust value of mobile node which is able to become the trusted nodes in future. The frame of discernment is U {T, ¬T}, m(A) represents the exact belief committed to A, according to the evidence associated with each node's opinion about the Suspicious node. If m(A) > 0 then A is called a focal element. The focal elements and the associated bpa define a body of evidence. To each subset of U is assigned a probability that represents the belief affected by the evidence. This confidence value is usually computed based on a density function m: 2U → [0, 1] called a basic probability assignment (bpa) function.

$m(\phi)=0, \ \sum_{A \subseteq U} m(A)=1$

From any neighbor node CH has got the information and the following probability assignments are given. If received trust value t>0.5, the node is treated as trusted. If received trust value t<0.5, node is treated as untrusted and the probability is assigned accordingly.

m1 ({T}) =0.8
m1 ({¬T}) = 0
m1 ({T,¬T}) =0.2 [This state is for Suspicious]

And the CH has the probability assignments on the same node

m2 ({T}) =0.6
m2 ({¬T}) = 0
m2 ({T, ¬T}) =0.4 [This state is for Suspicious]

5.2.1 The Dempster Combination Rule

Let m1 and m2 be the bpa associated with two independent bodies of evidence defined in a frame of discernment U. The new body of evidence is defined by a bpa, m on the same frame U.
$K=\sum_{B\cap C=\phi} m1(B)m2(C)$
$m (A)= m1\otimes m2=K^{-1}\sum_{B\cap C=A} m1(B)m2(C)$

The rule focuses only on those propositions that both bodies of evidence support. The new bpa regards for the bpa associated with the propositions in both bodies that yield the propositions of the combined body. The K of the above equation is a normalization factor that ensures that m is a bpa. The combination rule is commutative and associative. In our approach, the clusterhead computes the trust of each node according to each criterion (evidence) and combines them two by two. An example solution is illustrated in Table 2.

Table 2. An example of combining evidences using DS Theory

		m2		
m1		{T}:0.6	{¬T}:0	{T, ¬T}:0.4
	{T}:0.8	.24	0	.32
	{¬T}:0	0	0	0
	{T, ¬T}:0.2	.12	0	.08

Therefore,
m1⊗m2 ({T}) = (1) (0.24+0.32+0.12) =0.68
m1⊗m2 ({¬T}) = (1) (0) =0
m1⊗m2 ({T, ¬T}) = (1) (0.08) = 0.08

So the given evidence presented here by m_1 and m_2, the most probable belief for this Universe of discourse is T with probability 0.68. Any CH will execute this algorithm for getting the most probable belief after collecting recommendation trust from others and calculating the direct trust using the above said algorithm.

1. ANALYSIS AND SIMULATION

In order to quantify the trust metric used to calculate the trust and for a good prediction of threshold value (minimum value) for proper functionality of the ad hoc network we have simulated our proposal using the Prowler simulator in this research work because this simulator, easy to use, and it is available online [17]. We have used the Prowler network simulator to evaluate the protocol performance and specifically to measure the thresholds. It can incorporate arbitrary number of nodes on arbitrary and even dynamic topology. Prowler models all the important aspects of the communication channel and the application. The tool is implemented in MATLAB, thus it provides a fast and easy way to prototype applications, and has nice visualization capabilities. The non-deterministic nature of the radio propagation is characterized by a probabilistic radio channel model. The simulations are all performed using Prowler under the default radio model. The average radio range of transmission was a radius of 10m. However, the radio model in Prowler was set up to model the transmission range as an imperfect circle. The network setup consisted of 100 nodes dispersed in an area depicting 100 m X 100 m. The simulations are run on random networks model, where the nodes placements are changed randomly in uniformly square area. The nodes are deployed in a regular grid with random offsets. We assume that all the nodes start with uniform energy. To measure the performance of our system we identify availability as one of the most important parameter of security architecture for ad hoc networks. **Availability** is the ratio of No. of Trusted members and Total No. of member nodes in a system. We studied the impact of various threshold values of cooperative and uncooperative behavior of the nodes.

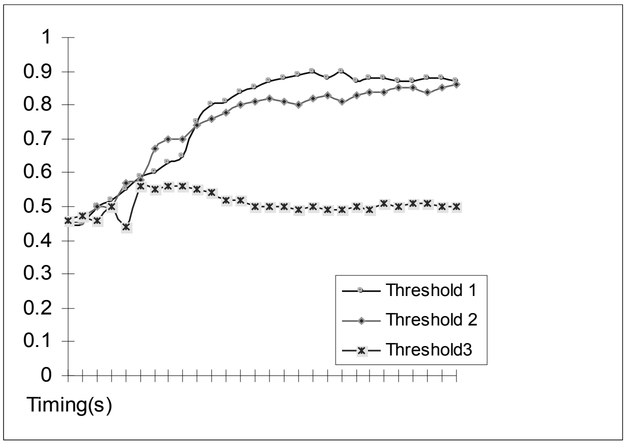

Fig 5: Availability of nodes at different Thresholds

From the Fig. 5, it can be clearly identified three possible cases:

- At beginning when only clusterheads are identified and new nodes are given *Suspicious* status and allow to send and receive packets, the security architecture establishes slowly and about 55% nodes are able to communicate securely.
- As time goes on and the Clusterhead starts to generate Trust Certificate CERT depending upon the trust value calculated by monitoring the behaviors the availability reaches up to 80%.
- After that the availability of trusted nodes reaches near to 90%.

We have compared three sets of threshold values for optimizing the thresholds.

- With a combination of 80% cooperative and 30% non-cooperative behavior we have plotted in the Threshold-1 graph
- With a 70% (good behaviors) and 40% (bad behaviors) combination in Threshold-2 graph
- A combination of 90% and 40% plotted at (Threshold-3) graph.

From Fig.5**,** it can be clearly seen that though we are decreasing the threshold for a co-operative behavior keeping the malicious rating same the Availability is decreased. So the threshold must be chosen carefully as a high value may cause low Availability but a low value may infringe the trustworthiness required for secure communication. So the value proved to be a good choice of threshold is 75% for proper data forwarding and 10% for packet dropped, 10% for packet misrouted and 20% for false packet injection. Due to find the trustworthiness depending upon the severity of malicious activity, weights given to proper forwarding was chosen high is w_1=0.8, the next higher weight given to packet dropped w_2=0.3, w_3=0.05 for packet misrouted and w_4=0.03 for falsely injected packets. The parameters, packet misrouted and packet falsely injected, were given low weights. In our simulation we have used the aforesaid weights. Note that these values are arbitrary at this time and may be adjusted according to system requirements.

We use mathematical model of Dempster-Shafer theory of evidence. The advantage of this theory is its capability to represent uncertainty which is the main problem of trust prediction. The originality of our work consists of combining different metrics for formulating trust and the use of DS theory in order to predict the trust of mobile node. In Fig. 6, it can be clearly seen that using DS Theory, the probable belief is more accurate. So far we have developed the trust evaluation algorithm to find out the threshold values for quantifying trust metric and good system performance.

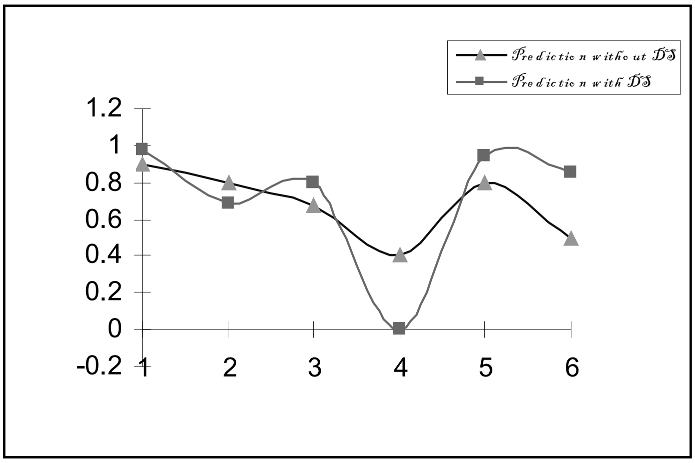

Fig 6: Prediction using DS Theory

1. DISCUSSION

The proposed trust based clustering framework along with a novel distributed leader election mechanism ensures that the clusterhead selection and cluster formation in the ad hoc network is secure. It is to be noted that initially a node given the status *Suspicious* node should be restricted to intra cluster communication until it gets Trust Certificate CERT. This certificate is also subject to review. As the trust value of a particular node depends on its participation towards proper functionality of the network each node must cooperate and the network can be prevented from inside malicious attacks. Moreover, we use mathematical model of Dempster-Shafer theory of combining evidences. The advantage of this theory is its capability to prediction. As we are combining different opinion collected from different member nodes, the DS Theory will provide the most probable belief and the prediction will be more accurate. It will help the clusterhead to give the status of a member node and an overall trusted environment framework will be created.

1. CONCLUSION AND FUTURE WORK

In today's scenario, importance of ad hoc networks is increasing day by day. Till date researchers mostly concentrate on routing issues and a little bit on various attacks. But there is a need of resilient system which can operate properly in the presence of malicious activities. So, there is increasing requirement for trust establishment. Some works have been done in this area but no model is sufficient to provide a complete framework. Moreover, it is more challenging to ensure that the security implementation conforms the global framework although there are certain link failures in the network.

In this paper we have proposed a new approach based on trust based self-organizing clustering algorithm. Only few works have been done in this field. The majority of security solutions were based on traditional cryptography which may not be well-suited with dynamic nature of ad hoc networks. We have used the trust evaluation mechanism depending on the behavior of a node towards proper functionality of the network. Our trust evaluation model gives a secure solution as well as stimulates the cooperation between the nodes of the network. We are not only restricting to direct observation for predicting trust but also recommendation from one hop neighbors of any node under review. The originality of our work consists of combining different metrics for quantifying trust and the use of DS theory in order to predict the trust of mobile node more accurately. In future we plan to compare our proposal with other existing proposals and to consider other issues like secure movement and location management of individual node to provide a better robust and secured solution.

References:

1. Pushpita Chatterjee, Indranil Sengupta and S.K. Ghosh.: A Trust Based Clustering Framework for Securing Ad Hoc Networks, in Proc. of ICISTM 2009, CCIS 31, pp. 313-324, 2009.
2. Vardhanrajan et al: Security for cluster based ad hoc networks, in Proc. of Computer Communications 27(2004), pp. 488-501.
3. Rachedi et al: Trust and mobility based clustering algorithm for secure ad hoc networks, in Proc. of ICSNC '06, October, 2006. ISBN: 0-7695-2699-3.
4. Marc Bechler and Hans-Joachim Hof and Daniel Kraft and Frank Pahlke and Lars Wolf: A Cluster-Based Security Architecture for Ad Hoc Networks, in Proc.of IEEE INFOCOM, 2004.
5. Hubaux, J. P., Buttyan, L. and Capkun, S. (2001): The Quest for Security in Mobile Ad Hoc Networks, in Proc. of ACM Symposium on Mobile Ad Hoc Networking and Computing, 146-155.
6. Zhou, L. and Haas, Z. J. (1999): Securing Ad Hoc Networks, in IEEE Network Magazine, 13(6).
7. P. Dempster: A generalization of Bayesian interface , Journal of Royal Statistical Society (1968), 205-447
8. G. Shafer: A Mathematical theory of Evidence, Princeton University Press, 1976
9. Garfinkel, S. (1995): PGP: Pretty Good Privacy, O'Reilly Associates, Inc.
10. Pirzada, A. A. and McDonald, C. (2003): A Review of Secure Routing Protocols for Ad hoc Mobile Wireless Networks, in Proc. Of 2nd Workshop on the Internet, Telecommunications and Signal Processing DSPCS'03, WITSP'03.
11. Rahman, A. A. and Hailes, S. (1997): A Distributed Trust Model, in Proc. of the ACM New Security Paradigms Workshop, 48-60.
12. M. Chatterjee, S.K. Das and D. Turgut: An on-demand weighted clustering algorithm (WCA) for ad hoc networks, in Proc. of IEEE GLOBECOM 2000, San Francisco, November 2000, pp. 1697- 1701.
13. S. Basagni: Distributed clustering for ad hoc networks, in Proc. of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310-315.
14. Virendra et al: Quantifying trust in Mobile ad hoc networks, in Proc. of KIMAS, 2005, April 18-21, 2005, Waltham USA.
15. N. Malpani, J. Welch and N. Vaidya. Leader Election Algorithms for Mobile Ad Hoc Networks, in Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, MA, August, 2000.

16. Sudarshan Vasudevan, Brian Decleene, Neil Immerman, Jim Kurose, Don Towsley: Leader Election Algorithms for Wireless Ad Hoc Networks in Proc. of DARPA Information Survivability Conference and Exposition, 2003.
17. Simon G, Volgyesi P, Maroti M, Ledeczi A.: Simulation based optimization of communication protocols for large-scale wireless sensor neworks, in Proc. of IEEE Aerospace Conference, Big Sky, MT, March, 2003.

Authors:

Pushpita Chatterjee is a PhD candidate in the School of Information Technology at Indian Institute of Technology, Kharagpur, India. Her area of research is routing security and trust computation in wireless ad hoc and sensor networks. She received her B.S and M.S. in Computer Science and Engineering from University of Calcutta, India and M.Tech in Computer Engineering from University of Calcutta, India.
E-mail: pushpitac@sit.iitkgp.ernet.in