ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Application of Indirect Trust Computation in MANET

Mr.Rahul.A.Jichkar[1], Dr.M.B.Chandak[2]

M.Tech Scholar, Department of Computer Science & Engineering, RCOEM, Nagpur, India[1]

Head of Department Dept. of Computer Science & Engineering, RCOEM, Nagpur, India[2]

Abstract: The mobile ad-hoc network consists of low power mobile devices communicating through radio signals. To minimize the power consumption, we form the clusters with an elected cluster head (Service Provider) based on any cluster head selection strategy. As the topology of this network is time variant due to node mobility, nodes are continuously leaving and entering the clusters, automatically registering with the cluster head to become the member of the cluster. But, there may be a scenario where a new node wants to access a service provided by the cluster head, at this time the newly entered node is unaware of the trustworthiness of the cluster head. To establish a trusted link between newly entered node and CH we have adopted an indirect trust computation technique based on recommendations form an important component in trust-based access control models for pervasive environment. It can provide the new node the confidence to interact with unknown service provider or CH to establish a trusted link for reliable accessibility of the service. In this paper, we will present some existing indirect trust based techniques and subsequently discuss our proposal along with its merits, demerits and future scope.

Keywords: MANET, Cluster Head, Cluster, Trust, Outlier Detection.

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are composed of fully autonomous wireless devices forming a temporary networks, these are suitable in the environment where the infrastructure is not fixed or we can say that infrastructure is time variant. In case of fixed hard-wired networks attacks are predictable with physical defence at firewalls and gateways, whereas, attacks on MANETs can come from any directions and may target any node. Due to dynamic topology of the networks any security solution with static configuration are not sufficient. Any node participating the network should not be directly trusted without verifying its trust information. If the trust information is available for each and every node in the network, then it is convenient to take precautionary measures to prevent the attacks using appropriate intrusion detection techniques. Moreover, it will be more sensible to reject or ignore hostile service requests. As the overall environment in MANET is cooperative by default, these trust relationships are extremely susceptible to attacks. So, in order to avoid the overhead of handling the network as a whole, nodes are grouped into clusters.

There are number of cluster formation strategies which are used to form the clusters, most of these techniques are based on the degree of connectivity of a particular node. A node having highest degree of connectivity i.e. it is surrounded by maximum neighbours, then the node is elected as cluster head. There are also some other parameters such as, energy, node-id, etc. which can be taken into consideration while electing a cluster head. For simplicity we will be considering the static mobility model with heterogeneous nodes i.e. each node will be having different capabilities. The node having more computing power and battery life can be elected as a cluster head, as it will handle inter cluster communication and computation.

In this paper, we will be focusing on the trust evaluation mechanism based on the second hand information, for that we are assuming a cluster formed by heterogeneous wireless devices of which only some devices is having the capability to become cluster head and each of this device will become cluster head for particular time slice which is analogous to round robin fashion.

II. RELATED WORK

The dynamism of pervasive computing environment allows ad hoc interaction of known and unknown autonomous entities that are unfamiliar and possibly hostile. In such environment where the service requesters have no personal experience with unknown service providers (here cluster heads), trust and recommendation models are used to evaluate the trustworthiness of unfamiliar entities. Recently, research in designing defence mechanisms to detect dishonest recommendation in these open distributed environments has been carried out [1-18]. The defence mechanisms against dishonest recommendations has been grouped into two broad categories, namely exogenous method and endogenous method [1].The approaches that fall under endogenous method use other external factors along with the recommendations (reputation of recommender and credibility of recommender) to decide the trustworthiness of the given recommendation. However, these approaches assume that only highly reputed recommenders can give honest recommendations and vice versa. In endogenous method, the recommendation seeker has no personal experience with the entity in question. It relies only on the recommendations provided by the recommender to detect dishonest recommendation. The method believes that dishonest recommendations have different statistical patterns from honest recommendations. Therefore, in this method, filtering of dishonest recommendation is based on analysing and comparing the recommendations themselves. In trust models where indirect trust based on recommendations is used only once to allow a stranger entity to interact, endogenous method based on the majority rule is commonly used. Dellarocas [13] has proposed an approach based on controlled anonymity to separate unfairly high ratings and fair ratings. This approach is unable to handle unfairly low ratings [14]. In [15], a filtering algorithm based on the beta distribution is proposed to determine whether each recommendation R*i* falls between q quartile (lower) and (1 − q) quartile (upper). Whenever a recommendation does not lie between the lower and upper quartile, it is considered malicious and its recommendation is excluded. The technique assumes that recommendations follow beta distribution and is effective only if there are effectively a large number of recommendations. Weng et al. in [16] proposed a filtering mechanism based on entropy. The basic idea is that if a recommendation is too different from majority opinion, then it could be unfair. The approach is similar to other reputation-based models except that it uses entropy to differentiate between different recommendations. A context-specific and reputation-based trust model for pervasive computing environment was proposed [17] to detect malicious recommendation based on control chart method. The control chart method uses mean and standard deviation to calculate the lower confidence limit (LCL) and upper confidence limit (UCL). It is assumed that the recommendation values that lie outside the interval defined by LCL and UCL are malicious, therefore discarded from the set of valid recommendations. It considers that a metrical distance exists between valid and invalid recommendations. As a result, the rate of filtering out the false positive and false negative recommendation is really high. Deno et al. [18] proposed an iterative filtering method for the process of detecting malicious recommendations. In this model [18], an average trust value (T_{avg}) of all the recommendations received (TR) is calculated. The inequality $| T_{avg} (B) − TR(B) | > S$, where B is the entity for which recommendations are collected from i recommenders (R) and S is a predefined threshold in the interval [0 1], is evaluated. If that inequality holds, then the recommendation is false and is filtered out. The method is repeated until all false recommendations are filtered out. The effectiveness of this approach depends on choosing a suitable value for S. These detection mechanisms can be easily bypassed if a relatively small bias is introduced in dishonest recommendations.

III. PROPOSED APPROACH

The main goal of Mobile Ad-hoc Network is to establish trusted connection amongst each other. We can define scenarios in MANET where a newly joined node wants to establish a secure connection with a particular CH from the set on CHs. To evaluate the trustworthiness of these CHs we have adopted indirect trust mechanism approach [19] and we have evaluated the performance and effectiveness of this technique for evaluating the trustworthiness of particular service provider (CH). The trust in short is computed on the basis of recommendations from the associated members which are frequently interacting with the service providers and the other members in the pervasive environment. In this approach, we define a scenario consisting of cluster formed using heterogeneous mobile nodes with static mobility model, in this cluster we take (for ex. 4) cluster heads, each of these CHs will perform their tasks in round robin fashion with fixed time slice i.e. at a time only one CH will be active. Now we divide our scenario into 3 phases:

A. Interaction Phase
B. Request Phase
C. Trust Evaluation Phase

A. *Interaction Phase*

In this phase we generate the interactions between the each cluster head and the member nodes and depending on the number of successful interactions and by considering some other communication parameters the member nodes will generate feedback values in the range from 1 to 10. These feedbacks or recommendations will be stored with the member nodes for each of the cluster head. The interaction phase can be visualized from fig 2.

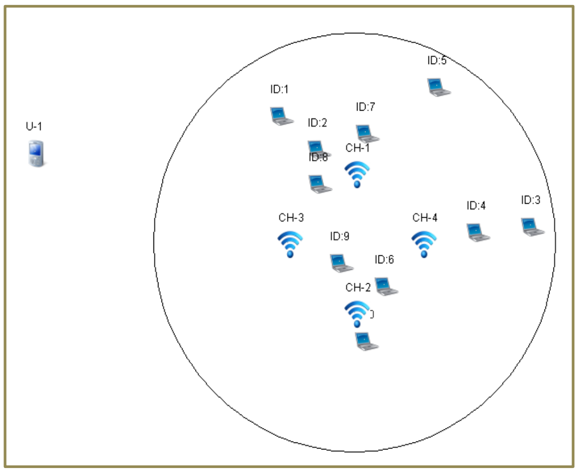

Fig. 1. MANET with Static Mobility Model

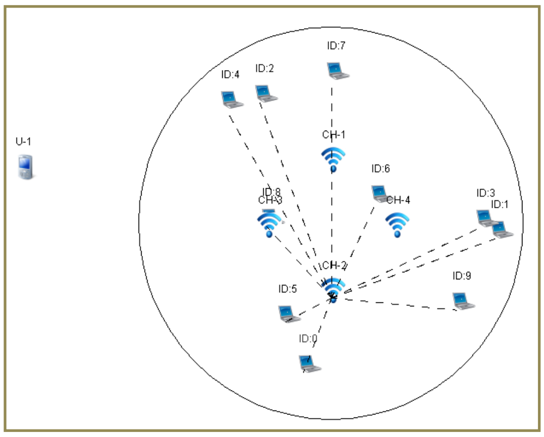

Fig. 2. Nodes interacting and generating recommendations for CH-2

B. *Request Phase*

Now let us assume that some user or node wants to access a secure connection with any of the cluster head which also acts as service provider for particular service such as gateway service, ftp server, etc. but the node is not sure

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

about the trustworthiness of the cluster heads. So when the node will enter the cluster it will request for trust ratings to all the cluster heads which then will serve this request during their respective active periods. We can visualize this from fig 3.

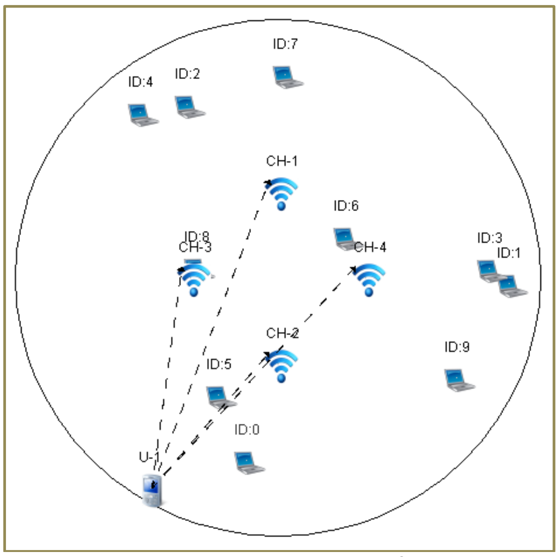

Fig. 3. Newly Entered Node requesting for Trust Value to all CHs.

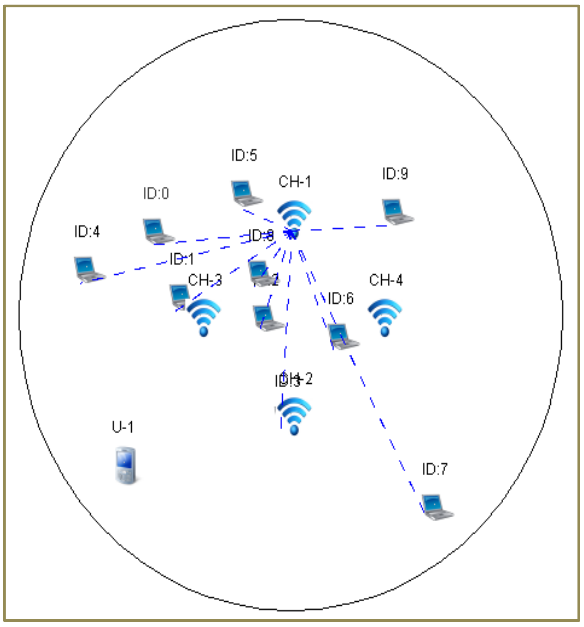

Fig. 4. CH-1 gathering recommendations from member nodes

C. Trust Evaluation Phase

After requesting the trust rating by the newly entered node, all the CHs will aggregate the recommendations from the member nodes in their respective active periods and will apply the indirect trust mechanism to evaluate the trust rating. The cluster head whose trust rating is greater than 0.5 will be treated as trusted but, if there are more than one trusted CHs then the CH with the highest trust rating will be chosen for establishing the secure connection. The node establishing secure connection with CH with highest trust rating can be seen in fig 6.

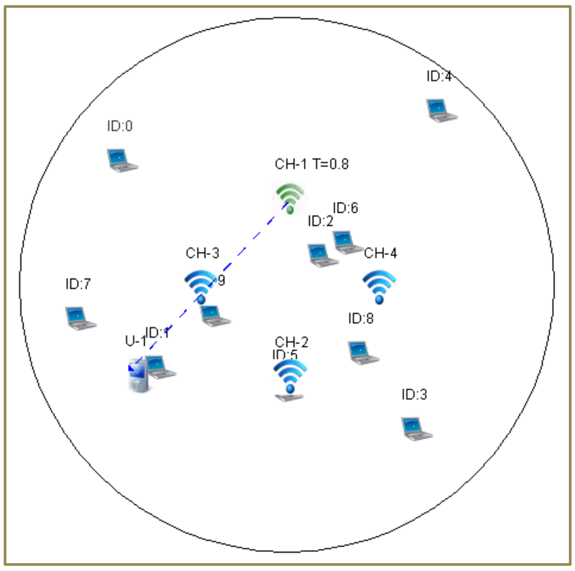

Fig. 5. Each CH reporting Trust Value to new node U-1

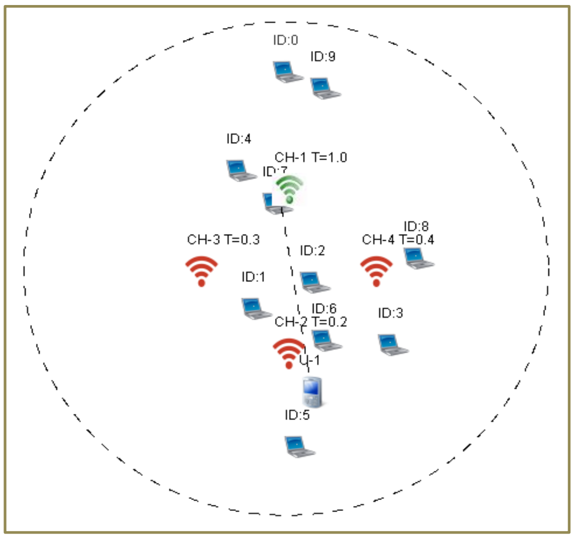

Fig. 6. New node U-1 establishes secure connection with CH having highest trust value

IV. WORK OF INDIRECT TRUST MECHANISM

The objective of indirect trust computation [19] is to determine the trustworthiness of an unfamiliar service provider from the set of recommendations that narrow the gap between the derived recommendation and the actual trustworthiness of the target service. In our approach, a dishonest recommendation is defined as an outlier that appears to be inconsistent with other recommendations and has a low probability that it originated from the same statistical distribution as the other recommendation in the data set. The importance of detecting outliers in data has been recognized in the fields of database and data mining for a long time. The outlier deviation-based approach was first proposed in [21], in which an exact exception problem was discussed. In [20], the author presented a new method for deviation-based outlier detection in a large database. The algorithm locates the outlier by a dynamic programming method. The approach (Algorithm 1) is based on the fact that if a recommendation is far from the median value of a given recommendation set and has a lower frequency of occurrence, it is filtered out as a dishonest recommendation. Suppose that an entity X requests to access service A. If service A has no previous interaction history with X, it will broadcast the request for recommendations, with respect to X. Let R denote the set of recommendations collected from recommenders.

$$R = \{r_1, r_2, r_3, \ldots\ldots, r_n\}$$

where 'n' is the total number of recommendations. Since smart attackers can give recommendations with little bias to go undetected, we divide the range of possible recommendation values into b intervals (or bins). These bins define which recommendations we consider to be similar to each other such that all recommendations that lie in the same bin are considered alike. b has an impact on the detection rate. If the bins are too wide, honest recommendations might get filtered out as dishonest. On the other hand, if the bins are too narrow, some dishonest recommendations may appear to be honest and vice versa. For this approach authors have tuned b = 10 such that Rc_1 comprises all recommendations that lie between interval [0 0.1], Rc_2 comprises all recommendations between interval [0.1 0.2], and so on for (Rc_3, . . . , Rc_{10}). After grouping the recommendations in their respective bins, we compute a histogram that shows count f_i of the recommendations falling in each bin. Let H be a histogram of a set of recommendation classes where

$$H(R) = \{\langle Rc_1, f_1\rangle, \langle Rc_2, f_2\rangle, \langle Rc_3, f_3\rangle, \langle Rc_4, f_4\rangle, \langle Rc_5, f_5\rangle, \\ \langle Rc_6, f_6\rangle, \langle Rc_7, f_{71}\rangle, \langle Rc_8, f_8\rangle, \langle Rc_9, f_9\rangle, \langle Rc_{10}, f_{10}\rangle\}$$

where f_i is the total number of recommendations falling in Rc_i. From this histogram H(R), we remove all the recommendation classes with zero frequencies and get the domain set (*R*domain) and frequency set (f)

$$Rdomain = \{Rc_1, Rc_2, Rc_3, \ldots\ldots, Rc_{10}\}$$

$$f = \{f_1, f_2, f_3, \ldots\ldots, f_{10}\}.$$

Definition 1. The dissimilarity function DF(x_i) is defined as

$$\text{DF}(x_i) = \frac{|x_i - \text{median}(x)|^2}{f_i} \qquad (1)$$

where x_i is a recommendation class from a recommendation set *x*.

Under this mechanism, the dissimilarity value of x_i is dependent on the square of absolute deviation from the median, i.e., $|\ x_i - \text{median}(x)\ |^2$. The median is used to detect deviation because it is resistant to outliers. The presence of outliers does not change the value of the median. In Equation 1, the square of absolute deviation from the median is taken to signify the impact of extremes, i.e., the farther the recommendation value x_i is from the median, the larger the squared deviation is. Moreover, the dissimilarity value of x_i is inversely proportional to its frequency. In Equation 1, $|\ x_i - \text{median}(x)\ |^2$ is divided by frequency f_i. In this way, if a recommendation is very far from the rest of the recommendations and its frequency of occurrence is also low, Equation 1 will return a high value. Similarly, if a recommendation is close to the rest of the recommendations (i.e., similar to each other) and its frequency of occurrence is also high, Equation 1 will return a low value.

```
Algorithm 1 Recommendation
Require: Set of Recommendations
Ensure: $Rdomain_{dishonest}$
1:  for $i = 1 \to 10$ do
2:      $Rc_i = i/10$
3:      $f_i$ = number of recommendations in interval $[i/10 - 0.1, i/10]$
4:  end for
5:  for $i = 1 \to 10$ do
6:      if $f_i <> 0$ then
7:          $Rdomain[k] = Rc_i$
8:          $H[k++] = \{Rc_i, f_i\}$
9:      end if
10: end for
11: $\overline{x} = \text{Median}(Rdomain)$
12: for each $k$ in $Rdomain$ do
13:     $\text{DF}[k] = \frac{|Rdomain[k] - \overline{x}|^2}{f_k}$    //calc deviation
14: end for
15: $\text{SRdomain} = \text{SortDesc}(Rdomain, \text{DF})$
16: $D_0 = \emptyset$
17: for $j = 1$ to size of (SRdomain) - 1 do
18:     $D_j \bigcup (\text{SRdomain}_j)$
19:     $\text{SF}_k = \text{SmoothingFactor}(D_j)$
20: end for
21: $\text{SF}_{max} = \max(\text{SF}(D_k))$
22: $f_{min}$ = min freq of $k$ in SRdomain with $\text{SF} = \text{SF}_{max}$
23: $Rdomain_{dishonest}$ = all $k$ in SRdomain with $\text{SF}_k = \text{SF}_{max}$ and $f_k = f_{min}$
24: return $Rdomain_{dishonest}$
```

For each Rc_i, a dissimilarity value is computed using Equation 1 to represent its dissimilarity from the rest of the recommendations with regard to their frequency of occurrence. All the recommendation classes in *R*domain are then sorted with respect to their dissimilarity value DF(Rc_i) in descending order. The recommendation class at the top of the sorted *R*domain with respect to its DF(x_j) is considered to be the most suspicious one to be filtered out as dishonest recommendation. Once the *R*domain is sorted, the next step is to determine the set of dishonest recommendation classes from *R*domain set. To help find the set of dishonest recommendation classes from the set of recommendations in *R*domain, Arning et al. [21] defined a measure called smoothing factor (SF).

Definition 2. A SF for each SRdomain is computed as

$$\text{SF}(\text{SRdomain}_j) = C(Rdomain - \text{SRdomain}_j) \times (\text{DF}(Rdomain) - \text{DF}(\text{SRdomain}_j)) \quad (2)$$

Where j = 1, 2, 3 . . . , m, and m is the total number of distinct elements in SRdomain. *C* is the cardinality function and is taken as the frequency of elements in a set {*R*domain − SRdomain_j}. The SF indicates how much the dissimilarity can be reduced by removing a suspicious set of recommendation (SRdomain) from the *R*domain.

Definition 3. The dishonest recommendation domain ($\text{Rdomain}_{\text{dishonest}}$) is a subset of Rdomain that contributes most to the dissimilarity of Rdomain and with the least number of recommendations, i.e., $\text{Rdomain}_{\text{dishonest}} \subseteq$ Rdomain. We say that SRdomain_x is a set of dishonest recommendation classes with respect to SRdomain, *C*, and DF (SRdomain_j) if

$$\text{SF}(\text{SRdomain}_x) \geq \text{SF}(\text{SRdomain}_j) \qquad x, j \in m$$

for all Rdomain, C, and SRdomain_j.

In order to find out the set of dishonest recommendation $\text{Rdomain}_{\text{dishonest}}$ from Rdomain, the mechanism defined by the proposed approach is as follows:

- Let R_{ck} be the kth recommendation class of Rdomain and SRdomain be the set of suspicious recommendation classes from Rdomain, i.e., SRdomain ⊆ Rdomain.
- Initially, SRdomain is an empty set, $\text{SRdomain}_0 = \{\}$
- Compute SF (SRdomain_k) for each SRdomain_k formed by taking the union of $\text{SRdomain}_k - 1$ and Rc_k

$$\text{SRdomain}_k = \text{SRdomain}_{k-1} \cup Rc_k \qquad (3)$$

where k = 1, 2, 3 . . . , m − 1, and m is the distinct recommendation class value number in sorted *R*domain.

- The subset SRdomain_k with the largest SF (SRdomain_k) is considered as a set containing dishonest recommendation classes.
- If two or more subsets in SRdomain_k have the largest SF, the one with minimum frequency is detected as the set containing dishonest recommendation classes.

After detecting the set $\text{Rdomain}_{\text{dishonest}}$, we remove all recommendations that fall under the dishonest recommendation classes.

An Illustrative Example

To illustrate how this deviation detection mechanism filters out unfair recommendations, this section provides an example that goes through each step of our proposed approach. Let X be a service requester who has no prior experience with service provider or CH. In order to determine the trustworthiness of CH, X will get registered with CH and will request for its trust value, CH in turn will request recommendations from its peer services who have previous interaction with X. Let R = {r1, r2, r3, ... , r_n } be a set of recommendations received by n = 10 recommenders for service requester R. After receiving the recommendations, they are grouped in their respective bins. Table 1 shows how the received recommendations are grouped in their respective classes. After arranging the recommendations in their respective recommendation class Rc_i, we remove the recommendation classes with zero frequencies and calculate DF (Rc_i) for each recommendation class using Equation 1. Table 2 shows the sorted list of recommendation classes with respect to their dissimilarity value.

In Table 2 the recommendation class Rc_6 has the highest deviation value, so it is taken as a suspicious recommendation class and is added to the suspicious recommendation domain (SRdomain), and its SF is calculated. Next we take the union of the suspicious recommendation domain SRdomain_1 and the next recommendation class in the sorted list, i.e., Rc_4 and calculate its SF using Equation 2. This process is repeated for each Rc_i of *R*domain until SRdomain = *R*domain − Rc_m, where m = 6.

Table 3 shows that the SF of SRdomain_2 has the highest value. Therefore, the recommendation classes {1.0, 0.8} in SRdomain_3 are considered as dishonest recommendation classes, and these recommendation classes are removed from the Rdomain.

TABLE I
FREQUENCY DISTRIBUTION OF RECOMMENTATION

Rc_i rc_i	Recommendation value	Frequency fi
Rc_1	0.1	2
Rc_2	0.2	1
Rc_3	0.3	0
Rc_4	0.4	3
Rc_5	0.5	0
Rc_6	0.6	2
Rc_7	0.7	0
Rc_8	0.8	1
Rc_9	0.9	0
Rc_{10}	1.0	1

TABLE II
RECOMMENDATION CLASSES SORTED WITH RESPECT TO THEIR DF

Rc_i	Recommendation value rc*i*	Frequency f_i	DF(Rc_i)
Rc_6	1.0	1	0.81
Rc_5	0.8	1	0.49
Rc_4	0.6	2	0.125
Rc_3	0.4	3	0.03
Rc_2	0.2	1	0.01
Rc_1	0.1	2	0

TABLE III
SMOOTHIN FACTOR COMPUTATION

SRdomain	Rdomain-SRdomain	DF(Rdomain-SRdomain)	SF
{1.0}	{0.8,0.6,0.4,0.2,0.1}	0.81	7.29
{1.0,0.8}	{0.6,0.4,0.2,0.1}	1.3	10.4
{1.0,0.8, 0.6}	{0.4,0.2,0.1}	1.425	8.55
{1.0,0.8, 0.6,0.4}	{0.2,0.1}	1.455	4.365
{1.0,0.8, 0.6,0.4,0.2}	{0.1}	1.465	2.93

V. PERFORMANCE EVALUATION

In this section, we evaluate our model in a simulated cluster based MANET environment. We carry out different sets of experiments to demonstrate the effectiveness of the proposed model against different attack scenarios (BM attack, BS attack, and RO attack). Results indicate that the model is able to respond to all three types of attack when the percentage of malicious recommenders is varied from 10% to 40%. We have also studied the performance of the model by varying the offset introduced by the malicious recommender in their recommended trust value. It was observed that the performance of the models decreases only when the percentage of malicious recommenders is above 30% and the mean offset between the honest and dishonest recommendation is minimum (0.2).

Experimental setup
We simulate a MANET environment using a Java based simulator, where nodes (offering and requesting services) are continuously joining and leaving the environment. The nodes are categorized into two groups, i.e., agents offering services as service provider nodes (SPN) and nodes consuming services as service requesting nodes (SRN). We conduct a series of experiments for a new SRN to evaluate the trustworthiness of an unknown SPN by requesting recommendation from other SPNs in the environment. All SPNs can also act as recommending agents (RA) for other SPNs.

The RA gives recommendations, in a continuous range [0 1], for a given SPN on the request of a SRN. The RA can either be honest or dishonest depending on the trustworthiness of its recommendation. An honest RA truthfully provides recommendation based on its personal experience, whereas a dishonest RA insinuates a true experience to a high, low, or erratic recommendation with a malicious intent. The environment is initialized with set numbers of honest and dishonest recommenders (N = 10 to 100).

A. Experiment 1: validation against attacks
To analyse the effectiveness of the proposed approach, three inherent attack scenarios (bad mouthing, ballot stuffing, and random opinion attack) for recommendation models have been implemented in the above defined simulation environment.

Bad mouthing attack
BM is one in which the intention of the attacker is to send malicious recommendations that will cause the evaluated trustworthiness of an entity to decrease. Let us suppose that the service provider asks for recommendations regarding an unknown service provider node CH-1. In this experiment we assume that a certain percentage of the recommenders are dishonest and launch a BM attack against (CH-1) by giving dishonest recommendations. It is assumed that the actual trust value of CH-1 is 0.7.

At the initial step of the simulation, the environment has 10% dishonest RA who attempt to launch a bad mouthing attack against A by providing low recommended trust values (between the range [0 0.3]). To elaborate the efficacy of the proposed approach, we vary the percentage of dishonest recommenders from 10% to 40%. Figure 1 a, b, c, d shows the SF calculated for each SRdomain.

It is shown that in each case the proposed approach is able to detect the set of bad mouthers giving low recommendation between 0.1 and 0.3. For example, when the percentage of dishonest recommenders is 10%, the SRdomains and respective SF values are as follows:

SRdomain 1 $\{0.1\}$,	8.64
SRdomain 2 $\{0.1, 0.2, 0.3, 0.8\}$,	13.62
SRdomain 3 $\{0.1, 0.2\}$,	16.12
SRdomain 4 $\{0.1, 0.2, 0.3, 0.8, 0.6\}$,	6.82
SRdomain 5 $\{0.1, 0.2, 0.3\}$,	20.4

Since the SF of SRdomain5 has the highest value, the recommendation classes {0.1, 0.2, 0.3} are considered as dishonest recommendation classes, and the recommendations that belong to these recommendation classes are considered as dishonest recommendations.

Ballot stuffing attack
BS is one in which the intention of the attacker is to send malicious recommendations that will cause the evaluated trustworthiness of an entity to increase. Let us suppose that the service provider asks for recommendations regarding an unknown service requester B. It is assumed that the actual trust value of B is 0.3. A certain percentage of recommenders providing the recommendation to the service provider are dishonest and gives a high recommendation value between 0.8 and 1.0, thus launching a BS attack. We evaluate the proposed approach by varying the percentage of dishonest recommenders from 10% to 40%. It is evident from the results that the model is able to detect dishonest recommendations even when the percentage of dishonest recommendations is 40%. When the percentage of dishonest recommendations is 40%), the SF values of each SRdomain are as follows:

SRdomain 1 $\{1.0, 0.9\}$,	5.038
SRdomain 2 $\{1.0, 0.9, 0.8, 0.1, 0.2\}$,	1.843
SRdomain 3 $\{1.0, 0.9, 0.8\}$,	5.47
SRdomain 4 $\{1.0\}$,	4.41
SRdomain 5 $\{1.0, 0.9, 0.8, 0.1\}$,	3.667

The proposed approach is able to detect the dishonest recommendations as SRdomain 3 with the highest SF value of 5.47.

Random opinion attack
RO attack is one in which the malicious recommender gives the recommendations randomly opposite the true behaviour of the entity in question. Let us suppose that the recommenders launch a RO attack while providing recommendations for a service provider node CH-1. The dishonest recommenders provide either very low recommendations (0.1 to 0.2) or very high recommendations (0.8 to 1.0). We vary the percentage of dishonest recommenders from 10% to 40% for the experiment.The proposed approach successfully detects random opinion attack and is able to filter out the dishonest set of recommenders in each case.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

B. Experiment 2: validation against deviation

The detection rate of unfair recommendations by varying the number of malicious recommenders cannot fully describe the performance of the model as the damage caused by different malicious recommenders can be very different depending on the disparity between the true recommendation and unfair recommendation (i.e., offset). The offset introduced by the attackers in the recommended trust value is a key factor in instilling deviation in the evaluated trust value of SPN. We have carried out a set of experiments to observe the impact of different offset values introduced by different malicious recommenders on the final trust value. We define mean offset (MO) as the difference between the mean of honest recommendations and the mean of dishonest recommendations. For the experiment, we have divided MO into four different levels L1 = 0.2, L2 = 0.4, L3 = 0.6, and L4 = 0.8. It is assumed that the actual trust value of SPN is 0.2, and the dishonest recommender's goal is to boost the recommended trust value of SPN (BS attack). The experiment was conducted in four different rounds by varying the MO level from L4 to L1 (i.e., from maximum to minimum). In each round, the recommended trust value is computed with different percentages of dishonest recommenders (10%, 20%, 30%, and 40%).

C. Comparison with existing approaches

To illustrate the effectiveness of the proposed deviation-based approach in detecting dishonest recommendations, we have compared our approach with other approaches proposed in the literature based on quartile [15], control limit chart [17], and iterative filtering [18] to detect dishonest recommendations in indirect trust computation. A set of experiments has been carried out by applying the approaches to detect dishonest recommendations in two different scenarios. For the first set of experiments, we assume that a certain percentage of the recommenders are dishonest and launch bad mouthing attack by giving recommendations between 0.1 to 0.3. For the second set of experiments, the dishonest recommenders are assumed to give a high recommendation value between 0.8 to 1.0, thus launching a ballot stuffing attack. In both set of experiments, the percentage of dishonest recommenders is varied from 10% to 45%. For comparison, we have used Matthews correlation coefficients (MCC) to measure the accuracy of all four approaches in detecting dishonest recommendations [22]. MCC is defined as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives. The formula used for MCC calculation is

$$\mathrm{MCC} = \frac{(\mathrm{TP} \times \mathrm{TN}) - (\mathrm{FP} \times \mathrm{FN})}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. MCC returns a value between − 1 and 1 (1 means perfect filtering, 0 indicates no better than random filtering, and − 1 represents total inverse filtering). To avoid infinite results while calculating MCC, it is assumed that if any of the four sums (TP, FP, TN, andFN) in the denominator is zero, the denominator is arbitrarily set to one.

VI. CONCLUSION

An application of indirect trust mechanism in MANET environment is proposed in the above work. The main focus in this present work was to detect dishonest recommendations based on their dissimilarity value from the complete recommendation set. Since median is resistant to outlier, we have proposed a dissimilarity function that captures how dissimilar a recommendation class is from the median of the recommendation set. The algorithm uses a smoothing factor which detects malicious recommendations by evaluating the impact on the dissimilarity metric by removing a subset of recommendation classes from the set of recommendations. Experimental evaluation shows the effectiveness of our proposed method in filtering dishonest recommendations in comparison with the base model. Results show that the proposed method is successfully able to detect dishonest recommendations by utilizing absolute deviation from the median as compared to the base technique which tends to fail as the percentage of dishonest recommendations increases. We have carried out a detailed comparative analysis with the base approach by varying the percentage and the offset introduced by the dishonest recommendations. Results that indicate improved performance of the proposed approach, which is able to produce 70% detection rate at a minimum offset of 0.2, have been shown. On the contrary, the base approach is unable to detect any dishonest recommendations at all. It is also shown that for different attacks (bad mouthing, ballot stuffing, and random opinion attack), the proposed method successfully filters out dishonest recommendations. A comparison between existing approaches and the proposed approach is also presented, which clearly shows the better performance of the proposed approach. In our future work, we will try to incorporate fuzzy or rule based engine to evaluate the trust value from the recommendation sets.

REFERENCES

[1] A Josang, R Ismail, C Boyd, "A survey of trust and reputation systems for online service provision," in Decis. Support Syst. 43(2), 618–644 (2007).

[2] [2] L Xiong, L Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," in IEEE Trans. Knowl. Data Engr. 16(7), 843–857 (2004)

[3] M Chen, JP Singh, "Computing and using reputations for internet ratings," in 3rd ACM Conference on Electronic Commerce (ACM, New York, 2001), pp. 154–162

[4] Z Malik, A Bouguettaya," Evaluating rater credibility for reputation assessment of web services," in 8th International Conference on Web Information Systems Engineering (Springer, Heidelberg, 2007), pp. 38–49

[5] S Ganeriwal, LK Balzano, MB Srivastava, "Reputation-based framework for high integrity sensor networks," in ACM Trans. Sensor Netw. 4, 1–37 (2008)14.

[6] R Zhou, K Hwang, "Powertrust: a robust and scalable reputation system for trusted peer-to-peer computing," in IEEE Trans. Parallel Distributed Syst. 18(4), 460–473 (2007)

[7] X Liu, A Datta, H Fang, J Zhang, "Detecting imprudence of reliable sellers in online auction sites," in IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (IEEE, Los Alamitos, 2012), pp. 246–253

[8] C Ziegler, J Golbeck," Investigating correlations of trust and interest similarity–do birds of a feather really flock together?," J. Artif. Intell. Res. (2005). doi:10.1.1.79.7225

[9] I Varlamis, M Eirinaki, M Louta, "A study on social network metrics and their application in trust networks," in 2010

International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (IEEE, Los Alamitos, 2010), pp. 168–175

[10] E Davoodi, M Afsharchi, K Kianmehr,"A social network-based approach to expert recommendation system," in Hybrid Artificial Systems. 7th International Conference on Hybrid Artificial Intelligent Systems, Salamanca, March 2012 (Springer, Heidelberg, 2012), pp. 91–102

[11] H Ma, I King, M Lyu, "Learning to recommend with social trust ensemble," in 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM, New York, 2006), pp. 203–210

[12] F Almenarez, A Marin, D Diaz, "A Cortes, C Campo, C Garcia, Managing ad-hoc trust relationships in pervasive computing environments, in Proceedings of the Workshop on Security and Privacy in Pervasive Computing," SPPC'04, Vienna, 20 April 2004

[13] C Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in 2nd ACM Conference on Electronic Commerce (ACM, New York, 2000), pp.150–157

[14] S Liu, J Zhang, C Miao, Y Theng, "A Kot, An integrated clustering-based approach to filtering unfair multi-nominal testimonies.," Comput. Intell. (2012). doi:10.1111/j.1467-8640.2012.00464.x

[15] A Whitby, A Josang, J Indulska, "Filtering out unfair ratings in Bayesian reputation systems," in 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems (IEEE, Washington, 2005), pp. 106– 117

[16] J Weng, C Miao, "A Goh, An entropy-based approach to protecting rating systems from unfair testimonies," IEICE Trans. Inf. Syst. 89(9), 2502–2511 (2006)

[17] SI Ahamed, M Haque, M Endadul, F Rahman, N Talukder, "Design, analysis, and deployment of omnipresent formal trust model (FTM) with trust bootstrapping for pervasive environments," J. Syst. Software 83(2), 253–270 (2010)

[18] MK Deno, T Sun, I Woungang, "Trust management in ubiquitous computing: a Bayesian approach," Comput. Commun. 34(3), 398–406 (2011)

[19] Naima Iltaf, Abdul Ghafoor and Uzman Zia, "A mechanism for detecting dishonest recommendation in indirect trust computation," in EURASIP Journal on Wireless Communications and Networking 2013

[20] Z Zhang, X Feng, "New methods for deviation-based outlier detection in large database," in Sixth International Conference on Fuzzy Systems and Knowledge Discovery (IEEE, Los Alamitos, 2009), pp. 495–499

[21] A Arning, R Agrawal, P Raghavan, "A linear method for deviation detection in large databases," in 2nd International Conference on Data Mining and Knowledge Discovery (AAAI, Portland, 1996), pp. 164–169

[22] BW Matthews,"Comparison of the predicted and observed secondary structure of T4 phage lysozyme," Biochim. Biophys. Acta 405, 442–451 (1975)