

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/302190343>

FuNN –an Interactive Tool to Detect Sybil Attack in MANET

Conference Paper · April 2016

CITATIONS

0

READS

219

2 authors, including:



Aditi Paul

Dronacharya College of Engineering

5 PUBLICATIONS 4 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



6. Rupinder Kaur, Navjot Singh, Dr Aditi Paul, “REVIVER - an Android Application to solve the issue of ‘Women Safety’”, NACORE16, January 17, 2016. [View project](#)

All content following this page was uploaded by [Aditi Paul](#) on 08 May 2016.

The user has requested enhancement of the downloaded file.

FuNN - an Interactive Tool to Detect Sybil Attack in MANET

SOMNATH SINHA AND ADITI PAUL

Faculty of Computer Science

Pacific Academy of Higher Education & Research University, Udaipur, India

Email: {ssin.mca; aditi23.mca}@gmail.com

Abstract: Detection of Sybil attack in mobile ad hoc network (MANET) has been a challenging issue in the context of network scalability, limited resource and complexity of the proposed methods. Literature review shows that most of the detection algorithms suffer from the above constraints and could not exhibit their proper efficiency and performance. This paper introduces a new Sybil detection method which utilizes network scalability and shows its efficiency within the available resource. In the proposed method fuzzy inference rule is used as tool to initially isolate those nodes whose behaviors do not conform with the genuine nodes. At the later stage we employ a trained Artificial Neural Network (ANN) to find out the Sybil nodes from the suspected nodes. The use of Fuzzy inference rule helps to avoid complex mathematical computations as this rule uses simple if...then clause based on nodes' attributes which can be easily extracted from a real network. On the other hand, the performance of ANN does not get affected in a scalable network since its learning efficiency increases with larger data set. The proposed algorithm does not need any extra hardware like antenna or receiver which may reduce the battery backup. The advantage of this technique is that, it can find out any number of Sybil nodes at one go and also minimize the chances of false positive. We have evaluated our scheme by using simulation and result shows a satisfactory detection rate with few false positive.

Keywords: Sybil attack, MANET, ANN, fuzzy inference rule, NS2

1. INTRODUCTION

Mobile ad hoc network (MANET) is an emerging field of wireless evaluation. Unlike wired network, MANETs are infrastructure-less which incorporates some limitations in their characteristics such as frequently changing topology [1], limited bandwidth and battery power etc. These types of networks are aimed to be implemented in situations such as rescue operations, battle field and emergency circumstances where there is no possibility of establishing traditional wired links. Such applications require openness and flexibility of MANET to cope up with the hostile environment. However, these characteristics cause MANET vulnerable to a wide range of security attacks. For example, an intruder can easily come into communication through radio links and breaks the authentication or a remote entity can malfunction by hampering the entity-identity mapping. Identity spoofing, eavesdropping, corrupting messages etc are widespread security attacks in MANET. Cryptographic authentication techniques can mitigate these types of attacks. Some attacks directly affect the routing protocol by dropping data packets or tunneling them to other locations. There are more severe attacks where some malicious nodes create illegitimate identities (Sybil) [2] either by stealing them or fabricating new ones which do not have real existence. These identities are then used to communicate with the legitimate nodes inside the network. These types of attacks rigorously disrupt the network performance by manipulating the routing table, corrupting hello packets and so on. In this paper, we have concentrated our motivation towards Sybil attack which is one of the most

powerful routing attacks. This attack can be launched externally or internally. In external attack, the malicious node comes into the radio range of the legitimate nodes and enters into it whereas in internal attacks the malicious node creates many Sybil identities either by compromising the existing true nodes or by generating arbitrarily new identities. External attacks can be prevented by authentication mechanism but it cannot mitigate internal attacks.

In Sybil attack a Sybil node can communicate directly with the legitimate node or becomes a third party between two legitimate nodes. They can be either stolen or fabricated identity and use them simultaneously or non-simultaneously. In simultaneous attack the new identity is replaced by the previous one thus only one identity is active at a time. This is called whitewashing of bad history. In the second type of Sybil attack, an attacker simultaneously uses all its identities for an attack. This type of attack causes interruption in the network. Whatever may be the dimension, a Sybil attacker enters into a system by using these fake identities and builds up basis for more severe attacks in order to disrupt the targeted system. Sybil node exploits the routing protocol and consumes intercepted packets to replay other attacks such as wormhole and black hole attack.

Since, Sybil attacks have a serious impact on the wireless ad hoc networks its detection becomes inevitable. It is not always desirable to apply authentication [3] because of its infrastructural, computational, and management overhead. Furthermore, cryptographic [3] methods are susceptible to node compromise, which is a serious concern, because most wireless nodes are easily accessible, allowing their memory to be easily

scanned. On the other hand, Received signal strength (RSS) [4] based localization techniques are said to have the potential to detect and localize Sybil nodes efficiently, but these require extra hardware set up such as transmitter, landmark detector, centralized server etc. Moreover localization techniques mostly involve complex mathematical calculation and large amount of statistical data. .

In the current study our aim is to introduce a Sybil detection technique which is easy to derive and does not require high set up cost. We name the detection technique as FuNN (Fuzzy Neural Network), since it is based on the concept of fuzzy inference rule and ANN. The motivation behind FuNN is to use simple tools that can be easily applied in a MANET keeping in mind the inherent constraints associated with it. Firstly, we have used the fuzzy inference rule to initially differentiate between the suspected nodes and the legitimate nodes in the network. For this we consider packet drop of individual node and calculate their deviation from the normal values during attack. These deviations are graded using fuzzy logic and then Mamdani inference rule is applied to categorize them as trust, distrust and enemy nodes. At the second stage we use a trained artificial neural network (ANN) that finally sorts out the Sybil nodes from the distrust and enemy nodes. The detection scheme is tested in simulator (NS2.35) and result shows a detection rates up to 90% with maximum of 10% false positive. We show graphically that, this approach not only traces the Sybil nodes with higher accuracy but also minimizes the chances of false positive. Our detection scheme neither use any localization method that requires any extra hardware [4] nor use any central authority (CA) [3], which incorporates high costing and maintenance hazards in a scalable network. One important aspect of the proposed scheme is that it performs well in a large scale network since ANN is itself a useful tool where large amount of data are available. Initially it seems to be difficult to train the ANN with a large volume of data set but once it is well feed, it can perform with higher accuracy. Moreover, the use of fuzzy logic in the first stage isolates the true nodes from distrust and enemy nodes and hence makes the second stage easier to execute. . In the current study, we make the following contributions:

- We design a Sybil attack model for MANET to show the impact of the Sybil attack on the network. For this we choose AODV routing protocol and consider its performance metrics such as network throughput, packet delivery ratio, average end-to-end delay and percentage packet loss. We show the variation of these parameters due to the

attack. We also study the variation of network performance when the number of malicious nodes and Sybil nodes increase. The result of all theses variations are depicted graphically which will help the readers to easily understand the behavior of the proposed attack model.

- We design an algorithm for FuNN which works in two stages a) fuzzy inference rule b) ANN which is discussed in the consecutive section.
- We have tested the efficiency of FuNN by applying it on the attack model in NS2.35 and have shown the result graphically.
- We analyze the result and its accuracy level and finally discuss the future scope.

The rest of the paper is organized as follows.

Section 2 represents related work in the field of Sybil attack detection mechanisms. Section 3 explains the attack model while section 4 describes the working of FuNN. Section 5 shows the simulation results. Section 6 analyses detection rate and the possibility of false positive. Section 7 concludes the paper.

2. RELATED WORKS

In the literature of Sybil detection mechanisms a considerable number of techniques have been proposed among which trusted certification is said to be one of the most prospective solution to prevent Sybil attack. Douceur [2] has proved that trusted certification is the only approach that has the potential to eliminate Sybil attack completely. However, there are a numbers of issues in this method related to implantation of certification authority as well as implementation of entity-identity mapping. Significant overhead and cost also restrict the use of this method in a dense network. Authors proposed a RSS based technique [4] that can perform attack detection and also localizes adversaries' positions. But this technique requires extra hardware set up and statistical calculation. Demirbas and Song [5] proposed a method that uses Received Signal Strength Indicator (RSSI) of messages. Upon receiving a message the receiver will associate the RSSI of the message with the sender identity, and later when another message with same RSSI but from a different sender is received, the receiver detect Sybil node. According to this method Sybil attacks can be detected with a100% true positive with a few positive alerts. However, a Sybil node can transmit message with different identities using different transmission power intensity to defeat this scheme.

Several designs of trust model are proposed in social networks by using different form of modified random walk [6]. However surveys the social network based Sybil defenses and states that Sybil defense methods can perform poorly when confronted with some real-world attacks that exhibit a very primitive structure [7]. They identified two main trends of Sybil defense in social networks. The first is based on random walk methods while the second considers community detection. This paper has also showed how the two approaches can go hand in hand to yield more robust Sybil defense protocols that are competitive with the state of the art. Authors proposed a fuzzy multi-agent security system for WSN [8] which can differentiate agents that can be trusted from those that cannot be trusted on the basis of fuzzy negotiations among agents present in the network. This system is able to recognize different types of attacks such as worm hole, grey hole and Sybil attack. In this system fuzzy logic is used to assign trust values for nodes and every node in the network is considered as agent. Another Sybil detection scheme [9] is anticipated in reputation based system that uses a non-monetary entry fee (i.e. fee is used as a form of work imposed on every newcomer) per identity to discourage Sybil attackers without using any costly method. This scheme performed better than CONFIDANT protocol in diving evil throughput and evil nodes' utility in the presence of whitewashing nodes. However, the drawback of this approach is that newcomers are not welcomed due to the free identities available in the network. A review of intrusion detection and protection mechanisms [10] shows that intruders often find new ways of attack and cause damage to computer systems and networks. According to this paper protection mechanism should learn from experience and use the existing knowledge of attacks to infer and detect new intrusive activities. Since the attacker may try to attack an existing protection scheme the protection mechanisms need to be robust enough to protect them and not to introduce new vulnerabilities into the system. Surveys of different trust management schemes in MANETs [11] showed they are much more challenging than the traditional

centralized environments due to changes in topology of MANETs. Researcher also represents [12] a Sybil tracking procedure which detect and isolates Sybil node in a p2p network. It uses the concepts of monitoring peers in the network to detect Sybil. Haribabu K et al. proposed [13] a Sybil detection that uses CAPTCHA and ANN in a peer-to-peer system where the neural network is trained by Sybil characteristics. In practical applications it is difficult to fabricate Sybil characteristics in a system. Moreover CAPTCHA behaves as an authentication mechanism which again incurs high overhead.

3. ATTACK MODEL

In this section we design an attack model of simultaneous Sybil attack where an attacker uses all its identities at the same time. We consider that the attacker spoofs the identity of a legitimate node. This type of attack is called masquerading attack where Sybil nodes compromise existing legitimate entities and use their identities (stolen). Thus, the proposed scheme considers simultaneous Sybil attack with stolen identities. The communication between the nodes is performed using standard AODV protocol in which the source node broadcasts request messages to its neighbors for finding paths to the destination. Sybil attacker provides wrong routing information to the source and makes the data traffic pass through it. While data packet is passed through the Sybil node, it also forwards them to the attacker. The attacker consumes all the data packets passes through it. The attacker and the compromise node vary their normal and Sybil behaviour periodically. We have deployed this malicious behavior in the attack model (fig. 1). Here node 0 and 33 are the attackers and node 1 and 38 are Sybil identities. Source node 6 broadcasts route request (RREQ) to find a route to destination node 5. Since our algorithm works for MANET we consider that node 6(source) and node 5(sink) are mobile. This assumption makes other nodes in the network relatively mobile with respect to the source and sink.

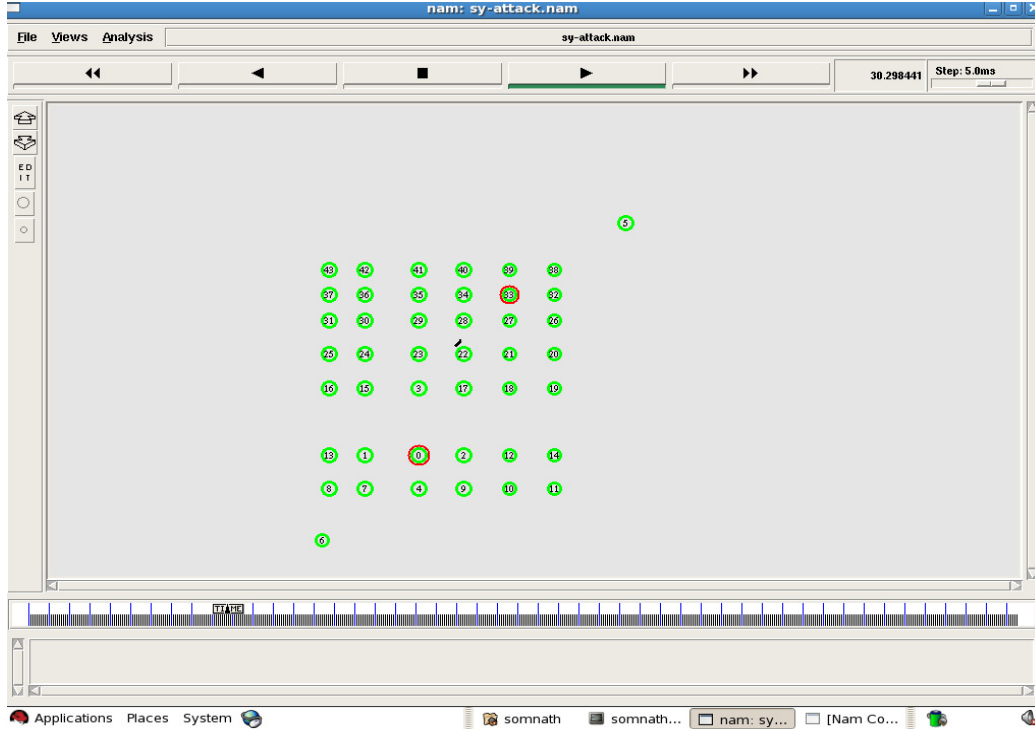


Fig.1 Topology of Attack Model

Fig.1 shows a MANET of 44 nodes in an area of 500x500 sq meters among which node 0 and node 33 are made Sybil attackers. The simulation parameters are shown in table 1. Total simulation time is 150s. Attack starts after 30 sec. Node 1 sends wrong routing information to node 6 by representing it as node 0 over the time interval of 20 sec. and increases its sequence number higher than the most recent value. Thus node 6 sends data packet to node 1 who

forwards incoming packets to node 0. Node 0 consumes these packets when they reach to it. The same thing happens for node 33 and node 38. In the next interval of 20 sec. the nodes become legitimates. We consider two-ray ground propagation model for communication between the nodes. The speed of the sink node is given 15m/s and that of source is 1m/s. Initial energy of all the nodes is set as 100 joules and transmission power 1.8 w.

Table 1.

Parameter	Level
Propagation Model	TwoRayGround
Transmission power	1.8w
Frequency	2.472×10^9 Hz
Initial energy	100 J
Collision threshold	100 dB
Carrier sense threshold	5.011872×10^{-12} w
Receive power threshold	5.82587×10^{-9} w
Idle Power	712×10^{-6} w
RxPower	35.28×10^{-3} w
TxPower	31.23×10^{-3} w
SleepPower	144×10^{-9} w
Number of Nodes	44
Protocol	AODV
MAC	802_11
Maximum packet in ifq	50
Topology	Flat Grid
Area covered	(500x500) sq.m.
Node movement (sink)	at 50 towards position 25, 20 at 100 towards position 490,480
Node movement (source)	at 10.0 towards position 20, 18

Simulation time	150s
Speed of the sink node	15m/s
Speed of the source node	1m/s
Starting time of attacker	30.0s
Attacker vary id in each	20.0s

The impact of the attack is shown in two dimensions which are:

- I. Performance of the network with time.
- II. Performance of the network with number of malicious attacker.

We consider some of the performance metrics of AODV protocol which are network throughput, packet delivery ratio (PDR), average end-to-end delay and percentage packet drop in order to show their variations due to attack. These are shown graphically in the following figures (from fig.2 to fig. 9).

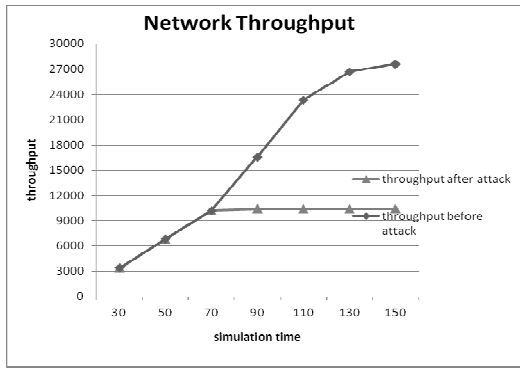


Fig.2. Variation of network throughput (kbps) with time (s) before and after attack.

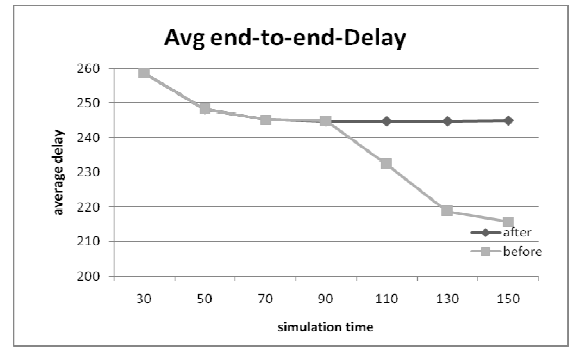


Fig.4. Variation of average end-to-end delay with time (s) before and after attack

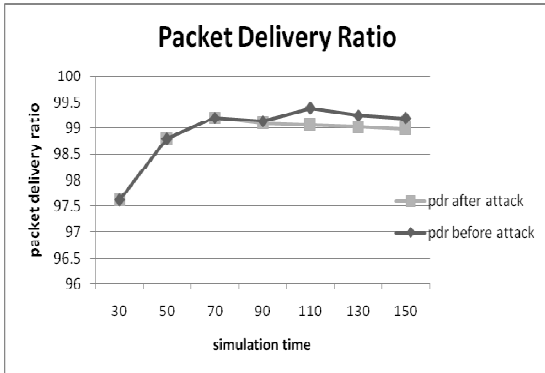


Fig.3. Variation of packet delivery ratio with Time (s) before and after Attack.

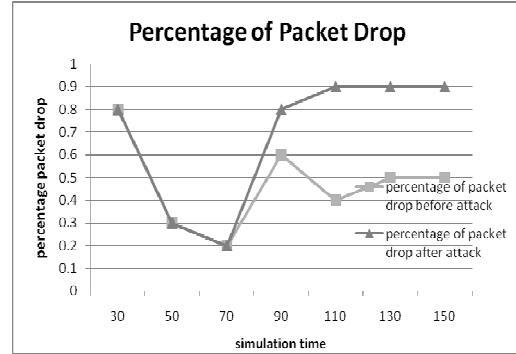


Fig.5. Variation of percentage packet drop with time (s) before and after attack.

We see (fig.2) that the network throughput decreases drastically after attack. This is because after 70 s the Sybil nodes start consuming packets. This causes less number of data packets reaching destination leading to abrupt fall in throughput. The same phenomenon happens for Packet Delivery Ratio (Fig.3) which decreases after 90s. Sybil nodes forward data packets

towards the attackers who consume them rather than delivering towards destination. This causes a fall in Packet Delivery Ratio.

We see that average end-to-end delay (fig.4) remains almost same (from 90 s onwards) after attack whereas it has a sharp decrease with time before attack. This is due to the fact that Sybil nodes attract

most of the data traffic towards the same route created by them which causes no such change in delay. Whereas in normal condition the delay decreases with time as alternate routes are found towards destination. The percentage of packet drop increases (fig.5) after attack, which is quite obvious as the Sybil attacker consumes data packet on its way

which causes network congestion and leads to more number of packets drop.

In the next dimension we show how the malicious attackers affect network performance. The figures below (fig.6 to fig.9) represent the variation of the same performance metrics with number of attackers.

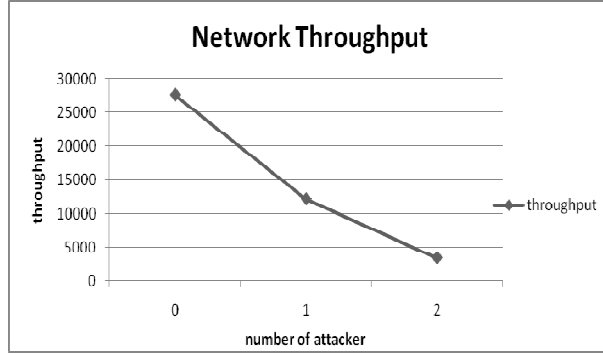


Fig.6. Variation of network throughput (kbps) with number of attackers.

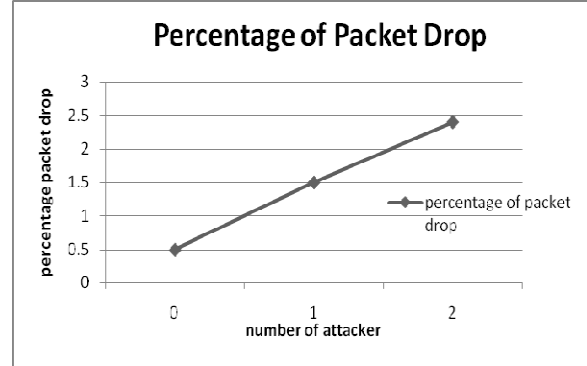


Fig.8. Variation of percentage packet drop with number of attackers.

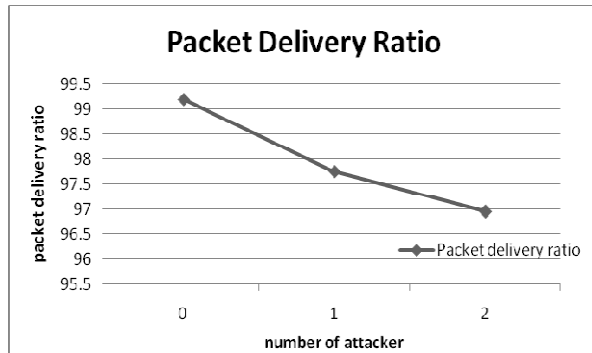


Fig. 7. Variation of percentage packet drop with number of attackers.

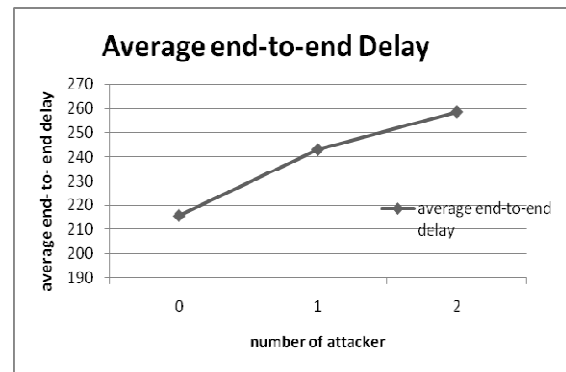


Fig.9. Variation of average end-to-end delay with number of attackers.

Here 0 implies no attack. We see that both network throughput and packet delivery ratio decrease as the attacker increase whereas average end-to-end delay and percentage packet drop increase. This conforms

to the previous result discussed. In the next section we will formulate the proposed algorithm for FuNN.

4. THEORETICAL MODEL FORMULATION

In FuNN we put together fuzzy inference rule and artificial neural network (ANN) for the purpose of double filtering. The contributions of each of them are discussed in the following subsections:

4.1. Application of fuzzy inference rule:

In our proposed system fuzzy logic is used to assign certain trust values to all the nodes in order to judge their behaviours during attack. The idea behind this is to identify those nodes whose behaviours are different from the behaviour of the true nodes. In fundamental crisp logic, an entity or node is treated as either Sybil or not. This overlooks the possibilities of the nodes whose behaviours lie between these two. In FuNN we address this issue by using fuzzy logic. Fuzzy logic sets up qualitative parameters by using continuous logic. In fuzzy logic value of a variable is not restricted to 1(Sybil) and 0(legitimate) as in crisp logic; rather it considers any value between zero and one. Here the probability of an event to be occurred can be defined

by a range of values which is not possible in crisp logic. Initially, we set a standard based on which we can identify the malicious behaviour (if any) of the nodes. For this we observe nodes' behaviour during communication (attack) and estimate how much they belong to Sybil nature. As (according to the graphs in section 3,) nodes' behaviour change if Sybil attack occurs in a system, we measure this change and define its intensity in three distinct categories. This is achieved by calculating the deviation of their attributes values (during attack) from the values in normal operation (no attack). In the proposed model these deviations are ranged as 0-0.25, 0.25-0.5, 0.5-1.0 and each of these deviation ranges is assigned a fuzzy linguistic value (table 2.). Then we employ fuzzy inference rule which is a simple rule base system that uses the fuzzy linguistic values and infers 'IF THEN' rules. One important aspect of the proposed algorithm is that, here we are not keen to find out whether a node is Sybil or not, rather we are evaluating the possibility a node being Sybil. Once this is done, we have three sets of nodes which are trust nodes, distrust nodes and enemy nodes having separate deviation range.

Table 2. Definition of fuzzy linguistic value

Deviation value	Linguistic value
0-0.25	Low
0.25-0.75	Medium
0.75-1	High

In table 2 low, moderate and high are fuzzy linguistic values which determine deviation of the nodes' attributes from their normal value (Table 2). In Table 3

we show the fuzzy linguistic variables and their linguistic values that are used to make inference rules.

Table 3. Defining fuzzy linguistic variable

Fuzzy linguistic variable	Linguistic value
Deviation	Low, medium, high
Node	Trust, distrust, enemy

Table 3 is used to formulate fuzzy inference rules as below:

- R1: If *deviation* is *low*
Then *node* is *trust*.
R2: If *deviation* is *medium*

- Then *node* is *distrust*.
R3: If *deviation* is *high*
Then *node* is *enemy*

These rules are used to categorize all the nodes in the network into trust, distrust and enemy levels. At the

second stage we separate trust nodes from distrust and enemy nodes and apply ANN to finally trace out the Sybil nodes from distrust and enemy ones.

4.2. Application of Neural network

The use of artificial neural network (ANN) in Sybil detection is a very limited literature. In [13] ANN is used to detect Sybil nodes in a peer-to-peer system where the neural network is trained by Sybil characteristics. However, it is difficult to fabricate Sybil characteristics in a system. Moreover the use of CAPTCHA as authentication mechanism incurs high overhead and hence does not fulfil our aim. In FuNN we eliminated these complications and train the ANN with true nodes' attributes. We designed a three layered feed-forward neural network with one input layer, one hidden layer and one output layer. Number of inputs taken is 5 which are:

- Packet drop
- Packet forward
- Received request/reply
- Sent request/reply packet
- Residual Energy

In order to recognize a Sybil node, the ANN was trained with the input pattern under normal condition i.e. when there is no attack and the targeted output was set as 0 (i.e. the probability of a node being Sybil is

0). The input pattern was first normalized by using min-max normalization into a specified range from 0.0 to 1.0. These values are then passed as input to the ANN and weights are adjusted accordingly by using back propagation algorithm. This algorithm is best suited for our detection method since it uses error-correction learning, where the desired output for the system must be known. In our system the targeted output i.e. the probability of a node being Sybil is taken as 0. The instantaneous error is calculated from the difference between calculated output and the desired output for a given input pattern. Each weight in the network is adjusted by correcting the present value of the weight with a term that is proportional to the present input and error at the weight. This process is repeated until the error minimizes to .003 or below. The learning rate and momentum constant are set to .8 and .2 respectively. Sigmoid function is used for calculating activations as because (unlike other activation functions) it has the effect of compressing the infinite range of *inputs* into the range 0 to 1 at output. The sigmoid function can be represented by the following equation

$$\text{Output} = \frac{1}{1+e^{-\text{input}}}$$

In the following subsection we give a pseudo code of the modified back propagation algorithm used for training and Sybil detection.

4.3. Neural Network Model (pseudo code for modified back propagation)

```

/*Variables for Error Calculations and Weight Adjustments */

Input to the input layer Ii[ ];
Input to the hidden layer Ih[ ];
Input to the output layer Io[ ];
Output of the input layer Oi[ ];
Output of the hidden layer Oh[ ];
Output of the output layer Oo[ ];
Input-hidden weights v[ ][ ];
Hidden-output weights w[ ][ ];

/*Calculation of input to hidden layer Ih[ ]*/

Prod1(transpose(v[ ][ ]), Oi[ ], no. of hidden, no. of input)
{
    Ih[ ]+=transpose(v[ ][ ])*Oi[ ];
}

/*Calculation of input to output layer Io[ ]*/

Prod2(transpose(w[ ][ ]), Oh[ ], no. of hidden, no. of output)
{
    Io[ ]=transpose(w[ ][ ])*Oh[ ];
}

```

```

    }
/*Computation of the output of neurons in the hidden layer and in the output layer*/

void MLP( input to input layer li[ ])
{
    Initialize Oi[ ]=li[ ];
    prod1(transpose(v[ ][ ]), Oi[ ],hidden, input);
    Oh[ ]=sigmoid(lh[ ]);
    prod2(transpose(w[ ][ ]), Oh[ ], hidden, output);
    Oo[ ]=sigmoid(lo[ ]);
}
/* Weight update*/

void change( w[ ][ ], v[ ][ ] )
{
    /*Hidden-output weight update*/

    d1[ ]=(original[ ]-Oo[ ])*Oo[ ]*(1-Oo[ ]);
    Y[ ][ ]=Y[ ][ ]+(Oh[ ]*d1[ ]);
    Changed_w[ ][ ]=(mtm_cnst*changed_w[ ][ ])+(lrm_rate*Y[ ][ ]);
    w[ ][ ]+=changed_w[ ][ ];

/*Back propagating error to hidden layer*/

    e[ ][ ]=e[ ][ ]+w[ ][ ]*d1[ ];

    /*Input-hidden weight update*/

    d2[ ][ ]=e[ ][ ]*Oh[ ]*(1-Oh[ ]);
    X[ ][ ]=X[ ][ ]+(Oi[ ]*transpose(d2[ ][ ]));
    Changed_v[ ][ ]=(mtm_cnst*changed_v[ ][ ])+(lrm_rate*X[ ][ ]);
    v[ ][ ]+=changed_v[ ][ ];
}
/* Calculation of error */

Error[ ]=(original[ ]-Oo[ ]).
E+=pow(error[ ],2);
E=E/2;
If (E<0.003)
change( w[ ][ ], v[ ][ ]);

```

Fig. 10 shows the structure of the neural network with five inputs. These values are fetched from the input file where the values are generated from the trace file after simulation after running awk script. After training the neural network we use it for detecting Sybil nodes. Input pattern of distrust and enemy nodes that came out from first stage are passed through the trained ANN. At output we get a certain probability of each of these nodes being Sybil. The higher the probability the greater is the chances that a node is Sybil. We consider an approximate probability of 60% and above as Sybil node in a network of 44 nodes. However, this upper limit may vary for different node densities.

4.4. Proposed algorithm

Start

- Create a MANET consisting of a group of mobile nodes with one source and one destination.
- Run simulation with and without attack using AODV routing protocol.
- Compare the node attribute values before and after the attack.
- Calculate packet drop deviation after attack.
- Select nodes having packet drop deviation.
- Assign trust values to the selected nodes using Fuzzy membership table.
- Categorize the nodes as trust, distrust and enemy using Fuzzy inference rule.
- Train the ANN with the five attribute values of the nodes as inputs before attack.

- Apply the input pattern of distrust and enemy nodes to the trained neural network and calculate output.
- The nodes with higher probability (above 60%) values are detected as Sybil.

Stop

In the next section we show the simulation result of the proposed algorithm and analyze its performance in terms of detection rate and false positive.

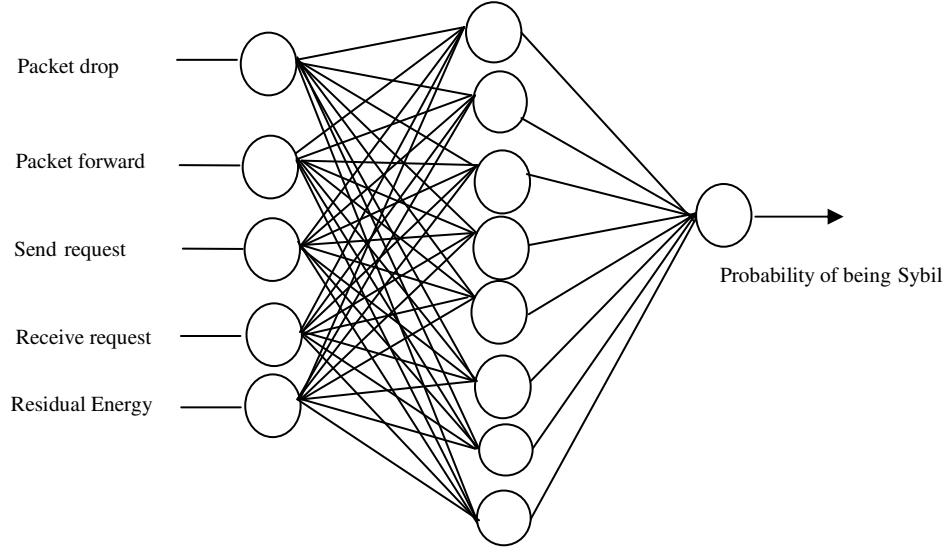


Fig.10. Structure of artificial neural network (ANN) used in FuNN

5. EVALUATION OF THE ALGORITHM

In this section we evaluate the proposed algorithm using the attack model (fig.1) in network simulator NS 2.35. We first run the network in normal condition i.e. when there is no attack. After simulation we fetch the values of the performance metrics for 44 nodes from the trace file (table 4). In the next run we consider the attack scenario where node 0 and node 33 are made Sybil attackers. According to attack model (table 1.) attack starts from 30th s. Attackers change identities periodically after each 20 s. After end of simulation we again fetch the value of the same performance metrics (table 5). At this point we start applying our algorithm.

Firstly, we compare the value of packet drop of each node except source and sink from Table 4 and Table 5. It is to be mentioned that packet drop plays an important role in Sybil attack because the compromised nodes take part in routing and forward data packets to the attackers who consume these packets. Moreover the attacker may also eaves drop information packets on its routes. These causes abrupt packet drop in the network. Hence we calculate packet drop deviation of each node from table 4 and 5 and observe that node 0, 13, 32 and 33 have deviation from normal values while rest of the nodes remain unchanged (table 6).

Table 4. Simulation Result before Attack

Packet Drop	Packet Forward	Send Request	Receive Request	Residual Energy
11	1	15	170	96.074093
0	0	13	145	96.063682
0	0	13	170	96.063682
0	1	13	231	96.122084
0	1	13	146	96.06369
0	0	14	90	96.087267
0	0	14	66	96.195876
0	0	12	128	96.069336
0	1	12	128	96.069336

0	0	13	148	96.063681
0	0	12	116	96.063678
0	0	12	89	96.063678
0	0	13	138	96.063681
10	2	15	122	96.144349
0	0	12	106	96.063678
0	2	13	184	96.063705
0	0	13	138	96.063681
0	1	13	232	96.069328
0	0	13	194	96.063681
0	0	13	149	96.063681
0	0	13	154	96.063681
0	0	13	201	96.063681
0	0	13	238	96.063682
0	0	13	236	96.063682
0	1	14	186	96.063701
0	1	13	148	96.063740
0	0	13	134	96.063681
0	0	13	162	96.063681
0	0	13	203	96.063681
0	0	13	194	96.063681
0	0	13	159	96.063681
0	0	13	127	96.063681
2	1	13	122	96.068894
12	1	14	137	96.121760
0	0	13	175	96.063681
0	0	13	173	96.063681
0	0	13	137	96.063681
0	0	13	103	96.063681
0	1	13	108	96.063708
0	0	13	123	96.063681
0	1	14	149	96.063695
0	2	13	156	96.063699
0	1	13	123	96.063768
0	0	12	91	96.063678

Table 5. Simulation Result after Attack

Packet Drop	Packet Forward	Send Request	Receive Request	Residual Energy
3	0	4	72	99.234219
0	3	1	64	99.234216
0	0	4	70	99.234207
0	1	4	88	99.253873
0	2	4	61	99.234229
0	0	4	48	99.237129
0	0	4	26	99.251055
0	0	4	50	99.234207
0	1	4	39	99.253852
0	2	4	65	99.234225
0	0	4	49	99.234207
0	0	4	38	99.234207
0	0	4	55	99.234207
0	0	4	48	99.234207
0	0	4	46	99.234207
0	0	4	67	99.234207
0	0	4	52	99.234207
0	0	4	84	99.234207
0	0	4	72	99.234207
0	0	4	54	99.234207
0	0	2	57	99.234200
0	2	4	75	99.234232
0	1	4	83	99.234223
0	0	4	80	99.234207
0	0	4	66	99.234207

0	0	4	52	99.234207
0	0	2	44	99.234200
0	0	2	56	99.234200
0	0	3	70	99.234203
0	0	3	64	99.234203
0	0	4	53	99.234207
0	0	3	41	99.234203
0	0	2	37	99.234200
18	2	2	46	99.253236
0	0	2	59	99.234200
0	0	2	57	99.234200
0	0	2	48	99.234200
0	0	2	40	99.234200
0	3	1	32	99.234221
0	0	2	39	99.234200
0	0	2	48	99.234200
0	0	2	52	99.23420087
0	0	2	43	99.234200
0	0	2	33	99.234200

Table 6. Deviation of packet drop after attack

Node	Packet drop before attack	Packet drop after attack	Deviation of packet drop
0	11	3	8
13	10	0	10
32	2	0	2
33	12	18	6

From table 6 and using table 2 we see that with respect to maximum deviation (which is 10) node 32 has a low deviation (0.2) whereas node 33 has deviation 0.6 which comes under medium deviation. Node 0 and node 13 has the deviation range above 0.75 which fall into the category of enemy node. Thus using fuzzy inference rule (as in section 3) we get three nodes out of the 44 nodes with trust levels as follows:

Node 33	distrust
Node 13, node 0	enemy

Now we employ second stage of FuNN where we pass these three nodes to the trained ANN. The probability values at the output of ANN we get are as follows:

Node 0	probability of Sybil character = .651398
Node 13	probability of Sybil character =.598140
Node 33	probability of Sybil character =.849404

Here we see that node 33 has probability of almost 85%. Thus we can infer node 33 as attacker. Node 0 has probability 65% which is higher than the probability of trust nodes. Thus node 0 falls under

enemy category and treated as Sybil node. For node 13 we see the probability is lower than 60% which makes it shift from distrust to legitimate node. One important observation about FuNN is that it double filters the suspected nodes in two steps hence minimizes the chances of false positive.

6. ANALYSIS AND DISCUSSION

This section describes the simulation result in order to analyse the detection efficiency of our proposed scheme under different scenarios. There are some attributes of the network that may affect the accuracy of the proposed Sybil detection scheme such as node density and speed of attacker nodes. We analyze the impact of these parameters through simulation. Although simulation result shows that proposed algorithm successfully detects Sybil nodes with almost 100% accuracy but speed of the attacker and the node density may change the detection rate as shown in fig 11 and fig 12. We consider two metrics to determine these change which are false positive rate and true positive rate. False positive is defined as a legitimate node incorrectly detected as Sybil attacker and true positive implies a malicious node detected correctly.

Our aim is to identify the variation of these metrics in presence of the said network constraint.

According to fig.11 we see that speed has a considerable impact on false positive at higher node densities. We have shown this variation for three different node densities. The reason is that as soon as a Sybil attacker moves with a certain speed, its distance from the compromised nodes changes. When the attacker node goes beyond the communication range of compromised nodes or other legitimate nodes the value of packet drops at this node change. It may also cause some legitimate nodes having high packet drop than that of the normal condition which yields false positive.

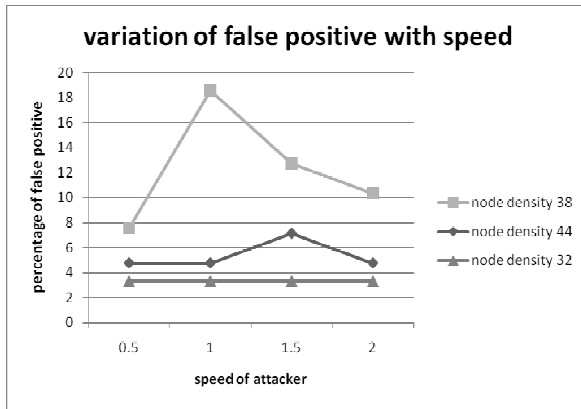


Fig.11. False positive rate with speed of attacker (m/s)

7. CONCLUSION

In the proposed method we addressed a special category of Sybil attack in which a Sybil node occupies multiple identities time to time. The algorithm is proposed for MANET where mobility is an important issue. We assumed various speed of the nodes to test the efficiency of the algorithm. We also considered the performance of the algorithm under different scalability of the network. We have used fuzzy inference rule to preliminary separate the true nodes from the suspected nodes. This reduces the over head of the second stage of the algorithm where we have to test a lesser number nodes with ANN.

One important advantage of the algorithm is that it incorporates the nodes' mobility which is a crucial parameter in MANET. The proposed detection scheme works under this constraint smoothly because of the flexibility of fuzzy inference rule and learning efficiency of the ANN with speed and scalability. Experimental result showed that the detection approach

At lower node density there is no change in false positive due to speed.

In fig. 12 we see that high node density produces high true positive. This is due to the fact that at high density number of connections increase which increase the nodes' frequency to send or receive packets. At higher density the communication between the nodes increase which also increase the chance to detect Sybil nodes. At lower density connections becomes poor which makes Sybil nodes unable to execute their action. This minimizes the chance to detect them. From the above analysis it is evident that the proposed scheme work better at high node density and of MANET. The detection accuracy will be improved at lower speed.

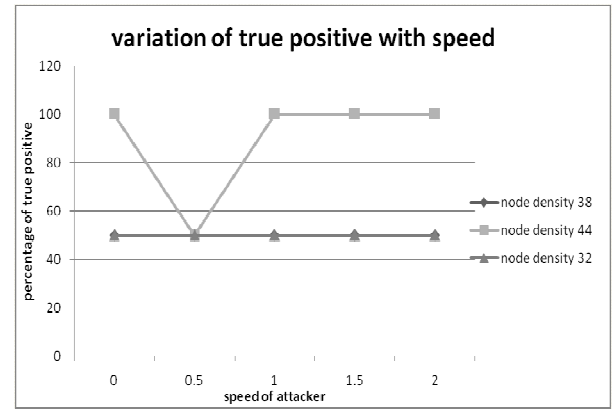


Fig.12. True positive percentage with speed of attacker (m/s)

achieves a true positive up to 90% with a false positive up to 10%.

In future work, we shall consider the dynamicity of the ANN where it will learn automatically with time and with change of different network topologies. In the current study we could be able to consider two attackers which may be increased to judge the algorithm's efficiency. We will also incorporate a preventive mechanism that will either destroy the enemy nodes or isolate them from network. The current study is based on MANET; however it can be extended to other domain of ad hoc network such as WSN as a future work.

REFERENCES

- [1] Chlamtac I., Conti M., and Liu J.N., "Mobile ad hoc networking: Imperatives and challenges, *Ad Hoc Netw.*", vol. 1, no. 1, pp. 13–64, 2003.
- [2] Douceur J.R., "The Sybil attack", presented at the Revised Papers from the First Int. Workshop on Peer-to-Peer Systems, 2002, pp. 251–260.
- [3] Hashmi S. and Brooke J., "Toward Sybil resistant authentication in mobile ad hoc networks", in *Proc of 4th Int Conf. on Emerging Security Inform., Syst. Technol.*, 2010, pp. 17–24.
- [4] Chen Y., Member, IEEE, Yang J., Student Member, IEEE, Wade Trappe, Member, IEEE, and Richard P. Martin, Member, IEE, "Detecting and Localizing Identity-Based Attacks in Wireless and Sensor Networks", in *IEEE transactions on vehicular technology*, vol. 59, no. 5, june 2010, pp 2418-2434.
- [5] Demirbas M., Song Y., "An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks", in *Proceedings of WoWMoM 2006*, pp. – 570.
- [6] Mohaisen A.; Univ. of Minnesota, Minneapolis, MN, USA; Hopper N., Yongdae K., "Keep your friends close: Incorporating trust into social network-based Sybil defenses", *INFOCOM 2011 Proceedings IEEE*, pp 1943 – 1951, 10-15 April 2011.
- [7] Alvisi L., Clement A., Epasto A., Lattanzi S., Panconesi A., SoK: "The Evolution of Sybil Defense via Social Networks, Security and Privacy (SP)", 2013 IEEE Symposium on Security and Privacy (SP), pp 382 – 396, 19-22 May 2013.
- [8] Shamshirband S., Kalantari S., Daliri Z. and Shing Ng L., "Expert security system in wireless sensor networks based on fuzzy discussion multi-agent systems", *Scientific Research and Essays*, Vol. 5, issue 24, pp. 3840-3849, 18 December, 2010.
- [9] Abbas S., Merabti M., Llewellyn-Jones D., "Identity-based Attacks Against Reputation-based Systems in MANETs", 12th Annual Postgraduate Symposium on Convergence of Telecommunications, Networking and Broadcasting (PGNet 2011), Liverpool, UK, pp 27-28, 2011.
- [10] Nadeem A., Gulshan-e-Iqbal, Howarth M.P., "A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks", *IEEE Communications Surveys & Tutorials*, Volume:15 Issue:4, pp. 2027 – 2045, 28 March 2013.
- [11] Cho Jin-Hee., Swami A. and Chen Ing-Ray, "A Survey on Trust Management for Mobile Ad Hoc Networks", *IEEE communications surveys & tutorials*, vol. 13, issue 4, fourth quarter, pp 562-583, 2011.
- [12] Trifaa Z., Khemakhemb M., "Sybil Nodes as a Mitigation Strategy against Sybil Attack", *International Workshop on Secure Peer-to-Peer Intelligent Networks & Systems (SPINS-2014)*, *Procedia Computer Science* 32 (2014) pp 1135 – 1140, June 2014.
- [13] Haribabu K., Arora D., Kothari B., Hota C., "Detecting Sybils in Peer-to-Peer Overlays using Neural Networks and CAPTCHAs", in *proceedings of International Conference on Computational Intelligence and Communication Networks*, 2010.