CHAPTER V

A CLUSTER - BASED FUZZY MODELING APPROACH TO PROVIDING A TRUST - BASED SECURED KEY APPROACH FOR MANET (TUFY)

5.0 INTRODUCTION

MANET adopts the property of self-organization with dynamic mobility [1]. This open complex behavior with hybrid topology leads to a demand for high security against multiple attacks [2] from malicious users. This chapter identifies the node behavior and incorporates the feature of heterogeneous trust membership values among mobile nodes using a variable secured key along with trust metrics. Each mobile node carries a TV TUFY, which is determined as a fuzzy value. These nodes also include the usage of an analytical security key SKi, which employs private key security for MANET nodes within a cluster. Experiments were conducted using the ns2 simulator [3] and a hard-coded real-time approach to evaluate the network interaction quality and the identification of malicious nodes in TUFY.

This chapter defines a fuzzy-based trust evaluation approach (TUFY) or MANET [82], where fuzzy theory is used to determine the nodes of high priority indices in a defined cluster to communicate or adhere as a hand-off node during packet transmission. The fuzzy-based trust implementation for MANET adopts a robust and effective security algorithm for mobile wireless nodes with dynamic mobility.
Any mobile node $Mi$ is considered to be trust-supportable only if the node possesses a TUFY value. Any $Mi$ can adopt the property of ($SK_i$, $Ci$, TUFY$_v$), where $SK_i$ is the private secret key used within the cluster, $Ci$ is the cluster identity and TUFY$_v$ is the TV assigned for an active transmission. Each $Mi$ maintains dynamic variables, such as the expiry time limit of the TV TEX(Time Expiry), the $SK_i$ and the data Transfer rate along with the identity of other mobile nodes $n$ in the session ($Mi[n]$). The application of fuzzy logic is adopted to combine these variables and to identify the priority indices of the session required, which can provide a sustainable secured path for communication. The design of a fuzzy-based priority scheduler TUFY chooses the inputs and determines the degree to which they belong to each of the appropriate fuzzy sets via membership functions.

This chapter discusses an improved trust based management system and the need for a fuzzy - based trust approach for MANET. Section 5.1 discusses literature survey with a specific focus on a fuzzy approach for providing trust between mobile nodes. Section 5.2 discusses Fuzzy – based Trust Evaluation over MANET. Section 5.3 focuses on an experimental approach for test-bed, and Section 5.4 evaluates the performance of TUFY and Section 5.5 summarizes the advantages of TUFY along with the future scope needed for TUFY.
5.1 LITERATURE REVIEW

Recent research works [4], [5], [6], [7] focused on providing security for wireless and ad hoc network demand support towards trust-based security approach in a dynamic MANET environment. There is a need for feasible implementation, improved volatile security features, consistent system updates and the fulfillment of user requirements. Although MANET [1] is considered to be capable of providing heterogeneous services, it has still not paved the way towards implementation of a real-time system. Many algorithms have been used to evaluate the trust of MANET, such as graph theory [8], Bayesian models and Markov chains [9]. However, uncertainty exists in most of the modeling approaches; hence, fuzzy theory is judged to be suitable for evaluating uncertainty and providing effective trust in MANET. The provision of trust is a soft security method that identifies and updates the dynamic behavior of nodes. One major advantage of trust-based security is that it requires less formal information about the entities involved and their related specifications.

The routing protocols of MANET, such as AODV [1] and DSR [10], employ intermediate nodes on a communication path to forward the packets of other nodes such that the mobile nodes can communicate beyond their wireless transmission range. However, due to constraints on the utilization of limited power and computation resources, some nodes may refuse to forward packets to conserve their limited resources (i.e., energy), resulting in traffic disruptions. Nodes
exhibiting such behavior are termed selfish nodes [11]. It has been proven that the presence of such selfish nodes (passive behavior) degrades the overall performance of a non-co-operative MANET [6].

Surveys on security [11], [12] and TV management [7], [13] shows that several approaches [14] have been proposed for collaborative reputation mechanisms to establish reputation ratings for self-organizing nodes with mobility. The authors [15], H. Luo and S. Lu, discussed a strategy-proof trust management procedure based on a node’s previous behavior. Although the management has low overhead, the honor definition for changing TVs has not been well-defined. [16] presented an authentication service to achieve network security by identifying and isolating dishonest users, but the authors did not define the rules for changing a node’s trust state between good and bad TVs.

Several trust relationships [2], [17] have been defined for a context-aware management protocol, but such protocols cause additional issues in trust relationships between nodes. [18] proposed a method for managing multicast key trees that match the network topology and thus reduce the communication overhead of rekeying. The impact factors of keying management servers were not considered due to dynamic node mobility. [9] proposed a two-step secure authentication approach for multicast MANETs with a Markov chain. A node’s TV was analyzed from the previous trust manner displayed in
this group. TUFY trust model can be demonstrated using a continuous-time Markov chain model.

The literature studies in [7], [14] proposed a dynamic trust evaluation framework based on machine learning. A survey on trust evaluation and dynamic routing protocols for MANETs was carried out [19], while a fuzzy reputation relationship was modeled in a P2P environment. Recommendations were computed with fuzzy operators in [17], which obtained an integration of the computational reputation model. This model can block malicious resource peers and prevent malicious recommendation behavior. Fuzzy logic trust management was discussed in [40], which inferred the trustworthiness of entities with uncertainty and imprecision. To monitor the transmission behavior, several trust-based security routing algorithms have been proposed to evaluate a node’s reputation.

SORI (Secure and Objective-based Reputation and Incentive)[20], [92] scheme is to encourage packet forwarding and to discipline selfish behavior. The scheme does not block a malicious node based on issues related to selective forwarding of packets or other malicious behavior. A token-based mechanism [10] is a unified network layer security solution used in MANETs, where each node carries a token to participate in communication. SPRITE [107], [110] is considered a simple, cheat-proof, credit-based trust management system for MANETs, which adopts a metric termed as credit to provide
incentives for mobile nodes to cooperate and report actions honestly. Chiu et al. [17] had proposed an auto-configuration certificate-based security model that, uses message authentication supported by a distributed CA (Certificate Authentication) to provide viable security in MANET. The presence of a distributed CA avoids the possibility that an intruder MN may produce messages or even change the messages already created with the purpose of breaking the protocol or rendering the service unavailable.

[13] proposed a new security routing protocol, CONFIDANT. Initially, introduced as a monitor to obtain the trustee’s transmission state, with the help of a reputation system and a trust manager component, it then implements the evaluation and update of the trust rating. However, when the time expires, the node will again return to act as a legitimate participant and may continue its misbehavior. Yi et al. proposed SAR (Security-aware Ad hoc Routing) [99], which classifies the nodes based on their trust level, such that nodes that have a similar classification share a secret group key. This approach lacks in key sharing among nodes that lead to failure and that can gain access to the secret group keys [27] presented a trust evaluation scheme dynamically based on a routing model (trust DSR). Five routing selection strategies have been proposed based on trust as an evaluation of the transmission links. Because its routing selection is limited on the routes that are obtained from the standard DSR, the final route selected is not necessarily the most trustworthy one. The
scheme given in [10], a trust model based on fuzzy recommendation similarity, was proposed to quantify and to evaluate the trustworthiness of nodes.

### 5.1.1 Need for Fuzzy Trust Management

The IETF MANET workgroup [25] has suggested that to provide enhanced security for mobile nodes with higher throughput (a desirable quality list), the following issues should be considered for a secured routing path: (a) distributed operation, (b) the absence of routing loops, (c) secured on demand session management, (d) proactive operation, (e) security during self-organization, (f) identification of inactivity period operation and (g) unidirectional link support. Fuzzy logic [26] provides a combined framework for dealing with node uncertainty and the tolerance of imprecise data inputs required for the subjective tasks of trust evaluation, packet forwarding review and credibility adjustment.

The methodology of trust invocation in MANET is represented by unique relationships that enable interaction among nodes. Hence, it defines all trust enabled mobile nodes that are authenticated for any communication while the remaining nodes are not involved in any communication. Fuzzy logic supports the ability to effectively handle uncertainty in decision making and imprecision; hence, it is ideally suited for reasoning about trust.
To establish trust using fuzzy set theory [61], [101], the notion of a membership function \( \sigma(x) \) and its relation must be determined. Let \( M_i = \{m_1, m_2, \ldots, m_n\} \) be a set of well defined mobile nodes in a MANET. Here we take \( x_i \) where \( i = \{1, 2, \ldots, n\} \) is a subject of the problem domain and its mapping relation TUFYv = \{f, h, a, v, r, p\}. Let \( \sigma(x) \) be the degree to which a fuzzy variable \( x \) is a member of a set. Node membership is represented as 1 and no membership as 0, while all intermediate values fall into the problem domain between 0 and 1. The membership function \( \sigma(x) \) maps \( x \) into the interval \([0, 1]\). TUFY \( (x_i) \) defines the fuzzy TV for a set of mobile nodes being authenticated based on its membership function \( \sigma(x) \).

Definition 1: For \( \exists \) mobile node \( M_i \), which should possess a defined fuzzy - based TV TUFYv, such that TUFYv \{f, h, a, g, r, p\} when it needs to communicate with another node within cluster \( C_i \) or another cluster \( C_j \).

**Proof**

For all mobile nodes \( M_i \) in cluster \( C_i \), where \( i = 1, 2, \ldots, n \), one can define a fuzzy based TV TUFYvi. For any mobile node \( M_i \) \( C_i \), there exists the relation \( M_i \times \text{TUFYvi} \), with TUFYvi. The same rule is also true for all \( M_j \) \( C_j \). TUFYv is a set of well-defined trust values ranging between 0 and 1, such that each TUFY value possesses \( f = \) Complete Trust \( (1.0) \), \( h = \) High Trust \( (0.8, 0.9) \), \( a = \) Acceptable Trust \( (0.6, 0.7) \), \( g = \)
Average Trust(0.4, 0.5), r = Below Average Trust(0.2, 0.3) and p=Poor Trust (0.0, 0.1).

Fig. 5.1 presents TUFY TVs based on varying node intensities, where nodes a, b, o and p possess Poor TVs. Nodes c, d, e, f and l possess High Trust values, g has Acceptable Trust, nodes h, i, j and k possess Average Trust, and nodes m and n have below average TVs.

**Fig. 5.1 Trust based fuzzy metrics adopted in TUFY**

Definition 2 : For any mobile node Mi, ∃ a fuzzy trust value TUFY (Xi) TUFY {SKi, Vi}, where for Vi, the trust value is (0.0, 0.1, ....1.0], and SKi is the secured key for transmission.
Proof

If $\exists$ is a set of fuzzy TVs for mobile node Mi, then any intersection of nodes falling within TUFYv should not be considered as a relation to TUFY (Mi), where $i = 1, 2, \ldots i, \ldots i+n$. Hence, for TUFY (Mi) $\cap$ TUFY (Mi+n) $\neq$ 0, $\exists$ a relation TUFYv between Mi and Mi+n as TRUE.

Any mobile node Mi in cluster Ci with TUFYv can identify its neighbor mobile node Mj, where TUFYv (Mj) TUFYv (Mi), and any node whose membership function $o(x) \leq$ TUFYv is considered to possess trust at time ‘t’.

5.2 FUZZY – BASED TRUST EVALUATION OVER MANET (TUFY)

The notion of trust [53] can be defined as a degree of belief about the future behavior of other systems and entities. Trust evaluations are based on the past experience of entities with other entities and observations of their related actions [36], [68]. Research work had been incorporated as a unique way of providing security in MANET that can be interpreted as a relation of TRUST among multiple nodes that are willing to participate in various shareable services and handle resource management [47]. Uncertainty prevails among mobile nodes when a node requires one or more communication members to establish trusted communication among peer nodes [Fig. 5.2].
To initiate trust among nodes, the fuzzy set plays its major role in simplifying the process of authenticating each node with a secured private key SKi. The trust needs to be checked and re-assigned when node A is in communication with node B. TUFY has to adopt the following procedures when node A is engaged in trust communication with node B [Fig. 5.3].

[a] A node maintains a TUFY for any defined event or activity between nodes.

[b] TUFYv for a node is updated when the node leaves or joins cluster Ci.

[c] For any mobile node, TUFYv (trust value) is independent of its cluster.

[d] The TUFY for node n may change in each cluster C.
[e] The TUFY update for node $n$ is dependent on its neighbor update and vote.

**Fig. 5.3 Establishment of TUFYv for nodes A and B**

The trustworthiness of the potential interaction among mobile nodes depend upon TUFYv, which works along with the inference rules for each of the trust dimensions. TUFY defines the set of rules [Fig. 5.3] (based on the node properties given in Table 5.1) for mobile nodes to establish communication or to transfer files.

<table>
<thead>
<tr>
<th>TUFYv</th>
<th>P</th>
<th>r</th>
<th>v</th>
<th>a</th>
<th>H</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>O</td>
<td>p</td>
</tr>
<tr>
<td>R</td>
<td>p</td>
<td>r</td>
<td>r</td>
<td>v</td>
<td>A</td>
<td>r</td>
</tr>
<tr>
<td>V</td>
<td>p</td>
<td>r</td>
<td>r</td>
<td>v</td>
<td>V</td>
<td>v</td>
</tr>
<tr>
<td>A</td>
<td>p</td>
<td>v</td>
<td>a</td>
<td>a</td>
<td>A</td>
<td>a</td>
</tr>
<tr>
<td>H</td>
<td>o</td>
<td>a</td>
<td>v</td>
<td>a</td>
<td>H</td>
<td>h</td>
</tr>
</tbody>
</table>

**Table 5.1 Property rules to be adopted for trust establishment**
Fig. 5.4 Steps involved in trust establishment

TUFY adopts the following rules for its update [Fig. 5.4]:

[a] Any mobile node Mi can participate in a trust vote among its cluster to receive a TUFY value.

[b] A node maintains its TUFY value until it reaches its TCL (Time in Cluster) or TEX, when a new mobile node joins the cluster or when an existing mobile node leaves the cluster.

[c] The secured key SKi is generated using the DH(Diffie-Hellman) key algorithm [28], which demands a higher TV based on its fuzzy measure TUFYv and TEX.

[d] The TEX of TUFYv for a mobile node is generated using a random time scale but is limited to 100 ms.

[e] A node Mi can reside in a cluster Ci that does not adopt any defined time limits. The fuzzy trust value TUFYv at any instance depends on TEX for its update.
5.2.1 Encapsulation of a Secured Key SKi

The security that is afforded by the TUFY scheme depends on the difficulty involved in encapsulating a secured secret key SKi through the DH algorithm [29]. The key agreement protocol belongs to a family of the n-party DH scheme, which is an extension of the 2-party DH key exchange [78]. This work adopts the n-party DH method among multiple nodes Mn that agree a priori. Each key SKi, randomly generated by Mi, can belong to oi q. Hence, the secured secret key SKi = Mi x {o1,... on }. Here, σ is the secret exponent of the key generated by Mi, ‘q’ is a prime number, and Mi is the identity of the node.

The system works only when nodes are assumed as honest and will break the operation (by blocking the user’s port) if a user is found to be malicious. The correlative factor of security for a mobile node, the potential TUFY security level, depends on the following factors:

1. The TUFY level of the secret key maintained is strong in the case of a serious brute force attack carried out by an anonymous user. Two types of keys are used in the IEEE 802.11 standard: 128-bit keys and 64-bit keys. A mobile host using a 128-bit key is more secure than a mobile host using a 64-bit key based on the TUFY level of its individual trust.

2. The TUFY TV is modified when a node moves from a known cluster Ci to an unknown Cluster Cj, its secret key is also modified accordingly.
3. A mobile host that frequently changes its secret key is more secure than a mobile host using a constant secret key for an active communication process.

4. If a mobile host has more neighbor hosts within its domain, then the potential of an attacker being within the domain is greater. Thus, the possibility of being attacked is greater.

5. In sufficient communication, bandwidth also affects the handshake of the secret key or TUFY values among nodes.

The TUFY trust certification (maintenance and distributed certification authority) approach considers that each mobile node \( M_i \) has a DH key pair \( \{SK_i, TUFY_i\} \), where \( SK_i = <Mi, Ci> \) and TUFYv is generated by the TUFY algorithm. These private and public keys are to be used in point-to-point communication over a set of MANET nodes required for transactions.

Any TUFYv in this approach can be verified using the system public key \( SK_i \), which is previously informed by each \( M_i \) in the network. The basic operations include: (a) secret key negotiation, where the secret key can be obtained by \( M_i \) in the system initialization or with the auto-configuration service, and (b) secret key updates, where \( SK_i \) is updated for each session in use.

The TUFYv certificate generated by \( M_i \) can be created along with the secret key \( SK_i \) arriving from cluster \( Ci \) or a subcluster of multiple mobile node \( M_i \). Hence, each mobile node \( M_i \) has its own \( SK_i \) generated TV, TUFYv for \( M_i = <Mi[id], SK_i, Ci, TEX> \). Here, \( Mi[id] \) denotes
the mobile node identifier, its secured key SKi, the cluster identity Ci and the TEX. Two procedures control the trust certificate validity. Equation (1) functions to provide an implicit trust certificate innovation and revocation stating that every Mi must renew its TV TUFYv at least after a period of TUFYr, where TUFYr denotes the TUFY renewal time and the trust expiration time is denoted by TEX < TUFYs + TUFYr for any node, where TUFYs denotes the TUFY start time. Equation (2) functions to provide an explicit TV TUFYv revocation.

5.2.2 TUFY Design and Functionality

Equation (1) defines the observed event, which defines the event and notifications received by Mi. OE is directly dependent on Rp.

\[ OE^a_{i=1...n} (A) \leftarrow \frac{\sum_{t=0}^{ttx-1} \alpha_t R^i_{p-n-i}}{\sum_{t=0}^{L-1} \alpha_{n-l}} \]  

(1)

where,

[1] α is an exponential function, α < =0 if n is less than a threshold number of nodes, else if >=1, to be computed.


\( OE^l_{i=1...n} \) observed entity of mobile node i

\( R^p_{n-i} \) retention probability of node i to achieve a TUFY

n : number of mobile nodes

ttx : time for node activity
\( f \) : functionality of node

\( SK_i \) : secured key used for transaction

Equation (2) defines \( TUFY_v \) for a node \( Mi \) based on the observed entity of the setup (from Equation (1) and using the transfer rate for the mobile node).

\[
TUFY_v = \frac{1}{n} \sum_{i=1}^{i \leq n} \max \left\{ \frac{\text{MiTr} - \min \{OE_{i^n}\}}{C_i} \right\}
\]  

(2)

Equation (3) defines weighted metric (or weight) of a mobile node \( TUFY_w \) can be determined based on multiple \( TUFY_v \) values obtained over a period of time \( TEX \). This metric determines the capacity of a mobile node to act as a coordinator node.

\[
TUFY_w = \max (\tilde{w}_i) = (w_1, w_2, \ldots w_n)
\]

(3)

Here, \( w \) is the weight assigned for a node based on its behavior as monitored over a period of sessions. To determine the weight of node \( i \) at a particular time, Equation (4) defines \( w \) as follows:

\[
\tilde{w}_i = \frac{1}{n} \sum_{i=1}^{n} \text{Mix} \int_{\frac{\text{TEX}}{N_k - S_i}} TUFY_v
\]

(4)

\( Sk \) : recent ‘i’ number of interactions made at time ‘k’

\( N_k \) : expected service count of node ‘i’ at time ‘k’

The mobile node \( Mi \) is observed at multiple time instances.
These factors determine the observations collected over a period of time; hence, the weight is determined. Although all of the neighbor nodes cooperate in assigning a TV to node Mi, its individual node behavior also contributes to the TV assignment.

[1] number of nodes in a cluster and their neighbor node trust values.

[2] frequency of opportunities for a TUFYv update when a mobile node degrades its trust.

[3] number of anomalous detections identified for a mobile node leading to a trust alert. TUFY alert.


The process of issuing or updating the TUFY value for a node involves the following steps [Refer Fig. 5.5]:

Step 1: A mobile node Mi requests its identification through the cluster coordinator.

Step 2: Each mobile node issues its existing TUFYv value to the TUFY coordinator.

Step 3: The TUFY coordinator requests that each mobile host within the cluster vote for mobile node Mi.

Step 4: Based on Equation (3), the coordinator identifies the TUFY value for node Mi.

Step 5: The co-ordinator assigns the value to node a. The co-ordinator node is determined based on Equation (4).

Step 6: A new mobile node k joins the cluster, and the cluster co-ordinator requests its identity.
Step 7: Steps 1 through 5 are taken to renew TUFYv for all nodes in the cluster.

Step 8: Node l leaves the cluster; Step 7 is repeated.

![Fig. 5.5 Process of initiating and updating TUFYv](image)

Nodes that rarely cooperate have a low SRR (Service Reputation Rate) and hence are considered to be less trustworthy (assigned a low TV). To maintain the credibility of the below average TUFYv obtained from neighboring nodes, it would be best to update the information reputation of the nodes. SRR denotes the Service Reputation Rate, which is obtained based on the below average metric obtained by monitoring the node at any time k and the weighted average of votes from other neighboring nodes (see Equation (4)).

### 5.3 EXPERIMENTAL TEST BED [79]

Experiments were carried out following two procedures: (i) a real hard - coded experimental approach and (ii) a simulated approach. A
hard-coded test bed was designed and tested by different researchers, where a socket based C++ program was developed and run over 15 MANET nodes. (refer to Table 5.2 for the configuration). The program was designed such that the MANET nodes could communicate at variable speeds. The experiment for TUFY was initially run using 15 mobile nodes over a peer-to-peer ad hoc setup on a small area of 50 feet x 70 feet. Five different clusters were created, each with five mobile nodes. The nodes were consistently mobile with variable speeds, while selfish nodes were introduced, with a minimum of one node in each cluster. Measurements were gathered separately using a simple BSD socket program for each mobile node. Round-trip times were gathered using ICMP echo packets typically measured within 100 ms, such that network delays were observable. The mobile nodes were relatively quiescent during the experiment, where signal processing and latency times were also observed at variable intervals of time. The experimental test-bed helps to identify the mean execution times for various TCP - based signal methodologies created for test purposes. Malicious user calls from five network stations have been tested to attempt to gain access rights for conference servers and to disturb conferencing.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>10, 15</td>
</tr>
<tr>
<td>MAC</td>
<td>IEEE 802.11 b/g/n</td>
</tr>
<tr>
<td>Node speed</td>
<td>10 kms, 20 kms</td>
</tr>
<tr>
<td>Protocol used</td>
<td>TCP / UDP</td>
</tr>
<tr>
<td>Network connectivity</td>
<td>BSD sockets</td>
</tr>
<tr>
<td>Pause time</td>
<td>60 ms</td>
</tr>
</tbody>
</table>

Table 5.2 Hard - coded test bed configuration

5.3.1 Overhead Analysis

Two tests were conducted to determine the overhead involved in using secure session key generation control and dynamic trust certificate agreement in TUFY. These test procedures include a determination of the raw processing time per routing packet for varying established key sessions and measurements of average route acquisition latency. The RTT (Round Trip and Time) between the establishment of sessions between multiple mobile conferees and clusters was examined. TUFY better handles secured trust-based session management, with variable established sessions.

5.3.2 Simulation Test Bed Analysis

Simulations were carried out using a network of mobile nodes placed randomly within a 1000 x 1000 meter area. The radio
propagation range for each node was 250 meters, and the selected channel capacity was 2 Mbps with a fixed data payload size of 512 bytes. A traffic generator was developed to simulate a variable bit rate of packets as the number of mobile nodes increased. Data sessions with randomly selected sources and destinations were simulated. Each source transmitted data packets at a minimum rate of 4 packets/s and a maximum rate of 10 packets/s. The traffic load was varied by changing the number of data sessions, and the effect on the scheduler was examined for different routing protocols.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency of operation</td>
<td>2.4 GHz - 5.5 GHz</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>15, 25, 50</td>
</tr>
<tr>
<td>Node speed</td>
<td>50 ms, 70 ms, 100 ms</td>
</tr>
<tr>
<td>Node placement</td>
<td>Random Uniform</td>
</tr>
<tr>
<td>Node pause time</td>
<td>0 – 10 ms, 0 – 50 ms</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random way point</td>
</tr>
<tr>
<td>Propagation model</td>
<td>Free space</td>
</tr>
<tr>
<td>Received power threshold</td>
<td>-81 dBm</td>
</tr>
<tr>
<td>Transmitted power</td>
<td>7.89 dBm</td>
</tr>
<tr>
<td>Transport layer</td>
<td>UDP, TCP</td>
</tr>
<tr>
<td>Network-layer protocols</td>
<td>ODMRP, AODV</td>
</tr>
<tr>
<td>MAC</td>
<td>IEEE 802.11</td>
</tr>
</tbody>
</table>

Table 5.3 Simulation parameters
The simulator introduced multiple malicious nodes for the test run. During the experimental run, a malicious nodes percentage of 12% was assumed (2 malicious nodes, while a few mobile nodes may be out of the communication range), by default, it was set at 25%, and 35%.

5.3.3 Variations in Mobility

Each node moved constantly with a defined speed [Table 5.3]. The directional movement of each node was randomly assigned, and when the nodes reached the simulation terrain boundary, they bounced back and continued to move. The node movement speed varied from 10 km/h to 25 km/h. In the mobility experiment, ten nodes were multicast members, while five nodes acted as a source to transmit the packets. The simulation for evaluating the fuzzy scheduler was implemented using ns2, a discrete event simulator.

5.4 PERFORMANCE ANALYSIS

The performance of TUFY was tested using different parameters, such as [i] intrusion against selfish malicious nodes, [ii] type of services for TUFY values, [iii] nodes against different TUFY values and, [iv] latency issues (delay, packet loss). The performance of TUFY was also compared with that of other trust schemes, such as TRUCE [24] and CONFIDANT.
**Fig. 5.6 Attacks detected for varying mobile nodes and time**

The performance of TUFY is shown in Fig. 5.6 and Fig. 5.7 TUFY outperforms trust schemes such as TRSTrust [102] and CONFIDANT [13] in the time taken to identify intruders attackers for varying increases in the number of attackers. It has been found that TUFY identifies intruders/malicious nodes based on their behavior [7], [30]. The time taken to identify attackers is found to be reciprocal to the number of nodes in use. Fig. 5.7 illustrates the throughput of mobile nodes with TUFY, where the packet drop is controlled with an update of the TUFY value; hence, the throughput increases.
Fig. 5.7 Packet drop ratio for varying periods of time

The throughput also depends on the packet delay or the round-trip time taken for data to be sent and received. Mobile nodes with TUFY confirm shorter delays for varying mobile nodes, thus explaining the higher throughput. [Fig. 5.8].

Fig. 5.8 Packet delay observed for different mobile nodes
Fig. 5.9 Intrusion node detection using CONFIDANT and TUFY

Fig. 5.9 demonstrates the time taken to identify intruder malicious nodes for the CONFIDANT scheme and the TUFY scheme. The performance of CONFIDANT for 10 mobile nodes was found to be nearly equivalent to that of TUFY for 50 mobile nodes. It should be recognized that CONFIDANT is inferior in memory utilization and node trust update time. This loss of time was modified in TUFY; hence, it uses a minimal time for node updates.

5.4.1 Performance of TUFY Based on Service Type

The performance of TUFY was measured for varying service types, such as Messenger, Streaming Media and email. Fig. 5.8 shows the time taken for mobile nodes based on TUFY values; nodes
dependent on high TUFY values demonstrate high throughput, while nodes with low TVs (TUFYv) demonstrate low throughput. TUFY identifies malicious nodes and prevents intruders from entering the system. Such malicious nodes with low TUFY values are not engaged in communication. The throughput is low due to high overhead; hence, packet delay [Fig. 5.8] and noticeable packet loss issues are observed.

Fig. 5.10 shows the throughput of TUFY for varying types of services. Different service types, such as e-mail chat, and audio play, require multiple sessions, which are highly dependent on TUFY trust metrics. It was found that the TUFY metrics defined as P have a low service priority, while the metric value H has a good throughput.

![Service Throughput for varying TUFY](image)

**Fig. 5.10** Throughput of TUFY measured for varying mobile nodes
**Fig. 5.11 Update of TUFY trust values based on node behavior**

Fig. 5.11 demonstrates the performance of TUFY for two different sets of mobile nodes A and B. The sets of nodes were organized such that Set A consists of 10 mobile nodes whose TUFYv > 0.5 while Set B consists of 10 nodes whose TUFYv < 0.5. Hard-coded program functionality was carried out for fixed time intervals. The updated TUFY values for the mobile nodes remained, on average, within the defined range. Both Set A and Set B mobile nodes with varying entities (leave, join) had TUFY updates, but on average, the values were within the defined range [Fig. 5.11]. Thus, the performance of TUFY has been demonstrated.

**5.5 CONCLUSION**

Recent works in research have used fuzzy theory in trust evaluation and routing decisions for MANET [44], [58]. The authors wish to summarize the following outcomes of TUFY:
(a) It works on a large set of mobile nodes to effectively identify intruders malicious nodes and also to update their reputation, so that such nodes are not used during route transfers.

(b) It is robust when compared with other similar schemes (Section 5.0, Fig. 5.6, Fig. 5.8) in identifying intruders in the minimal time.

(c) The sessions established among MANET nodes are highly secure for communication because TUFY works on top of the secured DH key exchange algorithm; hence, intruders will find it hard to execute an attack.

(d) It consumes minimal overload during key exchange and communication; hence, TUFY has a high throughput.

(e) The TUFY security scheme can also perform well in a worst-case scenario where most of the nodes are selfish and drop packets at varying intervals of time.

TUFY demonstrates a secure on-demand MANET security scheme in which nodes can work in the presence of selfish nodes and intruders. In addition to providing node authentication services and integrity checks, TUFY can mitigate against intruders and malicious nodes that selectively drop or modify packets they agreed to forward. Real-time experiments reveal that TUFY is capable of maintaining a high delivery ratio even when a majority of the MANET nodes are malicious.