IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-2 E-ISSN: 2347-2693

Smart, Secure, and Energy-Efficient Routing Algorithm for MANETs

Lingaraj K[1], Lokesh K S[2*]

[1,2]Rao Bahadur Y Mahabaleshwarappa Engineering College, Ballari, Karnataka, India

Corresponding Author: lokesh.kms@gmail.com

Available online at: www.ijcseonline.org

Received: 09/Jan/2017 Revised: 15/Jan/2017 Accepted: 13/Feb/2017 Published: 28/Feb/2017

***Abstract*- Energy consumption and security are two critical aspects of Mobile Ad Hoc Networks (MANETs). In MANETs, application security can be ensured through trust management, key management, firewalls, and intrusion detection. Additionally, secure communication is crucial in military applications where urgent and reliable transmission of sensitive information is required. However, most existing routing algorithms do not adequately address both energy efficiency and security in their design. Since these factors are essential for reliable communication in MANETs, it is imperative to incorporate them into routing algorithms. Enhancing security in routing protocols and employing cluster-based routing can significantly reduce energy consumption by preventing security attacks. To address these challenges, we propose a novel secured routing protocol called Cluster-based Energy Efficient Secure Routing Algorithm (CEESRA). This energy-efficient protocol leverages cluster-based routing, where trust scores are used to identify and mitigate intrusions effectively. By integrating intelligent agents for decision-making, CEESRA significantly reduces Denial of Service (DoS) attacks, ensuring a more secure and efficient routing process. Experimental results demonstrate that our proposed trust-based secured routing algorithm not only enhances security but also reduces energy consumption and routing delays, making it a robust solution for MANETs.**

***Keywords*- Smart, Secure, Energy, MANETs, AODV.**

I. INTRODUCTION

A Mobile Ad Hoc Network (MANET) enables communication through wireless links without relying on a fixed network topology. It consists of wireless mobile nodes that dynamically form a temporary network without centralized infrastructure, where communication occurs through multi-hop routing. The self-organizing and distributed nature of MANETs ensures network functionality through node cooperation, making it a crucial factor for effective and reliable communication. However, several key challenges exist in MANET design, including security, routing efficiency, medium access control, energy consumption, and reliability. Addressing these challenges requires a secure routing protocol capable of detecting and isolating malicious nodes, thereby enhancing network performance. Security plays a vital role in MANET routing protocols. Packet flooding-based security attacks increase energy consumption, cause network congestion, and may lead to Denial of Service (DoS) attacks. Trusted routing, a secure routing paradigm, mitigates risks by ensuring communication only through trusted nodes. In a trust model, each node is assigned a trust score, which is computed based on its behavior and interactions with other nodes. A trust evaluator gathers opinions from neighboring nodes (trustees) to assess a node's reliability. Trust management is essential for secure communication in MANETs, especially in infrastructure-less environments, as it helps improve both security and energy efficiency. Traditional security methods such as firewalls, intrusion detection systems, and cryptographic techniques have proven inadequate for securing MANET communication. One of the major challenges identified in research is the detection and isolation of malicious nodes, which is critical for secure routing. Given that MANETs are widely used in critical applications like disaster recovery, emergency response, military operations, and mobile conferencing, conventional routing protocols lack the necessary security features. This calls for the development of a robust and secure routing algorithm tailored for MANET environments.

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol is widely used in MANETs for on-demand route discovery. However, integrating security features into both route discovery and route reply processes by modifying packet formats is necessary. Furthermore, energy consumption and node mobility significantly impact routing efficiency. High mobility reduces network throughput and increases packet drop rates. The inclusion of all nodes in routing leads to excessive energy consumption, which can be mitigated through a cluster-based routing protocol, where cluster size is dynamically adjusted based on node distance and mobility. The formation of clusters is a crucial aspect of designing an efficient MANET routing scheme. However, in high-density networks, increased control overhead poses a challenge. Optimizing cluster formation requires considering

factors such as node count, inter-node distance, mobility patterns, and trust values. Large networks should be divided into smaller clusters, with each cluster electing a cluster head (CH) periodically. The CH serves as a central node for intra-cluster communication, while inter-cluster communication occurs through respective CHs. Efficient routing ensures that nodes near the CH consume less energy, while other nodes share the routing burden to prevent energy depletion of specific nodes. This can be achieved by implementing a Minimum Spanning Tree (MST)-based routing approach, which ensures that routing paths are optimized based on trust scores and energy metrics.

This paper proposes the Cluster-based Energy-Efficient Secure Routing Algorithm (CEESRA) to enhance MANET routing by adapting to frequent topology changes. Unlike existing routing algorithms that primarily rely on static nodes and distance metrics, CEESRA integrates energy efficiency and trust-based security mechanisms. The proposed protocol introduces a clustering algorithm at the routing layer, leveraging minimum spanning trees and intelligent agents for clustering, trust management, and routing decisions. The CEESRA protocol employs four categories of intelligent agents to enhance network efficiency:

1. Cluster Formation and Maintenance Agents – Manage dynamic cluster creation and reconfiguration.
2. Trust Management Agents – Evaluate trust scores, detect intrusions, and isolate malicious nodes.
3. Energy Management Agents – Optimize energy consumption and extend network lifetime.
4. Routing Decision Agents – Select optimal routing paths based on trust and energy parameters.

By integrating trust management as a supporting mechanism, CEESRA not only enhances security but also improves overall routing performance. Additionally, path length to the cluster head is considered a key metric for energy optimization, maximizing network longevity. The trust computation complexity is reduced by minimizing the number of nodes involved in secured communication. The Minimum Cost Spanning Tree (MST) algorithm is employed to determine the optimal cluster head, ensuring efficient routing by computing the cost of links based on trust scores and node distance. CEESRA also incorporates a fault-tolerant mechanism to address node and cluster head failures, ensuring continuous network operation. The proposed energy-aware trust-based security mechanism effectively identifies intruders and malicious nodes, preventing security breaches.

II. RELATED WORK

In recent years, network security and secure routing algorithms have become major research concerns in computer networks and communication due to the growing popularity and widespread use of the Internet. Various trust management and energy-efficient routing algorithms have been proposed, each focusing on either security or energy optimization. Most of these techniques are based on the Ad Hoc On-Demand Distance Vector (AODV) algorithm, ensuring uniform routing performance. However, the presence of malicious nodes disrupts this uniformity, compromising data delivery reliability. To address this issue, this section presents an overview of key secure routing algorithms in the literature.

2.1 Cluster-Based Secure Routing Algorithms

Numerous studies have explored cluster-based secure routing algorithms [3–7]. Among them, Dang et al. [8] proposed a distributed clustering scheme for Delay-Tolerant Mobile Networks. Their approach utilizes an exponentially weighted moving-average scheme to update node contact probabilities, enabling early detection of node failures and improving packet delivery ratio while reducing end-to-end delay. However, their model assumes all participating nodes are trustworthy, overlooking potential threats from malicious nodes. Hence, integrating security metrics into the routing process is essential.

Li et al. [2] introduced a secure routing algorithm that evaluates node trust based on packet forwarding ratios, using path trust scores for secure routing in small-scale networks. While effective, a cluster-based approach is preferable for managing routing in large-scale networks.

2.2 Trust-Based Secure Routing in MANETs

Yan et al. [1] proposed **a** dynamic trust evaluation method for securing software component-based systems**.** If applied to network environments**,** their model could enhance routing security by incorporating autonomic trust management into the routing process.

Bao et al. [9] introduced a scalable cluster-based routing protocol that relies on hierarchical trust values to identify malicious nodes. However, their approach was designed for Wireless Sensor Networks (WSNs)**. A** secure routing protocol for MANETs would require modifications, such as replacing sensors with intelligent agents for improved decision-making.

Wang et al. [10**]** developed a trust management framework that integrates vertical and horizontal trust aggregation methods using temporal intervals, statistical analysis, and clustering**.** Their approach effectively combines trust management and clustering, making it beneficial for service-oriented applications**.** However, applying this approach to network routing could further enhance routing and data delivery performance**.**

Nikulin [11] explored clustering techniques based on data compression and labels. While useful for data analysis**,** a

secure routing algorithm must incorporate security, energy efficiency, and distance metrics for effective node clustering.

2.3 Intrusion Detection and Secure Routing

Luo et al. [12] proposed a spectral clustering algorithm for intrusion detection, constructing an affinity matrix based on node similarity. Using eigenvector-based clustering, their model precisely detects intrusions without requiring labeled training data. However, their approach primarily focuses on cluster-based routing. If combined with trust-based security measures, network performance could be further enhanced by isolating low-trust nodes.

Jianliang et al. [13] implemented a K-means clustering algorithm for intrusion detection and routing performance analysis. Their simulation results demonstrated improved routing performance, but secure routing behavior remains a crucial factor for network reliability. Therefore, a trust-based mechanism is necessary for securing dynamic network environments.

2.4 Hierarchical and Cluster-Based Routing Protocols

Jemili et al. [14] developed a hierarchical routing protocol that optimizes routing by dynamically selecting less congested paths, ensuring fast data transfer. Their model improves packet delivery ratio, even under varying traffic conditions. However, security concerns were not addressed, limiting its effectiveness in adversarial environments.

Chen et al. [15] introduced a cluster-based secure routing protocol for MANETs, integrating trust modeling for dynamic node clustering. While effective, their approach could be further enhanced by incorporating energy-efficient routing techniques.

2.5 Secure Routing with Trust and Energy Optimization

Li et al. [16] developed the Ad Hoc On-Demand Trusted-Path Distance Vector (AOTDV) algorithm, integrating trust management into AODV-based routing. Their model selects routes based on trust values, ensuring secure data transmission. Through simulations, they demonstrated that AOTDV enhances security, reduces delay, and improves energy efficiency. However, their clustering approach only considers trust and distance metrics. A more optimal solution would integrate mobility analysis, intelligent agent-based communication, and inter/intra-cluster routing to accommodate diverse mobility environments.

2.6 Agent-Based Secure Routing Models

Das et al. [17] introduced the Secured Trust Model, utilizing intelligent agents for trust-based communication among nodes. Their approach effectively detects malicious nodes, thereby enhancing security. However, their model lacks cluster-based structuring, which is essential for scalable routing in large networks.

Serique et al. [18] proposed a secure routing model designed to eliminate malicious nodes, reducing packet drop rates and improving network security. While effective, integrating a cluster-based routing mechanism could further enhance overall performance.

Despite significant advancements in secure and energy-efficient routing, most existing approaches focus exclusively on either security or energy optimization. However, an optimal routing algorithm must consider:

- Trust-based security mechanisms for detecting and isolating malicious nodes.
- Energy-aware routing to optimize node lifetime and reduce energy consumption.
- Clustering techniques for scalable and efficient routing in large networks.
- Mobility-aware algorithms to adapt to dynamic network topologies.
- Agent-based decision-making for real-time routing adjustments.

To address these challenges, this paper proposes a Cluster-Based Energy-Efficient Secure Routing Algorithm (CEESRA) that integrates:

- Trust modeling and trust-based routing to enhance security.
- Dynamic cluster formation and maintenance for scalable network structuring.
- Intelligent agent-based decision-making for efficient route selection.
- Energy-aware routing mechanisms to optimize network performance.
- Failure analysis and fault tolerance for robust communication.

By incorporating these key components, the proposed CEESRA protocol ensures secure, reliable, and energy-efficient routing, making it ideal for dynamic and large-scale MANETs.

III. PROPOSED WORK

The proposed work consists of a trust based secure routing algorithm that works in three phases namely trust score evaluation, threshold setting and routing using the trust values. This proposed work focuses on two important aspects namely, cluster formation and maintenance and trust based secure routing with intrusion detection. This system collects trace data from the network periodically and performs clustering. Cluster head election is performed based on trust values and behavior analysis. In this work, the trace data are analyzed for intrusions in order to avoid DoS attacks in addition to evaluate the trust of nodes. The proposed cluster and trust based secure routing algorithm which is the main focus of this paper consists of four important phases namely, cluster formation, cluster head election, trust based secure routing and cluster reformation.

3.1 Proposed System Architecture

The architecture of the system proposed in this paper is shown in Fig. 1. It consists of seven modules namely Network trace data, Cluster formation module, Threshold manager, Secure Routing module, Rule base, Cluster reformation module and Administration module.

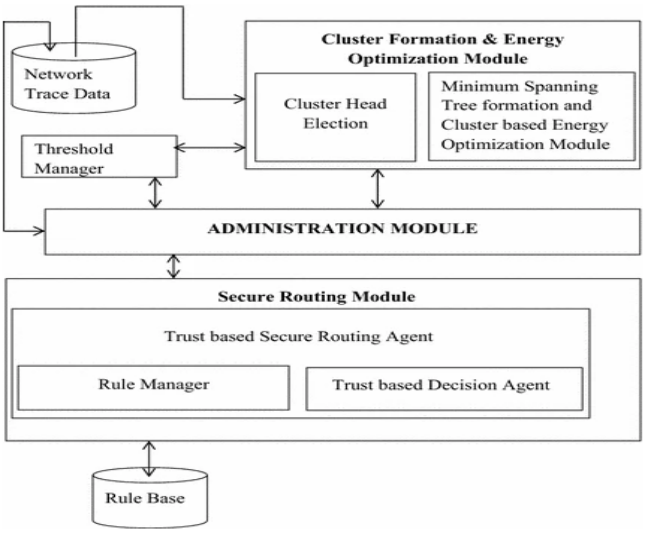

Fig 1: System Architecture

The trace data are collected from the network. The cluster formation module consists of three sub modules namely cluster head selection module, minimum spanning tree formation and cluster optimization module, each of which uses special algorithm to carry out the tasks. All these sub modules are responsible to form clusters for the network. The threshold manager is useful for setting a threshold to select the cluster head. The overall control of the system is with the administration module. This module initiates the communication and on receipt of the acknowledgement or expiry of the timer, it closes the communication. This module controls the cluster formation, cluster head election, failure analysis and is also responsible for secure communication. In order to provide effective security to the packets sent through the network, this module uses the services of the trust based secure routing module. The trust based secure routing module consists of two components namely the trust based decision agent and the rule manager. These modules use intelligent agents that fire rules present in the rule base for effective decision making on security and routing through the cluster heads.

3.2 Proposed Algorithms

In this work, three new algorithms are proposed for performing effective cluster formation, cluster head election and optimal routing. The cluster formation algorithm uses k-means clustering algorithm to form initial cluster and it is optimized by constructing a minimum spanning tree. In the cluster head election module, the nodes with high trust and energy with minimum mobility are chosen as the cluster head for each cluster. Finally, a secure routing algorithm which uses the cluster formation and cluster head election algorithms with additional steps for secure routing has been proposed in this paper. The steps of the cluster formation algorithm are given below:

3.2.1 Cluster Formation Algorithm

```
#include <iostream>
#include <vector>
#include <cmath>
#include <cstdlib>
#include "node.h"
#include "mobilenode.h"

#define CLUSTER_RANGE 100  // Maximum distance for cluster formation
#define ENERGY_THRESHOLD 10 // Minimum energy required for CH
#define TRUST_THRESHOLD 0.5 // Minimum trust value to be CH

using namespace std;

class Node {
public:
    int id;
    double trust;       // Trust value of the node
    double energy;      // Remaining energy of the node
    double mobility;    // Mobility factor of the node
    double x, y;        // Node position
    bool isClusterHead; // Cluster head flag

    Node(int id, double trust, double energy, double mobility, double x, double y) {
        this->id = id;
        this->trust = trust;
        this->energy = energy;
        this->mobility = mobility;
        this->x = x;
        this->y = y;
        this->isClusterHead = false;
    }
};

class ClusterFormation {
public:
    vector<Node> nodes;
    vector<Node*> clusterHeads;

    ClusterFormation(vector<Node> nodes) {
        this->nodes = nodes;
    }

    // Calculate distance between two nodes
    double calculateDistance(Node a, Node b) {
        return sqrt(pow(a.x - b.x, 2) + pow(a.y - b.y, 2));
    }

    // Select the best Cluster Head (CH) based on Trust, Energy, and Mobility
    void selectClusterHeads() {
        for (auto &node : nodes) {
            // Compute Cluster Head Score (Higher is better)
            double CH_Score = (0.5 * node.trust) + (0.4 * node.energy) - (0.1 *
node.mobility);

            if (CH_Score >= TRUST_THRESHOLD && node.energy >= ENERGY_THRESHOLD) {
                node.isClusterHead = true;
                clusterHeads.push_back(&node);
                cout << "Node " << node.id << " selected as Cluster Head." << endl;
            }
        }
    }

    // Assign non-CH nodes to the nearest Cluster Head
    void assignClusters() {
        for (auto &node : nodes) {
            if (!node.isClusterHead) {
                double minDistance = CLUSTER_RANGE;
                Node* bestCH = nullptr;

                for (auto *ch : clusterHeads) {
                    double distance = calculateDistance(node, *ch);
                    if (distance < minDistance) {
                        minDistance = distance;
                        bestCH = ch;
                    }
                }

                if (bestCH) {
                    cout << "Node " << node.id << " assigned to Cluster Head " <<
bestCH->id << endl;
                } else {
                    cout << "Node " << node.id << " could not be assigned to any
cluster!" << endl;
                }
            }
        }
    }

    // Perform the cluster formation process
    void formClusters() {
        selectClusterHeads();
        assignClusters();
    }
};
```

```
int main() {
    // Example Network with Random Trust, Energy, and Mobility values
    vector<Node> networkNodes = {
        Node(1, 0.8, 50, 2.0, 10, 20),
        Node(2, 0.7, 30, 1.5, 25, 30),
        Node(3, 0.5, 40, 3.0, 50, 50),
        Node(4, 0.9, 60, 0.5, 75, 80),
        Node(5, 0.6, 20, 2.5, 90, 100)
    };

    // Initialize and execute the clustering algorithm
    ClusterFormation clusterAlgo(networkNodes);
    clusterAlgo.formClusters();

    return 0;
}
```

In cluster formation, trust is not considered and all nodes are assumed to be trusted nodes. However, trust values are computed for each node before the routing process is started. The trust score evaluation method introduced in this paper is explained in the next subsection.

3.2.2 Trust Score Evaluation Process

In this model, we calculate the trust score for individual nodes based on the following two constraints. First, nodes which are genuinely sending their acknowledgement to neighbors whenever they received the packets are treated as first group. Second, the nodes which drop more packets are considered as group two nodes. Now, the initial trust score is computed using the Eq. 1 that represents the percentage of genuine acknowledgements.

$$TS_{1i} = \left(\frac{ACK}{RP}\right) \times 100 \quad (1)$$

where TS_{1i} = First trust score in percentage for ith node, ACK = No. of acknowledgements sent to the neighbors, RP = No. of packets received from neighbors.

The second trust score is computed using Eq. 2 which calculates the dropped packets.

$$TS_{1i} = 100 - \left(\left(\frac{DP}{TDP}\right) \times 100\right) \quad (2)$$

where TS_{2i} = Second trust score ircentage for ith node, DP = No. of packets dropped, TDP = Total number of packets dropped in network.

Finally, we calculate the overall trust score of the particular node by using Eq. 3.

$$TS_i = \frac{(TS_{1i} + TS_{2i})}{2} \quad (3)$$

3.2.3 Cluster Head Election Algorithm

The cluster head election algorithm is proposed in this work in order to find the cluster heads with high energy and trust but with minimum distance from all the participating nodes. The steps of the proposed cluster head election algorithm are as follows:

- Initialize all nodes and gather node parameters (trust, energy, mobility).
- Compute the Cluster Head (CH) **Score** using the formula: $CH\_Score = \alpha \times Tn + \beta \times En - \gamma \times Mn$
- Select the node with the highest CH Score in a given cluster range.
- Broadcast CH election results to cluster members.
- Re-elect CH dynamically if the current CH moves out or runs low on energy**.**

This algorithm is called frequently based on demand in order to handle the mobility, energy and trust of nodes which change dynamically over time. The trust based secure routing algorithm proposed in this paper is explained the next subsection.

3.2.4 Trust Based Secure Routing Algorithm

The proposed trust based secure routing algorithm performs effective communication using intelligent agents. It also uses the cluster formation and cluster head election algorithms in order to find the optimal and secure route to perform effective routing. The steps of the proposed Trust based Secure Routing algorithm is as follows:

```
#include <iostream>
#include <vector>
#include <map>
#include <cmath>

#define TRUST_THRESHOLD 0.6   // Minimum trust value required for routing
#define ENERGY_THRESHOLD 10    // Minimum energy required for participating in routing

using namespace std;
class Node {
public:
    int id;
    double trust;   // Trust score of the node
    double energy;  // Remaining energy
    bool isMalicious;  // Malicious node flag

    Node(int id, double trust, double energy) {
        this->id = id;
        this->trust = trust;
        this->energy = energy;
        this->isMalicious = (trust < TRUST_THRESHOLD); // Mark as malicious if trust is
low
    }
};

class RoutingTable {
public:
    map<int, double> trustScores;  // Stores trust scores for each node
    vector<int> trustedPath;       // Stores the best routing path

    // Update trust scores dynamically
    void updateTrustScore(int nodeId, double newTrust) {
        trustScores[nodeId] = newTrust;
    }

    // Check if a node is trusted
    bool isTrusted(int nodeId) {
        return trustScores[nodeId] >= TRUST_THRESHOLD;
    }
};

class TrustBasedRouting {
public:
    vector<Node> nodes;
    RoutingTable routingTable;

    TrustBasedRouting(vector<Node> nodes) {
        this->nodes = nodes;
        for (auto &node : nodes) {
            routingTable.trustScores[node.id] = node.trust;
        }
    }

    // Trust Score Calculation based on Packet Forwarding Ratio
    double   computeTrustScore(int   packetsForwarded,   int   packetsReceived,   double
historyWeight) {
        if (packetsReceived == 0) return 0;  // Avoid division by zero
        return ((double)packetsForwarded / packetsReceived) * historyWeight;
    }

    // Select the most trusted path using a modified AODV
    vector<int> selectTrustedRoute(int src, int dest) {
        vector<int> path;
        double maxTrust = -1;
        int bestNode = -1;

        cout << "Finding trusted path from Node " << src << " to Node " << dest << endl;

        for (auto &node : nodes) {
            if (!node.isMalicious && node.energy >= ENERGY_THRESHOLD) {
                double trustScore = routingTable.trustScores[node.id];
                if (trustScore > maxTrust) {
                    maxTrust = trustScore;
                    bestNode = node.id;
                }
            }
        }
```



```
        if (bestNode != -1) {
            path.push_back(src);
            path.push_back(bestNode);
            path.push_back(dest);
            routingTable.trustedPath = path;

            cout << "Selected Trusted Path: ";
            for (int nodeId : path) {
                cout << nodeId << " -> ";
            }
            cout << "Destination" << endl;
        } else {
            cout << "No trusted path available!" << endl;
        }

        return path;
    }

    // Update trust dynamically based on network behavior
    void updateTrust(int nodeId, double newTrust) {
        routingTable.updateTrustScore(nodeId, newTrust);
        nodes[nodeId - 1].trust = newTrust;
        if (newTrust < TRUST_THRESHOLD) {
            nodes[nodeId - 1].isMalicious = true;
        }
    }
};

int main() {
    // Example Network with Random Trust and Energy Values
    vector<Node> networkNodes = {
        Node(1, 0.8, 50),  // Trusted node
        Node(2, 0.7, 30),  // Trusted node
        Node(3, 0.4, 40),  // Malicious node (Low trust)
        Node(4, 0.9, 60),  // Trusted node
        Node(5, 0.5, 20)   // Potentially untrusted
    };

    // Initialize the Routing System
    TrustBasedRouting routing(networkNodes);
    // Select a trusted route from Node 1 to Node 5
    routing.selectTrustedRoute(1, 5);
    // Update trust values dynamically (simulate packet forwarding changes)
    routing.updateTrust(3, 0.2);  // Reduce trust for Node 3 (mark as malicious)
    routing.updateTrust(5, 0.7);  // Increase trust for Node 5

    // Re-select the trusted route after trust updates
    routing.selectTrustedRoute(1, 5);
    return 0;
}
```

The proposed secure routing algorithm considers trust values, distance and energy values for performing optimal routing. The additional constraint is that the mobility speed of a cluster head must be less than the average mobility speed of all the nodes.

IV. RESULTS AND DISCUSSION

This work has been implemented by using the NS2 simulator. For this purpose, 500 nodes were selected and a network area of 500 × 500 meters was used to perform the simulation. Moreover, the random way point mobility model was used in the simulation by considering different number of nodes namely 100, 200, 300, 400 and 500 nodes for carrying out the experiments. The mobility speeds were varied from 1 m/second to 20 m/second and the Ad Hoc on demand based Distance Vector (AODV) routing protocol is used as the base routing protocol for the simulation. The topology of the MANET was varied continuously by changing the cluster structure based on the mobility and the experiments were conducted using both AODV and AOTDV [6] routing protocols under homogeneous environments. The k-means clustering algorithm was used to perform the initial cluster and then the other parameters namely mobility, trust values and energy were used to change the cluster structure. This way the experiments were conducted for half-an hour to one hour duration for each experiment and the average values are used to perform the result comparison.

Table 1 shows the trust score variations between the proposed and existing algorithms.

Table 1: Average trust score in percentage for clusters

No. of nodes	No. of clusters	Trust score (%)		
		AODV	AOTDV [18]	CEESRA
100	5	70.1	77.5	83.5
200	10	66.2	79.0	87.0
300	15	55.6	79.9	79.9
400	20	51.2	78.9	88.3
500	25	61.0	73.1	78.4

From this Table 1, it is clear that all the trust score value of the proposed algorithm gradually increases when it is compared with the existing algorithms due to the formation of dynamic and optimal clusters.

Table 2 shows the delay analysis that makes a comparison between the conventional AOTDV [18] and the proposed secure routing protocol.

Table 2 Energy consumption analysis

Name of the algorithms	No. of packets sent					
	6000	8000	10,000	12,000	14,000	16,000
Energy consumption in AOTDV [18] (Joule)	0.69	1.7	2.8	3.1	3.3	3.6
Energy consumption in CEESRA (Joule)	0.64	1.63	2.73	3.02	3.17	3.44

From this table, it can be observed that elimination of the malicious nodes reduces the delay. Here, the energy consumption is reduced in the proposed algorithm since low mobility nodes are only considered for electing the cluster heads.

Figure 2 shows the packet drop ratio analysis between the existing AOTDV [16] protocol and proposed CEESRA with higher number of malicious nodes presence in the network for different mobility speeds.

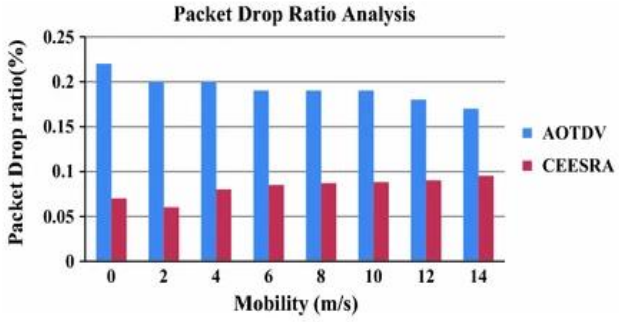

Fig 2: Packet drop ratio analysis with higher malicious nodes

From this Fig. 2, it can be observed that the packet drop ratio gradually decreases in this proposed CEESRA when it is compared with AOTDV [16] with the minimum number of malicious nodes present in the network.

Figure 3 shows the packet drop ratio analysis between the existing AOTDV [16] and the proposed CEESRA with minimum number of malicious nodes.

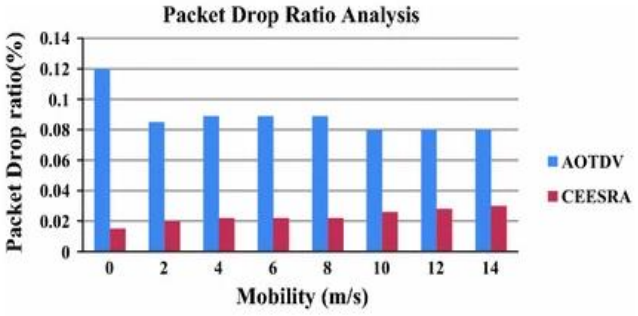

Fig 3: Packet drop ratio analysis under minimum number of malicious nodes

From this Fig. 3, it can be observed that the packet drop ratio decreases more in the proposed CEESRA when it is compared with the existing AOTDV [16] with the minimum number of malicious nodes present in the network. Figure 3 reduces the packet dropping ratio more than the scenario considered in Fig. 2. From this, it can be observed that the reduction in the number of malicious nodes also reduces the packet drop ratio.

Figure 4 shows the energy consumption analysis between the proposed CEESRA and the existing AOTDV [16] with different mobility speeds under maximum number of malicious nodes.

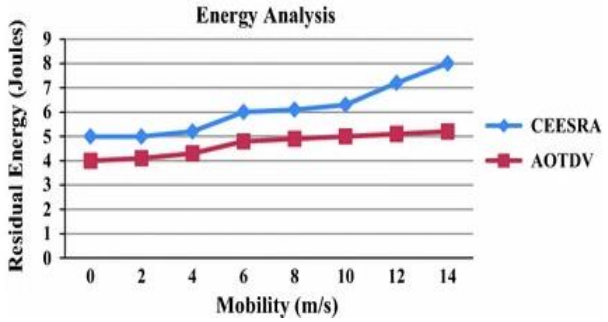

Fig 4: Energy analysis under maximum number of malicious nodes

From Fig. 4, it is proved that the residual energy in the proposed CEESRA is more than the residual energy in the existing AOTDV [16] algorithm. This is due to the use of cluster based routing in the proposed algorithm which reduces the number of hops for each packet sent from the source to the destination leading to reduction in energy consumption.

Figure 5 shows the energy consumption analysis between the proposed CEESRA and the existing AOTDV [16] with different mobility speeds under minimum number of malicious nodes.

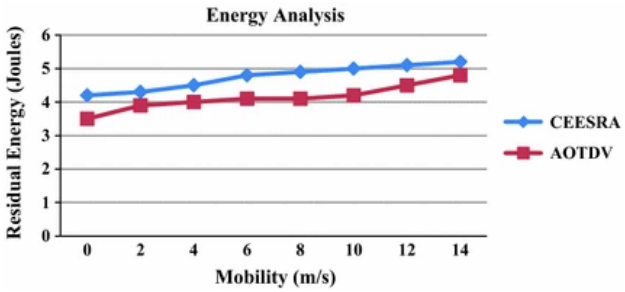

Fig 5: Energy analysis under minimum number of malicious nodes

From Fig. 5, it is proved that the residual energy in the proposed CEESRA is more than the residual energy in the existing AOTDV [16] algorithm. This is due to the use of cluster based routing in the proposed algorithm which reduces the number of hops for each packet sent from the source to the destination leading to reduction in energy consumption.

V. CONCLUSIONS

This paper introduced a novel energy-efficient and secure routing algorithm, Cluster and Energy Efficient Secure Routing Algorithm (CEESRA), designed to enhance trust-based security and energy optimization in Mobile Ad Hoc Networks (MANETs). The proposed algorithm effectively detects and mitigates malicious nodes through a trust score evaluation mechanism, ensuring secure data transmission. Additionally, a new trust score computation technique was developed to enhance trust assessment accuracy. The dynamic clustering technique used in CEESRA considers not only low-mobility nodes but also integrates trust values, energy consumption, and distance metrics to optimize routing efficiency. Extensive NS2 simulations were conducted, demonstrating that CEESRA outperforms existing routing techniques in terms of residual energy, packet drop ratio, security, and throughput. The key advantages of this approach include enhanced security, improved routing performance, and reduced energy consumption. For future work, fuzzy logic-based trust evaluation could be explored to further refine trust computation and enhance decision-making processes, leading to even more adaptive and intelligent secure routing in MANET environments.

REFERENCES

[1] Yan, Z., & Prehofer, C., Autonomic trust management for a component-based software system. IEEE Transactions on Dependable and Secure Computing, 8(6), pp.810–823, 2011.
[2] Li, F., & Wu, J., Uncertainty modeling and reduction in MANETs. IEEE Transactions on Mobile Computing, 9(7), pp.1035–1048, 2010.
[3] Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., & Kannan, A., Intelligent feature selection and classification techniques for intrusion

detection in networks: A survey. EURASIP-Journal of Wireless Communications and Networking, 2013(271), pp.1–16, 2013.
[4] Singh, V. P., & Kumar, K., Literature survey on power control algorithms for mobile ad-hoc network. Wireless Personal Communications, 60(4), pp.679–685, 2011.
[5] De Sanctis, M., Cianca, E., & Joshi, V., Energy efficient wireless networks towards green communications. Wireless Personal Communications, 59(3), pp.537–552, 2011.
[6] Rohokale, V. M., Inamdar, S., Prasad, N. R., & Prasad, R., Energy efficient four level cooperative opportunistic communication for wireless personal area networks (WPAN). Wireless Personal Communications, 69(3), pp.1087–1096, 2013.
[7] Madsen, T., Fitzek, F. H., Prasad, R., & Schulte, G., Connectivity probability of wireless ad hoc networks: definition, evaluation, comparison. Wireless Personal Communications, 35(1), pp.135–151, 2005.
[8] Dang, H., & Wu, H., Clustering and cluster-based routing protocol for delay-tolerant mobile networks. IEEE Transactions on Wireless Communications, 9(6), pp.1874–1881, 2010.
[9] Bao, F., Chen, I. R., Chang, M., & Cho, J.-H., Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. IEEE Transactions Network and Service Management, 9(2), pp.169–183, 2012.
[10] Wang, Y., & Li, L., Two-dimensional trust rating aggregations in service-oriented applications. IEEE Transactions on Services Computing, 4(4), pp.257–271, 2011.
[11] Nikulin, V., Weighted threshold-based clustering for intrusion detection systems. International Journal of Computational Intelligence and Applications, 6(1), pp.1–19, 2006.
[12] Luo, M., Li, X., & Xie, S., An intrusion detection research based on spectral clustering. In Proceedings of 4th international conference on wireless communications, networking and mobile computing WiCOM' 08, pp.1–4, 2008.
[13] Jianliang, M., Haikun, S., & Ling, B., The application on intrusion detection based on K-means cluster algorithm. In Proceedings of international forum on information technology and applications, IFITA'09 2009, vol.1, pp.150–152, 2009.
[14] Jemili, I., Chaabouni, N., Belghith, A., & Mosbah, M., A multipath layered cluster based routing for ad hoc networks. In 2012 5th international conference on new technologies, mobility and security (NTMS) 2012, pp.1–5, 2012.
[15] Chen, A., Xu, G., & Yang, Y., A cluster-based trust model for mobile ad hoc networks. In 4th International conference on wireless communications, networking and mobile computing, pp.1–4, 2008.
[16] Li, X., Jia, Z., Zhang, P., Zhang, R., & Wang, H., Trust-based on-demand multipath routing in mobile ad hoc networks. IET Information Security, 4(4), pp.212–232, 2010.
[17] Das, A., & Islam, M. M., Secured trust: A dynamic trust computation model for secured communication in multiagent systems. IEEE Transactions on Dependable and Secure Computing, 9(2), pp.261–274, 2012.
[18] Serique, L. F. S., & De Sousa, R. T., Evaluating trust in Ad Hoc network routing by induction of decision trees. IEEE Latin America Transactions, 10(1), pp.1332–1343, 2012.

