International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Mobile Ad Hoc Network Reactive Routing Protocol to Mitigate Misbehavior Node

Afsha Nishat[1], Dr. Mohd Abdul Bari[2], Dr. Guddi Singh[3]

Submitted: 09/09/2023 **Revised:** 23/10/2023 **Accepted:** 05/11/2023

Abstract: Mobile ad hoc networks (MANETs) are decentralised and self-organizing computer networks in which nodes are free to come and go and are randomly placed throughout a wireless communication region. The unpredictable and ever-changing topology of a network is a direct result of its mobile nodes. MANET's adaptability and ability to self-organize make it ideal for use in industries where mistakes could have serious consequences, such as medicine, transportation, and the armed forces, where even a single miscommunication could have lethal consequences. Therefore, it's crucial that all data transfers in this sector be encrypted. Routing is a vital component of data communication networks. To put it simply, a network protocol is a set of rules for However, in a hostile environment, the cooperative nature of the nodes in a network can't be assumed, and hence route selection in MANETs fails. When intermediate nodes drop or refuse to forward packets, it causes misbehaviour at the network layer. But because to resource limitations, packets may potentially be dropped by intermediate nodes. In this paper, we examine malicious routing behaviour in a mobile ad hoc network (MANET). By minimising all potential causes of packet drop owing to limited resources, our proposed "Reactive Routing Protocol to Mitigate Misbehavior node" can reliably identify the malicious node. The proposed technique mitigates misbehaving nodes with less overhead than existing secure knowledge algorithms, as shown by analytical and simulation findings.

Keywords: *MANETs, errant node, limited resources, and circuitous routes.*

1. Introduction

The nodes in a mobile ad hoc network [1] (MANET) are completely autonomous and heterogeneous, and they all communicate with each other via radio. MANET is well suited for deployment in industrial settings because of its autonomous features, such as self-configuration and self-maintenance. In contrast to wired and wireless infrastructure-based networks, however, the open wireless medium, heterogeneity, and multi-hop nature of a MANET make it more susceptible to security assaults. Furthermore, the MANET's inherent weakness towards security solutions and the lack of an explicit method to deploy security algorithms stem from the network's peer-to-peer nature, in which nodes in the network play the role of both router and host.

MANET is used in contexts where the main function is to save lives, such as in emergency medical sites, the military, and disaster relief. Since the transmission of even a single false message can have serious consequences, there must be a strong push to incorporate security measures into MANET's applications. In addition, nodes used for monitoring and detection have limited resources, such as limited battery life, making it unacceptable to stop the mission to refuel or swap out batteries. They must be efficient in energy use to prevent the network's nodes from failing. To be practical for deployment in industries, a

[1] *Research Scholar, Kaling University, Raipur.*

[2] *Assocate Professor, HOD Computer Science Department ISL College of Engineering, Hyd, Telangana, India.*

[3] *Associate Professor, Computer Science Department, Kalinga University, Raipur.*

MANET's security mechanism must have a low computational and administration overhead.

When two nodes in a network are within radio range of one another, they can communicate directly with one another; otherwise, they must use intermediary nodes. Assumption of cooperation and coordination among MANET nodes is used in the routing path selection [2]. The primary goal of secure routing protocols [3] is to ensure the safe delivery of data packets between communicating entities. Network layer misbehaviour typically results from intermediary nodes failing to properly forward packets or failing to properly drop packets. When an intermediate node agrees to forward a packet but later refuses to do so for malicious reasons, it is behaving badly. In addition to intentional interference, packet loss can occur when an intermediate node is overburdened, malfunctioning, or operating at capacity. So, we classified the bad actors into two categories: intentional misbehaving nodes that are purposefully causing problems by engaging in behaviours like Black hole attacks, Gray hole attacks, false reports, and partial drops. A misbehaving node may lose packets due to resource constraints or network factors such as a collision, low transmission power, low energy, buffer overflow, or TTL in the packet header expiring.It is a major issue that node-level packet loss in MANETs degrades overall network performance. Several protocols have been proposed in the literature with the goal of removing malicious nodes from the communication path by accounting for packet dropping but not taking into account the reasons for packet dropping. These protocols take a credit-based approach [4-7], a reputation-based approach [8-13], or an acknowledgement-based approach [14-16]. To correctly reduce the impact of a misbehaving node, we offer a secure routing approach that can prevent packet loss caused by malicious but accidental conduct.

The remaining sections of the paper are laid out as follows. Our study consists of three sections: section 2, which examines the associated literature work; section 3, which outlines the suggested work; and section 4, which consists of a performance analysis and conclusion.

2. Background

The primary function of the network layer in a MANET is to determine the path taken by data packets as they go from one communicating entity to another. For route [17] selection, some metrics including distance vector, delay, bandwidth, and energy are used. These protocols presume that a network's nodes will always agree with one another and work together harmoniously. Unfortunately, in a hostile environment, this consideration is not always true; malevolent nodes can readily affect the network activities by not following the protocol definitions. Using alterations to the control messages of a targeted routing protocol, malicious nodes [18] lure traffic toward themselves before dropping it. Some malicious nodes help with route selection but later stop helping with data forwarding and simply drop packets. In addition, rogue nodes send a bogus report to the source, claiming that the data was not sent on to the next-door node. We classify the aforementioned malicious nodes as deliberate misbehaving nodes because they behave maliciously on purpose by dropping packets. In addition, collision, congestion, a shortage of energy, or a lack of transmission power are all possible causes of packet loss in a MANET. We classify these nodes as accidentally misbehaving because they drop packets. It has been proposed in MANET [19] to use a variety of strategies to reduce the impact of misbehaving nodes. There are primarily three classes of these mechanisms. The first type is the credit-based method, which is predicated on the idea of providing incentives for nodes to reliably carry out networking tasks. This plan makes it possible to set up a digital currency or other form of electronic payment to accomplish this goal. In exchange for payment, nodes perform services for other nodes. Similar to how these nodes pay for other network nodes' packet forwarding services, these nodes employ a similar mechanism of payment. One major negative of credit-based schemes is the necessity for additional security measures, such as tamper-resistant hardware, in

order to ensure the integrity of digital currency and electronic payment systems. The second type is known as end-to-end acknowledgment, and it makes use of acknowledgements signals from the destination node to the source node to indicate that the data packets were successfully received. The monitoring and observation processes of network nodes identify the misbehaving nodes, falling under the third category of Reputation / trust based methods. One of a node's two options for determining trust is to integrate data from its neighbours, or to get information from them. The monitoring part of the node is crucial to the trust value calculation. All three approaches can identify and block malicious nodes, but only those that are acting dishonestly on purpose. In addition, Ayesha et al. [20] proposed a strategy to identify and stop both deliberate and accidental misbehaviour from individual nodes by promiscuous monitoring. Each network node continuously and unobtrusively examines the packet operations of its neighbours; if the number of dropped packets exceeds a predetermined threshold, the protocol investigates the cause. If a node doesn't drop a packet due to low battery, memory, or TTL, the protocol assumes it's behaving maliciously and notifies all other nodes to avoid communicating with it. There is a possibility of packet loss due to insufficient bandwidth using this approach, and the metric used for routing is minimum hop count. Reason for packet loss is only determined if the node's packet loss rate is greater than the set threshold. Which interns cause a network to experience more delay and routing overhead. As a result, in this effort, we're determining how to select routing paths so that the least amount of packets are lost owing to limited resources. We then employ the method for finding and eliminating nodes that are being bad. What follows is the primary result of our work that we bring to this paper. Identifying and preventing malicious nodes with limited resources through the analysis of packet loss. Promiscuous mode and authenticated key agreement for the identification of intentionally misbehaving nodes. Decisions about the routes to take in a network are made depending on the current buffer and energy states of each node.

3. Unintentional misbehaving node detection and prevention

Multiple hop communications frequently fail as a result of packet drops from intermediary nodes due to MANET resource constraints like energy and buffer. The router layer's congestion control and energy efficiency methods are usually responsible for reducing packet loss. The congestion in the intermediate node buffer is caused by routing protocols that use a distance vector, resource awareness, or priority-based metric. Here, we assess the amount of data dropped because of a node's buffer filling up or its power supply failing

3.1 Node Buffer enables the detection and prevention of inadvertent misbehavior on the part of nodes.

If two nodes in a MANET are within radio range of each other, they can communicate directly; otherwise, they must rely on an intermediary node. This means that nodes in a network must perform the dual role of host and router in order to transmit the data of other nodes. Different routing protocols have been developed in the literature to choose the most time- and resource-efficient path between communicating entities. Metric shortest path is the foundation of the majority of these routing systems. Some of the nodes are able to manage high volumes of traffic thanks to the protocols' optimization of their resources. However, this protocol causes certain nodes to become overloaded since they are receiving more traffic than they can process. The packets are temporarily stored in a buffer at each network node while they are being routed. The input queue, the processing module, and the output queue are the three main components of a buffer. If the routing protocol specifies that a packet must wait in the node buffer before being sent to another node, the packet will remain in the node buffer. Any node in the communication chain can act as a router to forward the data packets of other nodes if necessary. Each node's packets are stored in the buffer's input queue until the queue is full. The packets in the input queue are removed by the processing module, which then processes and decides on the packets in accordance with the routing protocol specification. Next,

packets enter the output queue to wait their turn to be transmitted. If the pace at which packets are arriving at the node is greater than the rate at which they can be processed, a queue will form in the node's buffer; once the queue length reaches the node's capacity, packets will begin to be dropped. If a packet is held in the buffer for longer than its TTL, the node will discard it. Not only does this cause packet loss, but it also wastes valuable bandwidth and slows down the network. In order to determine the state of a node, we develop an analytical model that takes into account the average queue length of the buffer and the average packet waiting time inside the queue.

The average waiting time of packets and the number of packets in the node's buffer are the two most important factors in assessing whether or not a node is losing packets. Considering the middleman, or intermediary node, which facilitates communication, we are able to quantitatively evaluate these parameters. It takes in data from various sources, stores it in a buffer of size, and then sends it on to its final destination in accordance with the requirements of the routing protocol. First, a node receives a batch of packets, which are queued in a buffer before being processed by a processor and then sent out. Without sacrificing generality, we calculate the median incoming packet count per node per time interval as follows.

$$R = \alpha R_c + \beta R_p \text{.....1}$$

We take its value to be where the average number of packets received over the preceding time interval, the average number of packets received over the current time interval, and are constants. If no packets have arrived in the preceding time frame, we will treat the arrival at time zero as event zero. We did a similar calculation to determine the typical amount of data sent by a node during a certain time period.

$$T = \alpha T_c + \beta T_p \text{.....2}$$

Where, previous interval's average number of packets transmitted and current interval's average number of packets transmitted. Congestion occurs and packets are dropped from the node buffer if the node receives more packets than it can send. Nodes process and send packets without dropping any of them if the rate at which they are received is lower than the pace at which they are sent. This only holds true if the buffer size is infinite, which is not the case in a mobile ad hoc network since intermediary nodes typically have a small memory footprint. A packet may be lost if its queue length is longer than the buffer's storage capacity, or if it has to wait in the buffer for longer than its TTL value allows. For this reason, we determined the typical amount of packets waiting in a node's buffer at any given moment.

$$Q_p = \left({}^{R}/_{(T-R)}\right) \text{.....3}$$

$$\text{Queue size } (Q_S) = Q_p * \text{kbytes} \text{.....4}$$

If the condition holds, the intermediate node will discard the packets. As an added bonus, we calculated the buffered packets' wait time as follows:

$$W_t = \left({}^{1}/_{(T-R)}\right) \text{.....5}$$

If the condition holds, the intermediate node will discard the packets. Algorithm 1 demonstrates analytical detection of packet loss by a node serving as a router for communicating entities.

Algorithm 1:- Unintentional packet drop detection due to buffer

Procedure for packet drop detection (Q_s, W_t, B_i, TTL)

{ if $(R > T) == true$

Unintentional misbehaving node //packets will drop from intermediate node

else if

$if\ (Q_s = B_i) \,||\, (W_t \geq TTL) == true$

Unintentional misbehaving node //packets will drop from intermediate node

else

go for Algorithm 3 //Packet process from node }

Authors are encouraged to submit their

The primary goal of this analysis is to identify the nodes that unintentionally misbehave as a router for various interacting entities due to a lack of buffer space. It is also used to determine the node's participation in routing operations and to regulate the traffic coming into the node. This architecture can stop packets from being lost because of buffer overflow or TTL expiration at an intermediary node.

3.2 The use of Node Energy for the detection of accidental misbehaving nodes

Packet loss at a relay node can also be caused by that node's limited availability of node energy.

However, in MANETs, the routing path must also take into account the restrictions of the network, which is not the primary goal of the routing mechanism. Energy is a major issue for MANETs since nodes have limited battery life and it might be difficult to swap them or recharge them mid-mission. Fast route building and reconstruction, little packet loss, and high throughput are all indicators of a successful routing mechanism. Why the energy-efficient routing method in MANETs is so revolutionary. Mobility, heterogeneity, a lack of infrastructure, and a dynamic network topology make managing a MANET more difficult than managing a network based on a static set of nodes and links. Path selection metrics in energy-efficient mechanisms aim to preserve network energy by considering factors such as communication energy costs, connection dependability, and node residual state.

Since the mobile ad hoc network's heterogeneous nodes double as routers and hosts, energy efficiency in routing is a crucial issue. The power of the node is depleted by the routing feature. Work [21], analysed that, power usage for receiving & transferring packets is roughly 800-1200 mw. Large, power-hungry applications can benefit from MANETs if their routing protocol is also efficient. In literature [22-24] many routing strategies developed depending on energy. The purpose of these protocols is to increase network longevity, network efficiency, and link reliability in order to prolong energy conservation. There are three main categories for these protocols. [25]

Method for determining the most direct path between two points via a network Protocol for determining the most energy-efficient path during route planning a strategy for locating the paths with the highest energy density Low-energy consumption To find the most reliable path, a routing protocol must take into account not only the energy consumption of each node, but also the status of the links between them and the nodes' current energy levels. In their work [26], the authors show how to implement a reliable routing scheme that uses an estimated number of transmissions to both save power and restore data that was

lost during transmission. The algorithms presented in [27,28,29] are designed to increase the network's lifetime by using the nodes' remaining energy, but they don't take energy efficiency into account. The energy usage is not taken into account by the proposed algorithms in [30,31,32]. These algorithms depend on the transmission power of the nodes. All of the aforementioned routing protocols reserve a specific routing path for communication based on a criterion, such as the highest energy node, the least energy consumed route, or the fewest retransmissions. If a node along the route is accepting a lot of traffic because of its high energy, or if it's located on a path with a low transmission count or low transmission energy, that node's ability to send packets will likely expire soon. Figure 1 depicts a network with what are known as bottle-neck nodes.

However, the issue of an intermediate node being a bottleneck node in a multi-link routing path was not taken into account by these protocols. When a node's energy threshold is used as a routing metric, connection dependability cannot be guaranteed by the routing mechanism. In addition, traffic builds up on links, and intermediary nodes become bottlenecks. If a node is receiving a large amount of traffic while running on its remaining power, it will drop the packets owing to a lack of buffer and will quickly run out of power. The network's efficiency will suffer as a result of this problem. Based on the existing residual status of network nodes with respect to energy, we present a unique energy-efficient measure using the knapsack algorithm to prevent the node from becoming a bottleneck.

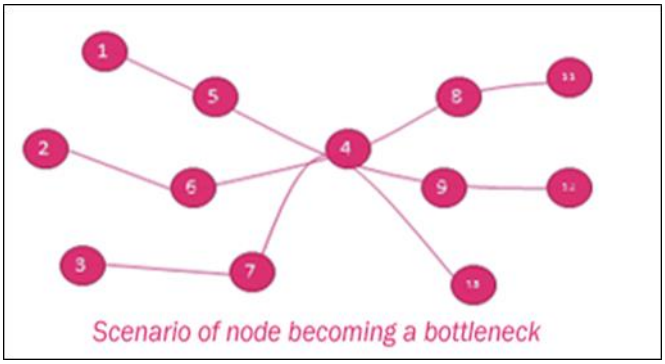


Fig 1: A potential scenario in which a node will become a bottleneck (node 4 is a bottleneck node)

Determine the energy status of a node in a network at the present time. Think of a multi-hop network where the nodes are mobile and have power packs and packet buffers. Since most network communication occurs in little data packages, that's what we'll focus on here. Let's say the packet has to go through an intermediate node, which causes a delay and uses up resources. To ensure that data is transmitted from nodes without interruption due to dropped packets, it is necessary to determine the nodes' current residual condition.

Each joule of energy is the most that can be transferred by a node's packets.

The packet throughput of the node must be maximised.

A node must not process any data from the packet that it has not been explicitly instructed to.

The amount of energy a node needs to receive, analyse, and transmit a packet can be used to determine its current residual condition. We assume that in multi-hop MANETs, multiple sources are trying to transmit packets through an active intermediate node that is equipped with a battery that can supply energy in the form of joules and a buffer that can store data in the form of bytes, and that this node can handle a maximum of 'packets' from all sources, taking into account the resources at its We assume that, in a network, a certain amount of packets must go through an active intermediate node in order for the communication to progress, and that these packets must either travel in full or

be discarded. We consider the following for topples with n in the positive range:

Maximum Each packet is bytes in size and requires processing energy equal to the node's joule total in order to move around the network. In order to determine how much power is needed to move a packet through a live intermediate node, we can use the formula

$$e(P_i) = \left(E_r + E_{p+}E_t\right).....6$$

Whereas, the three activities of receiving, processing, and sending a packet each demand a certain amount of energy. The amount of power left in a node after packet processing can be calculated as

$$E_r = E - e(P_i).....7$$

To compute a desired subset, the knapsack technique [33] appends a Boolean auxiliary array with the key Keep [i,e(P i). If the active intermediate node proceeds I th packet from the communication array Keep [i,e(P i),] then this Boolean array will become one; otherwise, it will become zero. In algorithm 2, you will find the method that will compute the current residual state of the node.

Algorithm 2

Knapsack (l, $e(P_i)$,n, E) {

For ($e(P_i)$=0 to E) S[0, $e(P_i)$]=0;

for(i=1 to n)

for($e(P_i)$ =0 to E)

If(($e(P_i) \leq E$)and(l[i]+L[i-1, E - $e(P_i)$] > L[i-1, $e(P_i)$])){

L[i, $e(P_i)$]=l[i]+L[i-1, E- $e(P_i)$];

Keep[i, $e(P_i)$]=1; }

else V [i, $e(P_i)$] = L [i-1, $e(P_i)$];

Keep [i, $e(P_i)$] =0; }

K= E ;

For (I = n down to 1)

If (keep [i, K]==1){

Output i;

K= K- $e(P_i)$;

Return L[n, E];

It is a route selection measure between communicating entities and is based on the computed current residual state of each node's status about energy. Its function is to determine routes between the entities. When a node in a network is selected to serve as an active intermediate node for communication, that particular node must do the computation necessary to determine its current residual state with the aid of the method that we have proposed. After then, this value must be receptively kept within a table. Following algorithm 3 involves comparing the actual CR value of the node to a predetermined threshold to decide whether or not the node needs to take part in the routing path. In an idle state, we are treating threshold values as the CR maximum, and in the least circumstance, we are treating them as the CR minimum.

Algorithm 3:- **Procedure for deciding present residual status of intermediate node**

Procedure for calculation of current residual condition of node (CR, CR_{Min}, CR_{max})

{ Compute the CR from algorithm 2

if $(CR \geq CR_{max})$

node participate in routing

else if $(CR_{min} < CR < CR_{Min})$

node act as a backup node

else $(CR < CR_{Min})$

4. Detection and Prevention of Intentional Misbehavior Nodes

4.1 Method for promiscuous monitoring to discover intentionally misbehaving nodes

Mischievous monitoring [34] is a component of the approach that we designed, the objective of which is to identify the nodes in the network that are losing packets and is one of the ways in which we intend to do this. Each node in a network is required to continually monitor the packet behavior of its immediate surrounding node, specifically the number of packets that are both received and transmitted. This is done so that the network can function properly. If the number of packets received by the neighbor node is indicated by the notation p r and the number of packets transmitted by the neighbor node is indicated by the notation p t, then the node will compare the monitored information and decide whether or not both values are the same. If they do match, for instance if p t = p r, then the node will conclude that the neighbor node is not a misbehaving node and go on to the next one. If the information that is being monitored does not match, as shown by p t > p r, then the difference is computed. If the information does match, then the difference is not computed. When the disparity exceeds the predetermined level, the neighbor node in issue is seen as behaving improperly since it has crossed the threshold. Control messages are used to relay this information to any remaining nodes in a network that are still functioning properly. On the other side, promiscuous monitoring is incapable of identifying a false misbehaving node on the routing path. As a result of this, our strategy makes use of the secure acknowledgment protocol in order to lessen the likelihood of being targeted by a fake malicious conduct attack.

Algorithm 4:- Promiscuous monitoring method between nodes S and D

Node 'S' promiscuously monitor its neighbor node 'D'

If monitor interval== true

For communication Session[i]

Monitor the received packets at neighbor node 'D' =p_r

Monitor the transmitted packets at neighbor node 'D' =p_t

If $p(_t == p_r)$

Node participates in communication

4.2 Authenticated key agreement among Entities

We present the idea of authenticated key agreements between communication entities as a means of mitigating the effects of a false misbehavior attack that originates from the routing path. Wireless Due to the open wireless channel, heterogeneity, and multi-hop peer-to-peer network environment that ad hoc networks operate in, these types of networks are susceptible to security breaches. It is unbearable to attempt to recharge or replace the batteries of network nodes while the mission is in progress because the nodes themselves are resource constrained. For example, nodes in a network have a limited amount of available energy. Therefore, the overhead of MANETs' security mechanisms needs to be as low as possible, both in terms of computation and management. To offer security, the most important pre-requisite, which is also the simplest and most relevant one, is to provide authentication between the organizations that are communicating through the use of a key agreement. Because the effectiveness of any cryptographic approach is dependent on the method it uses to maintain its keys. Authentication has two purposes: It prevents malicious actors from exploiting the network infrastructure and guarantees that the secret key is securely agreed upon by all involved parties. Yet, due to scarce resources, authentication services on MANETs might be challenging to implement. In this section, we will use chaotic maps to build a mutual authentication mechanism for MANETs using public-key cryptography. The performance findings show that compared to RSA and ECC, chaotic maps have a far lower overhead in terms of processing expenses.

Mutual authentication and key agreement among the companies communicating is a crucial prerequisite for ensuring security in any network. This is because it is practical and simple. This process is used to verify the identities of both parties and establish a shared secret key. This happens in the background when the data is sent. To prevent the formation of the secret key by any parties other than the source and the destination, mutual authentication, and key agreement is required. As MANET nodes are self-sufficient and can verify and establish authenticity independently, this approach is well-suited for usage in these networks. In addition to known-key security, key compromise, perfect secretes forward, implicit and explicit key authentication, unknown-key sharing, and key confirmation, the mutually authenticated key agreement method also includes several other desirable traits. Nodes in a net benefit from these characteristics because they enable them to fend off security threats and operate autonomously in situations with little supporting infrastructure. Nevertheless, the effectiveness of any cryptographic method relies on its key management, often known as mutual authenticated key agreement. In environments with limited access to resources, such as MANETs, the technique for a key agreement should have a minimal impact on either the amount of energy used or the amount of time spent computing. In the research literature, many key agreement procedures have been presented. These protocols are based on various cryptographic approaches, however, they do not take into account the computing cost. Therefore, MANETs require an authenticated key agreement procedure that places a minimal burden on the computing resources available. We provide a technique that, based on the Chebyshev polynomial, may establish the key agreement between the source and the destination. This will allow us to satisfy the criteria. The Chebyshev polynomial has a computation cost that is independent of the modular arithmetic of RSA and the scalar multiplication of ECC. In addition, the creation of our keys is based on a problem called Chaotic Maps-Based Discrete Logarithm, which an intruder may not be able to solve in a polynomial amount of time if they try.

Several security methods have been proposed in published works to prevent or defend against assaults [35] using a variety of cryptographic procedures [36]. The provision of authentication, authorization, the integrity of information, and non-repudiation is the primary objective of security. The plaintext is changed into ciphertext using an appropriate key and cryptographic procedures to

accomplish the purposes of security. These approaches are used to achieve the aims of security. Therefore, the processes of key management distribution and maintenance play an important part in these strategies. However, to function, cryptographic techniques require resources from the network, which has a negative influence on the network's performance. The cryptographic approach used in MANETs has a significant influence on the amount of energy used, as well as the processing and memory of the network nodes. Therefore, while creating security mechanisms in MANETs based on cryptographic techniques, one should strive to minimize the cost that these techniques impose on the performance of the network.

A variety of various security measures, each based on a different cryptographic method, have been developed to secure communication and keep the data from being accessed by unauthorized parties. According to the terminology underlying cryptography, "an encrypted document is produced by encrypting a plaintext document with a cipher text to produce a cipher text document, which is known as encryption; decryption is the reverse process of encryption; a cipher may utilize one or multiple keys," and "a cipher may utilize one or multiple keys.". "Encryption" refers to the process in which a plaintext document is encrypted with a cipher text to produce a cipher text document known as The primary components of a modern cryptographic system may be broken down into two categories: the public key and private key cryptography. The encryption and decryption processes of public key cryptography utilize separate keys, whereas the encryption and decryption processes of private key cryptography use the same key. In this study, we will not be discussing encryption using private keys since, compared to encryption using public keys, it requires significantly more processing power. Specific cryptographic standards, such as "trusted" authorities, a specific implementation of algorithms or/and protocols (including key sizes), generation of seeds and random numbers, parameters of algorithms, assurance of hardware and/or software support, and so on are used in the design of the public key.

Calculating the key, distributing it, and keeping it secure are the three most important aspects of every cryptographic system. Cryptanalysis, on the other hand, is the act of deciphering ciphertexts or particular codes by studying them and breaking them. With the use of a flaw in the way the system was implemented, it may be exploited to break through the cryptographic security mechanism and gain access to the encrypted text. A cryptographic system must be safe even if everything regarding the system, except the key, is known to the public, according to Kerckhoff's principle [37]. The strength of a public key cryptography system is dependent on the management of its key, as only the "key" information is kept secret from an intruder or an attacker. Chebyshev polynomials are the foundation for chaotic MAPs, which make it possible for communicative entities to reach a critical agreement. [38] (The theory of chaos) This subfield of mathematics is concerned with the behavior of dynamical systems that are extremely sensitive to the starting circumstances that they are subjected to. wherein A dynamical system is a system in which a function explains the time dependency of a point in a geometrical space. This may be done in a dynamic system by using a function. The following is how the Chebyshev polynomial is defined [39]:

cos(n) is a function that may be represented in the form of the polynomial cos().

$$\cos(n\theta) = T_n * \cos(\theta) \dots\dots\dots 8$$

$$\cos\big((n+1)*\theta\big) = 2*\cos(n\theta)*\cos(\theta) - \cos\big((n-1)*\theta\big)$$

$$T_{n+1}\cos(\theta) = 2*T_n\cos(\theta)*\cos(\theta) - T_{n-1}\cos(\theta)$$

$$T_{n+1}(x) = 2*x*T_n(x) - T_{n-1}(x) \quad \dots\dots\dots 9$$

The Chebyshev polynomial in T_n-x) is shown to be a polynomial in 'X' degree 'n' by Equation 9. One may utilize the semigroup feature of Chebyshev polynomials to

accomplish authentication [1] by following the steps below.

$$T_n(x) = 2 * x * T_{n-1}(x) - T_{n-2}(x) \ldots\ldots\ldots 10\,, \quad n \geq 2$$

In our prior work [1], we used the semi-group property of the Chebyshev polynomial [43] to offer authentication between communication entities. This is demonstrated in the equation below.

$$T_n(x) = 2\,x * T_{n-1}(x) - T_{n-2}(x) * (\text{mod } N) \ldots\ldots\ldots 11, \quad n \geq 2$$

In equation (11), it is impossible to compute the value of 'n' with the provided values of T n(x), X, and N. This is a property that is known as a Chaotic Maps-Based Discrete Logarithm issue. Where N is a large prime integer and X is in the range X ϵ (- ∞,+∞), it is impossible to compute the value of 'n'.

This characteristic is also known as the Chaotic Maps-Based Discrete Logarithm issue. It states that in the above equation (12), it is impossible to compute the value of 'T nm (X)' using the supplied values of T n(x), X, and N, in addition, to T_m (X).

$$T_m(T_n(X)) = T_n(T_m(X)) = T_{mn}(X) * (\text{mod } N) \ldots\ldots\ldots 12, n \geq 2$$

We tested both RSA and Chaotic Maps in an identical environment, which consisted of a dual-core 2.33 GHz CPU, 2 GB of DDR2 RAM, and a 160 GB hard drive capacity. We then evaluated the two programs' total performance. We performed a variety of experiments with the prime number size, which could be as big as 1024 bits, and we assessed the amount of time it took to compute using each method. As can be seen in figure 2, the outcomes of our research make it clear that the computing capacity of chaotic maps is noticeably less than that of RSA. This is shown to be the case. The amount of time spent calculating affects the performance of the network since it increases the latency from one end of the network to the other, eats up energy, and consumes space in the buffer.

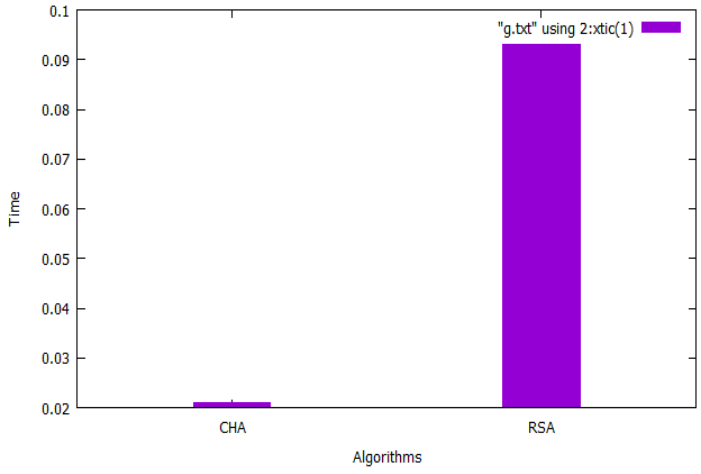


Fig 2 presents a comparison of the amount of processing time required by RSA and Chaotic Maps.

4.3 Proposed Two Party Authentication

Any form of communication between nodes in a distributed system such as a MANET needs to begin with the authentication of both parties. Due to the environment's limited resource availability, the authentication process that every node in the network must go through with its neighboring nodes is cumbersome. To provide authentication across different communication entities, two different node authentication methods have been developed and suggested [40]. Nevertheless, the complexity of the calculation is increased by these approaches more than necessary.

We proposed a low-overhead authentication approach for communicating entities termed chaotic MAPs. This method would use chaotic MAPs. Resistance to man-in-the-middle attacks, guessing data integrity attacks,

compromised session keys, and authenticated key exchanges are some of the security criteria that this method must fulfill to be considered secure. In addition, the operating mechanism of the scheme should be straightforward enough that the authentication procedure may be carried out by any node.

Traditional cryptographic techniques offer authentication based on a password, which may be obtained by registering with a reliable institution or through a secure channel. Additionally, a study [41] revealed the two-party authentication in MANETs via a hybrid crypto token that was received from a TTP (trusted third party) and constructed with the RSA, ECC approach. The cost of computing RSA is significantly higher than the cost of computing chaotic Maps, as was demonstrated in the previous section. Therefore, we recommend a "public key cryptography-based mutual authentication for MANETs," in which chaotic Maps serve as the underlying authentication mechanism. This is because chaotic Maps are more secure than traditional authentication methods. The suggested technique includes several phases.

Every node in the network is aware of its own identity since TTP provides each node with the same security token.

We assumed that every node acquired the token from TTP via way of a secure path, and a significant portion of our mechanism is based on chaotic maps, which are the fundamental building block for the idea of communicating nodes.

Table: **1**

X	Integer value
$k = T_t(x)$	Public key
PW	Password
ID_i	Identity of TTP
$H(.)$	Hash function

Every node in the network is responsible for computing both the public and the private keys using the chaotic maps equation.

Imagine a situation in which two nodes, A and B, are exchanging data. A and B each have a public and private key pair with the associated IDs. A and B are communicating with each other.

For a node to exchange data with a node in a reliable and risk-free way, the former must initially send the latter the authentication request that is detailed below.

Node 'A' is responsible for selecting a prime number and determining its worth using the other two nodes in the initial phase of the process. The next step is for a node to transmit the information to the node.

Second, after the message from the node has been successfully received. The node first decrypts using the key that it was given by TTP, and then it checks to see if the value of in the message that it has decrypted is the same as the value that it was given by TTP. If both of these requirements are satisfied, the node will recognize that the other party may be trusted as a genuine node. By computing the message's hash value and comparing it with a predetermined value, the node can identify whether or not the message has been altered while it was in transit. If both of the values are the same, then the node knows for certain that the message has not been altered in any way.

Third, the node selects a huge prime number and computes its value along with and, and, and. This completes the calculation.

The next step is for a node to transmit the information to the node.

Fourth, after having received the message from the node. It

calculates the value based on the message that it has received. After then, Node uses the key to decrypt the, and then it checks to see if the value of " in the message that it has decrypted is the same as the value it acquired from the TTP. If both of these requirements are satisfied, the node will recognize that the other party may be trusted as a genuine node. By computing the message's hash value and comparing it with a predetermined value, the node can identify whether or not the message has been altered while it was in transit. If both of the values are the same, then the node knows for certain that the message has not been altered in any way.

Fifth, after the two nodes have agreed upon the same session key, they will use it to encrypt and decode any future communications between them. This will take place once they have determined the key.

An explanation of the technique in its entirety may be found in the appendix.

4.4. Detection and prevention of false misbehaving nodes

If a malicious node purposefully misbehaves by making a false report that its neighbor node is not forwarding data or that data has not reached its intended destination, then the malicious node is acting in bad faith. We can prevent this sort of harmful behavior by utilizing an algorithm, which is a very effective tool.

Algorithm 5 :- False misbehavior node detection and prevention

1. Every destination node set predefined time interval (τ) for tracking the received packets
2. Initiate the tracking counter (S_τ) to count the number of packets received during the time interval
3. Create the Acknowledgement packet with fields as $\{(\tau, S_\tau\}$
4. Add the session key get from section 4.2 to Acknowledgement packet
5. Compute message digest as $d = H(\text{Ack Xor Sesion Key})$
6. destination unicast the packet $(\text{Ack} + d)$ to source through routing path
7. Source node validate the Acknowledgement packet by computing message digest by appending its key to received message

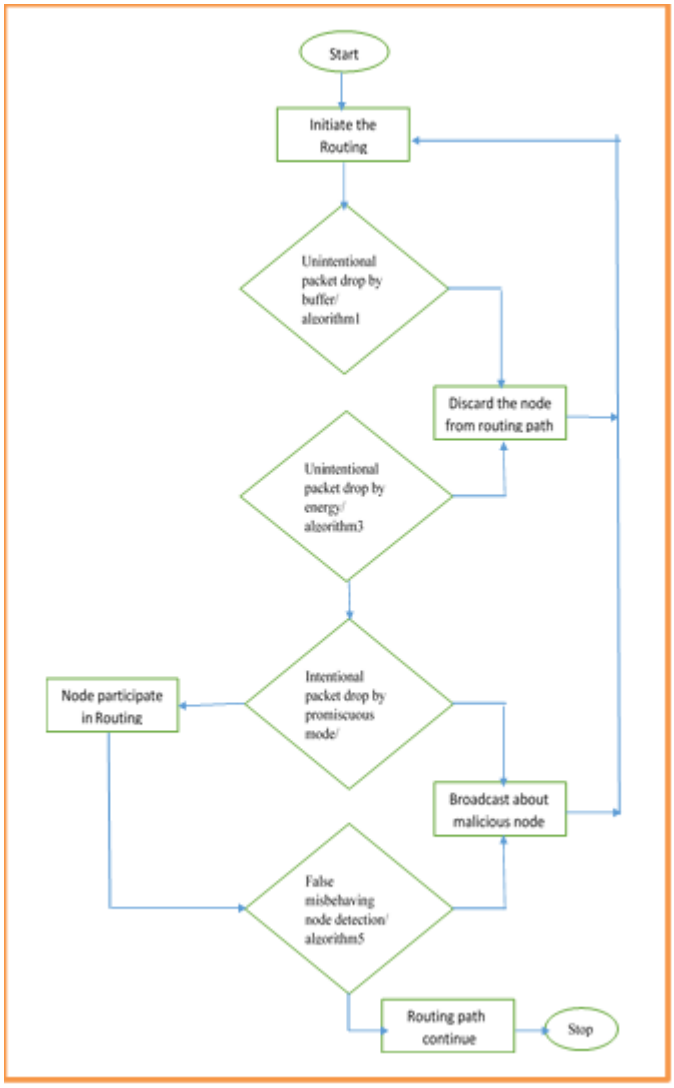
   $$d^1 = H(\text{Ack Xor Sesion Key})$$

   And compare the value with received (d)

   $$d^1 = d = ?$$
8. If both not matches then unintentional misbehaving node present in a communication path
9. Otherwise communication is proceeds

5. Proposed work

Our implementation of the AODV routing protocol is based on pre existing work, however we use algorithms 1 and 3 to remove maliciously misbehaving nodes from the routing path. Algorithms 4 and 5 in our study eliminate the deliberately disruptive nodes from the routing path. Workflow diagram for the proposed project:



6. Performance Evaluation

To test out how well our proposed model would work, we ran simulations in the NS2.34 simulator. The values used for the simulations are listed in table1. In our simulation, we take into account a dynamic number of nodes using a mobility model based on randomly generated waypoints and a 20 m/s pause period. Every node has an IEEE 802.11 MAC card capable of 2 Mbps data rates and a battery that can last for 10j of energy. The range of the radio transmission is fixed at 250m. Three hundred milliwatts is the receiving power while 600 watts is the transmitting power. Finally, source nodes produce 512-byte-packet CBR traffic. We averaged the results from three different simulation scenarios over the course of a thousand seconds.

Table 2. Simulation parameters of MANETs

Network Parameters	Values
Simulation Duration	1200 s
Number of Nodes	10 to 90
Link Layer	Logical Link
MAC	802.11
Mobility	Random way point

Network layer	SKA, ACK, Proposed
Communication	Two-Ray Ground
Queuing	Drop-Tail priority
Battery	20j
Traffic	Constant bit rate
Area of Network	1000m x1 000m

Implementation of Two Party Authentication

In the beginning, we evaluate the suggested RSA-based two-party key agreement model against an existing two-party agreement model based on RSA in terms of the amount of computing overhead. We examined the performance of the network with regard to end-to-end latency after implementing the RSA and chaotic maps-based key agreement between communicating entities in the underlying routing algorithms [42], such as AODV, AOMDV, and DSR. owing to the fact that the amount of time spent computing an algorithm directly affects the amount of time it takes communicating entities to complete a transaction. Delay is one of the most important factors to consider while trying to obtain improved performance or deliver quality service in a network. In particular for real-time traffic such as speech or multimedia transmission, a delay of less than a few seconds is required; otherwise, the quality of the communication traffic is inadequate. Our analysis in the prior section makes it abundantly evident that the processing time required by chaotic maps is far less than that required by RSA in particular in a static identical environment.

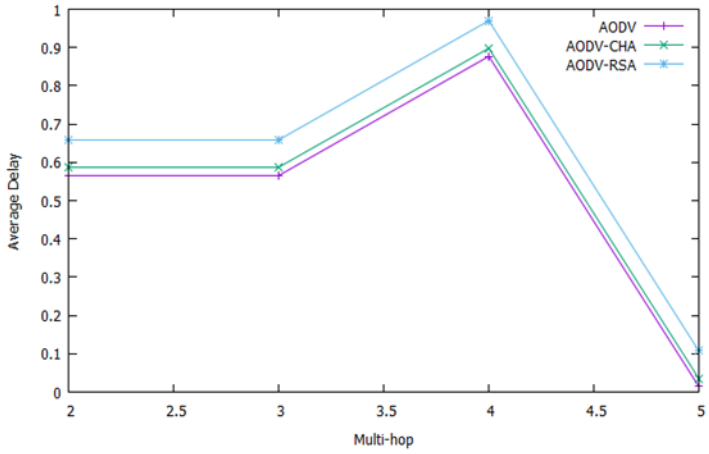


Fig 3: The AODV end-to-end latency when using the RSA and the chaotic map

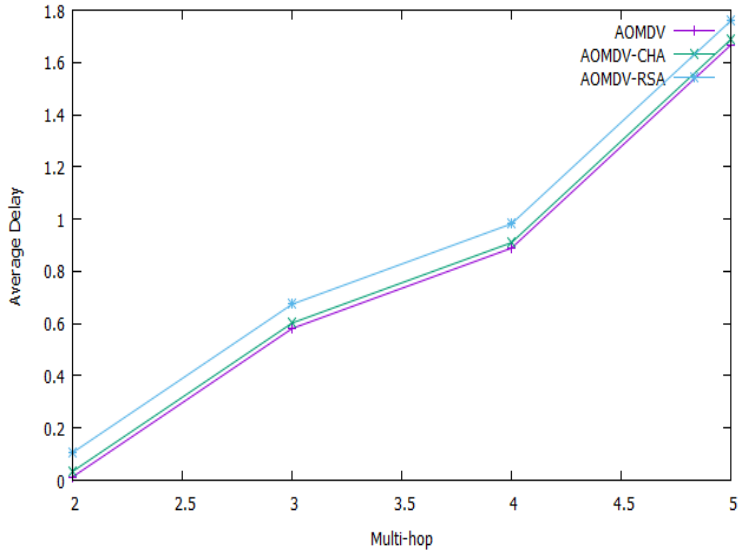


Fig 4: The end-to-end latency of the AOMDV protocol when using the RSA and the chaotic map

7. Results Discussion

Peer-to-peer networking is at the heart of MANETs, which means that individual nodes are responsible for directing traffic. As a result, the load on each node is quite high because each node is performing the task of a router. This means that each node is receiving packets, deciding what to do with each packet based on the protocol, and then forwarding the packets. Each of these steps requires the node to use its time, energy, and memory. Yet, providing security is the duty that is the most difficult to do and is very desirable owing to its characteristics; nonetheless, it must have a minimal impact on the performance of the network. Authentication is the most straightforward and hassle-free approach to achieving security in MANETs. So, it is possible to establish authentication using RSA and Chaotic Maps by having authenticated key agreements between the entities that are talking with one another. As a result, we have carried out an analysis of the RSA and Chaotic Maps-based key agreement in the MANETs setting. Figures 3, 4, and 5 make it very evident that the overhead of RSA is greater than that of the Chaotic Maps Key Agreement Protocol. When compared to the RSA algorithm key agreement, the computation time required by Chaotic Maps is significantly lower. As a consequence of our findings, we have demonstrated that a chaotic map is the most suited alternative to RSA when it comes to providing security for MANETs. According to the findings of the study [10], an adversary is unable to compute the chaotic maps authentication key in polynomial time. This was determined from a security point of view. Figure 5: Delay from beginning to conclusion of DSR measurements for RSA and chaotic maps The environment of wireless ad hoc networks can easily be penetrated by malicious actors. In order to offer security in the most practical manner possible, establishing authentication between communication entities through authenticated key agreements is the best option. Cryptographic methods such as RSA and Chaotic Maps are utilized in order to carry out the authentication process. In a setting with the same conditions, the computational cost of RSA is higher than that of Chaotic Maps. Under the context of MANETs, this has a direct bearing on the end-to-end delay, and therefore on other network factors like energy and buffer, and processor. We have come to the conclusion that authenticated key agreement provided by Chaotic Maps is the greatest alternative to RSA with regard to enhancing network performance while maintaining an adequate level of security.

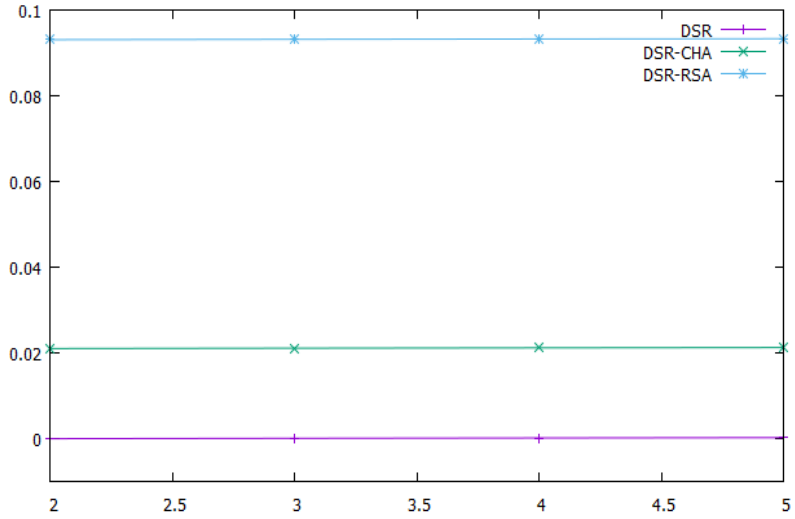


Fig 5: Delay from beginning to conclusion of DSR measurements for RSA and chaotic maps

7.1 Proposed Performance analysis

The performance of our proposed work is compared with that of an existing secure knowledge algorithm, an acknowledgment-based algorithm, and a monitor-based algorithm with regard to the percentage of packets that are delivered, the amount of time that it takes from beginning to end, and the overhead. Calculation of performance is done while variable nodes are present, along with intentionally misbehaving nodes and unintentionally misbehaving nodes.

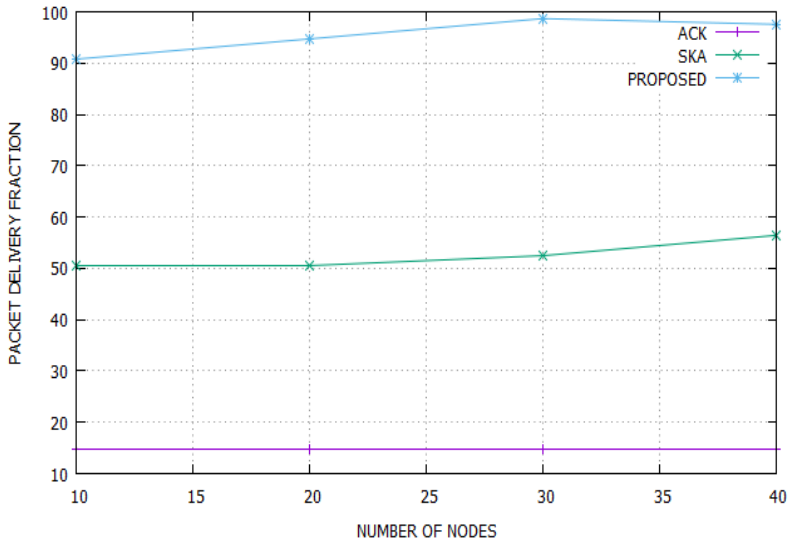


Fig 6 compares the proposed work's packet delivery fraction to that of existing ways

Our primary objective is to evaluate the performance of protocols in a network that contains both intentional and unintentional misbehaving nodes with regard to the percentage of packets delivered, the amount of overhead caused by routing, and the amount of time it takes to get from the beginning to end. Figure 6 demonstrates that the work that was recommended performed well since it locates the routing path by avoiding both purposeful and unintended misbehaving nodes in the beginning. On the other hand, the performance of the secure knowledge technique is lower since it eliminates the nodes that are accidentally misbehaving at a later stage. The acknowledgment-based solution had the worst performance because there was no provision for the elimination of the idea of unintentionally misbehaving node detection and prevention.

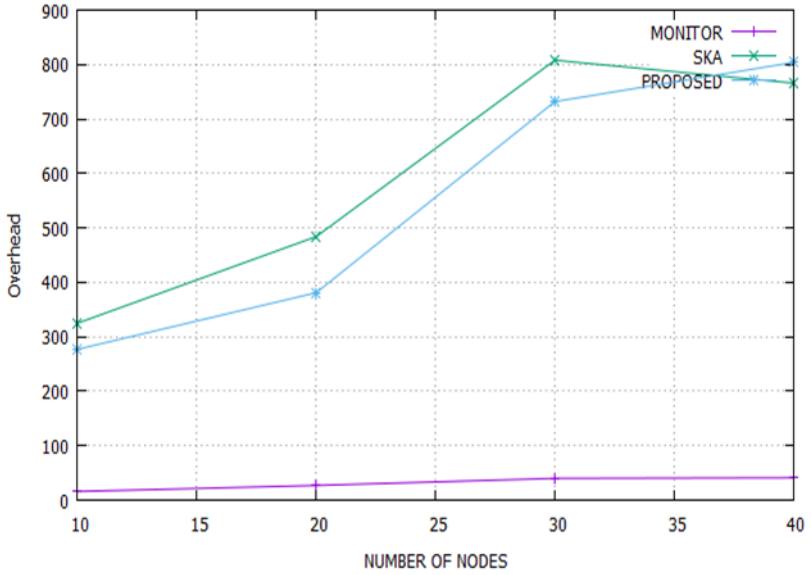


Fig 7 shows the suggested work routing overhead in contrast to the currently used methods.

As can be seen in figures 7 and 8, the proposed work will have a lower overhead for routing and a shorter delay than the present SKA. Because SKA makes use of a variety of control messages, it can determine whether a packet has been lost owing to factors such as energy, buffer, or TTL timeout.

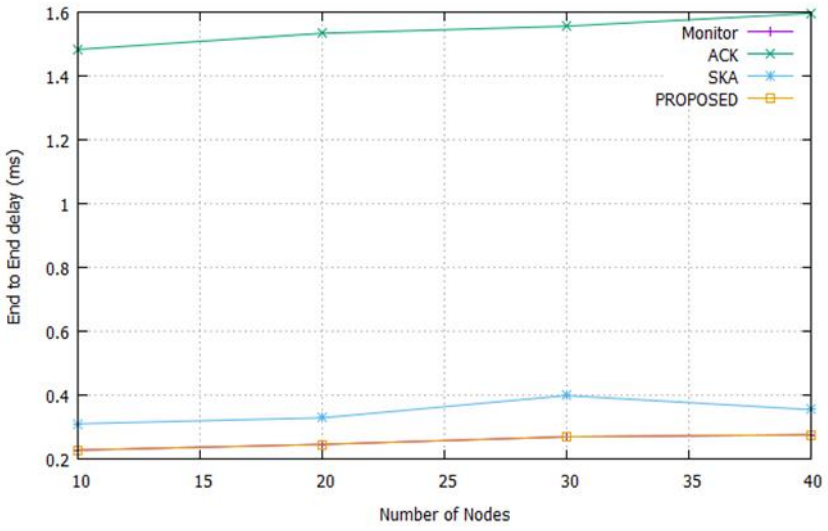


Fig 8. Delay from beginning to finish of the proposed task is compared to the existing ways

The work that is being proposed eliminates the unintentionally misbehaving node from the routing path by taking into account the packet drop that occurs as a result of limited resources. In addition, promiscuous monitoring helps reduce the impact of the intentionally misbehaving node. The authenticated key agreement helps discover and eliminate malicious false nodes that are misbehaving. By using chaotic maps, we are able to simplify the process of reaching critical agreements.

8. Conclusion

Due to its adaptability and self-organization, MANET is well-suited for use in mission-critical settings. One of the most important needs and difficult tasks is ensuring secure communication. Data communication makes use of a tool called routing. When designing a MANET, it is assumed that the nodes inside the network will work together to provide a successful path. Packet drops by individual nodes are a leading cause of routing misbehaviour, however limited network bandwidth is another possible cause. We classified the nodes that drop packets into two groups: those that do so intentionally, by engaging in harmful activities, and those that do so unintentionally, due

to limited resources. In light of the ubiquity of malicious behavior and the limitations imposed by resource availability, we provide a recommendation in this study for a routing approach to lessen the risk of packet loss. The work that is being proposed not only reduces the impact of misbehaving nodes, but also improves network performance in contrast to the techniques that are already in use.

References

[1] Sana, Afreen Begum, Farheen Iqbal, and Arshad Ahmad Khan Mohammad. "Quality of service routing for multipath manets." In 2015 International Conference on Signal Processing and Communication Engineering Systems, pp. 426-431. IEEE, 2015.

[2] Siddiqua, Ayesha, Kotari Sridevi, and Arshad Ahmad Khan Mohammed. "Preventing black hole attacks in MANETs using secure knowledge algorithm." 2015 International Conference on Signal Processing and Communication Engineering Systems. IEEE, 2015.

[3] Sultana, Towheed, Arshad Ahmad Khan Mohammad, and Nikhil Gupta. "Importance of the Considering Bottleneck Intermediate Node During the Intrusion Detection in MANET." In Research in Intelligent and Computing in Engineering, pp. 205-213. Springer, Singapore, 2021.

[4] Mohammad, Arshad Ahmad Khan, Ali Mirza Mahmood, and Srikanth Vemuru. "Intentional and unintentional misbehaving node detection and prevention in mobile ad hoc network." International Journal of Hybrid Intelligence 1, no. 2-3 (2019): 239-267.

[5] Mohammad, Arshad Ahmad Khan, Ali Mirza Mahmood, and Srikanth Vemuru. "Energy-Aware Reliable Routing by Considering Current Residual Condition of Nodes in MANETs." In Soft Computing in Data Analytics, pp. 441-452. Springer, Singapore, 2019.

[6] Mohammad, Arshad Ahmad Khan, Ali Mirza, and Srikanth Vemuru. "Cluster based mutual authenticated key agreement based on chaotic maps for mobile ad hoc networks." Indian Journal of Science and Technology 9 (2016): 26.

[7] Mohammad, Arshad Ahmad Khan, Ali Mirza, and Mohammed Abdul Razzak. "Reactive energy aware routing selection based on knapsack algorithm (RER-SK)." In Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2, pp. 289-298. Springer, Cham, 2015.

[8] Mohammad, Arshad Ahmad Khan, Ali Mirza, and Srikanth Vemuru. "Analytical Model for Evaluating the Bottleneck Node in MANETs." Indian Journal of Science and Technology 9 (2016): 31.

[9] Mohammad, Arshad Ahmad Khan, Ali Mirza, and Srikanth Vemuru. "Energy aware routing for manets based on current processing state of nodes." Journal of Theoretical and Applied Information Technology 91, no. 2 (2016): 340.

[10] Abdul, Arif Mohammad, Arshad Ahmad Khan Mohammad, P. Venkat Reddy, Praveena Nuthakki, Rakesh Kancharla, Rahul Joshi, and N. Kannaiya Raja. "Enhancing Security of Mobile Cloud Computing by Trust-and Role-Based Access Control." *Scientific Programming* (2022).

[11] Mohammad, Arshad Ahmad Khan, Et Al. "Efficient Mobile Sink Location Placement By Residual Status In WSN To Enhance The Network Lifetime." Journal of Engineering Science and Technology 16.6 (2021): 4779-4790.

[12] Sastry, M. Kiran, Arshad Ahmad Khan Mohammad, and Mohammad Abdul Arif. "Optimized Energy-efficient Load Balance Routing Protocol for Wireless Mesh Networks." *International Journal of Advanced Computer Science and Applications* 12.8 (2021).

[13] Arshad Ahmad Khan Mohammad, Towheed Sultana" Evaluation Of Performance Degradation Due To The Presence Of Bottleneck Intermediate Node In Communication Path In Manet" International Journal of Engineering and Advanced Technology (IJEAT)

ISSN: 2249-8958 (Online), Volume-8 Issue-4, April 2019

[14] Mohammad Mohammad, Aleem Mohammed and C Atheeq, “Prior Bush Fire Identification Mechanism Based on Machine Learning Algorithms” International Journal of Artificial Intelligence and Applications, Vol.13, No.4, July 2022

[15] C Atheeq, M Munir Ahamed Rabbani, “CACK—A Counter Based Authenticated ACK to Mitigate Misbehaving Nodes from MANETs” Recent Advances in Computer Science and Communications 12, 2021.

[16] C Atheeq, M Munir Ahamed Rabbani, “Secure Data transmission in integrated internet MANETs based on effective trusted knowledge algorithm” Indian Journal of Science and Technology. Vol. 9 No. 47, Dec 2016

[17] C. Atheeq, Ahamed Rabbani, “Secure Intelligence Algorithm for data transmission in Integrated Internet MANET” International Journal of Computer Science and Applications, Vol. 14, No. 2, pp. 142 – 163, 2017

[18] Krishna, N. ., R., A. ., John, N. M. ., & Kurian, S. M. . (2023). Training and Classification of PCA with LRM model for Diabetes Prediction . International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 08–15. https://doi.org/10.17762/ijritcc.v11i4s.6302

[19] Prof. Barry Wiling. (2017). Monitoring of Sona Massori Paddy Crop and its Pests Using Image Processing. International Journal of New Practices in Management and Engineering, 6(02), 01 - 06. https://doi.org/10.17762/ijnpme.v6i02.54

[20] Sharma, R., Dhabliya, D. Attacks on transport layer and multi-layer attacks on manet (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 5-11.