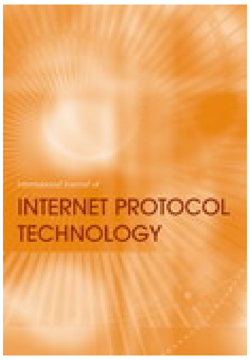


International Journal of
INTERNET PROTOCOL TECHNOLOGY

International Journal of Internet Protocol Technology

ISSN online: 1743-8217 - ISSN print: 1743-8209
https://www.inderscience.com/ijipt

RAD: reinforcement authentication model based on DYMO protocol for MANET

Rushdi A. Hamamreh, Mohammed R. Ayyad, Mohammed Abutaha

DOI: 10.1504/IJIPT.2023.10054906

Article History:

Received:	06 August 2020
Last revised:	17 June 2021
Accepted:	19 June 2021
Published online:	23 March 2023

RAD: reinforcement authentication model based on DYMO protocol for MANET

Rushdi A. Hamamreh* and
Mohammed R. Ayyad

College of Engineering,
Al-Quds University,
Jerusalem, Palestine
Email: rushdi@staff.alquds.edu
Email: mayyad@itce.alquds.edu
*Correspondence author

Mohammed Abutaha

College of Information Technology and Engineering,
Palestine Polytechnic University,
Hebron, Palestine
Email: M_abutaha@ppu.edu

Abstract: This article aimed to develop a new model based on DYMO protocol where a modification was proposed to route discovery and route maintenance processes. In route discovery process we made an authentication process between the nodes by using MD5 hashing algorithm, then we used reinforcement learning to improve the route maintenance process based on machine learning approach. At the end we used Diffie-Hellman key management to exchange the secret key to encrypt and decrypt the data between source *S* and destination *D*. When we tested the proposed protocol, the results show improvement in the performance of MANETs, despite the little increased in the end to end delay in comparison with DYMO protocol. This is due to the overheads in authentication and encryption processes.

Keywords: MANET; DYMO; authentication; reinforcement; security; encryption; hashing; key distribution; nodes; path.

Reference to this paper should be made as follows: Hamamreh, R.A., Ayyad, M.R. and Abutaha, M. (2023) ‘RAD: reinforcement authentication model based on DYMO protocol for MANET’, *Int. J. Internet Protocol Technology*, Vol. 16, No. 1, pp.46–57.

Biographical notes: Rushdi A. Hamamreh received his MS degree in Computer Engineering from the Saint Petersburg State Technical University in 1998 and his PhD degree in Distributed Information Systems and Networks Security from the Saint Petersburg State Technical University in 2002. Currently, he is an Associate Professor of Computer Engineering in the Department of Computer Engineering at Al-Quds University, Jerusalem.

Mohammad R. Ayyad received his BE degree in Computer Engineering from AlQuds University, Jerusalem, Palestine in 2008. He also received Master degree in Computer Engineering at the same University in 2020. Currently, he is a Head of Technical Support Department in AlQuds University.

Mohammed Abutaha received his PhD degree in Information Security from Nantes University, France in 2017 and MSc degree in Information Technology from Palestine Polytechnic University, Hebron, Palestine in 2011. Now, he is an Assistant Professor in the College of Information Technology and Computer Engineering. He has many researches and papers in information security field.

1 Introduction

MANET is a collection of mobile nodes; that construct an impermanent network without need of the help of centralised unit as in ordinary networks, and it have a node that connected to a sink and power source. These nodes doesn't have a wide range of transmission, so every node look for support from the neighbouring nodes in transmitting packets, in this case all nodes acts as transmitter and receiver, in this way, if any two nodes want to communicate together and they have been in same range, they can communicate directly, else, they need a node in the middle which will act as a router between these nodes to help them in communication (Gupta et al., 2018). The primary objective for MANET is to respond to the difficulties of the dynamically evolving topology, and create a proper and effective communication route between any two nodes with minimum cost tracking and the least bandwidth use. The issue with the development of routing protocols is not easy, because the environment of ad hoc networks present new difficulties that are not exist in traditional network. A series of routing protocol were created to resolve this issue, and the amount of these protocols continues to increase daily (Gupta et al., 2011).

1.1 Features of MANET

Most important features of MANET can describe as follows:

Autonomous terminal: Mobile nodes in MANET are autonomous nodes; each node can act as client or router, in other words, beside that node can act as sender or receiver, it can also perform as mediator between two other nodes to help in communication between two nodes, which found in wide range.

Distributed operation: Owing the decentralisation in MANET, the control of the operation is distributed for all nodes; each node in MANET must have the ability to sending and receiving data, path discovery and secure the discovered path.

Dynamic network topology: Nodes in MANET are moving rapidly, so MANET does not have fixed topology and the connections between nodes change in different periods, this mean that nodes will be disconnected from neighbouring nodes and search for new connection to other nodes in the new position (Naghshegar et al., 2008).

Fluctuating link capacity: Wireless connection nature has high bit error rate, one path can used in multiple sessions, and this will cause a noise, fading and interference in the channel used in communication between nodes and it will provide less bandwidth compared with wired network (Kumar et al., 2004).

Light weight terminals: Nodes in MANET usually have limited power, small memory size and small CPU processing ability, devices with these characteristics need a special optimised algorithm to deal with that limited resources.

1.2 Security requirements in MANETs

MANET security is considered a crucial requirement to protect the data exchanged throughout the nodes. In order for a MANET to be secured we should maintain security during different stages including linking, routing, data transmission and data forwarding stages. Numerous security requirements are applied providing different security solutions including:

Confidentiality: It implies that data get to nodes that have been approved to access it.

Integrity: It gives security of message packets while it passes between the nodes in the route.

Availability: A node must be able to gives all its responsibilities without take care of the security condition in that node.

Non-repudiation: The sender cannot deny that it sent the message, and receiver cannot deny that it received the message.

Authentication: It is important to prove that all participants in transmission process are real and not malicious by detecting their identities.

Authorisation: It is a process to determine the permissions for nodes to access the network resources.

Anonymity: The identity for the node must be private, it is not allowed for any node to distribute the other nodes identities.

1.3 Main attacks in MANET

Black hole: In this attack, a malicious node acts like a black hole and drop all the packets that pass through it. When this malicious node acts as connecting node in a network path, then it will separate this network into two disconnected networks.

1 *Grey hole*: It is a kind of black hole attack, but in this attack, the packets will drop selectively, not in random manner (Dhende et al., 2018).

2 *Man in the middle*: The malicious node takes its place between two nodes, and then it snoops the data transmitted between them. In some cases, the malicious node impersonates the source to send packets for the destination, or impersonates the destination to replay to the source (Sowah et al., 2019).

1.4 Routing protocols

Routing consists of many processes depending on different rules (protocols) and steps (algorithm), the main task of routing is giving information needed by routing algorithm to determine decisions about the path which must be selected (Ferdaus and Salihi, 2014).

There are numerous types of routing protocols, and these protocols can be divided into three categories according to the way they work.

1.4.1 Proactive routing protocols

Also called table driven protocols that attempt to make every node maintains one or more list called routing tables. These lists are periodically modified. When any update occurs in the networks topology, the node broadcasts a message to whole network. This message contains the information for the updates that happened in the network. Nevertheless, since up-to-date data is retained, it has higher overhead costs, as a consequence; network performance may be affected, but the real information is provided on network availability (Venkatesan et al., 2014).

Proactive routing protocols provide the paths for each node in MANET in advance, so whenever the path is needed it will be ready for use, this will decrease the average delay per packet, main proactive routing protocol is Destination-Sequenced Distance-Vector (DSDV) protocol.

In this protocol, each node in the network has a routing table that includes all routes to any destination and the number of hops in each route. Any updates happen in the network will cause a broadcast for new routing tables.

By using sequence number to tag each node, DSDV guarantee loop freedom in the network. Each sequence number determines the freshness of any route, so the highest sequence number shows the latest route to destination.

To decrease the number of broadcasting the updates, DSDV use two ways to define the update messages: full and incremental dump. In case of full dump, the message will include all information about routes, but in incremental dump, the message includes just the changed information since last dump (Mahdipour et al., 2009).

1.4.2 Reactive routing protocols (on demand)

These protocols also called on demand protocols. Only if needed by the destination node will use this routing protocol to create routes.

The way that these protocols work making it an efficient way to minimise the use of bandwidth by reducing the links that created for same routes compared with proactive routing protocols.

These protocols include two main processes: route discovery and route maintenance.

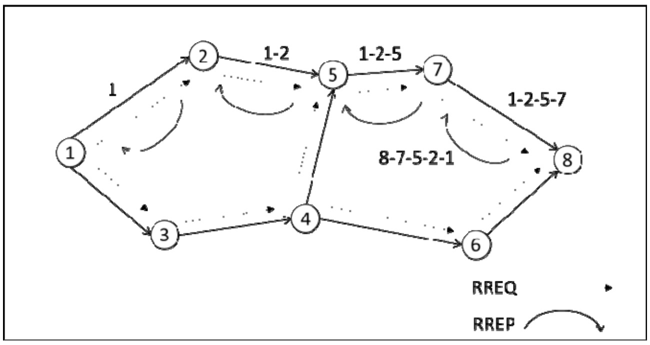
1 *Route discovery process*: In this process, every node prepares a list for direct neighbouring nodes and their link cost which called routing table, this list is created by sending Route Request (RREQ) packet for the neighbouring nodes, the nodes that receive that packets will send back a Route Replay (RREP) packet (Johnson and Maltz, 1996).

2 *Route maintenance process*: This process makes modifications in routing tables for the nodes when it detects a break in any surrounding links. This node sends Route Error (RERR) packets to the nodes, when other nodes receive that packet, they will start a new route discovery to avoid that break link (Khatri et al., 2010).

Main reactive routing protocols are: DSR, AODV and DYMO.

Dynamic source routing (DSR): It uses an algorithm called source routing, this routing protocol tries to find a path to destination node only when needed. The packet header has a list of all intermediate nodes that included in the path to the destination node, the main difference between this protocol and the table driven protocols is that every intermediate node will forward that packet to its next hop, which listed in the header, and there is no need to return to its routing table. In addition, if there are any updates in the network, there is no need to broadcast that updates, which will save the bandwidth for the network and battery power for nodes (Alaparthi et al., 2019). Figure 1 shows the RREQ and RREP packets in DSR.

Figure 1 DSR route discovery process



Ad-hoc on-demand distance vector (AODV): AODV tries to enhance DSR by including routing tables in the nodes, in this way data packets do not have to contain routes. AODV uses small messages defined as HELLO messages to determine local connectivity, which will shorten the time required to respond to routing requests, and activate updates if required. Sequence numbers are assigned to routes and routing table entries to take place of old cached routing entries (Liu et al., 2013).

Dynamic MANET on demand (DYMO): (DYMO) is a Dynamic MANET dependent on the demand, it is also defined to as successor of AODV or ADOVv2, it is a combination between DSR's characteristics and AODV's attributes, also, it is type of on demand (reactive) protocols.

DYMO protocol attempting to give a compelling and straightforward protocol, unlike AODV protocol, there is no need for unnecessary HELLO messages; process is purely based on sequence numbers assigned to all the packets. It is a reactive routing protocol that computes unicast routes on demand or when required. It employs sequence numbers to ensure loop freedom (Hamamreh and Salah, 2018).

Like all other reactive routing protocols, there are two processes should be utilised to discover paths, which are route discovery and route maintenance. The DYMO route discovery is very similar to that of AODV except for the path accumulation feature.

1.4.3 Hybrid routing protocols

Proactive and reactive protocols have advantages and disadvantages; Hybrid routing protocols combine the advantages of both.

Such protocols have ability for adaptation and respond to the area and location of the source and destination. This will split the network into different zones and then observes the location of source and destination.

Proactive routing protocols are used for data exchange if the source and destination are found in same zone, else, if source and destination did not found in same zone, then the reactive protocols are used (Al-Dhief et al., 2018).

1.5 Authentication techniques

Authentication is the process of determining whether someone or something is, in fact, who or what it declares itself to be. Authentication techniques give access control for network by checking if a node's credentials match that one in routing table for the neighbouring nodes. One of the main authentication techniques that can used in MANET is hashing.

A hashing algorithm is mathematical algorithm that transforms data of arbitrary size to a hash of a fixed size. It was created to be a single direction function, no way to invert this function so it uses to improve the security for packets during the path of network. Hence, the message is intended for a particular recipient only and the packets will be secured against attacking.

One of the main Hashing techniques is Message Digest 5 (MD5), the process of MD5 is to transform a message with variable length to fixed length message equal to 128-bit.

The original message M is divided into 512-bit blocks (16 words that have 32-bit), the length of message must be divisible by 512, so the algorithm uses padding to meet that condition. Padding is done by adding single '1' bit followed by the amount of '0' bits to produce a message have length in bits congruent to 448 modulo 512 (Shakya and Karna, 2019).

1.6 Encryption techniques

Encryption is the process of encoding a data in such a way that only authorised members can access it by decrypt that data, to encode the data, the sender and receiver need to share a key. Advanced Encryption Standard (AES) algorithm and Data Encryption Standard (DES) algorithm are examples for encryption techniques. These algorithms are fast but the problem is that they need to key-exchange process between the sender and receiver only without share it with the entire network.

There are many techniques to manage the secret key distribution in encryption process, most of them need trusted third party to distribute that key, MANET doesn't have that third party, so it needs a unique technique that can manage the key distribution in these circumstances which is Diffie-Hellman key exchange. Here is an explanation for the Diffie-Hellman key exchange process in points:

a) Source (S) and Destination (D) agree on Prime Number (P) and Generator of the prime number (G).

b) S randomly chooses private key X_S and D randomly chooses private key X_D.

c) S calculates Y_S, which is equal to G^{Xs} mod P.

d) S sends Y_S to D.

e) D calculates Y_D, which is equal to G^{Xd} mod P.

f) D sends Y_D to S.

g) S computes $K_S = Y_D{}^{Xs}$ mod P.

h) D computes $K_D = Y_S{}^{Xd}$ mod P.

i) Then, K is the shared secret key.

After this process S uses K to encrypt the data and send it to D, when D receives the encrypted data it can decrypt it using K.

1.7 Reinforcement learning

Reinforcement learning is an area of machine learning; it is about deciding to take the right action to maximise the rewards in a specific situation. In reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience (i.e., unsupervised leaning) (Szepesvári, 2010).

Main points in Reinforcement learning procedure (Bhatt, 2021):

- Precondition: the input must be the initial state where the model will begin.
- Post condition: there are many solutions for any problem, which called the output.
- Initial state rewards (R) = 0.
- Training: the training is depending on input, after the model generates the output, the user can make a decision to reward or punish the model depending on the output result.
- Every time the user takes a decision, the model continues to learn.
- The best route will depend on the maximum reward.

Types of Reinforcement: There are two types of Reinforcement:

1 *Positive*: positive reinforcement is giving a motivating action to the user when he takes the right decision; this will make this decision more likely to happen in the future.

2 *Negative*: negative reinforcement happens when the user takes the wrong decision; the result will be removing some stimulus or may be apply some punishments (penalties) on the use.

2 Previous work

Most important process in MANET is create path between source (S) and destination (D) to transmit the packets, this path must be the shortest to decrease the transmission time between S and D. In addition, this path must be secure to prevent any attack from malicious nodes, that shortest path which include security techniques will be the best path. To decrease the time for finding the best path, many researchers use some authentication techniques to create authenticated nodes, here are some examples of that:

Yang and Yoo (2014) proposed an authentication technique to provide secure communication by increasing the reliability of the nodes. They use cluster structure for authentication technique, they made a certificate authority by using the proposed techniques and cluster head, it will manage authentication information of member nodes. They confirmed the performance of the proposed technique by experiments (Yang and Yoo, 2014).

The research by Sharma and Gangal (2016) showed an effective node authentication structure for MANET that can readily provide a means for the malicious node to be detected. The primary alternatives in the research are use of Trusted Third Party (TTP), public-private key pair and authenticating malicious nodes (Sharma and Gangal, 2016).

Verma et al. (2016) suggested secure mechanism for authentication of nodes in the MANET, they proposed an authentication protocol based on digital signature with hash function to create a certificate that can exchanged between nodes.

Sen (2010) showed key exchanged protocol between nodes in MANET, this protocol based on multipath communication. After made the simulation, the results showed the effectiveness of that protocol even in network that have large number of malicious nodes (Sen, 2010).

The research by Subu et al. (2012) used a system with a trust model and SHA-1 key encryption. This system tries to detect the malicious nodes in MANET. The experiences for the nodes in network help to build a trust value. By using specific hashing techniques, which called SHA-1 the efficiency of trust, system is enhanced (Subu, 2012).

Lu and Pooch (2005) proposed an authentication protocol by using one-way hash chain to provide effective authentication for communications between any two nodes in MANETs. They also made an analysis for the security properties and performance. The results for that analysis are the protocol incurs low overhead penalty and achieves a trade-off between security and performance (Lu and Pooch, 2005).

The research by Tembhurkar and Singare (2015) proposed efficient initial access authentication protocol, it uses roundtrip messages to distribute the key between nodes. The main idea in this protocol is to provide a secure path between nodes to pass messages in safe way (Tembhurkar and Singare, 2015).

Nitnaware and Thakur (2016) suggested new strategy to detect and prevent black hole attack in DYMO, This research work attempts to develop a mitigation algorithm to avoid and prevent genuine nodes from malicious attack (Hamamreh and Salem, 2016).

3 Proposed model

The proposed protocol is based on DYMO protocol for route discovery and maintenance processes, and has two main phases: The Authentication phase, which is carried out by MD5 hashing and Diffie-Hellman algorithm for key management and it will use AES for Encryption process. The Reinforcement learning phase, which is carried out by rewards and punishments principles.

3.1 Model architecture

Our (RAD) proposed protocol consists of the following:

1 *Source node*: Which is the node that wants to send a message to another node (Destination node).

2 *Authentication*: In this phase, two main processes are carried out, hashing the MAC address for nodes that will participate in the transmission process using MD5 hashing algorithm, the second process generates and distributes the private key that will be used in AES process for encryption by using Diffie-Hellman key exchange algorithm and this will be found in the sink node.

3 *DYMO protocol*: Is the utilised protocol to find the route that the message will pass to reach the destination node.

4 *RRER*: The route request error will be the trigger to start the route maintenance process and apply the punishment for the malicious node.

5 *Reinforcement learning*: In this phase, two main actions are (carried out) performed, punishment for the malicious and selfish node by exclude it to revocation list, and rewards for the nodes that proves expected good behaviour for many cycles.

6 *Destination node*: Which is the node that will receive the message from Source node.

3.2 Model algorithm

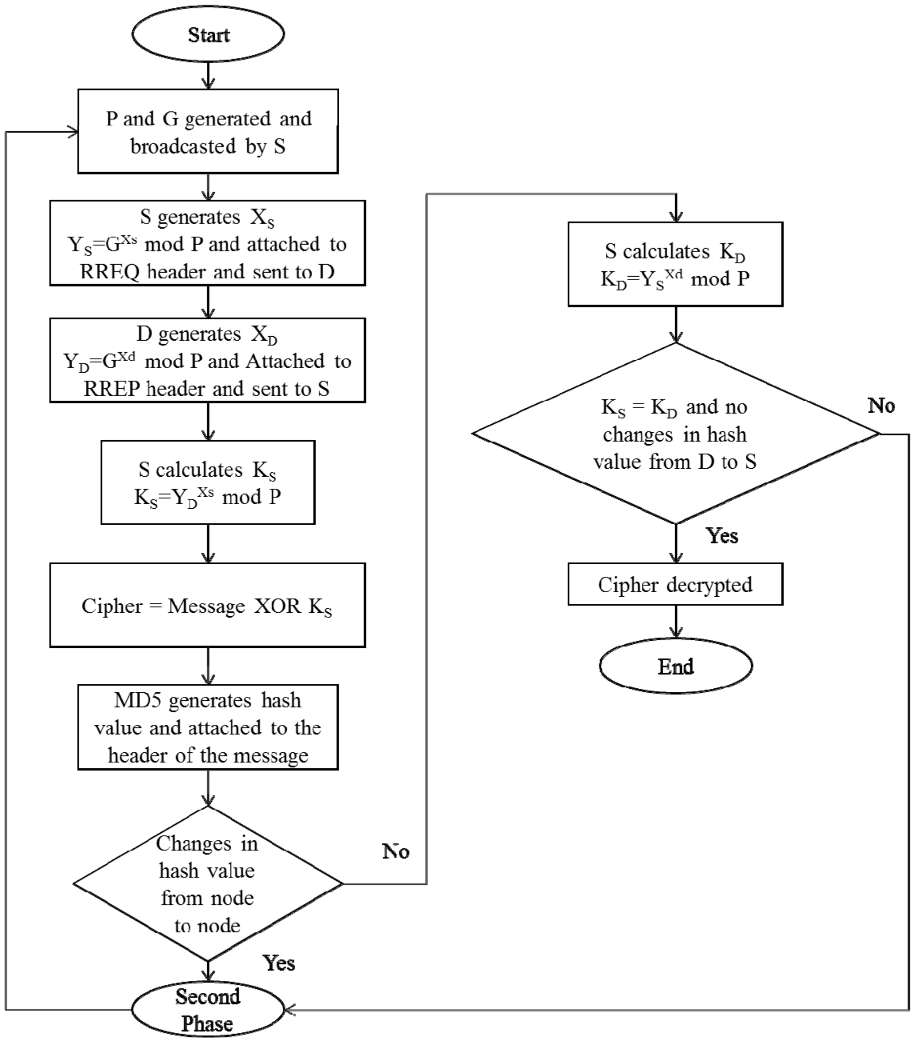
Proposed protocol (RAD) has two main phases: Authentication phase, and reinforcement learning phase.

3.2.1 The authentication phase

It has nine sequential steps to create safe transmission process. The steps of this phase are as follows:

- *Step 1*: Source *S* needs to contact Destination *D*, *S* generate two random numbers: Prime Number *P* and Generator of the prime number *G*, and then *S* Broadcasts *P* and G in the network.
- *Step 2*: *S* generates a number called private key X_S and use it to calculates Y_S using this formula: $Y_S = G^{Xs} \bmod P$, and then attaches Y_S to header of RREQ packet and sends it to *D*.
- *Step 3*: The route discovery process in DYMO protocol will start to determine the best route to *D* by broadcasting RREQ to the network.
- *Step 4*: D generates private key X_D, and calculates Y_D using this formula $Y_D = G^{Xd} \bmod P$, and then attaches Y_D to RREP packet and sends it back to *S* using the discovered route.

Figure 2 Flowchart for the authentication phase


Start
P and G generated and broadcasted by S
S generates X_S $Y_S=G^{Xs}$ mod P and attached to RREQ header and sent to D
D generates X_D $Y_D=G^{Xd}$ mod P and Attached to RREP header and sent to S
S calculates K_S $K_S=Y_D^{Xs}$ mod P
Cipher = Message XOR K_S
MD5 generates hash value and attached to the header of the message
Changes in hash value from node to node
No
Yes
Second Phase
S calculates K_D $K_D=Y_S^{Xd}$ mod P
$K_S = K_D$ and no changes in hash value from D to S
No
Yes
Cipher decrypted
End

- *Step 5*: *S* calculates number called encryption key K_S to use it in encryption process that will be 128-bit size, which is equal Y_D^{Xs} mod *P*.
- *Step 6*: AES divide the packet into blocks with 128-bit size and use this formula to get the encrypted message EM: EM =128-bit block XOR K_S.
- *Step 7*: MD5 hashing algorithm will use MAC Address for *S* and *D* to generate hash value to check the authentication for end-to-end transmission, and generate another hash value by using the MAC Address for intermediate nodes to check the authentication for node-to-node transmission.

In Source node, MD5 will make a summation for *S* and *D* MAC addresses and generate a hash value, this process will repeat in *D*, the hash values that generated in *S* and *D* will compared, if any changes appear between the two hash values, *D* will define as malicious and will be added to revocation list, this list contains the nodes that showed odd behaviours, this list will be found in the node itself. Same process will apply between node to node in the route.

We use MAC Address in hashing between *S* and *D*, because it is fixed and unique for each node

- *Step 8*: The hash values will attach to header of the encrypted message and send it to the discovered route.
- *Step 9*: Every time the message reaches for one of the intermediate nodes, the hash value for this node will compared with the hash value for previous node, if any change is detected, then RRER will generated and the second phase will start and path discovery process will start again.
- *Step 10*: When *D* receives the message, the hash value between *S* and *D* will compared to make sure that *D* is authenticated, then *D* will calculate K_D, which is equal Y_S^{Xd} mod *P*, and uses it to decrypts the message using AES in case that $K_D = K_S$, else the node will define as malicious node. Figure 2 shows a flowchart explaining the steps of the Authentication phase.

3.2.2 The reinforcement learning phase

It has five sequential steps to deals with malicious nodes. The steps of this phase are as follows:

- *Step 0*: *R* is the rank number for the malicious node, when any node enters the revocation list, *R* for this node will be equal 0.
- *Step 1*: When any node generates RRER, the next node will be recognised as malicious node, and then it will be added to revocation list.
- *Step 2*: The malicious node will exclude from the previous route, but it has chance to participate in new route.
- *Step 3*: Every time the malicious node gives a good behaviour in another route, *R* will be incremented.
- *Step 4*: We test performance of protocol when *R* equal 1, 3 and 5, then we find that R=3 will be the best value the cause less delay and more throughput as shown in Figures 3 and 4, so If *R* = 3, then the malicious node will exclude from the revocation list and will be recognised as normal node.

Figure 3 End to end delay (ms) for variable value of *R*

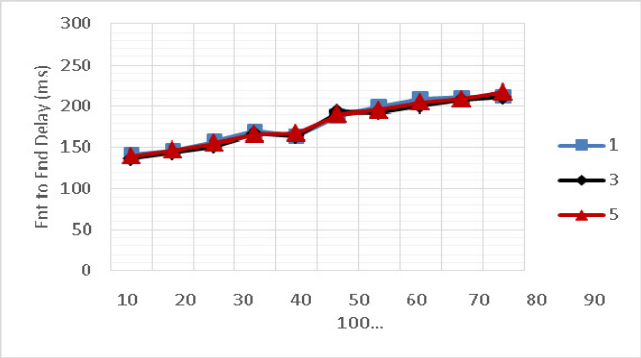


Figure 4 Throughput (Kbps) for variable value of R

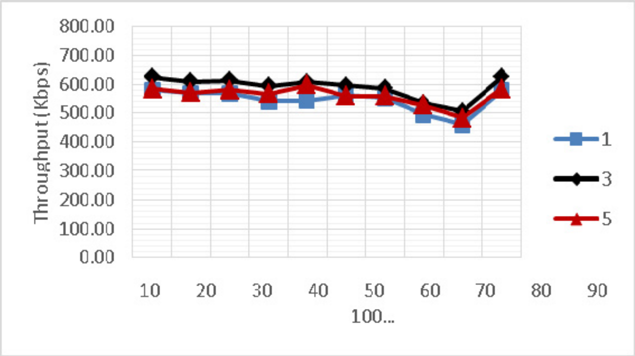
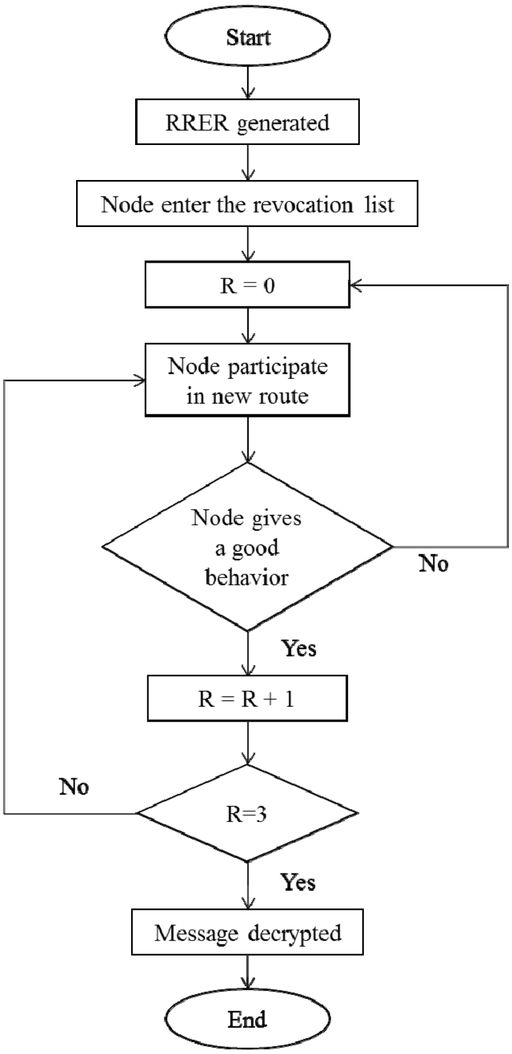


Figure 5 shows a flowchart explaining the steps of the reinforcement learning phase.

3.3 Mathematical model for reinforcement learning

In this section, we will present the mathematical model of our proposed protocol; we will use this model to calculate the rewards and the maximum utility that node can get from using the reinforcement learning in DYMO protocol.

Figure 5 Flowchart for reinforcement learning phase


Start
RRER generated
Node enter the revocation list
R = 0
Node participate in new route
Node gives a good behavior
No
Yes
R = R + 1
No
R=3
Yes
Message decrypted
End

The reward for taking action a in state ST is $R(ST,a)$:

$$R(ST,a)=\sum_{st=0}^{st'}P(ST'|ST,a)\,r(ST,a,ST')$$

where

- ST is set of states of paths.
- A is set of actions a (like selection of the path in DYMO).
- $P(ST'\,|\,ST,a)$ is the probability of transmission from *ST* to *ST'* given action a (success in transfer data to neighbour nodes in the DYMO path)

The maximum utility for taking action *a* in state *ST* is *U*(*st*):

$$U(ST)=\max\left(R(ST,a)+\sum_{st=0}^{st'}T(ST,a,ST')\,U(ST')\right)$$

where

- $T(ST,a,ST')=\sum_{ST=0}^{ST'}P(ST'\,|\,ST,a)$

4 Performance evaluation

we will create the simulation by using a software called 'the Network simulator version two (NS-2)'.

4.1 Simulation environment and parameters

In this article, we run NS-2 in windows 10 using VMware Workstation, we used the version ns 2.35 on Ubuntu 12.04 LTS 64 bit operating system to simulated our new proposed protocol and compare it with DYMO protocols, all that work on computer have Intel core i5 – 7200 CPU @ 2.5 GHz, the installed memory is 8 GB.

All scenarios were applied on an area simulating 3000 m × 1000 m based on Table 1 where Mobility model is Random way point and simulation time 900 s.

When the simulation begun, one node was selected as source node and another one was the Destination node. During the simulation, the selected nodes were not change.

The simulation was applied to RAD protocol depending on change the number of nodes from 10 to 100 nodes without change the speed that equal 30 m/s, and was applied when the number of nodes was 100 and the speed changed from 5–30 m/s, at the end we applied the simulation when the number of nodes is 100, the speed is 30 m/s and the time start from 0 to 900 s. The timing will start from 0, and the message will generate randomly by DYMO protocol library in NS2 using Poisson Distribution.

Table 1 Simulation parameters

Parameters	*Value*
Simulation tool	NS-2.35
Operating system	Ubuntu 12.04
Channel type	Wireless channel
No. of nodes	10, 20, 30, 40, 50, 60, 70, 80, 90, 100 nodes
Antenna model	Omni directional
Interface queue size	50 packets
Transmission range	250m
Speed	5, 10, 15, 20, 25, 30 m/s
Simulation time	900 s
Mobility model	Random way point
Examined protocol	DYMO, RAD
Simulation area	3000 m*1000 m
Bandwidth	2 Mbps

4.2 Performance metrics

In this simulation, we used three parameters to test the performance of the proposed protocol by made a comparison between the RAD protocol simulation results and the DYMO protocol simulation results.

a) *Packet delivery ratio (PDR)*: It represents the ratio of the number of packets received successfully in destination node to the number of packets sent by source node per unit of time.

$$PDR = \frac{\sum_{i=1}^{N} Ri}{\sum_{i=1}^{N} Si}$$

R: Received data packets

S: Sent data packets

b) *End to end delay*: It represent the average time needed to deliver data packets from source node to destination node including all delays in route.

$$E2ED = \frac{\sum_{i=1}^{N} (TRi - TSi)}{\sum_{i=1}^{N} Ri}$$

TR: Receiving time

TS: Sending time

R: Received data packets

c) *Throughput*: Is the average number of bits that successfully delivered per unit of time.

$$TH = \frac{\sum_{i=1}^{N} Ri}{\sum_{i=1}^{N} TRi}$$

TR: Receiving time

R: Received data packets

4.3 Simulation result

In this part, we applied the previous environment parameters and get the results, we will show these results and discuss them.

The results compared the behaviour for DYMO protocol and RAD protocol in three scenarios: performance metrics with network size (number of nodes), nodes speed and with the simulation times.

4.3.1 Results of performance metrics when value of R (Rank) equal 1, 3 and 5 vs. network size

We applied the parameters in a network includes variable number of nodes start from 10 nodes until 100 nodes that moves in speed of 30 m/s, as we mentioned (The Reinforcement learning phase) the malicious nodes will stay in the revocation list for *R* = 3 cycles, which is the best value to get maximum throughput as shown in Table 2.

Table 2 Throughput (Kbps) for variable value of R

Number of nodes	*Value of R*		
	1	*3*	*5*
10	579.21	624.47	584.73
20	568.30	609.44	570.42
30	568.08	613.53	579.31
40	540.98	593.98	565.70
50	542.29	607.27	598.35
60	562.20	596.38	557.98
70	553.36	586.83	558.89
80	494.08	533.61	528.20
90	458.68	506.17	482.07
100	578.21	624.47	584.73

When $R = 1$, the number of malicious nodes in the network will increase, then the throughput will decrease. When $R = 5$, the number of malicious nodes in the network will decrease, but the throughput will decrease because the number of available nodes in network will decrease then the rout discovery process will not be effective. As shown in Figure 4 $R = 3$ is the best value to get the maximum throughput.

4.3.2 *Scenario 1: results of performance metrics vs. network size*

We applied the parameters in a network includes variable number of nodes start from 10 nodes until 100 nodes that moves in speed of 30 m/s. Figure 6 shows PDR for DYMO and RAD protocols.

PDR for RAD is better than DYMO when the number of nodes increased over 20 nodes (the size of network increased), but at small number of nodes (less than 20 nodes) DYMO showed slightly better results than RAD.

Figure 6 Packet delivery ratio (%) at speed of 30 m/s by different numbers of nodes

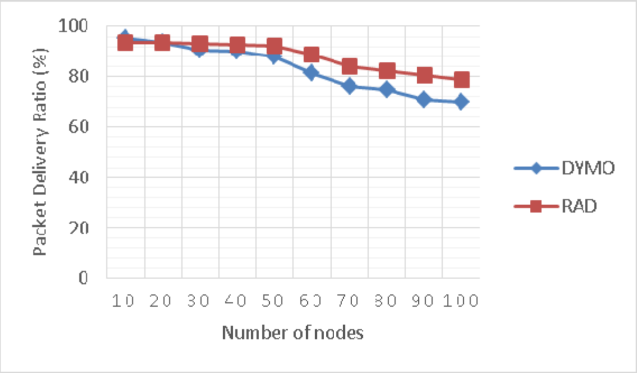


Figure 7 shows throughput for DYMO and RAD protocols. Throughput for RAD is better than DYMO when the number of nodes increased over 50 nodes, when number of nodes less than 50 nodes DYMO and RAD protocols showed almost identical results.

Figure 7 Throughput (Kbps) at speed of 30 m/s by different numbers of nodes

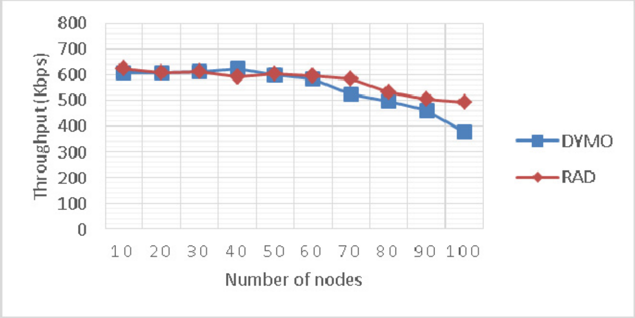
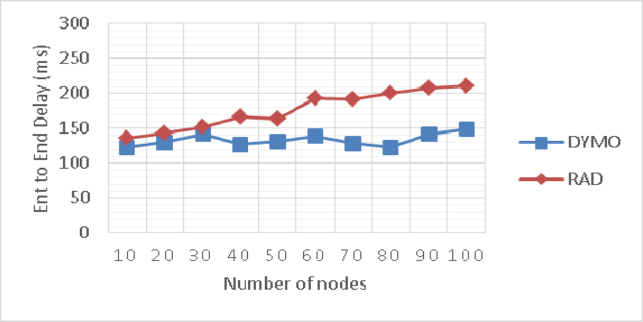


Figure 8 shows End to End delay for DYMO and RAD protocols. DYMO is better than RAD with less delay, this was as a result of cryptography process in RAD, but also the delay increased in DYMO because when the number of nodes increased, the probability of malicious nodes appearance increased, then DYMO protocol needs many route maintenance processes which consumed time.

Figure 8 End to end delay (ms) at speed of 30 m/s and different numbers of nodes



4.3.3 *Scenario 2: results of performance metrics vs. nodes speed*

We applied the parameters in a network includes 100 nodes have different speeds start from 5 m/s until 30 m/s. Figure 9 shows PDR for DYMO and RAD protocols, where PDR for RAD is better than DYMO at speeds over 20 m/s, at speeds slower than 20 the differences in PDRs is slightly small.

Figure 9 Packet delivery ratio (%) at 100 nodes and different speeds

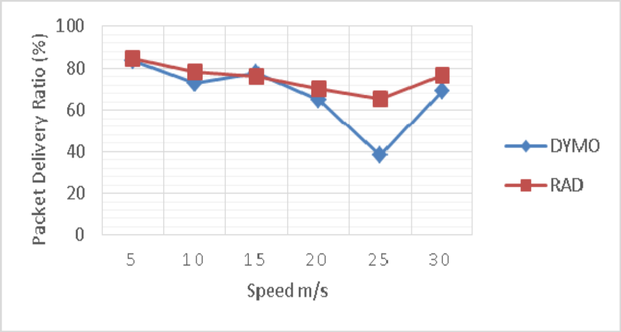


Figure 10 shows throughput for DYMO and RAD protocols. Throughput for RAD is better than DYMO, when the speed increased over 25 m/s the throughput increased for DYMO and RAD, but RAD still have the best throughput.

Figure 10 Throughput (Kbps) at 100 nodes and different speeds

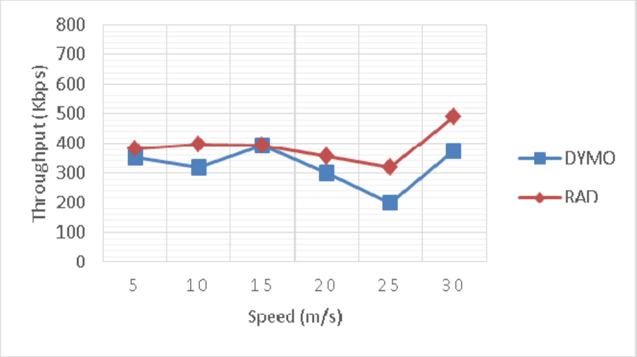
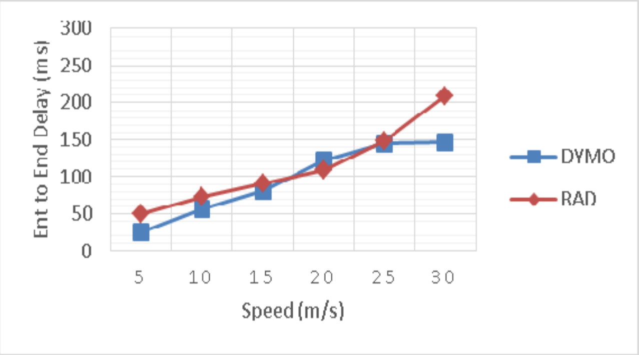


Figure 11 shows End to End delay for DYMO and RAD protocols. DYMO is better than RAD with less delay, as we mentioned before this was as a result of cryptography process in RAD, between the speeds 15 and 25 m/s RAD gives closed results to DYMO

Figure 11 End to end delay (ms) at 100 nodes and different speeds



4.3.4 *Scenario 3: results of performance metrics vs. simulation time*

We applied the parameters in a network includes 100 nodes moved at speed 30 m/s, we notice the behaviour for the nodes during the simulation time. Figure 12 shows PDR for DYMO and RAD protocols, at the beginning of simulation, DYMO gives better results than RAD, after 400 seconds RAD start to gives better results, this improvement in the results for RAD protocol was mainly due to reinforcement learning which improves the nodes behaviour and excludes the malicious nodes with time.

Figure 13 shows throughput for DYMO and RAD protocols. At the beginning of simulation, DYMO gives better results than RAD, after 600 seconds RAD start to gives better results than DYMO due to reinforcement learning.

Figure 12 Packet delivery ratio (%) vs. simulation time at speed of 30 m/s and 100 nodes

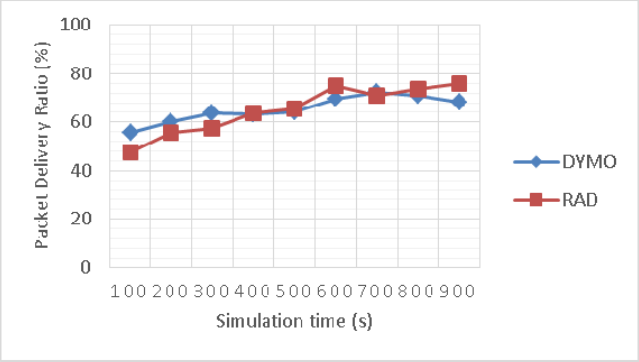


Figure 13 Throughput (Kpbs) vs. simulation time at speed of 30 m/s and 100 nodes

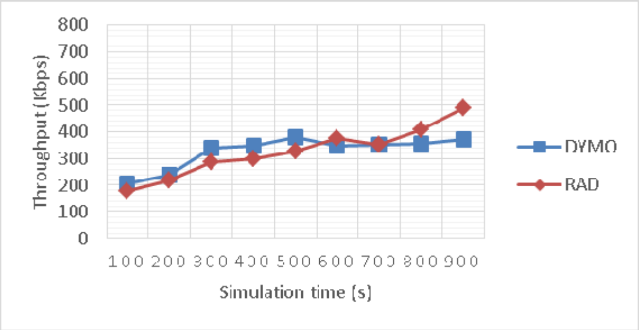
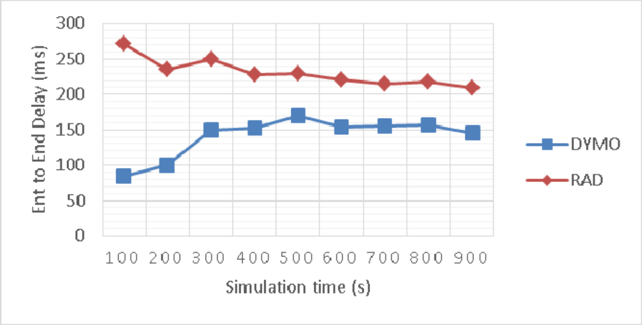


Figure 14 shows End to end delay for DYMO and RAD protocols. DYMO always have less delay time than RAD, but the delay time for RAD decreased with time and increased for DYMO.

Figure 14 End to end delay (ms) vs. simulation time at speed of 30 m/s and 100 nodes



4.4 *Trade-off between security and performance*

To calculate the overhead that resulted from using RAD protocol, the extra average delay that RAD protocol added to network must be determined.

For scenario 1: Extra delay = RAD delay – DYMO delay
= 113.2183 – 96.1033
= 17.115 ms
For scenario 2: Extra delay = 43.022 ms
For scenario 3: Extra delay = 89.789 ms

The results have shown that we can get a significant increase in the encryption strength at a very small overhead for our RAD protocol compare with basic protocol DYMO, we believe this small overhead which was incurred by added time of encryption, decryption and authentication is worth the effort to increase the security level in MANET. As a result of this extra delay and operations, the power consumption will increase for the nodes.

4.5 *Security analysis*

MANET have a unique topology that make it exposed to many attacks that we mentioned before in chapter one, RAD protocol solve some of these attacks like man in the middle and black hole attack.

- *Black hole attack*: RAD protocol try to produce a network with authenticated nodes by using hashing, each node in this network try to act in good way that keep it away from blocked in the revocation list. In black hole attack the malicious node receives a packet and doesn't resend it to the next node in the route, this will break the route and generate a route maintenance process which will increase the end to end delay, also it will consume the bandwidth because many RRERs and RREQs will exchanged between nodes this will decrease the throughput for the network. As shown in the result, RAD protocol increases the average throughput, but the increasing in the end to end delay is occurred due to the encryption process, this process solves the man in the middle attack.
- *Man in the middle attack*: in this attack the malicious node tries to snoops the packets in the route, but when using the encryption process, it is impossible for the malicious nodes to decrypt the cipher packet, to prove that, here is a calculation for the time needed to decrypt the cipher packet:

We consider that the malicious node will use the brute force to decrypt the cipher packet, in this process the malicious node checks all possible keys to find the correct one. Just consider the following:

- Possible number of key combinations for AES – Diffie Hellman = 11.56 × 10^{76} (Bernstein, 2005).
- Supercomputer: *K* Computer
- Speed: 10.51 Pentaflops 10.51 × 10^{15} Flops [Flops = Floating point operations per second]
- Flops required per combination check = 1000 (very optimistic but just assume for now)
- Combination checks per second = (10.51 × 10^{15}) / 1000 = 10.51 × 10^{12}
- Seconds in a Year = 365 × 24 × 60 × 60 = 31,536,000
- No. of Years to crack AES – Diffie Hellman with 128-bit Key = (11.56 × 10^{76}) / [10.51 × 10^{12}) × 31,536,000]

  = (1.099 × 10^{64})/31,536,000

  = 3.484 × 10^{56} years

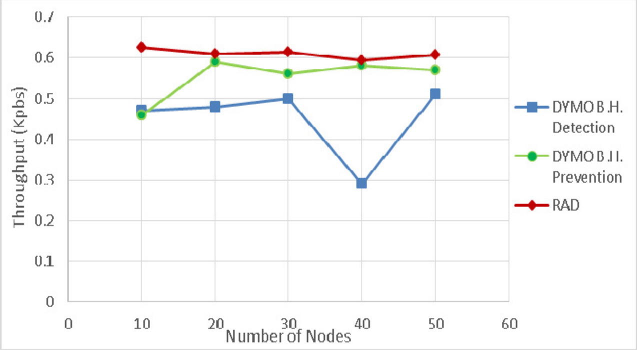
4.6 *RAD performance vs. previous protocols*

In this section we will compare RAD performance results with other protocol that tries to improve DYMO performance.

In 'Black hole attack detection and prevention strategy in DYMO for MANET' (Nitnaware and Thakur, 2016), it compares the normal DYMO with 2 other protocols, black hole detection and black hole prevention.

Figure 15 normalises RAD throughput in scenario one and compare it with the normalised result in that research.

Figure 15 RAD compared with DYMO black hole detection and prevention protocols



The figure shows that RAD protocol improves the average throughput and gives better results than the DYMO black hole detection and prevention protocol.

5 Conclusions

The goal of this work is to develop a new model based on DYMO protocol where a modification was proposed to route discovery and route maintenance processes. In route discovery process we made an authentication process between the nodes by using MD5 hashing algorithm, then we used reinforcement learning to improve the route maintenance process based on machine learning approach. At the end we used Diffie-Hellman key management to exchange the secret key to encrypt and decrypt the data between Source *S* and Destination *D*.

When we tested the proposed protocol, the results show improvement in the performance of MANETs, despite the little increased in the end to end delay in comparison with DYMO protocol. This is due to the overheads in authentication and encryption processes.

References

Alaparthi, S. et al. (2019) 'Dynamic source routing protocol – a comparative analysis with AODV and DYMO in ZigBeebased wireless personal area network', *Proceedings of the 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, India, pp.1042–1046.

Al-Dhief, F. et al. (2018) 'MANET routing protocols evaluation: AODV, DSR and DSDV perspective', *MATEC Web of Conferences*, Vol. 150, pp.1–6.

Bernstein, D.J. (2005) *Understanding Brute Force*, Thesis, The author was supported by the National Science Foundation under grant CCR– 9983950.

Bhatt, S. (2021) *Youplus: Reinforcement Learning.* Available online at: https://www.includehelp.com/ml-ai/main-points-of-reinforcement-learning-in-artificial-intelligence.aspx (accessed on 15 June 2021).

Dhende, S. et al. (2018) 'A survey on black hole attack in mobile ad hoc networks', *Proceedings of the 4th International Conference on Recent Advances in Information Technology (RAIT)*, IEEE, India, pp.1–7.

Ferdaus, J. and Salihi, R. (2014) 'Routing: internet routing protocols and algorithms', *Academic Paper*, pp.1–20.

Gupta, A., Sadawarti, H. and Verma, A. (2011) 'Review of various routing protocols for MANETs', *International Journal of Information and Electronics Engineering*, Vol. 1, pp.251–259.

Gupta, A., Verma, P. and Sambyal, R. (2018) 'An overview of MANET: features, challenges and applications', *Proceedings of the National Conference on recent advancement in Computer science and IT*, Vol. 4, No. 1, pp.122–126.

Hamamreh, R. and Salah, O. (2018) 'An intelligent routing protocol based on DYMO for MANET', *International Journal of Digital Information and Wireless Communications*, Vol. 8, No. 3, pp.195–202.

Hamamreh, R. and Salem, A. (2016) 'Efficient mechanism for mitigating multipleblack hole attacks in MANETs', *Journal of Theoretical and Applied Information Technology (JTAIT)*, Vol. 83, No. 1, pp.156–164.

Johnson, D. and Maltz, D. (1996) 'Truly seamless wireless and mobile host networking. Protocols for adaptive wireless and mobile networking*', Personal Communications*, IEEE, Vol. 3, pp.34–42.

Khatri, P. et al. (2010) 'Performance study of ad-hoc reactive routing protocols', *Journal of Computer Science*, Vol. 6, pp.1159–1163.

Kumar, A., Jacob, L. and Ananda, A.L. (2004) 'SCTP vs. TCP: performance comparison in MANETs', *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, IEEE, USA, pp.431–432.

Liu S., Yang, Y. and Wang, W. (2013) 'Research of AODV routing protocol for ad hoc networks', *Proceedings of the AASRI Conference on Parallel and Distributed Computing and Systems*, Vol. 5, pp.21–31.

Lu, B. and Pooch, U. (2005) 'A lightweight authentication protocol for mobile ad hoc networks', *Proceedings of the International Conference on Information Technology: Coding and Computing*, IEEE, USA, Vol. 11, pp.546–551.

Mahdipour, E., Rahmani, A. and Aminian, E. (2009) 'Performance evaluation of destination-sequenced distance-vector (DSDV) routing protocol', *Proceedings of the International Conference on Future Networks*, IEEE, Thailand.

Naghshegar, A., Darehshoorzadeh, A. and Dana, A. (2008) 'Dynamic topology control scheme in MANETs for AODV routing', *Australasian Telecommunication Networks and Applications Conference*, IEEE, Australia.

Nitnaware, D. and Thakur, A. (2016) 'Black hole attack detection and prevention strategy in DYMO for MANET,' *Proceedings of the 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, India.

Sen, J. (2010) 'A robust and efficient node authentication protocol for mobile ad hoc networks', *Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation*, IEEE, Indonesia.

Shakya, A. and Karna, N. (2019) 'Enhancing MD5 hash algorithm using symmetric key encryption', *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, pp.18–22.

Sharma, N. and Gangal, A. (2016) 'Mobile node aauthentication in MANET using enhanced cluster based aucres algorithm', *Far East Journal of Electronics and Communications*, pp.1–12.

Sowah, R., Mills, G. and Koumadi, K. (2019) 'Detection and prevention of man-in-the-middle spoofing attacks in MANETs using predictive techniques in artificial neural networks (ANN)', *Journal of Computer Networks and Communications*, pp.1–14.

Subu, N., Jayapal, S. and Sridharan, D. (2012) 'A trust system in manet with secure key authentication mechanism', *Proceedings of the International Conference on Recent Trends in Information Technology*, IEEE, India, pp.261–265.

Szepesvári, C. (2010) 'Algorithms for reinforcement learning', *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 4. Doi: 10.2200/S00268ED1V01Y201005AIM009).

Tembhurkar, M. and Singare, Y. (2015) 'Design of an efficient initial access authentication over MANET', *Proceedings of the International Conference on Industrial Instrumentation and Control (ICIC)*, IEEE, India.

Venkatesan, T.P., Rajakumar, P. and Pitchaikkannu, A. (2014) 'Overview of proactive routing protocols in MANET', *Proceedings of the 4th International Conference on Communication Systems and Network Technologies*, IEEE, India, pp.173-177.

Verma, U., Kumar, S. and Sinha, D. (2016) 'A secure and efficient certificate based authentication protocol for MANET', *Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp.1–7.

Yang, H.S. and Yoo, S.J. (2014) Authentication Techniques for Improving the Reliability of the Nodes in the MANET', *Proceedings of the International Conference on IT Convergence and Security*, IEEE, China.