International Journal of Information Security, Privacy and Digital Forensics
Vol. 8, No. 2, December, 2024

SECURED MOBILE AD-HOC NETWORK (MANET) USING LIGHTWEIGHT ENCRYPTION SCHEME

S.A. Akinboro
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
akinboro@unilag.edu.ng

R. Isimeto
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
risimeto@unilag.edu.ng

F.A. Oladeji
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
foladeji@unilag.edu.ng

D.O. Tunwashe
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
tunwashed@gmail.com ,

D.D. Owadokun
Department of Computer Sciences, University of Lagos, Lagos, Nigeria
horwadee3@gmail.com

Abstract –The integration of Advanced Encryption Standard (AES) encryption into the Routing protocol known as Ad Hoc On-Demand Distance Vector (AODV) enhances the security and confidentiality of control messages exchanged between nodes. This work explores the implementation of AES-128 bits encryption in AODV called Lightweight AODV (LwAODV) to protect critical routing information from unauthorized access and eavesdropping. AES-128 bits encryption was used to encrypt the payload Route Request (RREQ) packets. The encryption uses randomly generated symmetric key for each packet. The system decrypts the incoming encrypted payload RREQ packet by retrieving the decryption key from the packet header. The decrypted payload RREQ packet is then processed and used for route discovery. Simulation was performed using network simulator 2 and object tool command language. The LwAODV was benchmarked with the traditional AODV, using packet delivery ration, packet drop ratio, throughput and control overhead as metrics. The result showed that the LwAODV performed almost the same as the traditional AODV. This makes LwAODV better and safer because of the extra layer of security through the lightweight encryption.

Keywords – AES-128 bits, Lightweight, Encryption, LwAODV, AODV, RREQ

1 Introduction

Wireless networks are becoming increasingly popular because they allow users to communicate with one another regardless of their geographical location. Electromagnetic or radio waves can connect several nodes. Connecting wireless and wired networks together is easier. An example of a centralized medium that nodes in a wireless network can connect to one another over is an access point or base station. As base stations are used for both authentication and communication between two nodes, cellular networks are regarded as infrastructure-dependent networks (Kalia and Sharma, 2016).

Ad hoc networks are wireless networks with no centralized infrastructure that operate via multi-hop radio transmission between nodes. Additional subcategories of ad hoc networks are mobile ad hoc networks (MANET) and static ad hoc networks (SANET). The lack of a base station or central controller makes routing more difficult than it is in cellular networks. Because nodes in static ad

hoc networks have fixed geographic locations, host mobility is not possible in these networks. On the other hand, because nodes can freely move within and outside of the network, the topology of a MANET is dynamic and subject to sudden changes. MANET is susceptible to several security risks, such as eavesdropping, identity spoofing, and other attack vectors like sinkholes, wormholes, and black holes (Panda and Pattanayak, 2018).

An eavesdropping hazard arises when unauthorized individuals are able to access or modify data through the usage of wireless communication in MANET nodes. To transfer data to destination nodes, nodes in a MANET rely on one another as there is no central infrastructure to manage connection between nodes. Because of this, a malevolent attacker node can change connection connections or remove data that has been sent. One of the biggest risks to MANETs is the Denial of Service (DoS) assault, which is the name given to this kind of attack. In this attack, a malicious attacker node poses as the one who would be the destination or, in certain situations, as the one who would have the best path to the target. upon arrival of the packet at the fraudulent node, also known as the "black hole node." it becomes discarded (Jhaveri and Patel, 2015). As a result, communication inside this network is impossible. Despite its adaptability, MANETs are vulnerable to security breaches because of open medium, dynamic topology, and decentralized monitoring, among other factors that seriously impair the network's fundamental operation (Rajeswari, Kulothungan, and Kannan, 2016). Effective routing protocols have been developed to meet the particular requirements of these networks, including OLSR, DSR, and AODV (Ur-Rehman and Ullah, 2010).

In order to strengthen MANET security, this research focuses on incorporating a lightweight encryption technique within the AODV route discovery protocol. The goal is to create an encryption system that defends against frequent security risks in an efficient and effective manner. The second section discusses the related works, section 3 provides conceptual framework and algorithms, and section 4 was results and discussions while section 5 is the conclusion

2 Related Work

(Jhaveri and Patel, 2015), suggested a strategy based on the greatest sequence number that rogue nodes might produce. The series of numbering-based detection method (SNBDS) modifies each node's routing table in two distinct ways: "Node Status," which logs a node's activity, and "The final Reply Time," which pulls the destination node's most recent sequence number from the node's latest RREP. This technique considers three different attack types with different spoof routing characteristics. Every node computes an initial value before beginning the routing process. Any node that has its target sequence number value of RREP packets above a threshold is considered suspicious. A decoy query packet with a fake destination location is transmitted to the target node to find out how it is doing and the state of the suspicious node. Upon receiving a bait request packet, a node's status will change to: harmful node replaced by suspicious node, no packages at that node, it is either received or forwarded. Get as far away as possible. Latency and processing cost may be increased by the threshold calculations made for every node. A baiting method based on a distinct node ID was presented by (Kalia and Sharma, 2016). Sending bait queries to all neighbouring nodes is the initial step towards identifying a black hole node. A bait request contains an SSN (source sequence number) and a source ID. When it receives the answer, the transmitting node searches for a reply with a greater DSN than the SSN. This indicates that the response is coming from a black hole since no node in the network has a DSN larger than the SSN of the original node. The source node alerts all neighbouring nodes to the presence of a black hole node in the network. This strategy has a drawback in that if an intelligent blackhole node detects that the source of her RREQ is being requested by another node, it can detect this and refuse to answer. False black hole alarms can be set off by intelligent black hole nodes, which can then utilize them to isolate particular network nodes. The scheme for a collaborative bait detection strategy (CBDS) is the basis of a strategy developed by (Sathya and Priyadharshini, 2016). Three categories exist in CBDS for black hole identification. phase bait, defence by reversal, and reactive defence. The source node uses its ID to make a fake request to one of its randomly chosen neighbours during the decoy phase. The RREPs of the spoof RREQs are utilized to compile a list of questionable nodes for the converse trace stage. Next, to determine the extent to which a malicious node is in the route, nearby nodes enter promiscuous mode. Each of the black hole nodes identified in the network sends a black hole alert to nearby nodes. When using reactive defence, the source node repeats the decoy phase after assessing if the PDR is under a present threshold. This technique's drawback is that it forces the node into promiscuous mode, which not all nodes find acceptable. Certain nodes are vulnerable to promiscuous mode passive attacks because they do not want unauthorized users to intercept their broadcasts. By utilizing the black hole alarm functionality to isolate network nodes, intelligent black hole nodes might begin broadcasting fictitious black hole alerts. The methodology put out by (Singh, Srikanth, and Kumar, 2016) begins by saturating the network with erroneous requests. Every node's answer is regarded as malicious if messages are being sent to the desired node, it may be discovered while doing so

by the malicious node. Because it is a military model, the suggested model includes a localization mechanism that shows the locations of the black hole nodes. This approach has a drawback in that it introduces fake requests into the network, which may cause congestion. According to (Rajeswari, Kulothungan, and Kannan, 2016), the suggested approach depends on a unique kind of node known as a guard node to assist in identifying black hole nodes inside the network. Promiscuous mode nodes that keep an eye on other nodes' actions inside the network are known as guard nodes. Tables kept on guard nodes document node activity on the network. Every node has a trust value that is based on how it behaves in the network; if a node only transmits RREPs and not RREQs, its trust value will decrease. A node will be quarantined or banned if its trust score is less than a certain level. All nearby nodes will receive an alert from the guard node upon detecting the presence of a black hole node. Special node types This system's shortcomings include (preserve nodes) and a large number of preserve nodes throughout the network. Furthermore, there are a lot of tables in this system, which increases overhead. A model that just relies on the valid bits specified in RREP was proposed by (Deshmukh, Chatur, and Bhople, 2016). This model works on the assumption that the attacker node transmits her RREP without knowing which valid bits to send. When the original node gets the RREP, it checks the valid bit. If it is set to 1, it uses that path; if not, it evaluates and discards the RREP from the blackhole node because the attacking nodes attempting to breach the network employ the same protocol and scrutinize it beforehand, one of the model's limitations is its impractical assumption. Thus, upon noticing this legitimate bit, each blackhole intelligent node will transmit her RREP on any request that includes it. the valid bit is set. Blackhole nodes in the network are found using the built-in queries using the suggested model in (Sathish *et al.,* 2016). A source initiates a broadcast of a request that any node on the network can make. Blackhole nodes are those that respond to fraudulent queries. The average answer that the DSN provided for the queries that were posed is kept on file by the node of the origination. A request is sent to a desired node by a source node in this architecture. When a response is received, it compares the response DSN to the average DSN that has been recorded, and the destination node is deemed to be a black hole node. If absolutely nothing else, the knot is typical. Preventive steps are included in the recommended method to decrease the effect of black hole nodes on the network by using digital signatures and trust values. (Kumar and Doohan, 2016) suggested utilizing IDS nodes to find malicious nodes. Developing methods to identify black hole attacks and lessen the real impact of rogue nodes was the primary objective of this effort. According to this method, nodes may identify black hole attacks based on anomalous sequence number values in RREP packets. Every communication between nodes in the network is watched over and intercepted by a dedicated IDS node. A packet containing the ID of the malicious node will be transmitted over the network in an alert packet to block subsequent transactions, should the IDS identify a node response sequence number during monitoring that over the set threshold. This solution has two limitations: rigid limits, which, considering the constantly changing nature of MANETs and the installation of additional IDS nodes, are unsuitable, which may result in significant computing cost. Poor system performance might result from inadequately equipped IDS nodes, elevated routing overhead and the identification of rogue nodes. (Dhende *et al.,* 2017) presented the SAODV model, a model that uses the opinions of its neighbours to identify nodes that are blackholes and gray holes. In SAODV, every node has two table neighbours. two lists: an opinion list (OL) used to categorize nodes based on their network activity and a list (NL) comprising the IDs of nearest nodes. Following receipt of the path Request Response, the source node notifies its neighbour via an Opinion message asking for the assertion made by the node that its path is the shortest. If all other nodes respond with a NO message, then the node is a black hole node. Maybe a node is a Gray hole node in certain circumstances if it answered with a YES message while the other nodes replied with a NO message. Apart than that, the knot is typical. When a black node is found, the network receives a notice alert. The large overhead of space allotted for the OL table and the overhead of exchanged messages are not the only drawbacks of this paradigm. How clever blackhole nodes may transmit incorrect information. What enables them to separate oneself on a regular basis is posed to them. (Ganesh and Tamilselvi, 2019), suggested a productive method for finding the best path to employ for data transmission, avoiding blackhole nodes in the process. The source node chooses neighbouring nodes in this configuration. It can access its one-hop neighbour by utilizing that location as the recipient's address in her RREQ packet and attracts the malicious node. The neighbour's spoof destination address and the public key-encrypted message are included in the RREQ packet that the source node first transmits. The request packet is forwarded by the node if it can find a path to the given destination; otherwise, it replies to the packet with an encrypted message. A black

hole node is unable to send unprotected messages it can only reply to RREQ packets with reconfigured routing information. When the encrypted message is missing from the response packet, the destination node simply throws it away and notifies the original node about the illegal node. Blackhole nodes utilize data regarding the remaining nodes in the network and can influence packet delivery rates and circumvent security measures by refusing to react to fake her RREQs. In order to protect against clever blackhole attacks, (Yasin and Abu, 2018) suggested a technique that makes use of timers and baiting messages. The two stages of the suggested strategy are luring and non-neighbour reaction. During the baiting phase, a bait timer with a random value of B seconds is present on each node. Once the timer reaches B, a bait request is created and submitted with a bogus ID that is generated at random. By default, a black-hole node behaviour is to respond to a route request by claiming to have the optimal path, even in cases when there isn't one. After receiving the baited request, the blackhole node replies to the source node, pretending to have a route; the source node blacklists the blackhole node upon receiving this reply. Thanks to the broadcasting of hello messages, every node in the neighbour reply phase of the non-neighbour reply is aware of the nodes to its right and left. Upon receiving a reply, the origin node checks to see if the black-hole list contains the Node with the shortest Path (NWSP) ID. If it isn't, the ID and those in the neighbour list are compared to determine if it is included in the list. To prevent communication with unknown nodes, the source node discards the reply if NWSP is not a neighbour node. A novel strategy known as "detecting and eliminating black holes" (DEBH) was presented by (Dorri, Vaseghi, and Gharib, 2018). To find malicious nodes, this method makes use of extra blackhole check (BCh) tables and data control messages. Every node keeps track of its neighbours in a BCh table, which is kept up to date by the neighbours' historical actions, there are two fields in the BCh table. H. A Boolean value and node ID "trust table" In the trust table field, a value of '0' denotes harmful behaviour, whereas '1' denotes trust. An intermediate node has to include a BCh table in every RREP packet it transmits back to the source node. Prior to transmission, a safe path is chosen based on each node's BCh table after all answers from intermediary nodes have been received. A data control packet is sent along the path to verify the path's legitimacy when data on the chosen path is received. In the event that a black hole node breaches the route, it consistently misses data control packets, identifies the malicious node, or selects the path. Overhead may grow from this method's extensive usage of control packets. To assist other nodes, every node keeps up her BCh table. During the routing procedure, this may cause more delays.

3 Description of the proposed LwAODV model

In MANETs, the Ad-hoc On-demand Distance Vector (AODV) protocol is a popular reactive routing mechanism. The standard was designed to efficiently establish and preserve routes amongst network hubs, even in scenarios where the network's topology is continuously shifting and the nodes are highly mobile. The basis of AODV is the distance vector approach, which calculates the quickest route between nodes in terms of hops. Because AODV is on-demand, routing only begins when a node needs to send data to a destination node that is not in its immediate neighbourhood, in contrast to typical distance vector protocols. The AODV protocol as shown in figure 1 works as follows:

1. A node broadcasts a Route Request (RREQ) message to a target node in order to transmit data. The RREQ message contains the Internet Protocol (IP) number of the original and destination nodes, a transmitted ID, and an IP address distinctive sequence number.

2. Upon receipt of an RREQ message, a node verifies if it possesses a route to reach the intended node. It transmits the RREQ message to its neighbours after appending its own IP address if it lacks a route. Additionally, by storing the IP address and The original node's serial number, and the node builds a reverse path back to it.

3. A node decides which node has the largest sequence number and discards the others after getting RREQ information for the same node of destination from various sources.

4. A Route Reply (RREP) message is a communication sent back to the original node through the node that is the destination in response to an RREQ message that it has received. The RREP message contains the sequence number, hop count, IP address of the final node, and IP address of the original node.

5. As soon as the RREP information is returned to the initial node, each intermediate node updates its routing record with the new path. When nodes get RREP information for a target node after previously sending RREQ messages to that node, they delete the RREP message.

6. The node that originated the request can now send messages containing data to the node that it

intended since the existing route has been built. If the source node receives a Route Error (RERR) message stating that the chosen path is no longer feasible, sending a new RREQ response will initiate a new route discovery process.

AODV also includes a route maintenance mechanism to handle link failures and changes in network topology. When a node detects a link failure, it sends a RERR message to all the nodes that have a route through it. The affected nodes then use the RREQ message to initiate a new route discovery process.

In the LwAODV protocol, each packet is encrypted using a randomly generated symmetric key. This key is included in the packet header, allowing receiving nodes to decrypt the packet. When a node receives an encrypted Route Request (RREQ) packet, it retrieves the decryption key from the header, decrypts the payload, and processes the RREQ to participate in route discovery. The encryption ensures that only authorized nodes can access and process the packet contents, enhancing the security of the route discovery process in Mobile Ad-Hoc Networks (MANETs). The LwAODV maintains the efficiency of the original AODV protocol while adding a critical layer of security through encryption.

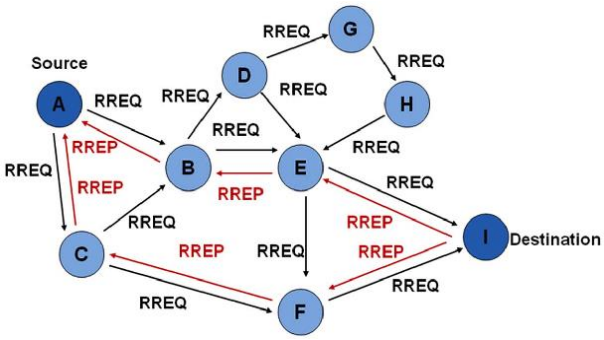


Figure 1: AODV route discovery process

Figure 1 illustrates the route discovery process in the Ad-hoc On-demand Distance Vector (AODV) routing protocol. When a node, referred to as Node A, wants to send data to another node, Node I, but lacks a direct route, Node A broadcasts a Route Request (RREQ) message. This message contains the IP addresses of both Node A and Node I, a transmitted ID, and a unique sequence number.

As intermediate nodes receive the RREQ message, they check their routing tables for a route to Node I. If an intermediate node does not have a route, it appends its IP address to the RREQ and forwards it to its neighbours. This action establishes a reverse path back to Node A by recording the IP address and sequence number. Multiple RREQ messages may reach Node I via different paths, and Node I selects the RREQ with the highest sequence number, indicating the freshest route, while discarding the others.

Node I then sends a Route Reply (RREP) message back to Node A through the chosen intermediate nodes. The RREP message includes the sequence number, hop count (number of hops to reach Node I), and the IP addresses of Node I and Node A. As the RREP travels back to Node A, each intermediate node updates its routing table with the new route. If an intermediate node receives an RREP for a destination after previously sending RREQs, it discards the RREP.

Once Node A receives the RREP, it begins sending data packets to Node I using the established route. If a link failure occurs, the intermediate node detecting the failure sends a Route Error (RERR) message to the nodes using the failed link. Nodes receiving the RERR message initiate a new route discovery process by sending a new RREQ.

3.2 Proposed LwAODV Algorithm

The LwAODV algorithms comprises of four sub-algorithms for sending route request, receiving route request, sending reply and receiving route reply respectively. The lightweight encryption algorithm is AES with 128-bit key sizes.

ALGORITHM 1: ALGORITHM TO SEND REQUEST // SEND RREQ

```
START
    Set key for node
    Generaterandomkey(unsigned char *key)
    {
        int result;
        do {
            result = RAND_bytes(key, AES_BLOCK_SIZE);
        if (result <= 0) {
            memset(key, 0, AES_BLOCK_SIZE);  // Clean up the key by
```

```
            overwriting it with zeroes
        }
    }
    while (result <= 0)
    unsigned char key[AES_BLOCK_SIZE];  // Generate the random key
    generateRandomKey(key);
    std::map<int, unsigned char*>pskKeys;   // PSK key storage
    pskKeys[rq->psk_key] = new unsigned char[AES_BLOCK_SIZE];
    memcpy(pskKeys[rq->psk_key], key, AES_BLOCK_SIZE); // Store the key in the PSK
    key storage
    Encrypt RREQ packet
    void encryptPacket(Packet *p, unsigned char *key)
    {  // Function to encrypt a packet using AES-128 with the given key
        AES_KEY aesKey;
        AES_set_encrypt_key(key, 128, &aesKey);
        AES_encrypt((unsigned char *)p->accessdata(), (unsigned char *)p-
        >accessdata(), &aesKey);
    }
    encryptPacket(p, key);  // Encrypt the packet
    Broadcast it to intermediate nodes
End
```

ALGORITHM 2: ALGORITHM TO RECIEVE REQUEST // RECEIVE RREQ

```
START
    If (receiving node knows the key)
    {
        decrypt the encrypted value
     }
    int pskKey = rq->psk_key;   // Retrieve the PSK key from the packet header
    if  (pskKeys.find(pskKey) == pskKeys.end())
    {
        Packet::free(p);   // Key not found, discard the packet or handle the error
        return;
      }

    void decryptPacket(Packet *p, unsigned char *key)
    { // Function to decrypt a packet using AES-128 with the given key
        AES_KEY aesKey;
        AES_set_decrypt_key(key, 128, &aesKey);
        AES_set_decrypt_key(key, 128, &aesKey);
    }
    decryptPacket(p, pskKeys[pskKey]); // Decrypt the packet using the PSK key
    // ...  // Process the decrypted packet
    Packet::free(p);
        else
    drop packet
End
```

ALGORITHM 3: ALGORITHM TO SEND ROUTE REPLY

```
START
    void sendReply(nsaddr_tdst) { // Function to send a reply packet

        Packet *p = Packet::alloc();// Allocate a RREP packet
        struct hdr_cmn *ch =HDR_CMN(p);
        struct hdr_ip *ih = HDR_IP(p);
        struct hdr_aodv_reply *rp =      HDR_AODV_REPLY(p);
```

```
        aodv_rt_entry *rt = rtable.rt_lookup(dst);
        assert(rt);
        unsigned char key[AES_BLOCK_SIZE]; // Generate the random key
        generateRandomKey(key);
        pskKeys[rp->psk_key] = new unsigned char[AES_BLOCK_SIZE]; // Store
        the key in the PSK key storage
        memcpy(pskKeys[rp->psk_key], key, AES_BLOCK_SIZE);
        encryptPacket(p, key);  // Encrypt the packet
        // Fill out the RREP packet
         // ...
        // Send the encrypted RREP packet
        Scheduler::instance().schedule(target_, p, 0.);
    }
End
```

ALGORITHM 4: ALGORITHM TO RECIEVE ROUTE REPLY // RECEIVE RREQ

```
START
    void recvReply(Packet *p) { // Function to receive a reply packet
        struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
        int pskKey = rp->psk_key; // Retrieve the PSK key from the packet header
    if (pskKeys.find(pskKey) == pskKeys.end())
    { // Find the corresponding PSK key in the storage
        Packet::free(p); // Key not found, discard the packet or handle the error
        return;
     }
    decryptPacket(p, pskKeys[pskKey]);

    // Process the decrypted packet
     // ...
     // Free the packet
    Packet: free(p);
    }
End
```

4.0 Simulation setup

The experiment was performed in ns-2 simulation environment. The parameters are showed in table.1. To efficiently analysis the performance of the protocol, the automate.sh file was used to randomly generate ten mobility scripts which were inputted into the aodv.tcl file. The aodv.tcl file also generates ten network animator files and trace file. The analysis.awk script uses the trace file to compute the average of the performance analysis of the ten nodes and print them on the excel sheet as shown in table 4 and figure 2.

Table 1: Parameters Table

Parameters	Values
Number of nodes	10-100
Speed	10-100m/s
Simulation time	100s
Grid dimension	100m x100m

4.1 Simulation Results and Discussion

The simulation was performed with ten simulation runs with other parameters in table 1 using packet delivery ration, packet drop ratio, throughput and control overhead as metrics. The LwAODV was benchmarked with the traditional AODV protocol.

In tables 2 and 3 the number of packets sent is randomly generated and the number of node is sequential. The results of the evaluation using packet delivery ration, packet drop ratio, throughput and control overhead are shown in tables 2 and 3. Summary of the average results for AODV compared with LwAODV protocol is shown in table 4 and figure 2. The packet delivery ratio is affected by factors such as network congestion, jitter,

delay and unpredictable mobility of nodes. This is responsible for the average packet dropping ratio of 23.187 for LwAODV and 22.656 for AODV. The result of the proposed LwAODV is almost the same as the traditional AODV protocol. This makes LwAODV protocol better and safer because of the extra layer of security through the lightweight encryption.

Table 2 Lightweight encryption simulation result table

Simulation Number	Packet Sent	Number of Nodes	Received Packets	Packet Delivery (Ratio %)	Packet Dropping Ratio (%)	Throughput (Kbs)	Control Overhead
1	1600	10	1096	68.5	31.5	26.52	34
2	1000	20	900	90	10	84.99	4
3	500	30	499.69	99.94	0.06	42.74	109
4	2000	40	928.75	46.44	53.56	85.56	33
5	2600	50	1779.38	68.44	31.56	76.69	42
6	1500	60	1499.06	99.94	0.06	84.88	15
7	800	70	164.5	20.56	79.44	84.62	2
8	1600	80	1192	74.5	25.5	84.24	19
9	2600	90	2600	100	0	76.75	64
10	1600	100	1597	99.81	0.19	84.64	18
Average	1740	55	1225.64	76.813	23.187	65.326	38.1

Table 3: AODV Simulation Result Table

Simulation Number	Packet Sent	Number of Nodes	Received Packets	Packet Delivery (Ratio%)	Packet Dropping Ratio (%)	Throughput (Kbs)	Control Overhead
1	1600	10	1096	68.5	31.5	58.53	58
2	1000	20	`617.5	61.75	38.25	52.42	18
3	500	30	500	100	0	84.92	6
4	2000	40	1267.5	63.38	36.62	53.99	25
5	2600	50	2596.75	99.88	0.12	85.42	95
6	1500	60	1257.19	83.81	16.19	71.01	40
7	800	70	548	68.5	31.5	58.51	9
8	1600	80	1098	68.62	31.38	58.65	42
9	2600	90	2600	100	0	84.62	2
10	1600	100	944	59	41	49.82	130
Average	1740	55	1252.494	77.344	22.656	65.789	42.5

Table 4: Average results of AODV Versus LwAODV

AODV	Avg. Packet Sent	Avg Packet Received	Avg Delivery Ratio	Avg Dropping Ratio	Avg Throughput	Avg Control Overhead
	1740	1252.494	77.344	22.656	65.789	42.5
LwAODV	1740	1225.64	76.813	23.187	65.326	38.1

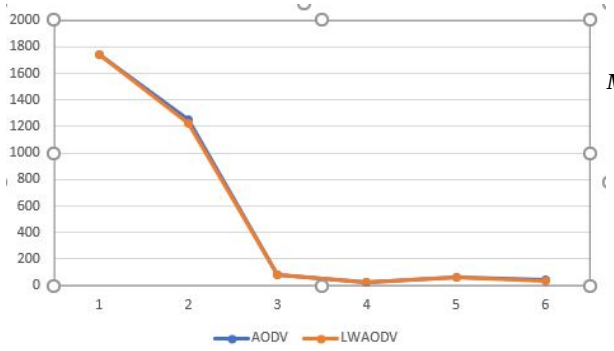


Figure 2.: AODV Avg. VS EAODV Avg. graph

5 Conclusion

The process of applying effective and low-resource cryptographic algorithms that are appropriate for devices with limited memory, processing capacity, and energy resources is known as lightweight encryption. The Advanced Encryption Standard (AES) is one commonly used lightweight encryption technique. AES is a symmetric encryption method that offers effective and secure decoding and encoding of data. The AES algorithm works with fixed-size data blocks, typically 128 bits in size, and supports aligned key sizes of 128, 192, or 256 bits. In this research, we have been able to integrate AES-128 bits encryption into AODV called Lightweight AODV (LwAODV) routing protocol which offers a robust and secure mechanism to protect the confidentiality and integrity of control messages exchanged between nodes. The LwAODV was benchmarked with the traditional AODV, using packet delivery ration, packet drop ratio, throughput and control overhead as metrics. The result of simulation showed that the LwAODV performed almost the same as the traditional AODV. This makes LwAODV better and safer because of the extra layer of security through the lightweight encryption.

Future works, can include, key Management which is investigating efficient key management mechanisms for the AES encryption in AODV. It entails secure key generation, distribution, and updating procedures. Also, adaptive encryption which is investigating the feasibility of incorporating adaptive encryption techniques into the AODV protocol can also be considered.

REFERENCES

Panda, N. & Pattanayak, B.K., (2018). Analysis of Blackhole Attack in AODV and DSR. *International Journal of Electrical & Computer Engineering (2088-8708)*, *8*(5).

Ur-Rehman, S. & Ullah, I., (2010). Analysis of Black Hole attack on MANETs Using different MANET routing protocol..

Kalia, N. & Sharma, H., (2016). Detection of Multiple Black hole nodes attack in MANET by modifying AODV protocol. *International Journal on Computer Science and Engineering*, *8*(5), pp.160-174.

.

Sathya, M. & Priyadharshini, M., (2016). Detection of Multiple Black hole nodes attack in MANET by modifying AODV protocol. *International Journal on Computer Science and Engineering*, *8*(5), pp.160-174.

Sathya, M. & Priyadharshini, M., (2016). Detection and removal of black hole attack in mobile ad-hoc networks using cooperative bait detection method scheme. *International Journal of Scientific & Engineering Research*, *7*(3), pp.81-85.

Singh, B., Srikanth, D. & Kumar, C.S., (2016). March. Mitigating effects of black hole attack in mobile Ad-Hoc NETworks: Military perspective. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)* (pp. 810-814). IEEE.

Rajeswari, A.R., Kulothungan, K. & Kannan, A., (2016). GNBAODV: guard node based–aodv to mitigate black hole attack in MANET. *International Journal of Scientific Research in Science, Engineering and Technology*, *2*(6), pp.671-677.

Deshmukh, S.R., Chatur, P.N. & Bhople, N.B., (2016). AODV-based secure routing against blackhole attack in MANET In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 1960-1964). IEEE

Dhende, S., Musale, S., Shirbahadurkar, S. & Najan, A., (2017). SAODV: Black hole and gray hole attack detection protocol in MANETs In *2017 International conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 2391-2394). IEEE

Sathish, M., Arumugam, K., Pari, S.N. & Harikrishnan, V.S., (2016). Detection of single and collaborative black hole attack in MANET In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2040-2044). IEEE.

Jhaveri, R.H. & Patel, N.M., (2015). A sequence number based bait detection scheme to thwart grayhole attack in mobile ad hoc networks. *Wireless Networks*, *21*, pp.2781-2798

Kumar, S. & Doohan, N.V., (2016). A modified approach for recognition and eradication of extenuation of gray-hole attack in MANET using AODV routing protocol. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)* (pp. 1-5). IEEE

Yasin, A. & Abu Zant, M., (2018). Detecting and isolating black-hole attacks in MANET using timer based baited technique. *Wireless Communications and Mobile Computing, 2018*.

Tamilselvi, P. & Ganesh Babu, C., (2019). An efficient approach to circumvent black hole nodes in manets. Cluster Computing, 22(Suppl 5), pp.11401-11409.

Dorri, A., Vaseghi, S. & Gharib, O., (2018). DEBH: detecting and eliminating black holes in mobile ad hoc network. Wireless Networks, 24, pp.2943-2955