Received: November 9, 2016

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Secure Light Weight Encryption Protocol for MANET

Banoth Rajkumar[1*] **Gugulothu Narsimha**[2]

[1]*Jawaharlal Nehru Technological University, India*
[2]*Jawaharlal Nehru Technological University College of Engineering, Nachupally, Karimnagar, India*
* Corresponding author's Email: banothrajkumar0381@gmail.com

Abstract: This paper proposes an incorporation mechanism of primitives to provide complete cryptography services for Mobile Ad-Hoc Networks (MANETs). The proposed approach is the Secure Light Weight Encryption Protocol for MANET, where an algorithm for providing availability with DoS resilience has been implemented to avoid TCP SYN flooding packets in the network and passes other packets and also an authentication code and hash function is generated. Later the intermediate nodes are allowed to recode the encoded data in the network. Through which the confidentiality and non-repudiation, key management, integration is achieved.

Keywords: Mobile Ad-Hoc networks, Security, DoS resilience algorithm, Encoding and decoding.

1. Introduction

1.1. MANET

A mobile ad hoc network (MANET) is an infrastructure less self-configuring network with a set of mobile nodes which communicate with each other without any base station or access point in a multi-hop manner. A MANET can be a router to relay packets and connected by wireless links. . The routers can freely move in random and organize themselves arbitrarily so the network's wireless topology may change rapidly and unpredictably. This network may operate in a standalone fashion, or may be connected to the larger Internet. Applications of MANETs include communications in battlefields, disaster rescue operations, and outdoor activities [1, 2].

MANET characteristics like open network environment, lack of centralized control, dynamic topology, and so on make it susceptible to security attack whereas traditional security mechanisms could not meet MANET security requirements because of limited communication bandwidth, computation power, memory and battery capacity, and dynamic deployment environment [3, 4].

1.2 Secure Light Weight Encryption in MANET

Maintaining security in wireless ad-hoc networks is a critical issue due to its characteristics causing attacks paying the way for hackers to pretend as one of the cooperative network node. Hence a security protection should ensure MANET integrity to prevent external as well as internal attacks from the node against network.

Many security mechanisms are based on data encryption where a ciphertext is generated by coordinating message with a secret key so that it can be reviewed only with original key. This encryption mechanism protects the network from unauthorized user accessing to secured communication [4].

In addition, this encoding can reduce energy consumption due to lesser transmissions in encoded data. Confidentiality is achieved by encrypting the packet payload using symmetric-key encryption algorithms [5].

However, a fixed secret key is vulnerable to deciphering by capturing sufficient packets or by launching a dictionary attack [4].

1.3 Problem Identification

In our previous paper, we have proposed a trust based light weight authentication routing protocol in MANET. Initially a multipath route discovery technique is utilized that selects the path with maximum packet success ratio as optimal path for data transmission. For each node in the chosen path, global trust value is estimated based on direct and indirect trust values of the node. If the trust value of any node is below threshold value, then it is authenticated using the secret sharing technique. This authentication technique enhances the reliability, redundancy and network lifetime. In [5], p-coding based lightweight encryption scheme has been proposed for providing confidentiality for network-coded MANETs in an energy-efficient way. P-Coding consumes minimal energy consumption compared to other encryption schemes due to its light weighting scheme. However, the only online computation overhead occurs.

1.4 Contribution

In this paper, we have proposed light weight techniques to provide confidentiality and integrity to the messages. For light weight encryption and decryption, we have used DoS Resilience Algorithm. Also the encoded messages from the source can be recoded in the intermediate nodes to reconstruct the messages. Compared to the previous works, QoS of our proposed work has been improved.

1.5 Organization

Rest of this paper organized as follows. Section 2 relates our work with the previous work. Section 3 describes our proposed solutions. Results of our work are discussed in section 4. This paper concluded with section 5.

2. Literature Review

P. Zhang et al [5] proposed P-Coding, a lightweight encryption scheme for providing confidentiality for network-coded MANETs in an energy-efficient way. The P-Coding enable the source to permute randomly the symbols of each packet before performing network coding operations. Without knowing the permutation, Eavesdroppers could not locate coding vectors for correct decoding without the knowledge of permutation, and hence any meaningful information cannot be obtained. P-Coding consumes minimal energy consumption compared to other encryption schemes due to its light weighting scheme. However, the only online computation overhead occurs.

P. H. Yu and U. W. Pooch [4] presented a new, efficient, low-bandwidth cost and security-enhancing data encryption *i*-key protocol for MANET using dynamic re-keying while end-to-end communication. The secret *i*-key is generated taking advantage of previous data as the seed as well as next packet encryption before delivery. Hence this *i*-key technique enables decryption of message is done only by the original sender and authorized client assuring the privacy of their communication. However, the complexity of the encryption system and the size of the ad-hoc network have a negative effect on performance.

J. Liu et al [6] presented a lightweight key distribution scheme relying on inherent security property of network coding. The scheme combines a simple XOR network coding operations and message authentication codes (MACs) so as to attain data confidentiality and assure the integrity of the distributed keys, respectively. However, computational overhead arises from XOR operations.

A. Bhosle and Y. Pandey [7] proposed securing routing protocol AODV utilizing Symmetric Encryption algorithm AES so as to secure the data as well as preserves the confidentiality. However, the detailed description is not provided.

A. Kumar et al [8] proposed a novel integration mechanism of primitives for providing complete cryptography services for resource constraint Mobile Ad-Hoc Networks (MANETs). The performance of secure MANETs was evaluated by considering software; throughput, jitter & end to end delay; and hardware parameters; area consumption in terms of gate equivalents (GE).

C. Li and G. Wang [9] proposed a commodity integrity detection algorithm (CIDA) relying on Chinese remainder theorem (CRT), to compare the hash value of each commodity identifier and the product of these values with those values that have been stored in a database in advance. Then the proposed algorithm was compared with a typical commodity detection protocol, called trusted reader protocol (TRP). However it has computational complexity.

S. Zhao et al [10] proposed an integrated KM-SR scheme addressing interdependency cycle problem. This scheme provides security features including confidentiality, integrity, authentication, freshness, and non-repudiation by utilizing identity based cryptography.

K. Shanthi et al [11] proposed the protocol namely the hop by hop key agreement convention, where each and every node creates a key (session key) thus for the data encryption These approaches when compared with the proposed work the hop by hop agreement convention suffered from computation complexity as well as the communication overhead. It focused only on the secured authentication of messages the remaining well being criteria were in constrained manner.

3. Proposed Solution

3.1 Overview

As an extension to this work, we provide confidentiality and integrity to the messages which uses light weight techniques.

Here, we use an algorithm for light weight encryption and decryption where nibbles are circulated in a counter so as to avoid sum nibble attack. Then, we implement an algorithm for providing availability with DoS resilience [8]. Also, message authentication code is generated and a hash function is applied to it. Then encryption is performed at source and decryption at destination. By this method, we can attain confidentiality and non-repudiation in addition to authentication.

The encoded messages from the source when passing through intermediate nodes find difficult to reconstruct the messages as they are unaware of the current key in use. Hence intermediate nodes recode the encoded message from the source [4]. Also a random perturbation key is used to avoid single key failure. This transparency property enhances the efficiency of coding. As a whole, availability, confidentiality, key management, integration is achieved.

3.2 DoS Resilience Algorithm

The main aim of the DoS resilience algorithm is to stop the TCP SYN flooding packets in the network and pass other packets. TCP HDR and ipaddr are the built in structures for TCP and IP header information. The maximum limit of packet receiving is threshold without stopping the any service that is set by administrator according to accessible hardware resources. Active open & passive open are two processes which makes a particular side prepared to send a packet or receive a packet. client(), router(), server() and block() are the functions which will handle the client, router, server and block processes.

```
Count()
    {
    }
    static int i=0;
    return ++i;
    block(struct ipaddr *) { }
    packet_receive(TCP HDR *packet)
    {
    int no_of_pkts;
    no of_pkts=count();
    if (no_of_pkts > threshold)
    block(struct ipaddr → src_addr)
    else
    {
    client(active_open);
    router(passive_open);
    server(passive_open);
    client: send(router, syn)
    router: recv(router, syn) & send(c1ient, cookies+
syn+ack)
    client: recv(cookies+syn+ack)
    client: send(cookies+ack)
    router: recv(cookies+ack)
    If (cookie==valid)
    router: send(server, ack)
    else
    block(struct ipaddr → src_addr)
    }
    }
```

In order to accomplish authentication, key management, confidentiality and non-repudiation the network implements the following algorithm.

Step 1: Generating MAC over message '*M*'
Step a: Source: MAC=MAC(*K*, *M*).
Step b: Source: Apply hash over MAC and generate HMAC.
HMAC=H(MAC(*K*,*M*))
Step 2: Source: Encryption Process
Source: message_for_dest = $\mathrm{E}_{\mathrm{K}_{\mathrm{S}}^{2}}$ (M) || HMAC.
Step 3: Destination: Decryption Process
Destination: Message_to receive = message_for_dest
Destination: (*M* || HMAC) = $\mathrm{D}_{\mathrm{K}_{\mathrm{S}}^{2}}$ (Message_to_receive)
Step 4: Destination: reply the step a and step b of step 1 over *M* and generate HMAC
If (HMAC = = HMAC) then return SUCCESS
else return FAILURE

3.3 Intermediate Nodes Recoding Scheme

The basic aim of P-Coding implement the permutation encryptions on coded messages, as shown in Fig. 1. After Permutation Encryption Function (PEF) operations, ciphers of the messages and corresponding Global Encoding Vector (GEV) will be mixed and reordered together. The P-Coding initially consists of three phases: source encoding, intermediate recoding, and sink decoding.

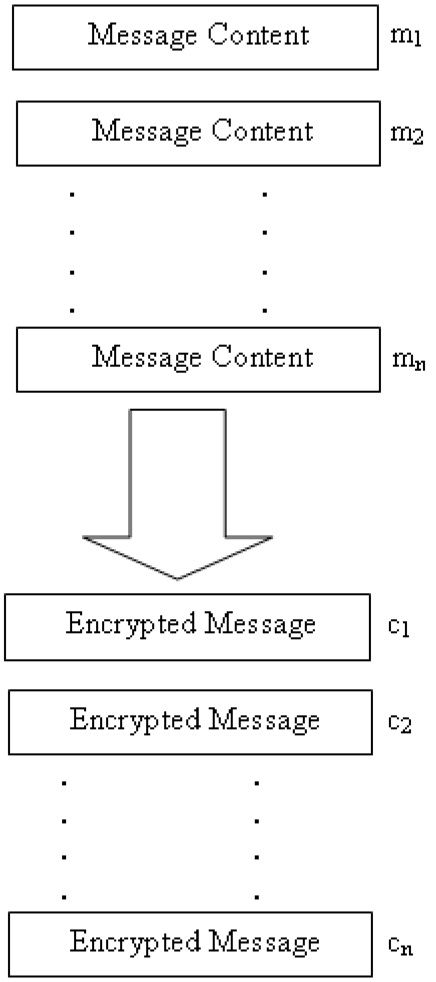

Figure.1 P-coding messages

A Key Distribution Center (KDC) responsible for symmetric key establishment without loss of generality. Hence the source node and sink nodes can share a PEF key k at the bootstrap stage of P-Coding.

Source Encoding

Assume a situation where a source s has h messages which are denoted by column vectors $x_1, \ldots, x_h$, which have to be sent out. It first prefixes these h messages with their corresponding unit vectors. Then the source node s performs linear combinations on these messages with randomly selected Local Encoding Vector (LEV). For example, with LEV $\beta(e_i)$ of output link e_i, the coded message is $y(e_i) = [\beta(e_i), \beta(e_i)X]$, where $X = [xT_1, \ldots, xT_h]T$. Lastly, the source node performs permutation encryption on every message $y(e_i)$ to get its cipher text $c[y(e_i)] = E_k[y(e_i)]$.

Intermediate Recoding

The symbols of messages and corresponding GEVs are reorganised via PEF. The intermediate nodes do not have knowledge about the key being used, which is rather difficult for them to rebuild source messages. On the other side, as permutation encryptions are exchangeable with linear combinations, intermediate recoding can be transparently performed on the encrypted messages:

$$\begin{aligned} c[y(e_i)] &= c[\sum_{e'\in\Gamma-(v)} \beta_{e'}(e) y(e')] \\ &= \sum_{e'\in\Gamma-(v)} \beta_{e'}(e) c[y(e')] \end{aligned} \tag{1}$$

Here, $c[y(e_i)]$ represents cipher text for the input $y(e_i)$.

Sink Decoding

For every sink node by receiving a message $c[y(e_i)]$ from its incoming link $e_i \in \Gamma-(v)$, it decrypts the message by executing permutation decryption on it:

$$D_k\{c[y(e_i)]\} = E_k^{-1}\{E_k[y(e_i)]\} = y(e_i) \tag{2}$$

The term $D_k\{c[y(e_i)]\}$ denotes the decryption of the cipher text for the input $y(e_i)K_k^{-1}$ refers to the inverse version of the encrypted key.

Once h linearly independent messages $y(e_1), \ldots,$ $y(e_h)$ are collected, the sink derives the following matrix representation

$$y = \begin{bmatrix} y(e_1) \\ . \\ . \\ . \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g(e_1), g(e_1)X \\ . \\ . \\ . \\ g(e_h), g(e_h)X \end{bmatrix} = [G, GX] \tag{3}$$

Finally, the source messages can be recovered by applying Gaussian eliminations on Y:

$$Y = [G, GX] \rightarrow [I, X] \tag{4}$$

GX represents the Gaussian form with the product X, X takes of the values g ($e1$) to $g(eh)$. I to be the identity values used for the term G.

3.3.1 Key Perturbing Function Algorithm

During the key perturbing function the data D is divided into generations G.

$$\mathbf{D} = \{\underbrace{x_1, \ldots\ldots x_n}_{G_1}, \ldots\ldots \underbrace{x_{n-1}, \ldots\ldots x_{(n-1)(n-2)}}_{G_n}\}$$

For every generation G_i, let the Permutation Encryption Function (PEF) key be used in G_i as k_i. Normally the source S implements the following steps:

(1) S chooses a random permutation ρ_i of length

n, termed as the perturbing key.

(2) Using the equation $k_i = \rho_i \circ k_{i-1}$, the S updates k_i, where, $\circ$ represents the product of two permutations.

(3) S encrypts ρ_i and sends the ciphertext of ρ_i to every sink and update k_i.

When the perturbing key ρ_i is casually selected every generation and communicated securely between the source nodes and sink nodes, through this it can effectively prevent the single generation failure in the network. Though, it also certainly incur some space overhead since perturbing key should be transmitted in each generation. Every perturbing key is of length n, the same with a labelled packet where this scheme will incur 100% space overhead when no further measures are taken.

```
    Input: a permutation k of length n, integers n, m, s, d with $1 \le m \le n$, $s \in [0, n - m + 1]$ and $d \in [0, m! - 1]$
    Output: a perturbed permutation $\tilde{k}$ of length $n$
foreach $i \in [1, m - 1]$ do        // to create the sequence $(a_1, \ldots, a_{m-1})$
    $a(i) \leftarrow d\%(i + 1)$;
    $d = d/i + 1$;
end
foreach $i \in [1, m - 1]$ do        // to create the sequence $(b_1, \ldots, b_{m-1})$
    $b(i) \leftarrow m - a(m - i)$;
end
foreach $i \in [1, n]$ do            // Initialization
    $\rho(i) \leftarrow i$;
end
foreach $i \in [1, m - 1]$ do        //to compute the partial permutation
    $\rho(s - 1 + i) \leftrightarrow \rho(s - 1 + b(i))$;
end
foreach $i \in [1, n]$ do            // to perturb the current key $k$ using $\rho$
    $\tilde{k}(i) \leftarrow \rho(k(i))$;
end
    return $\tilde{k}$;
```

The above algorithm aims to perturb the key using five limits, of which the current PEF key is k, n represents the length of the labelled packet, m represents the partiality of the perturbing key, s and d are selected randomly from their respective domains to denote the perturbing key. This algorithm employs symmetric encryptions in order to secure the transmission of perturbing key (s, d) from the source node to sink nodes. One more possible way is to let the source node and sink nodes share a common Pseudo Random Number Generator, then the perturbing key (s, d) will be generated by the source and sink nodes in a distributed manner.

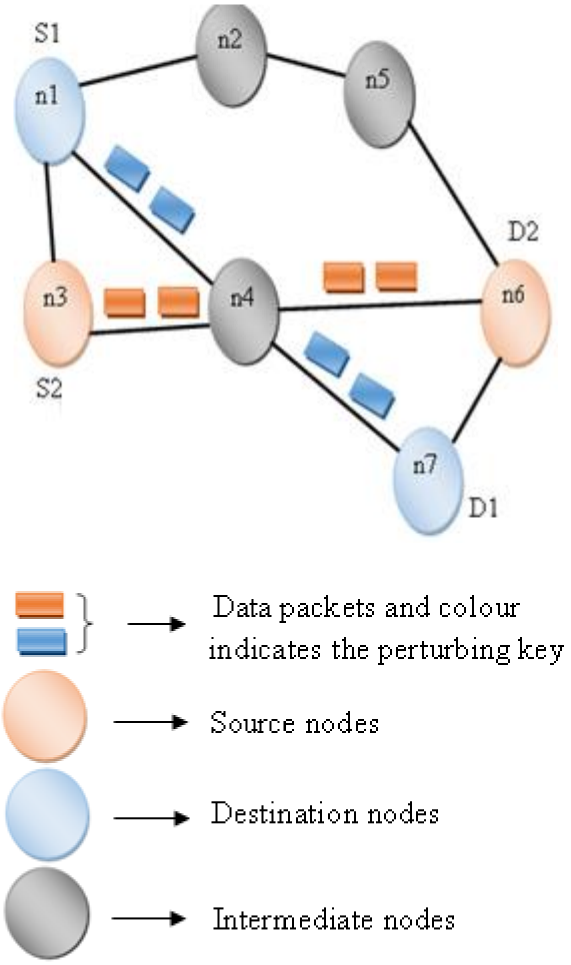

Figure.2 Key perturbing Function

In the above fig 2, the data packets are transmitted from the source nodes to destination nodes through the intermediate nodes present in the network. The data packets which are transmitted are represented with a particular colour. These colours represents the perturbing key. Through these perturbing key in the data packets, helps network to have decent transmission between the nodes. Since only the destination nodes can encrypt the data packets in order to have a secured transmission in the network.

3.4 Overall Algorithm

The overall operation of the approach can be described in the following three steps:

Step 1: Initially in the network implements the DoS resilience algorithm which stops the TCP SYN flooding packets and allow the other packets to pass in the network. Here in the DoS resilience active open & passive open are two processes through which a particular node is arranged to send a data packet or to receive a data packet. The functions which will handle the client, router, server and block processes are client (), router (), server () and block ().

Step 2: Through the DoS resilience algorithm the data packets between the source nodes and destination nodes are transmitted. When the data packets are transmitted the Intermediate Nodes Recoding scheme is implemented. In this intermediate nodes recoding the intermediate nodes are able to recode the encrypted key of the data packets which are transmitted to the destination nodes.

Step 3: Finally the data packets are transmitted between the source nodes and destination nodes in the network. During this transmission the intermediate nodes recoding and also Key Perturbing Function Algorithm is implemented. In this key perturbing function, the data packets are assigned with a key where only the destination nodes can decrypt the key and accept the received data packet from the source nodes in the network.

4. Simulation Results

4.1 Simulation Model and Parameters

We use Network Simulator Version-2 (NS2) [14] to simulate our proposed algorithm. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. We use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link breakage.

Table 1. Simulation Settings

No. of Nodes	50
Area Size	1000 X 1000m
MAC protocol	802.11
Simulation Time	50 sec
Traffic Source	CBR
Packet Size	512
Attackers	2, 4, 6, 8 and 10
Rate	250Kb.
Propagation Model	TwoRayGround
Antenna Type	OmniAntenna
Speed	5, 10, 15, 20 and 25m/s

In our simulation, mobile nodes move in a 1000 meter x 1000 meter region for 50 seconds simulation time. We have varied the node speed as 5,10,15,20 and 25m/s. The transmission range is 250 meters. The simulated traffic is Constant Bit Rate (CBR). Our simulation settings and parameters are summarized in table 1.

4.2 Performance Metrics

We compare our Secure Light Weight Encryption Protocol for MANET (SLWEP) with the P-Coding [5].

We evaluate mainly the performance according to the following metrics.

Average Packet Delivery Ratio: It is the ratio of the number .of packets received successfully and the total number of packets transmitted.

Resilience: It is the ratio between number of packets dropped and the number of packets sent.

Average Packet Drop: It is the average number of packets dropped by the misbehaving nodes.

End-to-End Delay: It is the amount of time taken by the packets to reach the destination.

4.3 Results

A. Varying the Speed

The speed of the nodes is varied as 5,10,15,20 and 25 m/s with 2 attackers.

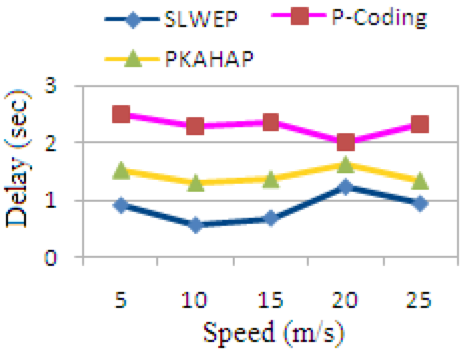

Figure.3 Speed Vs Delay

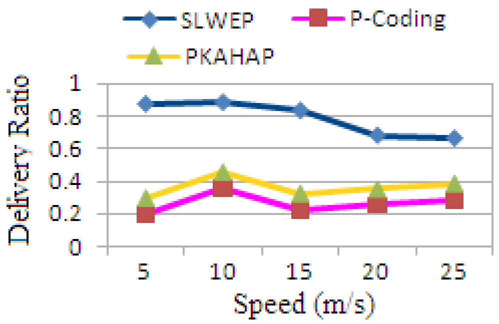

Figure.4 Speed Vs Delivery Ratio

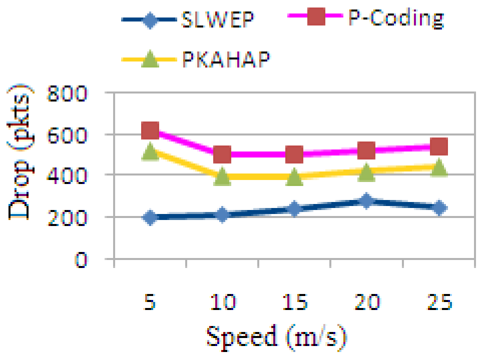

Figure.5 Speed Vs Drop

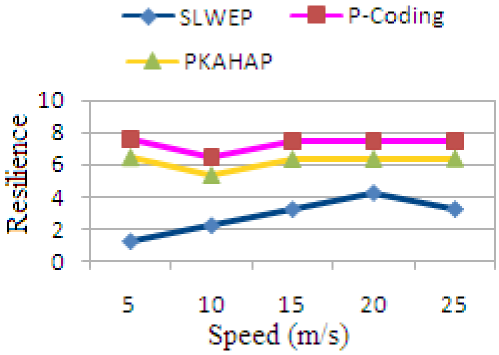

Figure.6 Speed Vs Resilience

From figure 3, it is possible to infer that the delay of our proposed work SLWEP is 93% lesser when contrasted with the P-Coding as well as to the PKAHAP protocol.

From figure 4, we can observe that the delivery ratio of our proposed SLWEP is 66% higher than the existing P-Coding and to the PKAHAP protocol.

From figure 5 we can infer that the drop of our proposed work SLWEP is 75% less when compared with the existing P-Coding and PKAHAP approaches.

From figure 6 it is possible to infer that the resilience of our proposed SLWEP is 64% less than the existing P-Coding protocol and PKAHAP approaches.

B. Varying the Attackers

The number of attackers are varied from 3 to 6 keeping the speed as 25m/s. The performance results are given in the following figures.

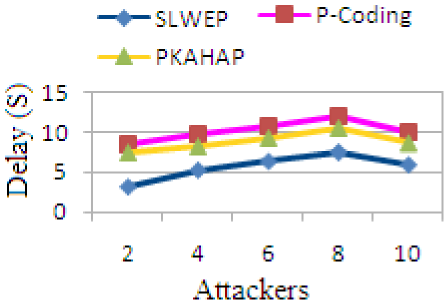

Figure.7 Attackers Vs Delay

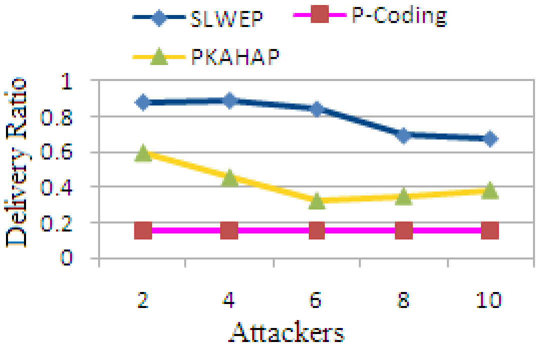

Figure.8 Attackers Vs Delivery Ratio

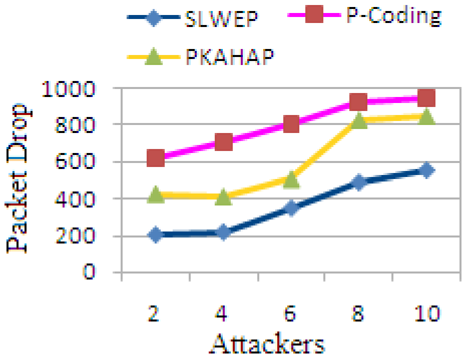

Figure.9 Attackers Vs Drop

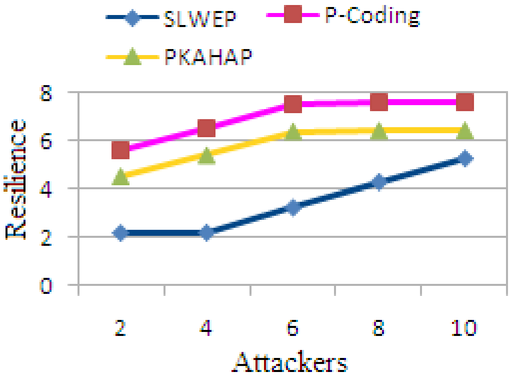

Figure.10 Attackers Vs Resilience

Figures 7 to 10 demonstrate the results of delay, delivery ratio, drop and resilience, respectively for proposed SLWEP, existing P-Coding and PKAHAP approaches. The figures depict that SLWEP outperforms P-Coding, PKAHAP approaches in terms of delay by 59%, in terms of delivery ratio by 66%, in terms of drop by 55% and in terms of resilience by 52%.

5. Conclusion

Here an approach has been proposed for the MANETs known as Secure Light Weight Encryption Protocol for MANET. This approach implements the DoS resilience algorithm in order to avoid the TCP SYN flooding packets which may affects the network and allow the other data packets to flow smoothly. Later also an authentication code and hash function is generated. Also intermediate nodes are allowed to recode the encoded data in the network and a random perturbation key is used to avoid single key failure. The advantages of this approach are

- confidentiality
- non-repudiation
- key management
- integration
- authentication

References

[1] Y. C. Tseng, J. R. Jiang and J. H. Lee, “Secure Bootstrapping and Routing in an IPv6-Based Ad Hoc Network”, *Proceedings of the International Conference on Parallel Processing Workshops* (*ICPPW’03*), pp. 375-382, 2003.

[2] M. Tajamolian, M. Taghiloo and M. Tajamolian, “Lightweight Secure IP Address Auto-Configuration Based on VASM”, *International Conference on Advanced Information Networking and Applications Workshops*, pp. 176-180, 2009.

[3] X. Zhao, Z. You, Z. Zhao, D. Chen and F. Peng, “Availability Based Trust Model of Clusters for MANET”, *7th International Conference on Service Systems and Service Management* (*ICSSSM*), Tokyo, pp. 1-6, 2010.

[4] P. H. Yu and U. W. Pooch, “Security and Dynamic Encryption System in Mobile Ad-Hoc Network”, *Mobile Ad-Hoc Networks: Protocol Design*, 2011.

[5] P. Zhang, C. Lin, Y. Jiang, Y. Fan, and X. Shen, “A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks”, *IEEE Transactions On Parallel And Distributed Systems*, Vol. 25, No. 9, pp. 2211-2221, 2013.

[6] J. Liu, A. R. Sangi, R. Du and Q. Wu, “Light Weight Network Coding based Key Distribution Scheme for MANETs”, *Network and System Security*, Vol. 7873, pp. 521-534, 2013.

[7] A. Bhosle and Y. Pandey, “Applying Security to Data Using Symmetric Encryption in MANET”, *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, No. 1, 2013.

[8] A. Kumar, K. Gopal and A. Aggarwal, “A Complete, Efficient and Lightweight Cryptography Solution for Resource Contrainst Mobile Ad-Hoc Networks”, *2nd IEEE International Conference on Parallel, Distributed and Grid Computing*, pp.854-860, 2012.

[9] C. Li and G. Wang, “A Light-Weight Commodity Integrity Detection Algorithm Based on Chinese Remainder Theorem”, *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1018-1023, 2012.

[10] S. Zhao, R. D. Kent and A. Aggarwal, “An Integrated Key Management and Secure Routing Framework for Mobile Ad-hoc Networks”, *Tenth Annual International Conference on Privacy, Security and Trust*, pp. 96-103, 2012.

[11] K. Shanthi and D. Murugan, “Pair-wise key agreement and hop-by-hop authentication protocol for MANET”, *Wireless Networks*, pp. 1-9, 2016.