International Journal of Engineering Research and Development
eISSN : 2278-067X, pISSN : 2278-800X, www.ijerd.com
Volume 2, Issue 4 (July 2012), PP. 46-52

Implementation of Enhanced Data Encryption Standard on MANET with less energy consumption through limited computation

S.Sudha[1], V.Madhu Viswanatham[2], K.Brindha[3], L. Agilandeeswari [4], G.Ramya[5]
[1,3,4,5]School of Information Technology and Engineering, [2]School of Computing Science and Engineering
VIT University, Vellore-632 014, TamilNadu, India

***Abstract*— A mobile ad hoc network (MANET) is a self-organizing system of mobile nodes. The nodes in MANET are free to move arbitrarily. The nature of the mobile ad hoc networks (MANETs) makes them very vulnerable to an adversary's security threats. Providing security through cryptographic algorithms in these networks is very important. To provide an information security in MANET symmetric encryption algorithms play a main role among all of the cryptographic algorithms. Encryption algorithms used to provide information security are known to be computationally intensive. This algorithm consumes a significant amount of computing resources such as memory, processing time and battery power. A mobile node, that consists of very limited resources, especially limited battery power, is subject to the problem of more energy consumption due to encryption algorithms. Designing an energy efficient security algorithm requires an understanding of the common encryption schemes related to the energy consumption. This article presents an Enhancement to Data Encryption Standard Algorithm in terms of less energy consumption through limited computation by reducing number of rounds and increasing key size.**

***Keywords*— Encryption, Cryptography, MANET Security, EDES**

I. INTRODUCTION

Cryptography is the method of writing the message in secret code and ancient art by converting message into a form non-recognizable by its attackers while stored and transmitted. In cryptography, encryption is the method of transforming information by using an algorithm (called cipher) to make it unreadable by any one except those acquire special knowledge, usually referred to as a key. The output of this method is encrypted information called as a cipher text. The reverse of encryption is the decryption which makes the encrypted information readable again. A variety of encryption algorithms are commonly used in information security. Encryption algorithm can be classified into Symmetric (single key) and Asymmetric (pair of keys) encryption. In Symmetric encryption, a single key is used in encryption and decryption process. The source node uses the key to encrypt the plaintext message and sends cipher text to destination node. The destination node applies the same key to decrypt the cipher text message and recover the plaintext. The key should be securely shared between entities before it starts its transmission. [1] The strength of the Symmetric key encryption algorithm depends on the key size used.

[2] Generally Encryption algorithms utilize significant amount of computing resources such as processing time, memory, and battery power. In MANET, since all the mobile nodes are handheld devices like mobile phones, PDA (Personal Digital Assistant) and Laptops, battery power is issue to the problem of energy consumption due to encryption algorithms. Battery technology is slowly increasing than any other technologies. So, it is necessary to provide a suitable algorithm which requires limited computation and energy consumption with improved security. The DES, one of the most commonly used encryption algorithm for networks, but it is not suitable to MANET, because it requires more computational time and energy in encryption process.

So in this article, we proposed an Enhancement to DES (EDES) which consider the limitation in CPU time, Memory and battery utilization in mobile nodes.[1]The symmetric key is shared in more secured way by using DH (Diffie Hellman key exchange algorithm) key protocol. The rest of the article is organized as follows: Section2 describes the architecture of the EDES. EDES algorithm, experimental setup, Implementation results, and performance are explained in section 3, 4, 5 and 6. Final conclusion is drawn in section 7.

II. ARCHITECTURE OF ENHANCED DATA ENCRYPTION STANDARD

Fig1 and Fig2 show the general architecture for encryption and decryption of the EDES. The proposed new algorithm uses 64 bit plaintext and resultant 64 bit cipher text as like DES algorithm, but it uses key size of 112 bits (2 keys each of which 56 bit length). As shown in the figure, each and every alternative round uses 48 bit sub keys generated from different keys k1, k2. [3]The total number of rounds reduced from 16 to 8. S-box mapping in EDES also vary from DES.

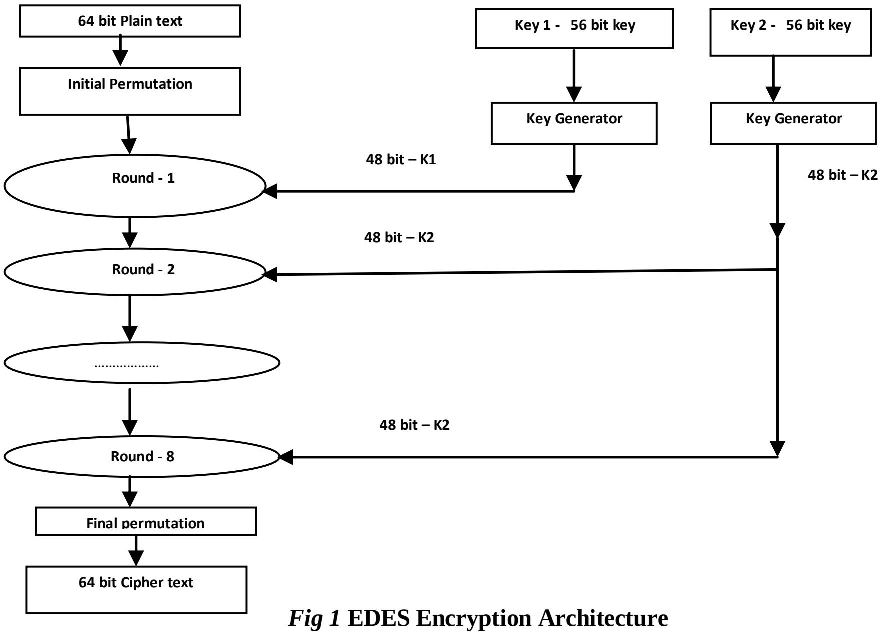


Fig 1 **EDES Encryption Architecture**

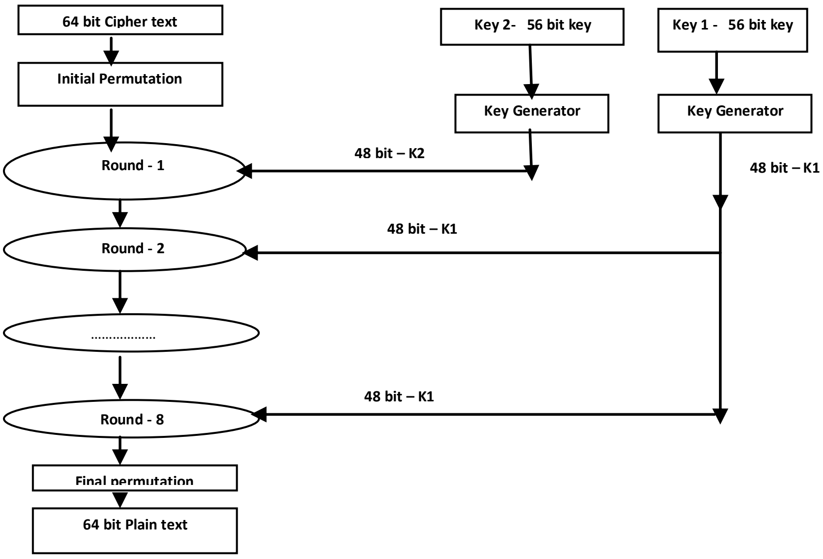


Fig 2 **EDES Decryption Architecture**

A. EDES Round Structure

After the initial permutation it divide the input into two halves each of which consists of 32 bits, as for any Feistel cipher structure it can be describe as: Li = Ri–1, Ri = Li–1 XOR F(Ri–1, K1i| K2i). Function takes input of 32 bit of right halve and 48 bit sub key generated from K1 or K2. Right halve 32 bit expanded to 48 bit by using expanded permutation, then it adds to sub key K1 or K2 using XOR. The 48 bit divided into 8 frames, each of which contains 6 bit of each. The first 2 frames passes to S-Box 1, the next 2 frames 3 & 4 passes to S-Box 2, similarly 7th and 8th frame passes to S-Box 4. Final 32 bit output interchange with L halves. Figure 3 shows the single round structure of EDES algorithm.

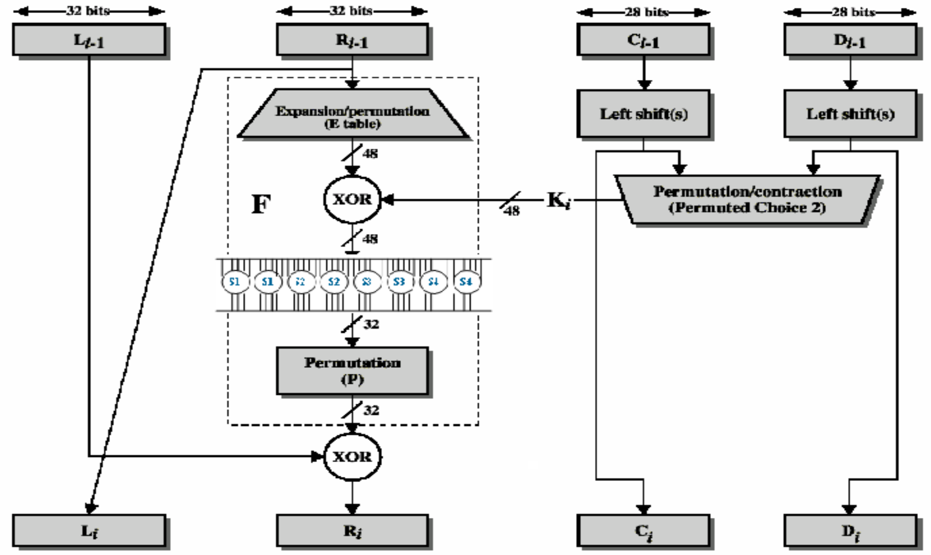


Fig – 3 Single Round of EDES Algorithm

III. ALGORITHM

Symmetric key [1] is shared by both the communication parties using Diffie Hellman key exchange algorithm. [4]

A. Key Exchange between Sender and Receiver

1. Sender selects the large prime number P and generates the multiplicative group G and selects the generator g which is primitive root of G.
2. Sender select the random no X (X < P) and calculate $Y_a = g^X \bmod P$.
3. Sender send (g, P, Y_a) to receiver.
4. Receiver select random number Y (Y < P) and calculate $Y_b = g^Y \bmod P$.
5. Receiver sends Y_b to sender.
6. Sender computes the secret key $K = (Y_b)^X \bmod P = g^{XY}$
7. Similarly Receiver computes the secret key $K = (Y_a)^Y \bmod P = g^{YX}$ during decryption process.

B. Encryption

1. The given input has 64 bit plain text is divided into 2 halves of 32 bit each.
2. Two 56 bit keys (K1 & K2) are used to generate sub keys for each round.
3. 8 sub keys of size (48 bit key) generated from left circular shift and permutation using 2 keys K1 & K2.
4. Sub key are XOR'ed with 48 bit plaintext after the Expansion/Permutation.
5. S-box contain 8 sub frames of 6 bit each as like DES, but 2 sub frames mapped with one S box table.
6. Output of 8th round after the swapping and final permutation it produces 64 bit cipher text.

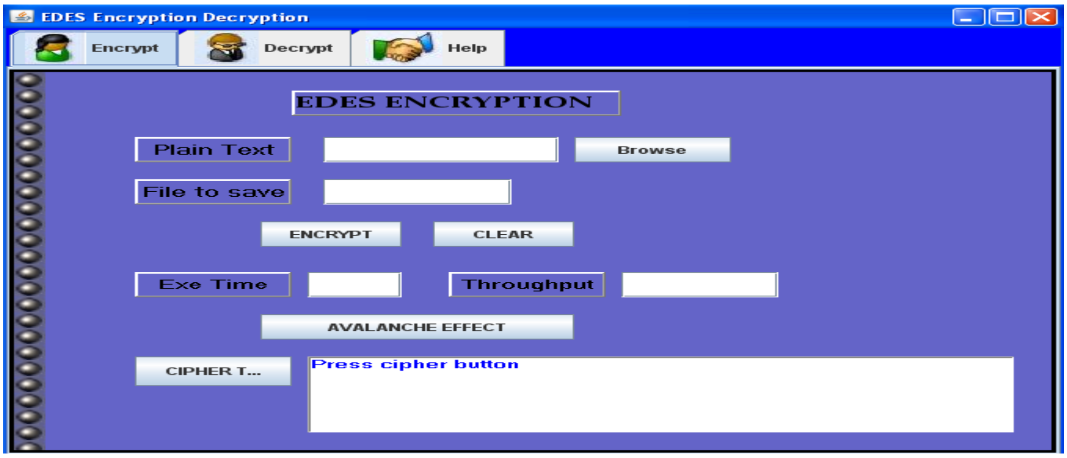
C. Decryption

Decryption is the reverse process of encryption process as like DES. So the Round 1 takes the sub key from K2 in similar way the 8th round takes the sub key from K1.

IV. EXPERIMENTAL SETUP

EDES encryption algorithm have been implemented in Java and the various experiment tests has been carried out using the laptops with the configuration of 2.39 GHz, Intel Pentium core -2 Duo processors with 2GB RAM on windows XP Professional version 2. The algorithm is implemented in different size of text files; the performance of algorithm is tested with various parameters like execution time, throughput, avalanche effect and brute force attack.

The execution time is considered the time taken to encrypt message from a plaintext to cipher text. Throughput of the EDES algorithm is calculated using the execution time of an EDES encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the execution time [8]. The Figure 4 shows the software screen shot (GUI) of the experiment. The user can choose any size of the text file for encryption or decryption. The execution time and throughput of the encryption and decryption of the various file will be displayed after its completion of encryption or decryption process. The strengthness of EDES algorithm is measured using the avalanche effect option in this software.



Press cipher button

Fig 4 **Software Screen shot**

V. IMPLEMENTATION

DES and EDES encryption algorithm compared with different text file size. The experiment results are shown below.

A. Comparison of Encryption and Decryption Execution time

The Table I and Table II show the execution time for DES and EDES encryption and decryption algorithm with different text file size.

Table I Encryption Execution time in Milliseconds

Encryption Algorithm	Various size of Text Files			
	24KB	58KB	1040KB	3118KB
DES	813	1859	28453	87781
EDES	719	1547	24547	73641

Table II Decryption Execution time in Milliseconds

Decryption Algorithm	Various size of Text Files			
	24KB	58KB	1040KB	3118KB
DES	344	719	11000	32359
EDES	313	531	7422	21891

The Fig 5 and 6 shows the average execution time of the encryption and decryption algorithm with different text file size. The result shows the improvement in EDES in terms of less energy consumption through the less execution time of encryption and decryption.[5]

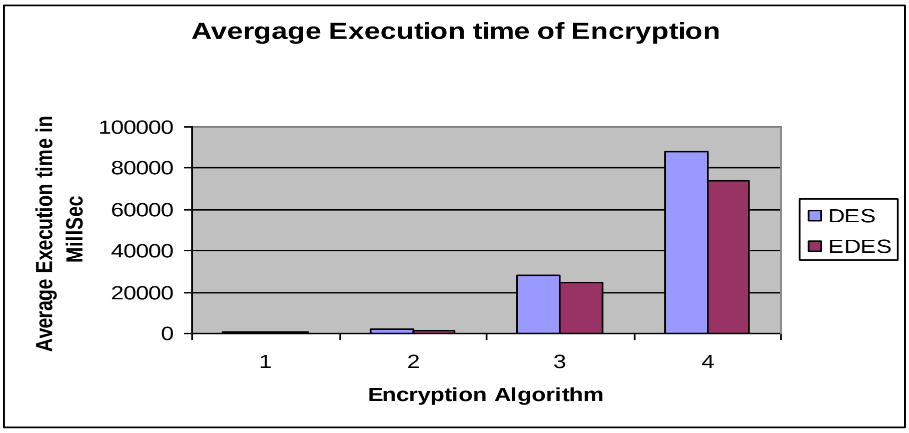


Fig 5 **Average Execution time of Encryption Algorithm**

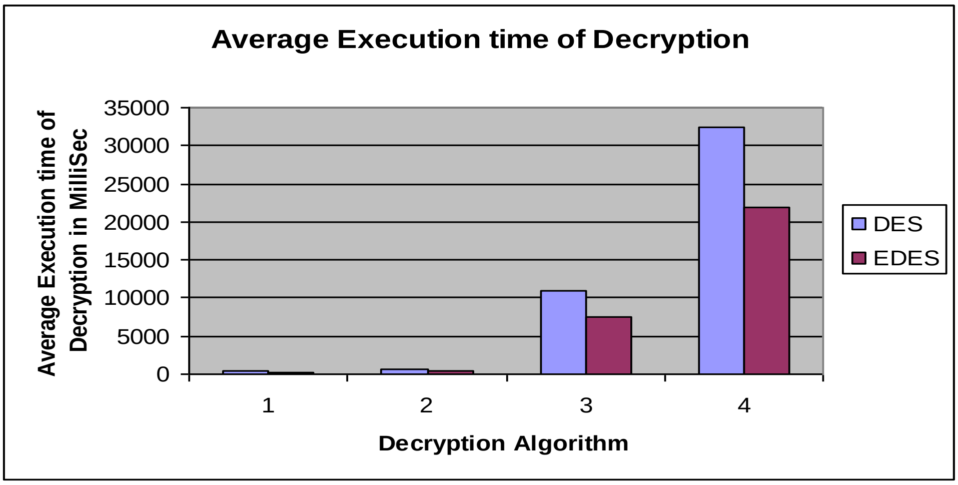


Fig 6 **Average Execution time of Decryption Algorithm**

B. Comparison of Throughput in Encryption and Decryption algorithm

The Table III and IV show the comparison of throughput in DES and EDES Encryption and Decryption algorithm.

Table III Throughput in KB/Milliseconds

Encryption Algorithm	Various size of Text Files			
	24KB	58KB	1040KB	3118KB
DES	29	31	37	36
EDES	32	38	43	43

Table IV Throughput in KB/Milliseconds

Decryption Algorithm	Various size of Text Files			
	24KB	58KB	1040KB	3118KB
DES	68	82	96	98
EDES	75	111	143	145

The Fig 7 and 8 shows the average throughput of the encryption and decryption algorithm with different text file size. The result shows the performance improvement in EDES in terms of average throughput compared to DES.

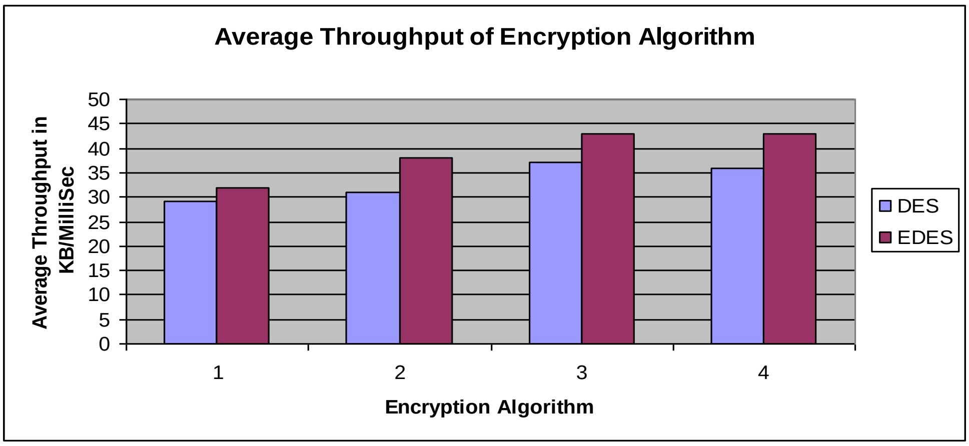


Fig 7 **Average Throughput of Encryption Algorithm**

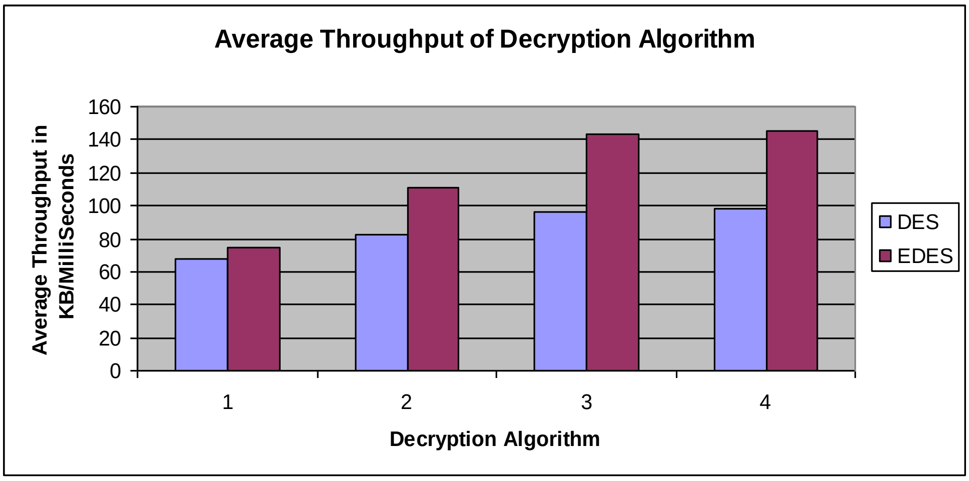


Fig 8 **Average Throughput of Decryption Algorithm**

VI. PERFORMANCE OF EDES

A. Less Energy Consumption through Limited computation.

The proposed EDES algorithm involves only 8 rounds, so it requires lesser computation and it saves battery utilization than DES algorithm which shown in our experimental results.[7], [8]

B. Brute force attack

EDES uses 112 bit key size, it is impossible for an attacker to try all possible 2112 in a feasible amount of time. Suppose the key of DES can be cracked in some less amount of time, but it needs more time to crack EDES. Therefore, EDES is secured against brute force attack than DES.[8]

C. Avalanche effect.

The avalanche effect is clear if, when an input is changed to some extent, the output changes drastically. In any type of cipher method, such a small change in either the plaintext or the key should make the drastic change in the cipher text. [8] Thus, the avalanche effect is more in EDES compared with DES. It is also proved in the experimental results.

D. Key Management & Complexity

112 bit key securely exchanged with the help of diffie-hellman key exchange algorithm reduces the possibility of hackers cracking the keys during the transmission.

EDES algorithm provides more complexity than DES, because it uses 2 keys K1, K2 alternatively.

E. Improved execution time and Throughput

EDES encryption and Decryption takes less execution time compared with DES for various sizes of text files as shown in the experimental result. [6]

Average throughput of Encryption and Decryption in EDES also improved compared with DES as shown in the fig 6 and 7.

F. Known plaintext Attack

Known plain text is one of the attacks for cryptanalysis where the hacker uses the sample pair of plaintext and cipher text to recover the information about the key. It has been proved that DES version with less than 16 rounds is more vulnerable to known plaintext attack [8]. Even though the proposed algorithm uses only 8 rounds, it is difficult for the hacker to crack the information about the key in feasible amount of time due to increased key size.

VII. CONCLUSION

The proposed new algorithm EDES uses limited computation by reducing the number of rounds used in DES algorithm by which energy consumption is reduced and security is improved than DES in terms of brute force attack, differential crypto analysis attack by increasing the key size. Symmetric key also shared by using one of the public key crypto systems Diffie-Hellman key exchange algorithm based on discrete logarithm problem. Since the less energy consumption and security is the major requirement in the MANET, the EDES algorithm is one of the most suitable encryption algorithms for MANET nodes. [2] Eventhough the algorithm tested with only laptop but it is applicable for all handheld devices which require limited computation and energy consumption.

As the future work, we decided to implement this algorithm with different file format like image, video and audio and it also compared with various symmetric encryption algorithms.

REFERENCES

[1]. W.Stallings, Cryptography and Network Security: Prentice Hall, 2005, PP. 58-309.

[2]. M.Umaparvathi, Dr.Dharmishtan K Varughese , “Evaluation of Symmetric Encryption Algorithms for MANETs,” 2010 IEEE Transactions.

[3]. Coppersmith D, "The Data Encryption Standard (DES) and Its Strength against Attacks.” IBM Journal of Research and Development, PP. 243 – 250, May 1994.

[4]. Hardjono, "Security in Wireless LANS and MANS," Artech House Publishers 2005.

[5]. S.Hirani, "Energy Consumption of Encryption Schemes in Wireless Devices Thesis," university of Pittsburgh, April 9, 2003. Retrieved October1, 2008, at: portal.acm.org/citation.cfm? Id=383768.

[6]. M.Haleem, C.Chetan, R.Chandramouli and K.P.Subbalakshmi,“Opportunistic Encryption: A Trade-Off between Security and Throughput in Wireless Networks,” IEEE Transaction on Dependable and Secure Computing, vol. 4, no. 4, pp.313-324, Oct 2007.

[7]. Ruangchaijatupon, P. Krishnamurthy, “Encryption and Power Consumption in Wireless LANs,” The Third IEEE Workshop on Wireless LANs - September 27-28, 2001- Newton, Massachusetts.

[8]. Behrouz A.Forouzan, Cryptography and Network Security: Special Indian Edition 2007, Tata McGraw-Hill, pp. 177.