A NEW-FANGLED SYMMETRIC BLOCK CIPHER USING ZIG-ZAG SCAN PATTERNS

Kalavathi Alla[1], Sai Jyothi B

[1,2]Vasireddy Venkatadri Institute of Technology,
Jawaharlal Nehru Technological University, Kakinada.

Abstract

In today's world of Information Technology, information storage and distribution is very important. To secure the data in the network, one need to protect the data using cipher algorithms. This paper presents a novel symmetric block cipher which ciphers data using zig-zag scan patterns. The input data is divided into manageable size of blocks. Each block of data is taken as an input and produces the corresponding cipher text. The data is arranged in a corresponding matrix of manageable size using the zig-zag scan pattern. Then again it is divided into number of 2x2 sub matrices. Cipher text can be retrieved from each 2x2 sub matrix by reading column wise. The Cipher is tested with time complexity, frequency analysis, and with homogeneity testing.

Keywords: *Symmetric Ciphers, block Ciphers, Cryptography, Zig-Zag Scan Patterns, DES*

---***---

1. INTRODUCTION

Cryptography is a science of secret writing which protects data and transmits secret messages in a distributed network from one peer to another peer. With the vast development in the field of information technology, there is a need to strengthen and develop new security algorithms. Modern algorithms are based on mathematical theory. These algorithms are developed by assuming computation hardness and hard to break practically [4]. Therefore cryptic ciphers are computationally secure by using integer factorization. Cryptic ciphers can be developed either by using symmetric key or asymmetric key cryptography. The current research focuses on symmetric key cryptography[6]. Symmetric cryptography works with both stream and block ciphers. A block cipher inputs a block of characters into the encryption algorithm and outputs a block of cipher characters. Whereas, stream ciphers encrypts bit by bit character by characte[7]r. This paper proposes a novel block cipher algorithms which arranges the letters of the plain text in Embedded Zerotree Wavelet zig-zag scanning order (EZWZBS) and then divide the matrix into 2x2 sub matrices to produce the cipher text. The algorithm is designed in such a way to overcome with all possible security attacks in block ciphers. Algorithm illustration and scanning procedure with an example is explained in subsequent sections II and III. The experimental results are discussed in section IV.

2. PROPOSED ALGORITHMS

The proposed algorithm is developed using the principles of block ciphers. Encryption and decryption algorithms are described in this section.

2.1 Encryption Algorithm

Step1 : Input a text file as a secret data into the encryption algorithm.
Step 2: Read the text file and find the relevant ascii value and convert it to binary and store this string in a variable secretbin.
Step 3: Divide the secretbin into blocks of manageable size $(2*n)^2$, where n=1,2,3,4,… Therefore, the size of each block becomes 4/16/36/64/100/144………..
Step 4: For each block, do the following steps.
Step 5: Find the size of matrix using the equation 2*n, where n=1,2,3…so that 2x2,4x4,6x6 matrices can be generated.
Step 6: The binary bits of each block can be taken from most significant bit (MSB) to least significant bit (LSB) to fit in a corresponding matrix by using EZW zig-zag scanning pattern.
Step 7: Now divide the square matrix of 2*n (where n=1,2,3…) into $(2*x)^2$ (where x=1/2,1,3/2,2,5/2,…) number of 2x2 sub matrices. Both n and x are shared between the sender and receiver.
Step 8: Read bits of each 2x2 sub matrix diagonally and append to a string cipher.
Step 9: Find the relevant ascii character.

2.2 Decryption Algorithm

Step 1: Input the encrypted file to the decryption algorithm.
Step 2: Read the input file and find the ascii value of each character, and convert it to binary and also store this binary stream in a string plainbin.
Step 3: Divide the plainbin string into blocks of manageable size $(2*n)^2$, where n is a shared secret key between the tewo peer entities physically. And n=1,2,3,4,….. So that the size of each block becomes 4/16/36/64/100/144…
Step 4: For each block of the cipher text Do

Step 5: Find the size of matrix using 4*n, where n=1,2,3… so that 2x2,4,x4,6x6,…. matrices can be generated.
Step 6: Find the number of sub matrices using $(2*x)^2$ where x=1/2,1,3/2,2,5/2…
Step 7: Find the size of square matrix using $(2*n)^2$ where n=1,2,3,4….. and store in the variable size=m.
Step 8: write bits of the plainbin string diagonally on each 2x2 sub matrix.
Step 9: Combine all the 2x2 sub matrices sequentially to form a square matrix.
Step 10: Read the bits of the square matrix using EZW zig-zag scanning pattern and append to a string "retrieved".
Step 11: Find the relevant ascii characters of string "retrieved".

3. IMPLEMENTATION PROCEDURE

To illustrate the (EZWZBS) EZW zig-zag scanning pattern, let us consider a secret message "CM". The corresponding 8 bit binary equivalent of ASCII character 'c' is "01000011" and for character 'M' is "01001101". Therefore the binary representation of plain text message "CM" is 0100001101001101. Now, divide the binary string into manageable size of blocks using the formula $(2*n)^2$ Where n=1,2,3…….. The binary bits of each block are stored in a square matrix of size 2*n using EZW scanning order.

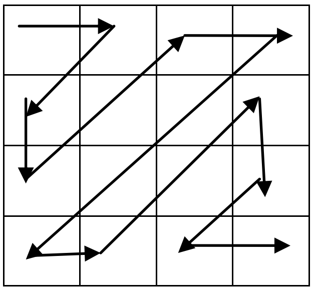

Fig 1: EZW Zig-Zag Scanning Order

The plain text message is "CM"

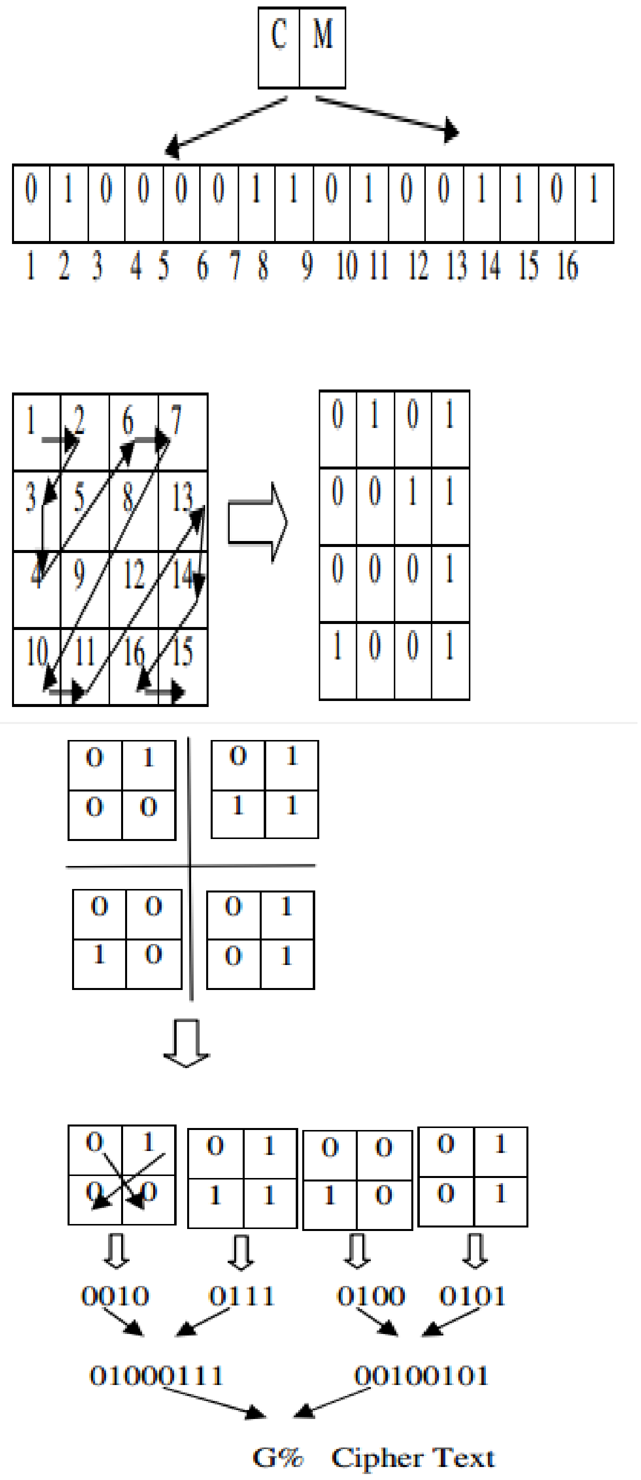

Fig 2: Encryption procedure

After arranging bits of plain text in a square matrix divide them into 2*n number of sub matrices. Read bits of each 2x2 sub matrix diagonally from left to write and top to bottom. Append all these bits to a cipher string. Find the relevant ascii character of cipher string to retrieve the cipher text. As each cryptic algorithm is reversible in nature, apply vice versa procedure to decrypt the received cipher text.

4. EXPERIMENTAL RESULTS

The EZWZBS algorithm is tested for its efficacy in terms of encryption and decryption times, frequency analysis, Avalanche effect which includes both strict avalanche effect and bit independence criteria. Algorithm is also compared with the existing algorithms DES with 64 bit and AES with 256 bits.

4.1 Time Complexity Analysis

To find out the efficacy of EZWZBS algorithm time complexity is calculated based on the time difference between the processor clock ticks of execution times from start and end times. We measure the time in terms of milliseconds. Table 1 shows the time complexity comparison of proposed algorithm with existing DES and AES algorithms with variable file sizes [3]. Figure 3 shows the graphical representation of proposed algorithm execution times with existing DES and AES algorithms[5].

Table 1: Comparative execution times in milliseconds

S.No	Input File Size (in bytes)	Execution Time in(milliseconds)		
		DES	AES	Proposed Algorithm
1	64	6	12	9
2	128	12	25	16
3	192	19	37	17
4	256	25	50	31
5	1024	100	199	95
6	2048	200	400	190
7	2054	200	400	191
8	5096	497	993	473
9	982732	95750	191500	91171
10	5456704	531661	1063322	506237

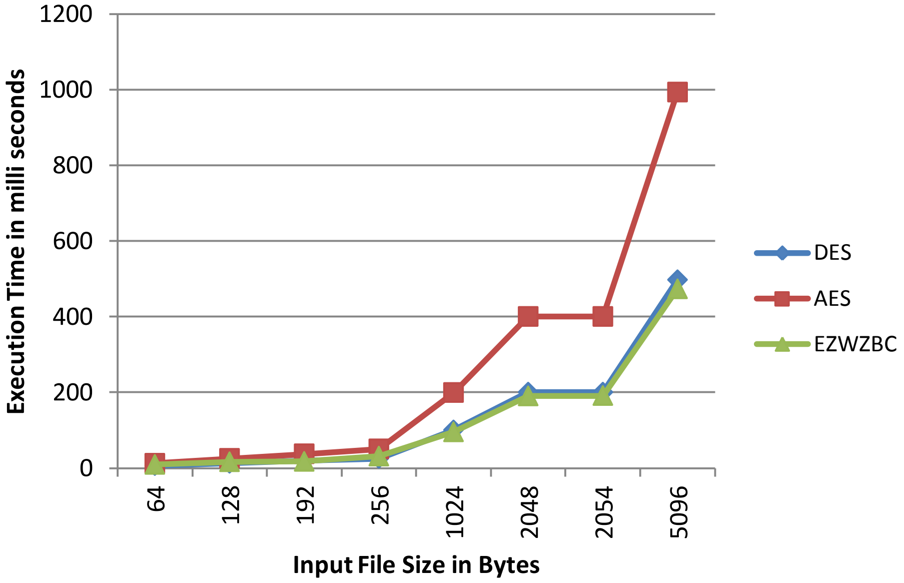

Fig 3: Execution times of DES, AES and EZWZBC

4.2 Statistical Analysis

Cryptanalyst benefits from inherent characteristics of plain text language to lodge a statistical attack. The cryptanalyst identifies the most frequently repeating character in the cipher text, and assumes that the corresponding plain text character is E using the standard frequency of English language. He may also do the same for most frequently repeated diagrams and trigrams in the cipher text. After finding such a few pairs, the crypt analyst can find the key and use it to decrypt the cipher text. To overcome with this kind of attacks, the cipher should hide the statistical characteristics of the language. The proposed EZWZBC is tested with 8 different types of files that we have analyzed in section 4A from such type of attacks. The following figures 4 to 7 shows the distribution of characters in the source file, and cipher text file according to DES, AES, EZWZBC[5],[7],[11].

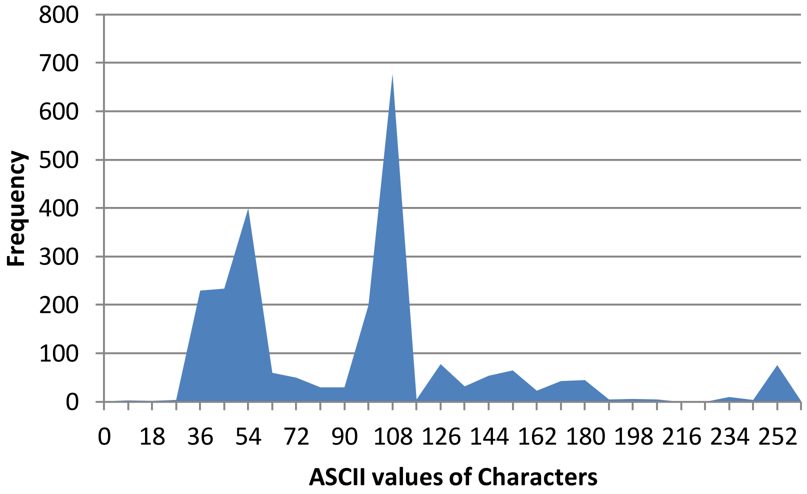

Fig 4: Frequency distribution of characters in input file

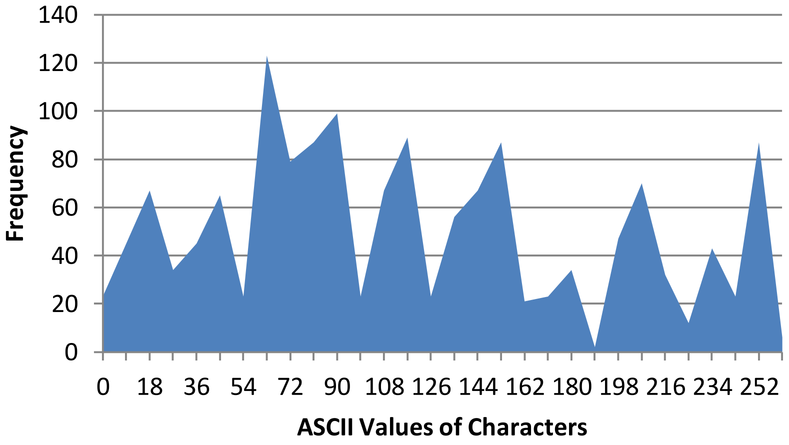

Fig 5: Frequency Distribution of characters for encrypted file using DES

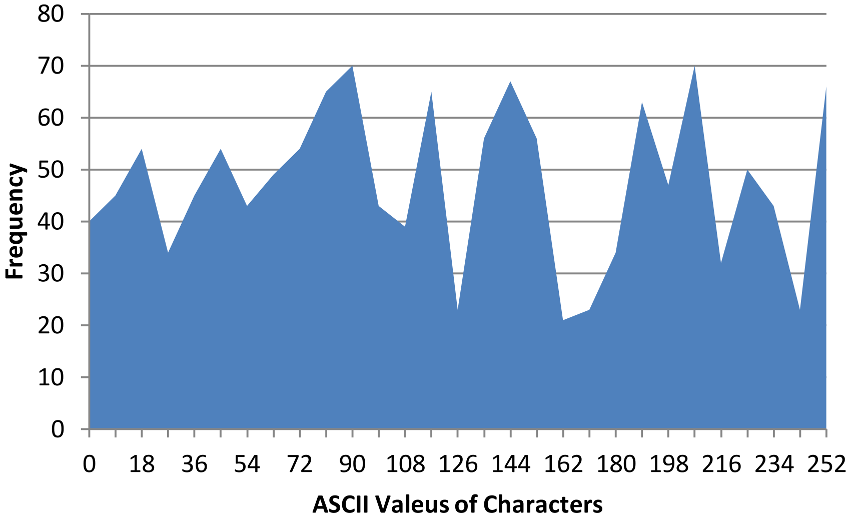

Fig 6: Frequency distribution of characters for encrypted file using AES

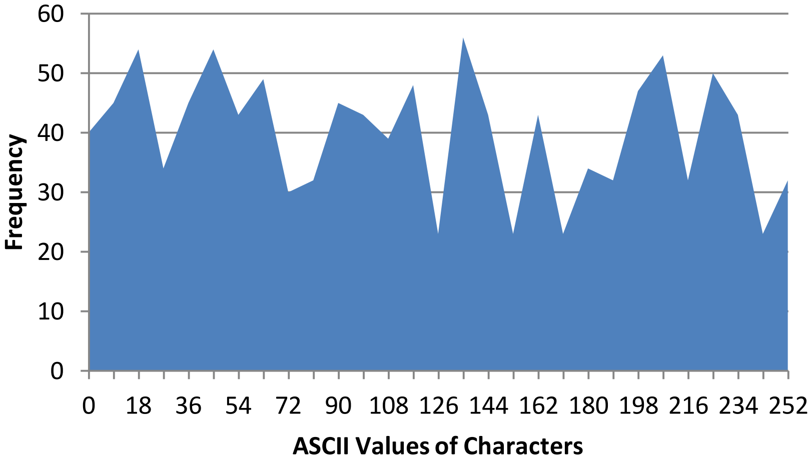

Fig 7: Frequency Distribution of cipher text characters using EZWZBC

4.3 Avalanche Analysis

This is a most desirable property of all cryptographic algorithms and hash functions. Due to this effect when an input changes slightly, the output changes significantly, that is at least half of the bits will be changed when compared with the original cipher text. In high quality block ciphers, such a small change in key or plaintext causes a drastic change in the cipher text. Constructing a cipher to exhibit a substantial avalanche effect is one of the primary design objectives. The proposed EZWZBC algorithm showed good avalanche effect and is also compared with the existing algorithms DES and AES[5],[7],[11].

Table 2: Comparison of Execution times

S.No	Input File Size (in bytes)	No of bits changed when there is a a bit change in plain text or key.		
		DES	AES	EZWZBC
1	64	43	41	33
2	128	90	83	69
3	192	134	126	93
4	256	185	167	155
5	1024	673	658	642
6	2048	171	1320	1317
7	2054	1380	1318	1326
8	5096	5042	3264	3258

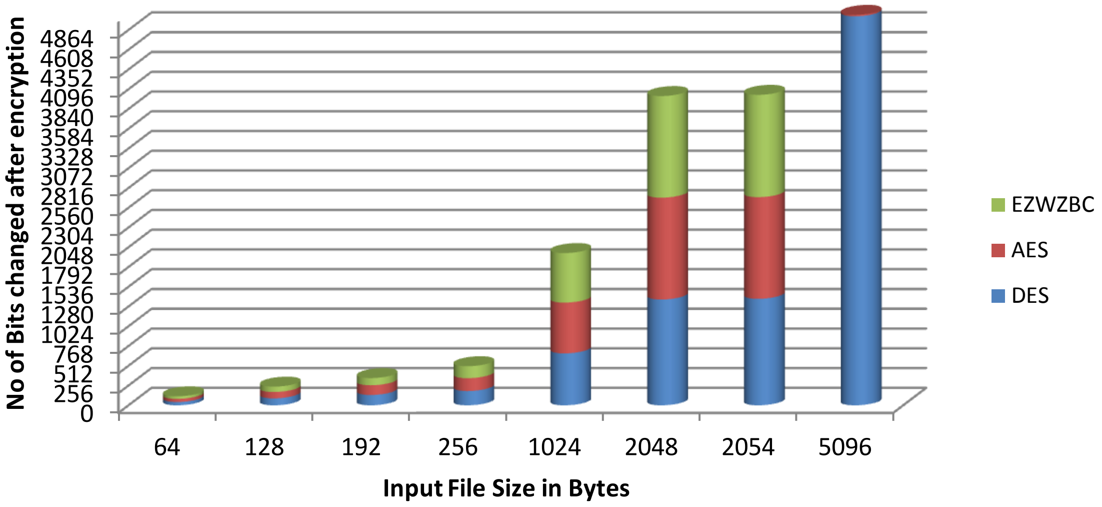

Fig 8: Graphical representation of execution times

4.4 Non- Homegenity Analysis

In Cryptography, Chi-square analysis is used to examine the homogeneity between the source file and encrypted file(cipher file) or frequency of each character in source file and encrypted file. This is also known as Pearson’s test. In this analysis, the frequency of occurrence of plain text characters in a source file are compared against the frequency of occurrence of these characters with encrypted file using the following formula.

$$\chi^2 = \Sigma\{(f_s - f_e)^2/f_e\},$$

where f_s and f_e are frequency of each character with respect to the source and encrypted files.

Table 3: Chi square values to test for non homogeneity

.No	Input File Size (in bytes)	Chi-Square Value		
		DES	AES	EZWZBC
1	64	953	1113	1021
2	128	1834	2122	1976
3	192	2988	3428	3274
4	256	4213	4385	4289
5	1024	16987	17563	17435
6	2048	34863	35121	35016
7	2054	34934	35153	35097
8	5096	68537	70289	69979

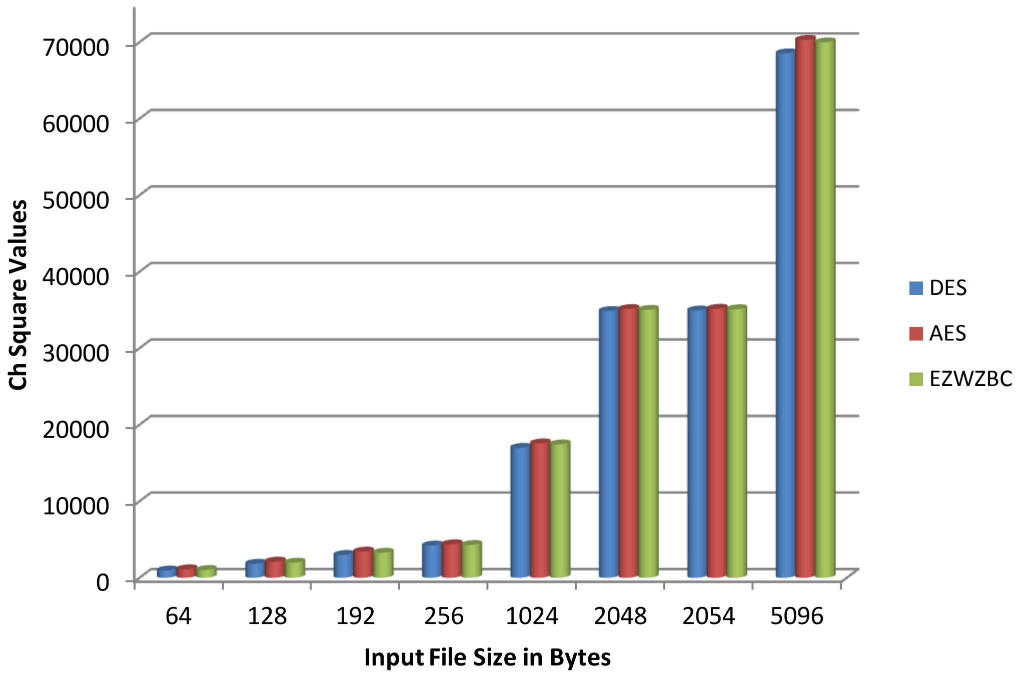

Fig 9: Graphical representation of chi square test.

5. CONCLUSION

The proposed EZWZBC algorithm is implemented in Java. The algorithm is very simple and easy to understand. This algorithm is tested with various performance metrics in terms of its execution speed, statistical analysis, avalanche analysis and chi-square analysis. Algorithm ensures high security in message transmission and is also compatible with existing industry accepted standards like DES and AES algorithms.

ACKNOWLEDGMENTS

The authors would like to thank the college management Vasireddy Venkatadri Institute of Technology for their academic and financial support to carry out this work.

REFERENCES

[1]. American National Standard for Financial Services 1998, "Triple Data Encryption Algorithm Modes of Operation." American Bankers Association, Washington, D.C. X9.52- July 29, 1998.

[2] Barker W, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication, 2008, 800-67.

[3]. Biham E. and Shamir A., Differential cryptanalysis of the full 16- round DES, Lecturer Notes in computer Science, 1993, 494-502.

[4]. Douglas R. Stinson, CRYPTOGRAPHY: Theory and practice, Chapman & Hall/ CRC Press, 2002, pp. 161-280.

[5]. Federal Information Processing Standards Publication 46-3, "Data Encryption Standard (DES)."U.S. DoC/NIST, October 25,1999..

[6]. Feistel H., Cryptography and Computer Privacy, Scientific American, vol. 228, no. 5, 1973.

[7]. FIPS PUB 46-3, Federal Information Processing Standards Publication, 1999.

[8]. Grabbe J., Data Encryption Standard: The DES algorithm illustrated, Laissez faire City time, vol. 2, no 28, 2003.

[9]. Grinstead Ch. and Snell L., Introduction to Probability, American Mathematical Society, 1997, pp. 325-360.

[10]. J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES algorithm Submission, FIPS 197, 1999.

[11]. Jawahar Thakur, Nagesh Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography algorithms simulation based performance analysis" International Journal of Emerging Technology and Advanced Engineering, Volume 1, Issue 2,December (2011).

[12]. Kalavathi Alla, Gowri Shankar, Secure Transmission of Authenticated Messages using New Encoding Scheme and Steganography in CCSEIT 2012 at Coimbatore Proceedings were published in ACM Digital Library.

[13]. Kalavathi Alla, R. Sivarama Prasad, A New Approach to Telugu Text Steganography, IEEE ISWTA , Malaysia, October 2011.

[14]. Kalvathi Alla, Sivarama Prasad Hindi Text Steganography using Matraye, Core Classification and HHK Scheme, IEEE ITNG 2010, held at Las Vegas, USA.

[15]. Kalvathi Alla, Sivaram Prasad, Presented a paper on Evolution of Hindi Text Steganography, IEEE ITNG 2009, held at Las vegas, USA, April 27-29, 2009.

[16]. Kalavathi Alla, N. Bhagya Sri, An Efficient Key Exchange for Secure Peer Communications, International

Journal of Advances in computer, Electrical and Electronics Engineering, Volume 2 Special Issue, Dec 2012.

[17]. Kalavathi Alla, Sivaram Prasad A New approach to Hindi Text Steganography using Hindi Karak Kriyaye, International Journal of Information Assurance and Security, Vol 6, Issue 6 2011.

[18]. Kalavathi Alla, Sivaram Prasad Information Hiding using Telugu Text Steganography, International Journal of Intelligent Information Processing, Volume 6, January 2012.

[19]. Kalavathi Alla, Sivaram Prasad A Novel Hindi Text Steganography using Letters and Letter Diacritics, International Journal of Computer Science and Network Security, Volume 8, issue 12, December 2008

[20]. V.S.Shankar Sriram, Abhishek Kumar Maurya, G.Sahoo,"A Novel Multiple Key Block Ciphering Mechanism with Reduced Computational Overhead" in "International Journal of Computer Applications", Vol.1 (No.17):25–30, February 2010.