ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

A Universal Session Based Bit Level Block Encoding Technique to Enhance the Information Security

Joyita Goswami (Ghosh)[1], Manas Paul[2]
[1]Dept. of Computer Application, JIS College of Engineering, Kalyani, West Bengal, India, joyitagoswami@gmail.com
[2]Dept. of Computer Application, JIS College of Engineering, Kalyani, West Bengal, India, manaspaul@rediffmail.com

Abstract: In this paper a session based symmetric key cryptographic technique has been proposed and it is termed as Matrix Based Block Encoding Technique (MBBET). MBBET consider the input plain text as a stream of finite number of binary bits. This binary bit stream is chopped into manageable-sized blocks. The bits of these blocks fit column-wise from top to bottom into a square matrix of suitable order n. Now bits are collected from the square matrix along diagonally upward starting from (n , n) cell in a right to left trajectory to form the encrypted binary string and from this string cipher text is formed. Combination of values of block length and no. of blocks of any particular session generates the session key for MBBET. For decryption the cipher text is considered as a stream of binary bits. Processing the session key information, this binary string is split into manageable-sized blocks. The bits of the block fit diagonally upward starting from (n , n) cell in a right to left trajectory into a square matrix of suitable order n. Now bits are taken column-wise from the square matrix to form the decrypted binary string and from this string plain text is formed. A comparison of MBBET with existing and industrially accepted TDES and AES has been done.

Key words: AES, MBBET, Session Based Key, Symmetric Key Cryptography, TDES.

1. INTRODUCTION

With the global acceptance of the Internet our communication becomes faster and easier. There is always a chance of manipulation of our important data and interception of any secret data. Hence it is very important to secure our essential information from eavesdroppers. Day by day it is becoming more and more complex to maintain the network security. As a result continuous research works [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] are going on in this field of cryptography to enhance the network security. Various algorithms are developed in this field but each of them has their own merits and demerits.

Based on symmetric key cryptography a new technique has been proposed where the plain text is considered as a stream of binary bits. Bit positions are shuffled to generate the cipher text. A session key is generated using plain text information. The plain text can be regenerated from the cipher text using the session key information.

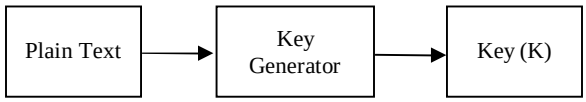
Section 2 of this paper contains the proposed scheme with block diagrams. Section 3 deals with the algorithms for encryption, decryption and key generation. Section 4 explains the proposed technique with an example. Section 5 shows the results and analysis on different files and the comparison of the proposed technique with TDES [11], AES [12]. Conclusions are drawn in section 6.

2. THE SCHEME

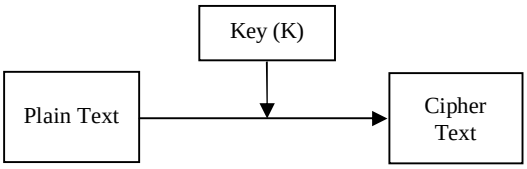
The MBBET algorithm consists of three major components

- Key Generation
- Encryption Mechanism
- Decryption Mechanism

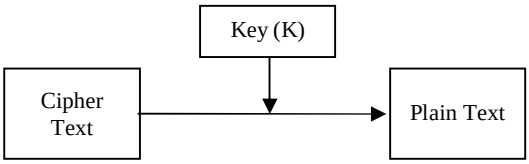
Key Generation:



Encryption Mechanism:



Decryption Mechanism:



3. PROPOSED ALGORITHM

3.1 Encryption Algorithm:

Step1. The plain text i.e. the input file is considered as a stream of finite no. of binary bits.

Step2. This binary string breaks into manageable-sized blocks with different lengths like 4 / 16 / 64 / 144 / 256 / 400 / …….. [$(4n)^2$ for n = 1/2, 1, 2, 3, 4, 5, …….] as follows:

First n_1 no. of bits is considered as x_1 no. of blocks with block length y_1 where $n_1 = x_1 * y_1$. Next n_2 no. of bits is considered as x_2 no. of blocks with block length y_2 where $n_2 = x_2 * y_2$ and so on. Finally n_m no. of bits is considered as x_m no. of blocks with block length y_m (= 4) where $n_m = x_m * y_m$. So no padding is required.

Step3. Square matrix of order $\sqrt{y}$ is generated for each block of length y. The binary bits of the block from MSB to

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

LSB fit column-wise from top to bottom into this square matrix.

Step4. From the square matrix bits are taken diagonally upward starting from (√y , √y) cell in a right to left trajectory to generate the encrypted block of length y.

Step5. The cipher text is formed converting the encrypted binary string into characters.

3.2 Decryption Algorithm:

Step1. The encrypted file i.e. the cipher text is considered as a stream of binary bits.

Step2. Processing the session key information, this binary string breaks into manageable-sized blocks.

Step3. Square matrix of order √y is generated for each block of length y. The binary bits of the block from MSB to LSB fit diagonally upward starting from (√y , √y) cell in a right to left trajectory into the square matrix.

Step4. The decrypted binary string is generated taking the bits column-wise from the square matrix.

Step5. The plain text is reformed converting the decrypted binary string into characters.

3.3 Generation of Session Key:

During encryption a session key is generated for one time use in a session of transmission to ensure much more security to MBBET. This technique divides the input binary bit stream dynamically into 16 portions, each portion is divided again into x no. of blocks with block length y bits. The final (i.e. 16^{th}) portion is divided into x_{16} no. of block with block length 4 bits (i.e. $y_{16} = 4$). So no padding is required. Total length of the input binary string is

$$x_1 * y_1 + x_2 * y_2 + \ldots\ldots\ldots + x_{16} * y_{16}.$$

The values of x and y are generated dynamically. The session key contains the sixteen set of values of x and y respectively.

4. EXAMPLE

To illustrate the MBBET, consider a two letter's word "Go". The ASCII values of "G" and "o" are 71 (01000111) and 111 (01101111) respectively. Corresponding binary bit representation of that word is "0100011101101111". Consider a block with length 16 bits as

0	1	0	0	0	1	1	1	0	1	1	0	1	1	1	1

Now these bits from MSB to LSB fit column-wise from left to right into this square matrix of order 4 as follows:

0	0	0	1
1	1	1	1
0	1	1	1
0	1	0	1

The encrypted binary string is formed taking the bits diagonally upward starting from (4 , 4) cell in a right to left trajectory from above the square matrix as follows:

1	0	1	1	1	1	0	1	1	1	0	1	0	1	0	0

The equivalent decimal no. of two 8 bit binary numbers 10111101 and 11010100 are 189 and 212 respectively. 189 and 212 are ASCII values of the characters ½ (Vulgar Fraction One Half) and Ô (Latin Capital Letter O with Circumflex) respectively. So the word **Go** is encrypted as **½Ô**.

For decryption, exactly reverse steps of the above are followed.

5. RESULTS AND ANALYSIS

In this section the comparative study between Triple-DES (168bits), AES (128bits) and MBBET has done on 20 files of 8 different types with file sizes varying from 330 bytes to 62657918 bytes (59.7 MB). Results are generated for analysis and comparison of encryption time, decryption time, Character frequencies, Chi-square values, Avalanche effects, Strict Avalanche effects and Bit Independence. All implementation has been done using JAVA.

5.1 Analysis of Encryption and Decryption Time:

Time taken is the difference between processor clock ticks at the starting and at the end of execution. Since the CPU clock ticks taken as time, there might be a slight variation in actual time. This variation is insignificant and may be ignored. Table 1 and Table 2 show the encryption and decryption times for Triple-DES (168bits), AES (128bits) and MBBET against the different files.

Table 1: Encryption times for TDES, AES and MBBET

Sl. No.	Source File Size (in bytes)	File type	Encryption Time (in seconds)		
			TDES	AES	MBBET
1	330	Dll	0.001	0.001	0.003
2	528	Txt	0.001	0.001	0.005
3	96317	Txt	0.034	0.004	0.021
4	233071	Rar	0.082	0.011	0.052
5	354304	Exe	0.123	0.017	0.073
6	536387	Zip	0.186	0.023	0.118
7	657408	Doc	0.220	0.031	0.168
8	682496	Dll	0.248	0.031	0.179
9	860713	Pdf	0.289	0.038	0.206
10	988216	Exe	0.331	0.042	0.237
11	1395473	Txt	0.476	0.059	0.261
12	4472320	Doc	1.663	0.192	0.356
13	7820026	Avi	2.626	0.334	0.632
14	9227808	Zip	3.096	0.397	0.473
15	11580416	Dll	4.393	0.544	0.777
16	17486968	Exe	5.906	0.743	1.784
17	20951837	Rar	7.334	0.937	1.586
18	32683952	Pdf	10.971	1.350	1.985
19	44814336	Exe	15.091	1.914	2.917
20	62657918	Avi	21.133	2.689	5.269

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

Table 2: Decryption times for TDES, AES and MBBET

Sl. No.	Source File Size (in bytes)	File type	Decryption Time (in seconds)		
			TDES	AES	MBBET
1	330	Dll	0.001	0.001	0.003
2	528	Txt	0.001	0.001	0.006
3	96317	Txt	0.035	0.008	0.024
4	233071	Rar	0.087	0.017	0.062
5	354304	Exe	0.128	0.025	0.081
6	536387	Zip	0.202	0.038	0.135
7	657408	Doc	0.235	0.045	0.188
8	682496	Dll	0.266	0.046	0.206
9	860713	Pdf	0.307	0.060	0.221
10	988216	Exe	0.356	0.070	0.268
11	1395473	Txt	0.530	0.098	0.283
12	4472320	Doc	1.663	0.349	0.394
13	7820026	Avi	2.832	0.557	0.684
14	9227808	Zip	3.377	0.656	0.518
15	11580416	Dll	4.652	0.868	0.847
16	17486968	Exe	6.289	1.220	1.948
17	20951837	Rar	8.052	1.431	1.736
18	32683952	Pdf	11.811	2.274	2.166
19	44814336	Exe	16.253	3.108	3.208
20	62657918	Avi	22.882	4.927	5.756

Proposed MBBET takes very less time to encrypt/decrypt than Triple-DES and little bit more time than AES. Fig. 1(a) and Fig. 1(b) show the graphical representation of encryption time and decryption time against file size in logarithmic scale. The graphs show that the encryption/decryption time increases with the increase of source file sizes. There is not much difference observed between the encryption and decryption times, which indicate that the complexity of computation for all the processes is approximately similar.

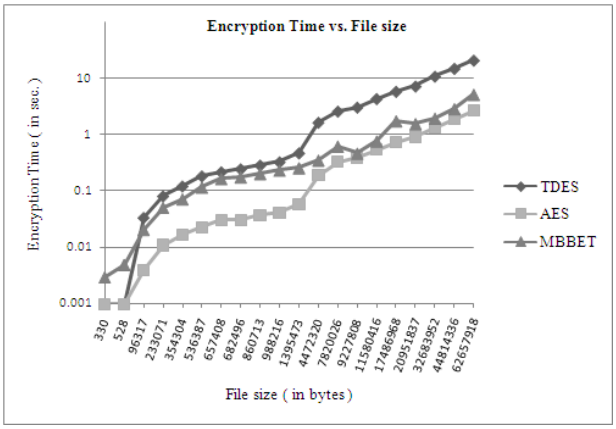


Fig. 1(a): Graphical representation of encryption times against file sizes in logarithmic scale

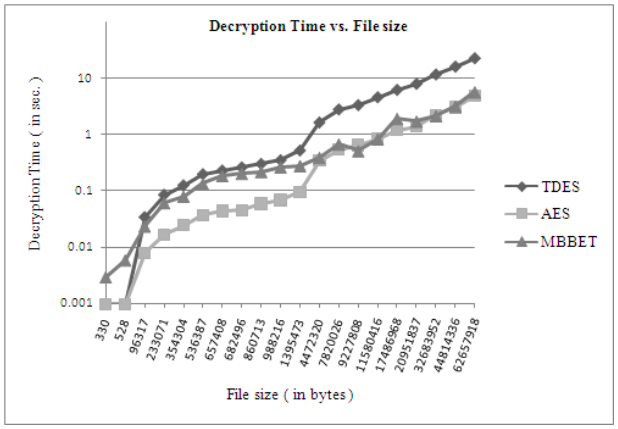


Fig. 1(b): Graphical representation of decryption times against file sizes in logarithmic scale

5.2 Analysis of Character Frequencies:

Analysis of Character frequencies for text file has been performed for T-DES, AES and proposed MBBET. Fig. 2(a) shows the spectrum of frequency distribution of characters in the plain text. Fig. 2(b), 2(c) and 2(d) show the spectrum of frequency distribution of encrypted characters in cipher text using T-DES, AES and MBBET.

In source files, some characters appear with very high and with very low frequencies and some characters appear with zero frequency. In encrypted files all characters with ASCII values ranging from 0 to 255 appear with certain frequencies and all these characters are approximately equally distributed over a certain range. Therefore it is very difficult to detect the actual message for a cryptanalyst. Since the frequency spectrum is smoother, so the degree of security of proposed MBBET is very high and is comparable with that of TDES and AES.

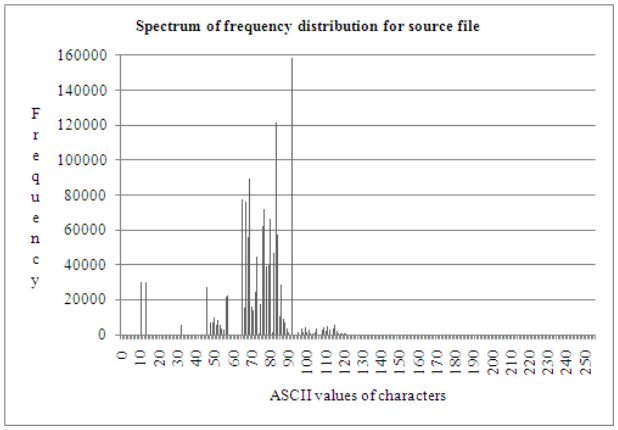


Fig. 2(a): Graphical representation of the spectrum of frequency distribution of characters for source file

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

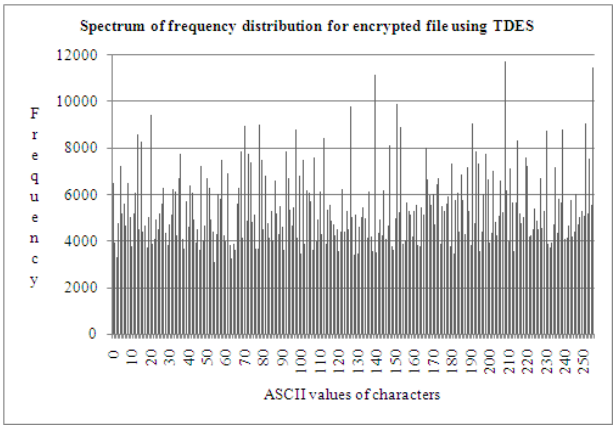


Fig. 2(b): Graphical representation of the spectrum of frequency distribution of characters for encrypted file using TDES

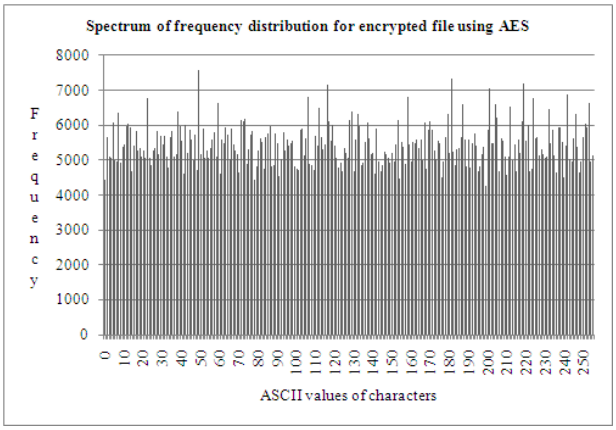


Fig. 2(c): Graphical representation of the spectrum of frequency distribution of characters for encrypted file using AES

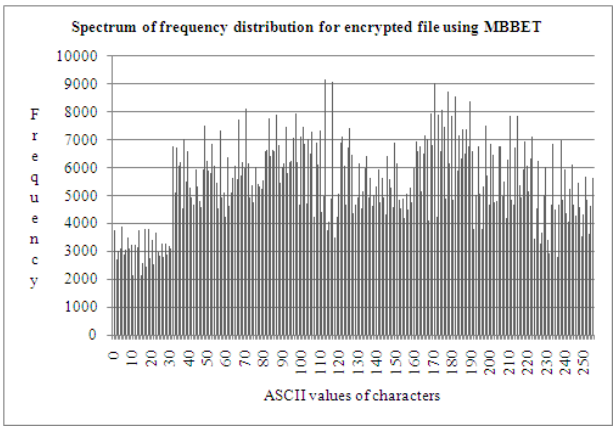


Fig. 2(d): Graphical representation of the spectrum of frequency distribution of characters for encrypted file using MBBET

5.3 Tests for Non-homogeneity:

The test for goodness of fit (Pearson χ^2) has been performed between the source files and the encrypted files. The large Chi-Square values (compared with tabulated values) may confirm the high degree of non-homogeneity between the source files and the encrypted files. Table 3 shows the Chi-Square values for Triple-DES (168bits), AES (128bits) and proposed MBBET for the different files.

Table 3: Chi-Square values for TDES, AES and MBBET

Sl. No	Source File Size (bytes)	File type	Chi-Square Values		
			TDES	AES	MBBET
1	330	Dll	922	959	838
2	528	Txt	1889	1897	1868
3	96317	Txt	23492528	23865067	20607612
4	233071	Rar	997	915	938
5	354304	Exe	353169	228027	192328
6	536387	Zip	3279	3510	3220
7	657408	Doc	90750	88706	85541
8	682496	Dll	29296	28440	25765
9	860713	Pdf	59797	60661	55016
10	988216	Exe	240186	245090	251170
11	1395473	Txt	5833237390	5545862604	5598716397
12	4472320	Doc	102678	102581	97200
13	7820026	Avi	1869638	1326136	1080704
14	9227808	Zip	37593	37424	35680
15	11580416	Dll	28811486	17081530	16317171
16	17486968	Exe	8689664	8463203	7778167
17	20951837	Rar	25615	24785	25688
18	32683952	Pdf	13896909	13893011	14803965
19	44814336	Exe	97756312	81405043	690015596
20	62657918	Avi	3570872	3571648	3789041

Fig. 3 graphically represents the Chi-Square values on logarithmic scale for T-DES, AES and MBBET. The calculated Chi-square values are much higher than the tabulated Chi-square values. Fig. 3 indicates that the Chi-Square values of MBBET are at par with and sometimes better than that of T-DES and AES. From this point of view it is concluded that the degree of security of the proposed MBBET is very high and very much comparable with that of the techniques TDES and AES.

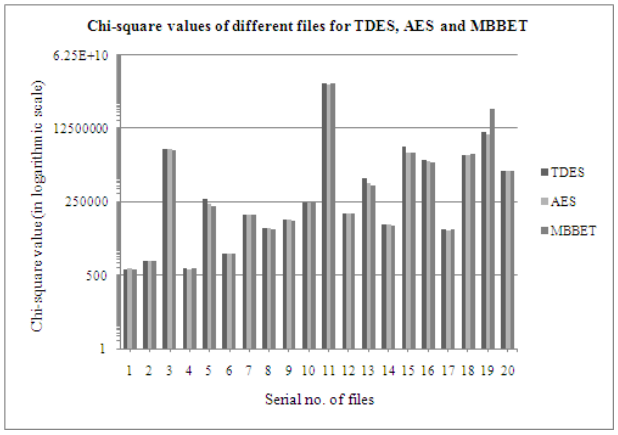


Fig. 3: Chi-Square values for TDES, AES & MBBET in logarithmic scale

5.4 Studies on Avalanche Effects, Strict Avalanche Effects and Bit Independence Criterion:

Avalanche, Strict Avalanche and Bit Independence are cryptographic test methods which measures the degree of security of any cryptographic technique. The bit changes among encrypted bytes for a single bit change in the original message sequence for the entire or a relative large number of bytes. The ratio of calculated standard deviation

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

with expected value gives the Avalanche and Strict Avalanche values on a 0.0 – 1.0 scale. The values of Avalanche, Strict Avalanche and Bit Independence closer to 1.0 indicate the high degree of security.

Table 4, 5 and 6 show the Avalanche values, the Strict Avalanche values and the Bit Independence values respectively. For all three techniques, the Avalanche and Strict Avalanche values varies from 0.9772 to 1.0.

Table 4: Avalanche values for TDES, AES and MBBET

Sl. No.	Source File Size (in bytes)	File type	Avalanche values		
			TDES	AES	MBBET
1	330	dll	0.99591	0.98904	0.97721
2	528	txt	0.99773	0.99852	0.98648
3	96317	txt	0.99996	0.99997	0.99442
4	233071	rar	0.99994	0.99997	0.99658
5	354304	exe	0.99996	0.99999	0.99647
6	536387	zip	0.99996	0.99994	0.99761
7	657408	doc	0.99996	0.99999	0.99784
8	682496	dll	0.99998	1.00000	0.99662
9	860713	pdf	0.99996	0.99997	0.99763
10	988216	exe	1.00000	0.99998	0.99778
11	1395473	txt	1.00000	1.00000	0.99752
12	4472320	doc	0.99999	0.99997	0.99525
13	7820026	avi	1.00000	0.99999	0.99886
14	9227808	zip	1.00000	1.00000	0.99962
15	11580416	dll	1.00000	0.99999	0.99844
16	17486968	exe	1.00000	0.99999	0.99898
17	20951837	Rar	1.00000	1.00000	0.99848
18	32683952	pdf	0.99999	1.00000	0.99963
19	44814336	exe	0.99997	0.99997	0.99876
20	62657918	Avi	0.99999	0.99999	0.99966

Table 5: Strict Avalanche values for TDES, AES and MBBET

Sl. No.	Source File Size (in bytes)	File type	Strict Avalanche values		
			TDES	AES	MBBET
1	330	Dll	0.98645	0.98505	0.97866
2	528	Txt	0.99419	0.99311	0.97884
3	96317	Txt	0.99992	0.99987	0.98225
4	233071	Rar	0.99986	0.99985	0.99568
5	354304	Exe	0.99991	0.99981	0.99762
6	536387	Zip	0.99988	0.99985	0.99779
7	657408	Doc	0.99989	0.99990	0.99792
8	682496	Dll	0.99990	0.99985	0.99878
9	860713	Pdf	0.99990	0.99993	0.99993
10	988216	Exe	0.99995	0.99995	0.99868
11	1395473	Txt	0.99990	0.99996	0.99827
12	4472320	Doc	0.99998	0.99995	0.99754
13	7820026	Avi	0.99996	0.99996	0.99884
14	9227808	Zip	0.99997	0.99998	0.99956
15	11580416	Dll	0.99992	0.99998	0.99894
16	17486968	Exe	0.99996	0.99997	0.99941
17	20951837	Rar	0.99998	0.99996	0.99947
18	32683952	Pdf	0.99997	0.99998	0.99962
19	44814336	Exe	0.99991	0.99990	0.99984
20	62657918	Avi	0.99997	0.99998	0.99976

Table 6: Bit Independence values for TDES, AES and MBBET

Sl. No.	Source File Size (in bytes)	File type	Bit Independence values		
			TDES	AES	MBBET
1	330	Dll	0.49180	0.47804	0.43913
2	528	Txt	0.22966	0.23056	0.21851
3	96317	Txt	0.41022	0.41167	0.42774
4	233071	Rar	0.99899	0.99887	0.98962
5	354304	Exe	0.92538	0.92414	0.93613
6	536387	Zip	0.99824	0.99753	0.99558
7	657408	Doc	0.98111	0.98030	0.98534
8	682496	Dll	0.99603	0.99560	0.98770
9	860713	Pdf	0.97073	0.96298	0.97948
10	988216	Exe	0.91480	0.91255	0.94082
11	1395473	Txt	0.25735	0.25464	0.24981
12	4472320	Doc	0.98881	0.98787	0.96749
13	7820026	Avi	0.98857	0.98595	0.98784
14	9227808	Zip	0.99807	0.99817	0.98972
15	11580416	Dll	0.86087	0.86303	0.86849
16	17486968	Exe	0.83078	0.85209	0.85937
17	20951837	Rar	0.99940	0.99937	0.98816
18	32683952	Pdf	0.95803	0.95850	0.96862
19	44814336	Exe	0.70104	0.70688	0.82855
20	62657918	Avi	0.99494	0.99451	0.99776

Fig. 4(a), 4(b) and 4(c) show graphical representation of the Avalanche, Strict Avalanche and Bit Independence values respectively for TDES, AES and MBBET. Figures indicate that the Avalanche, Strict Avalanche and Bit Independence values of MBBET are at par with that of TDES and AES. So it is concluded that the degree of security of MBBET is very high and is comparable with that of TDES and AES techniques.

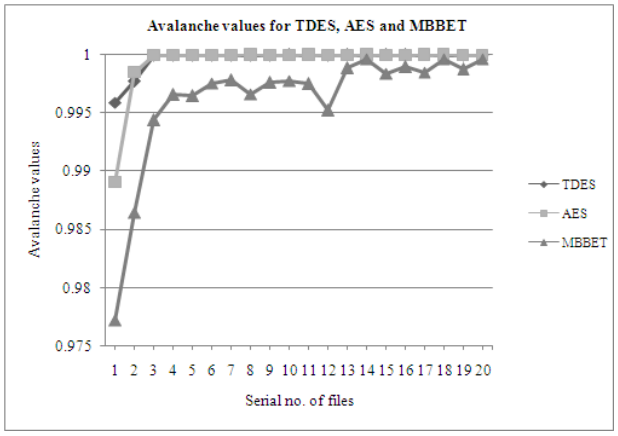


Fig. 4(a): Pictorial representation of Avalanche values of input bit streams for TDES, AES and MBBET

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.6, Pages : 64-69 (2013)
Special Issue of ICETEM 2013 - Held on 29-30 November, 2013 in Sree Visvesvaraya Institute of Technology and Science, Mahabubnagar – 204, AP, India

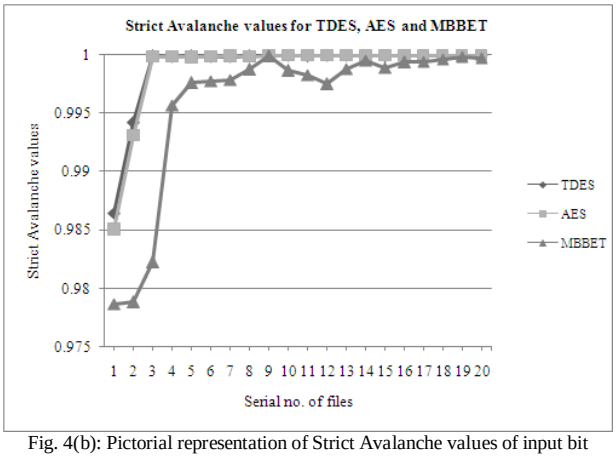


Fig. 4(b): Pictorial representation of Strict Avalanche values of input bit streams for TDES, AES and MBBET

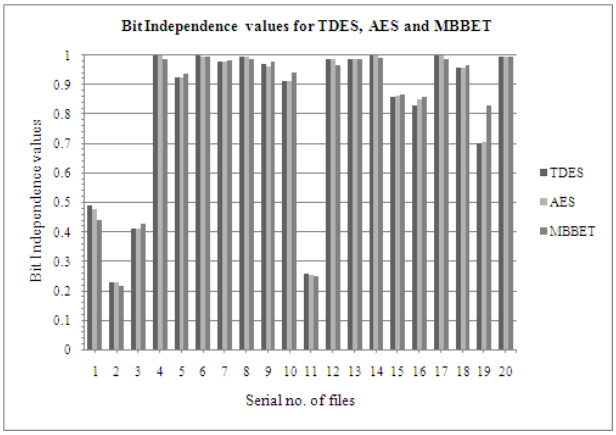


Fig. 4(c): Pictorial representation of Bit Independence values of input bit streams for TDES, AES and MBBET

6. CONCLUSION

The MBBET, the proposed algorithm presented in this paper is simple to understand and easy to implement. The session based key varies from session to session for any particular file which may enhance the security features. Results and Analysis section indicates that the MBBET is comparable with industry accepted standards T-DES and AES. The performance of MBBET is significantly better than T-DES algorithm. For large files, MBBET is at par with AES algorithm. This proposed technique is applicable to ensure high security in message transmission of any form.

Change the size of the secret key to make it complex is the future scope of this work.

ACKNOWLEDGEMENT

The authors express deep the sense of gratitude to the Department of Computer Application, JIS College of Engineering, Kalyani, West Bengal, India, for providing the facility to carry out the work.

REFERENCES

[1] B. K. Mandal, D. Bhattacharyya and S. K. Bandyopadhyay, "Designing and Performance Analysis of a Proposed Symmetric Cryptography Algorithm", *International Conference on Communication Systems and Network Technologies (CSNT 2013)*, 6-8 April 2013, Gwalior, India, Pages: 453 – 461.

[2] H. Cheng and Q. Ding, "Overview of the Block Cipher", *Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC 2012)*, 8-10 December 2012, Harbin, China, Pages: 1628 – 1631.

[3] M. Niemiec and L. Machowski, "A new symmetric block cipher based on key-dependent S-boxes", *4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2012)*, 3-5 October 2012, St. Petersburg, Russia, Pages: 474 – 478.

[4] V. Gupta, G. Singh and R. Gupta, "Advance cryptography algorithm for improving data security", *International Journal of Advanced research in Computer Science and Software Engineering*, January 2012, Vol. 2 – Issue 1.

[5] S. Som, N. S. Chatergee and J. K. Mandal, "Key Based Bit Level Genetic Cryptographic Technique (KBGCT)", *7th International Conference on Information Assurance and Security (IAS)*, 5-8 December 2011, Melaka, Malaysia, Pages: 240-245.

[6] A. H. Navin, A. R. Oskuei, A. S. Khashandarag and M. Mirnia, "A novel approach cryptography by using Residue Number System", *6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2011)*, November 29 - December 01 2011, Seogwipo, South Korea, Pages: 636 – 639.

[7] S. Samanta and M. Ghosh, "AN APPLICATION OF ROTATION BASED BIT LEVEL ENCRYPTION STRATEGY", *INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES AND APPLICATIONS IN ENGINEERING, TECHNOLOGY AND SCIENCES*, July – December 2011, Vol. 4, Issue 2, Pages: 142 – 143.

[8] A. Singh and R. Gilhotra, "DATA SECURITY USING PRIVATE KEY ENCRYPTION SYSTEM BASED ON ARITHMETIC CODING", *International Journal of Network Security & Its Applications (IJNSA)*, May 2011, Vol. 3, No. 3, Pages: 58-67.

[9] S. Dasgupta , S. Mazumder and P. Paul, "Implementation of Information Security based on Common Division", *International Journal of Computer Science and Network Security (IJCSNS)*, February 2011, Vol. 11 No. 2, Pages: 51-53.

[10] S. Gupta , S. Chatterjee and M. Bhattacharyya, "Variable Length Cumulative Key Size and Seed Based Symmetric Key Cryptographic Algorithm Using Composite 3-D Transposition Substitution and Chaining Technique", *International Conference on Advances in Communication, Network, and Computing (CNC '10)*, 04-05 October 2010, Calicut, Kerala, India , Pages: 109-113.

[11] "Triple Data Encryption Standard", *FIPS PUB 46-3 Federal Information Processing Standards Publication, Reaffirmed*, 1999 October 25 U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology.

[12] "Advanced Encryption Standard", *Federal Information Processing Standards Publication 197*, November 26, 2001.