Triple SV: A Bit Level Symmetric Block Cipher Having High Avalanche Effect

Rajdeep Chakraborty
Department of Computer Science & Engineering
Netaji Subhash Engineering College
Kolkata, West Bengal, India

Sonam Agarwal
Department of Computer Science
Netaji Subhash Engineering College
Kolkata, West Bengal, India

Sridipta Misra, Vineet Khemka, Sunit Kr Agarwal
Department of Information Technology
Netaji Subhash Engineering College
Kolkata, West Bengal, India

J. K. Mandal
Department of Computer Science and Engineering
University of Kalyani
Kalyani, West Bengal, India

Abstract— **The prolific growth of network communication system entails high risk of breach in information security. This substantiates the need for higher security for electronic information. Cryptography is one of the ways to secure electronic documents. In this paper, we propose a new block cipher, TRIPLE SV (3SV), with 256-bit block size and 112-bit key length. Generally, stream ciphers produce higher avalanche effect but Triple SV shows a substantial rise in avalanche effect with a block cipher implementation. The CBC mode has been used to attain higher avalanche effect. The technique is implemented in C language and has been tested for feasibility.**

Keywords- Avalanche Effect; Block Cipher; Cipher Block Chaining (CBC) mode; Cryptography.

I. INTRODUCTION

With the transformation of classical [1] methodology in business to conventional methodology, security of electronic documents is in need. A Cipher is something that converts the actual document into a format that cannot be recognized by anyone except the sender and intended receiver. One of the vital considerations for a good cipher is its *avalanche effect.* This effect obviates the brute force attack to a greater extent. Modern techniques of encryption either use a single symmetric key or two asymmetric keys. The algorithm is called as Symmetric Key Cryptography if only one key is used on both ends of the communication and it is called as a Public Key Cryptography if two distinct keys are used. The proposed technique is based on Private Key Cryptography in Cipher Block Chaining mode [1,2] with high avalanche effect i.e. a one-bit change in a plaintext affects all following cipher-text blocks [2,3].

The Section II of this paper deals with the proposed technique. A concept of key-generation [1] is given in Section III. Results and comparisons are illustrated in Section IV. Conclusions are drawn in Section V.

II. TRIPLE SV (3SV)

The Triple SV is a block cipher that uses secret key encryption. This algorithm takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher-text bit string of the same length. The proposed block size is 256 bits.

The Key comprises 112 bits. Fig. 1 summarizes the overall structure of Triple SV.

A. Modes of Operation

Like other block ciphers, Triple SV by itself is not a secure means of encryption but must instead be used in one of the several modes of mode of operation, like Electronic codebook (ECB), Cipher-block chaining (CBC), Propagating cipher-block chaining (PCBC), Cipher feedback (CFB), and Output feedback (OFB). We have designed and implemented Triple SV in CBC mode

In the CBC mode, each block of plaintext is XORed with the previous cipher-text block before being encrypted. This way, each cipher-text block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an initialisation vector must be used in the first block.

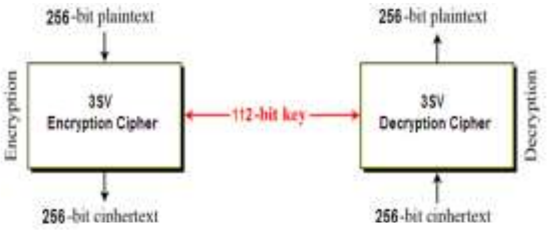

Figure 1. Overview of the Triple SV

A one-bit change in a plaintext affects all following cipher-text blocks. A plaintext can be recovered from just two adjacent blocks of cipher-text. As a consequence, decryption can be parallel zed, and a one-bit change to the cipher-text causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext. Fig. 2 and Fig. 3 represent the encryption and the decryption process of CBC mode.

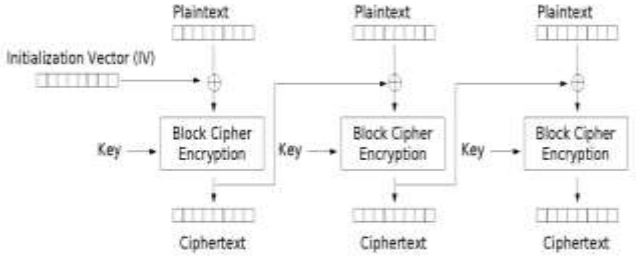

Figure 2. The Cipher Block Chaining (CBC) mode (encryption)

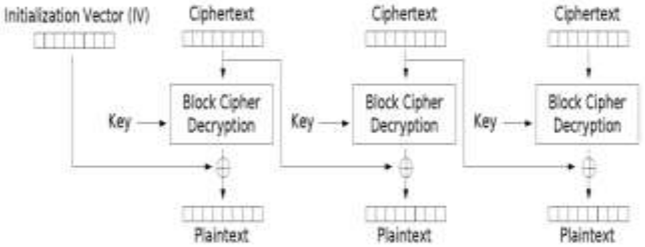

Figure 3. The Cipher Block Chaining (CBC) mode (decryption)

B. *Encryption*

The algorithm's overall structure (for encryption) is shown in Fig. 4.There are basically seven similar levels of processing. In addition there is also an initial and final inversion operation. The seven similar levels of processing have identical structure but differ in the number of consecutive ***n*** bits out of the input 256 bits to each level, which are coupled together and treated as a single entity while being processed inside each level. The values that *n* take in the 7 distinct levels are 2, 4, 8, 16, 32, 64, 128, (i.e. 2 $^{\text{(level number)}}$), respectively. Hence the seven levels of processing are named as 2-bit level, 4-bit level, 8-bit level, 16-bit level, 32-bit level, 64-bit level, and 128-bit level, respectively.

1) *n-Bit Level Structure*

Fig. 5 explains the entire construct of the n-bit level. Each level basically comprises three major functions, namely, Far Swapping, Near Swapping and Expansion Function, and a XOR Function.

The 256-bit input to the level first undergoes an n-bit far swap. The 256-bit output of the n-bit far swap is then introduced to an n-bit near swap, which again generates a 256-bit output.

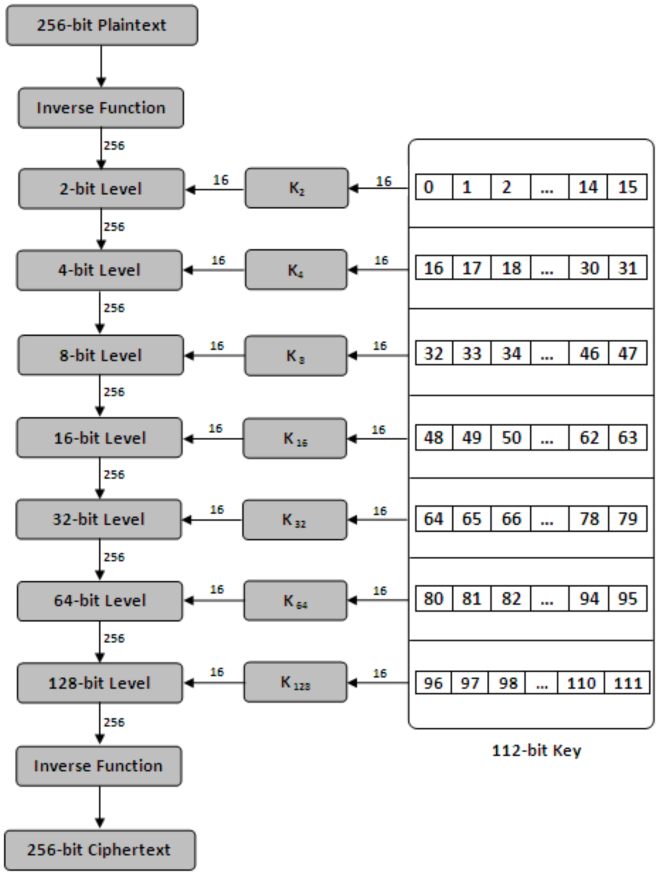

Figure 4. Triple SV Encryption Overview

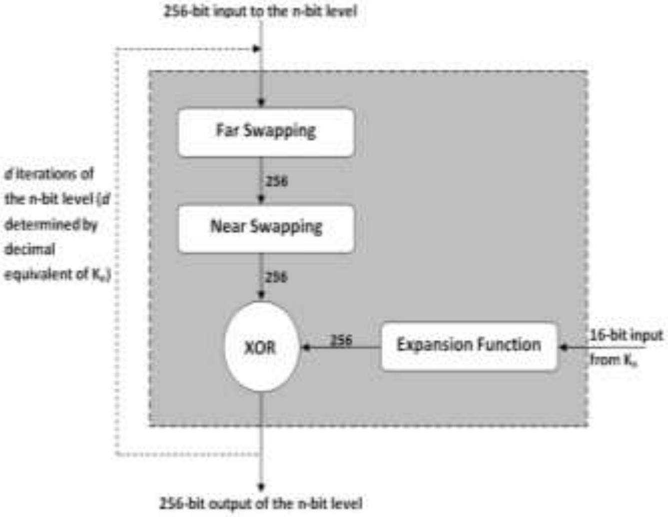
256-bit input to the n-bit level
Far Swapping
256
Near Swapping
256
d iterations of the n-bit level (d determined by decimal equivalent of Kₙ)
XOR
256
Expansion Function
16-bit input from Kₙ
256-bit output of the n-bit level

Figure 5. n-BIT Level Structure (Encryption)

Meanwhile, the 16-bits string, K_n enters the level and get expanded into a 256-bit string with which the 256-bit output of the near swap gets XORed to produce a 256-bit intermediate. This intermediate is again fed into the same level to carry out the procedure all over again. This iterative operation of the level continues for *d* time, where *d* is a positive integer, the value of which is determined by the decimal equivalent of the string K_n.

After *d* complete iterations of the level, the 256-bit output is the output of that level and is carried to the next *n*-bit level for similar series of operations.

a) n-Bit Far Swap Function: The *n*-bit far swap function has been diagrammatically depicted in Fig.6. The *n*-bit far swap function is a simple function. Firstly, the 256-bit of the incoming string are grouped into distinct *n*-bit groups, where n is 2, 4, 8, 16, 32, 64, or 128, depending on the level at which we are operating. The groups are formed by starting from the first bit and grouping together the first ***n*** consecutive bits, then the next ***n*** consecutive bit, and so on. These distinct *n*-bit groups behave as individual entities at that particular level.

For *n*-bit far swapping, the first *n*-bit group gets swapped (interchanged) the last (farthest) *n*-bit group. The second *n*-bit group gets swapped with the penultimate *n*-bit group, and so on.

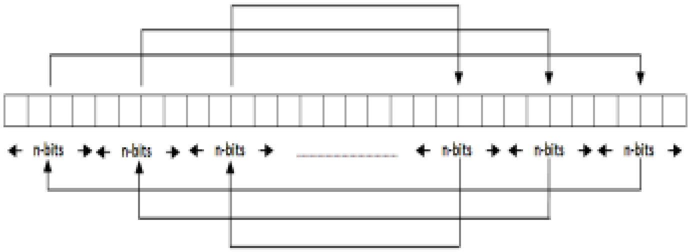

Figure 6. n-Bit Far Swap Function

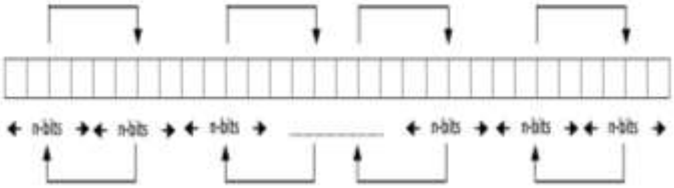

Figure 7. n-Bit Near Swap Function.

b) n-Bit Near Swap Function: The *n*-bit near swap function is quite similar to *n*-bit far swapping function, with a subtle change in swapping pattern, as demonstrated in Fig. 7. For *n*-bit near swapping, the first *n*-bit group gets swapped (interchanged) the second (nearest) *n*-bit group. The third *n*-bit group gets swapped with the fourth *n*-bit group, and likewise the penultimate *n*-bit group is swapped with the ultimate *n*-bit group.

c) Expansion Function (for encryption): The Expansion Function, in comparison to the earlier functions is a little more complex. For a certain n-bit level, the Expansion Function transforms the 16-bits string, K_n into a 256-bit string, which is used as an input to the XOR Function. Fig. 8 summarizes the expansion function for encryption.

The function takes the 16-bit K_n string as input. Next, it determines the number of iteration of the particular level (*a*). Then the K_n is given *a* left-rotations and the modifies string makes the first 16 bits of the expanded string. Next, another left-rotation is given to the first 16-bits to produce the next 16-bits, and so on.

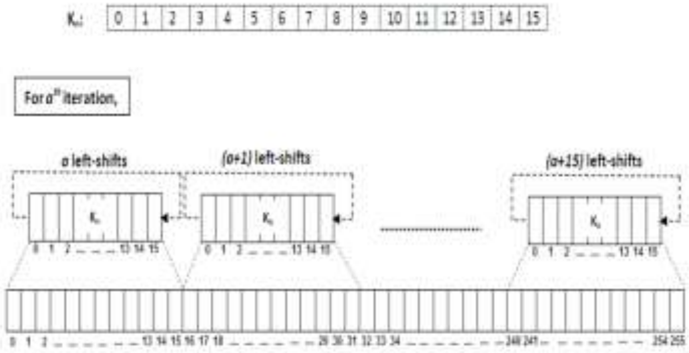

Figure 8. Expansion Function (for encryption)

Sixteen such modifications of the 16-bit string finally produce the 256-bit string for that particular level and that particular iteration. Thus, a 256-bit output is generated by the expansion function, which then gets XORed with the 256-bit output of the n-bit near swapping of that particular iteration of the level.

C. Decryption

The decryption algorithm is just the reverse of the encryption algorithm. In case of decryption, the 256-bit cipher text fed to the cipher first undergoes 128-bit level, then 64-bit level and so on till 2-bit level. Fig. 9 shows the decryption algorithm.

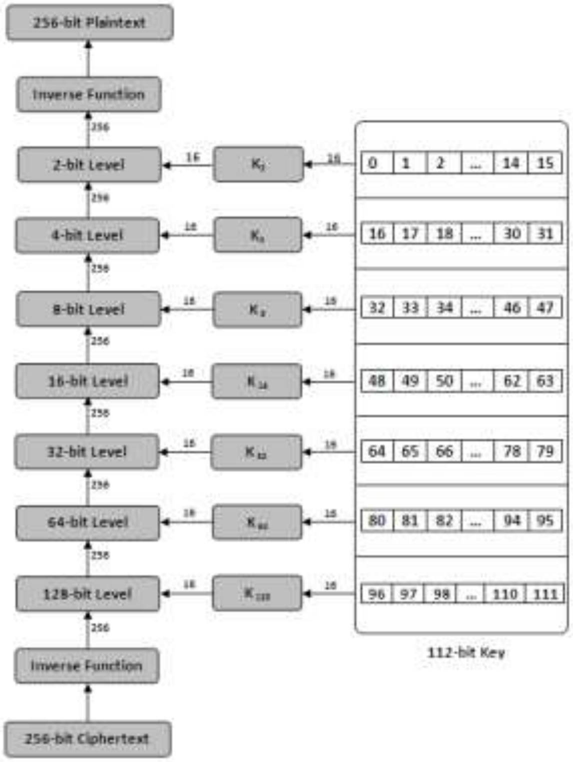

Figure 9. Triple SV Decryption Overview

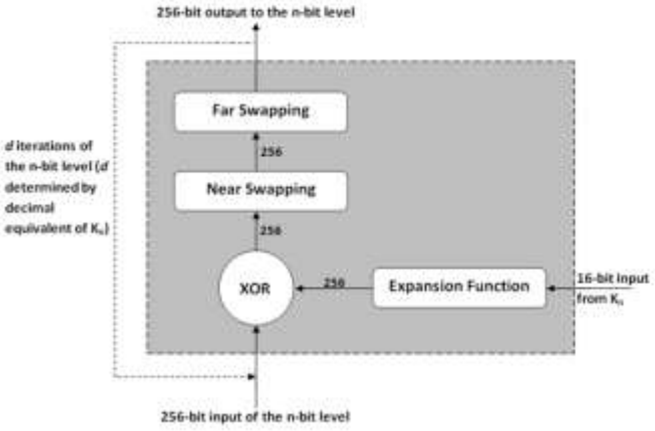
256-bit output to the n-bit level
Far Swapping
256
Near Swapping
256
XOR
256
Expansion Function
16-bit input from Kn
d iterations of the n-bit level (d determined by decimal equivalent of Kn)
256-bit input of the n-bit level

Figure 10. n-Bit Level Structure (Decryption)

Even the order of operations inside each level is reversed with the expansion function operating first, then the n-bit near swap and then the n-bit far swap, as depicted in Fig. 10.

All the individual function retains exactly the same functionality as in case of encryption. The only function that gets a little modified in case of decryption is the Expansion Function.

Fig. 11 clearly explains the functioning of the expansion function in case of decryption.

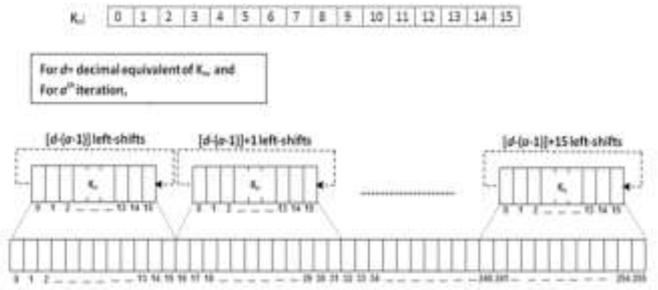

Figure 11. Expansion Function (for Decryption)

III. KEY STRUCTURE

Key length is one of the two most important security factors of any encryption algorithm—the other one being the design of the algorithm itself. The effective key length of Triple SV is 112 bits, giving 2^{112} possible combinations. The 112-bit key is completely user defined and is provided by the user in the form of numbers of iteration that each of the n-bit levels would have while the encryption or decryption process progresses. The 112 bits of the key have been logically divided into seven 16-bit binary sequences, each of which relates to a particular *n*-bit level. The association is elucidated below.

- Bit number 1 to 16 form string K2, and is associated with 2-bit level.
- Bit number 17 to 32 form string K4, and is associated with 4-bit level.
- Bit number 33 to 48 form string K8, and is associated with 8-bit level.
- Bit number 49 to 64 form string K16, and is associated with 16-bit level.
- Bit number 65 to 80 form string K32, and is associated with 32-bit level.
- Bit number 81 to 96 form string K64, and is associated with 64-bit level.
- Bit number 97 to 112 form string K128, and is associated with 128-bit level.

IV. RESULTS AND COMPARISONS

A. The Avalanche Effect:

Calculation of Avalanche Effect,

$$\boldsymbol{Avalanche\ Effect = \frac{No.of\ flipped\ bits\ in\ ciphertext}{No.of\ bits\ in\ the\ ciphertext} \times 100\%}$$

Avalanche Effect refers to a desirable property of any cryptographic algorithm where, if an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g., more than half the output bits flip).

Table I compares the avalanche effect ratio for 3SV, TDES and RSA algorithm and Fig.12 gives the graph of the same which are obtained after calculating the respective Avalanche Effect by making a change of a few (approx 3) characters in each file.

It is observed that the proposed technique is showing an average avalanche ratio percentage of 95.77%, which is way higher than that obtained using Triple DES or RSA. High avalanche ratio ensures higher security from brute force attack.

B. Frequency Distribution

The results [4] shown in Fig. 13, Fig. 14, Fig. 15 are obtained after calculating the respective Frequency Distributions of the source file 'ANSI.DOC' and the corresponding encrypted files using Triple SV, RSA and Triple DES.

TABLE I. COMPARISION OF AVALANCHE RATIO FOR 3SV, TDES AND RSA ALGORITHM

Index	**Filename**	**File Type**	**Size (In Kb)**	**Avalanche Ratio**		
				3SV	***TDES***	***RSA***
1	POINT	CPP	1	0.9860	0.9874	0.1651
2	Wuident	TEXT	1.2	0.9902	0.1427	0.1381
3	winword2	DOC	1.72	0.9977	0.0403	0.0042
4	Footer	HTML	1.73	0.9739	0.1496	0.1221
5	VCIRC	CPP	2.46	0.9873	0.1827	0.0712
6	CIRCLE	CPP	3	0.9850	0.0763	0.2144
7	Options	BMP	3.8	0.7337	0.2502	0.0105
8	Default	HTML	4.31	0.9797	0.0958	0.0948
9	Winword	DOC	4.5	0.9715	0.1939	0.0124
10	safe_easier	HTML	7.5	0.9844	0.0573	0.0981
11	best_road	HTML	8.16	0.9820	0.0976	0.0577
12	GREP2MSG	C	9	0.9854	0.0364	0.0357
13	desktop_icon_04	BMP	9.05	0.9766	0.1021	0.0234
14	start_windows	HTML	10	0.9824	0.0247	0.1349
15	WINWORD8	DOC	10.5	0.9718	0.2341	0.0719
16	TASM2MSG	CPP	17.2	0.9854	0.0361	0.0455
17	ANSI	DOC	23	0.9874	0.0394	0.0070
18	Greenstone	BMP	25.9	0.9884	0.0101	0.0060
19	Dberr	TEXT	27	0.9858	0.7562	0.0062
20	eula.txt	TEXT	32	0.9868	0.0205	0.0075
21	Setup	BMP	234	0.9773	0.0343	0.0087
22	CLASSLIB	DOC	255	0.9853	0.0006	0.0011
23	watermark_300x	BMP	351	0.7295	0.0251	0.0183
24	Ntbtlog	TEXT	443	0.8425	0.0004	0.0003
25	Setuplog	TEXT	758	0.9865	0.0003	0.0013

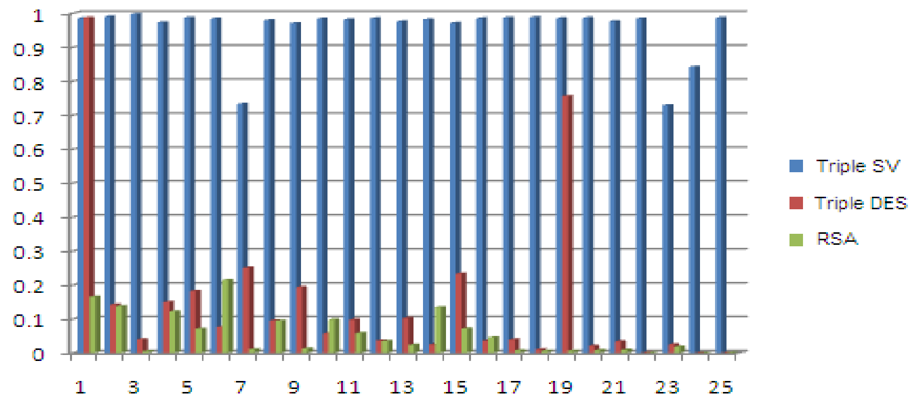

Figure 12. Graph for Comparison of Avalanche Ratio for 3SV, TDES & RSA

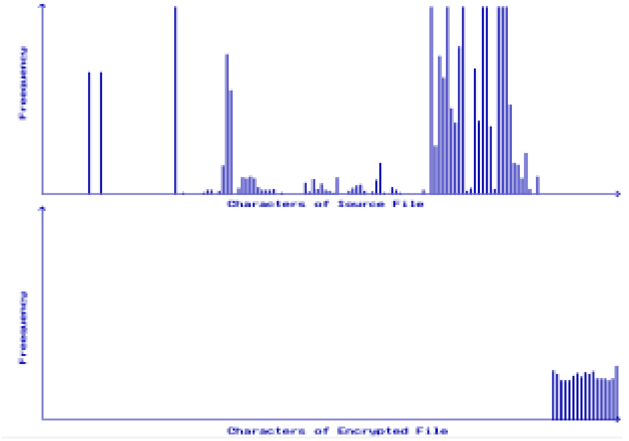

Figure 13. Graph for Distribution of Characters for Source file '*ANSI.DOC*' and Triple SV

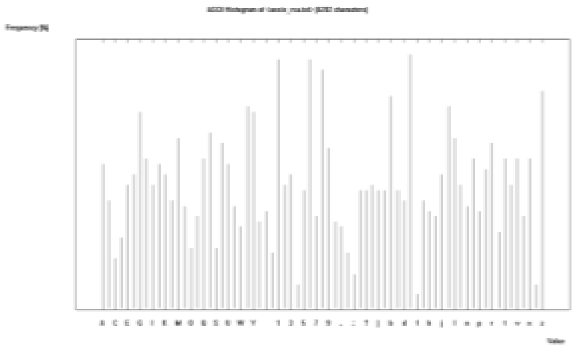

Figure 14. Graph for Distribution of Characters for RSA

From Fig. 13, it is substantiated that the frequency distribution of the proposed technique is well distributed over a range that is exclusive to the range of the source file. This makes statistical cryptanalysis [1,5,6,7] immensely improbable.

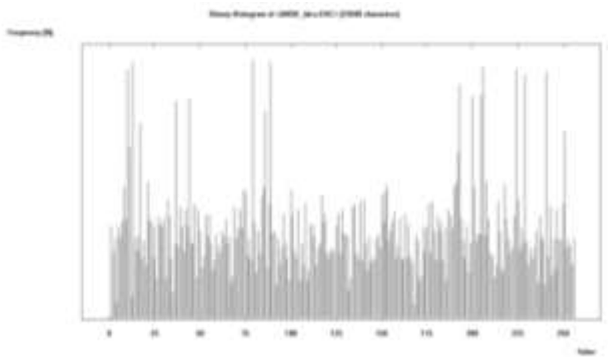

Figure 15. Graph for Distribution of Characters for TDES

C. Non-Homogeneity Test

Another way to analyse the technique is to test the non-homogeneity of the source and the encrypted file. The Chi-Square test has been performed for this purpose. Table II and Fig. 16 show the file size and the corresponding Chi-Square values for ten different files. The Chi-Square values for the proposed technique are comparatively lower than those obtained by RSA or TDES. The value of degree of freedom is on an average 127. Hence the source and the corresponding encrypted files are considered to be heterogeneous.

D. Time Complexity Analysis

This section compares the time complexity of the Triple SV with that of RSA and TDES by taking the encryption and decryption times into consideration.

The graphical analysis of the encryption and decryption time of Triple SV, RSA and TDES has been depicted in Fig. 17 and Fig. 18 respectively where x-axis represents the 25 test files (of increasing sizes varying from 1 KB to 758 KB) and y-axis represents the time taken for encryption/decryption (in seconds). The time complexity of the proposed technique is well comparable to RSA and TDES.

V. Conclusions

The cryptographic algorithm, Triple SV is a symmetric block cipher using a 256-bit block and 112-bit key. From the above discussions it can be inferred that Triple SV is potentially a promising algorithm, which can find its efficient implementation in different fields. Triple SV has a way better Avalanche Effect than any of the other existing algorithms and hence can be incorporated in the process of encryption of any plain text. The high avalanche ratio and a key size of 112 bits ensure sound security from brute force attacks. The implementation in CBC mode ensures low predictability and tougher cryptanalysis. Even the time complexity of the proposed algorithm is considerably viable and even better than RSA and TDES at many instances.

TABLE II. THE CHI SQUARE VALUES AND DEGREE OF FREEDOM FOR TRIPLE SV, TRIPLE DES AND RSA ALGORITHMS

Index	**Filename**	**File Type**	**Size (In KB)**	**Chi-Square Value**			**Degree Of Freedom**		
				3SV	***TDES***	***RSA***	***3SV***	***TDES***	***RSA***
1	POINT	CPP	1	1221	15417	15281	127	248	251
2	Wuident	TEXT	1.2	1443	60322	34433	126	254	255
3	winword2	DOC	1.72	2365	2413761	2457	101	223	245
4	Footer	HTML	1.73	2072	30826	24387	127	254	255
5	VCIRC	CPP	2.46	2923	39301	59753	127	255	255
6	CIRCLE	CPP	3	3071	57862	52331	127	255	255
7	Options	BMP	3.8	5019	841431	39438	127	254	254
8	Default	HTML	4.31	5143	45397	37919	127	255	254
9	Winword	DOC	4.5	9476	1314485	89517	125	252	255
10	safe_easier	HTML	7.5	8917	124503	119531	127	255	254
11	best_road	HTML	8.16	9694	144300	98956	127	255	255
12	GREP2MSG	C	9	10693	352680	295390	127	255	255
13	desktop_icon_04	BMP	9.05	8289	863031	76607	127	255	254
14	start_windows	HTML	10	11507	151721	114562	127	255	255
15	WINWORD8	DOC	10.5	11498	6891550	88675	127	254	255
16	TASM2MSG	CPP	17.2	20424	491314	52537	127	255	255
17	ANSI	DOC	23	26159	369304	374085	127	255	255
18	Greenstone	BMP	25.9	30007	330192	319512	127	255	255
19	Dberr	TEXT	27	31450	268039	263938	127	255	255
20	eula.txt	TEXT	32	33595	406353	286747	127	255	254
21	Setup	BMP	234	268322	42142512	7993738	127	255	255
22	CLASSLIB	DOC	255	300645	3929874	2622070	127	255	255
23	watermark_300x	BMP	351	242589	5249601	4218862	127	255	254
24	Ntbtlog	TEXT	443	524360	5484097	3694341	123	255	255
25	Setuplog	TEXT	758	897065	5551341	5395921	127	255	255

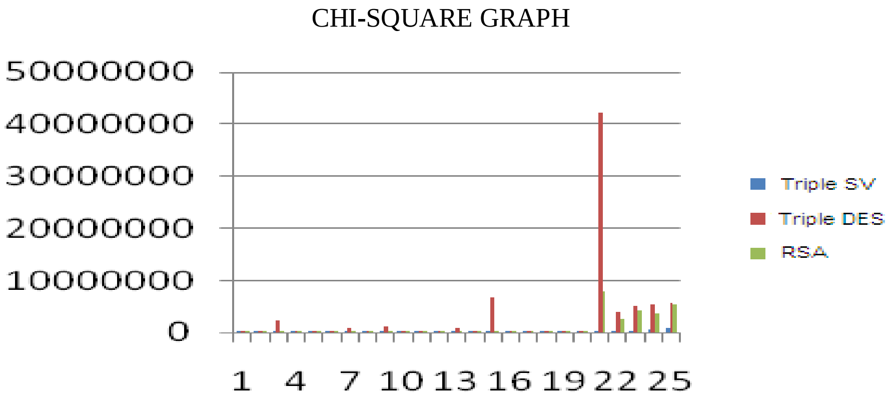

Figure 16. Graph for Comparison of Chi Square Values for Triple SV, Triple DES and RSA

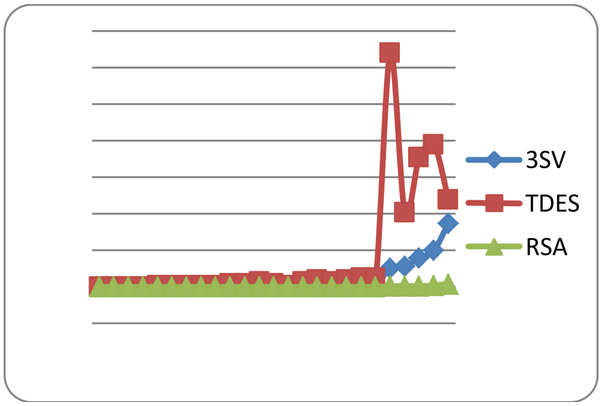

Figure 17. Graph for Comparison of Time Complexity for Encryption Time for Triple SV, Triple DES & RSA

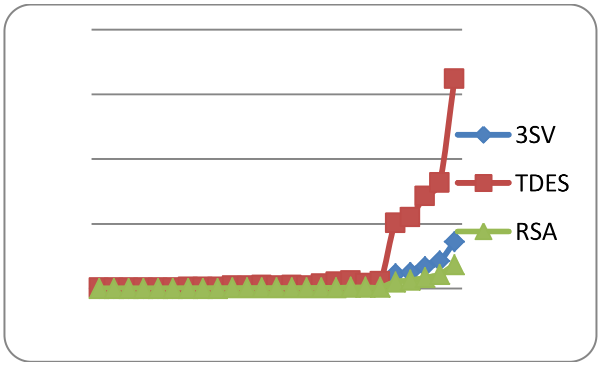

Figure 18. Graph for Comparison of Time Complexity for Decryption time for 3SV, TDES & RSA

ACKNOWLEDGMENT

The authors would like to acknowledge the Department of Computer Science and Information Technology, Netaji Subhash Engineering College, garia, Kolkata, India and the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nodia, West Bengal, India.

REFERENCES

[1]. B. Schneier, “Applied cryptography” John Wiley & Sons Inc., New York, New York, USA, 2nd edition, 1996.

[2]. http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation#Cipher-block_chaining_.28CBC.29

[3]. Sriram Ramanujam and Marimuthu Karuppiah, “Designing and algorithm with high avalanche effect,” *International Journal of Computer Science and Network Security*, VOL.11 No.1, January 2011.

[4]. The software cryptographic tools for educational purpose available at http://www.cryptool.com/.

[5]. W. Stallings, “Cryptography and Network Security: Principles and Practices,” Prentice Hall, Upper Saddle River, New Jersey, USA, Third Edition, 2003.

[6]. U.S. Department of Commerce/National Institute of Standard and Technology, FIPS PUB 197, Specification for the Advanced Encryption Standard (AES), November 2001. Available at http://csrc.nist.gov/encryption/aes.

[7]. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. “Handbook of Applied Cryptography”, CRC Press, Boca Raton, Florida, USA, 1997.

[8]. http://en.wikipedia.org/wiki/Data_Encryption_Standard.

[9]. Rajdeep Chakraborty, Dr. J.K.Mandal, “A Microprocessor-based Block Cipher through Rotational Addition Technique (RAT)”, ICIT – 2006 18-21 December, 2006, Bhubaneswar, INDIA.

[10]. “Data Encryption Standard,” Federal Information Processing Standards Publication No. 46, National Bureau of Standards, January 15, 1977.