DNA Secret Writing With Laplace Transform

Sukalyan Som
Department of Computer Science
Barrackpore Rastraguru Surendranath College,
Kolkata, West Bengal, India

Moumita Som
Department of Computer Science
Barrackpore Rastraguru Surendranath College,
Kolkata, West Bengal, India

ABSTRACT
A symmetric key cryptographic system has been proposed and it is termed as DSWLT. This proposed technique is very fast, suitable for encryption of large files. DSWLT consider the plain text (i.e. the input file) as binary string with finite no of bits. The input string converted to DNA nucleotides using DNA coding and then the DNA codes are converted to positive integers. Laplace transform is applied considering these numbers to be the co-efficient of the expansion. To provide multilevel security the resultant coefficients are converted to their binary equivalent and another level of encryption with cumulative XOR is performed and respective MSBs found at every iteration are taken to construct the cipher text. Decryption is performed in the reverse manner. Experimental results are tested, analyzed and a comparison with existing and industrially accepted TDES and AES has been performed.

General Terms
Network Security, Cryptography.

Keywords
DNA, DNA Cryptography, Laplace Transform, Symmetric key Cryptography, Cumulative XOR, Most Significant Bit, Serial Test, Monobit Test, Frequency Test

1. INTRODUCTION

1.1 DNA Encoding
Watson stated that the DNA strands can be useful to encode information [1]. Even though DNA cryptography is emerging and effective disciple of cryptography but it is not as much effective than traditional cryptography. It can be combined with existing cryptographic schemes to provide enhanced security [2] [3] [4]. DNA cryptography and steganography is a new field born from Adleman's research [5] in DNA computing and from Viviana Risca's project on DNA steganography [6].
Deoxyribo Nucleic Acid (DNA) is a long linear polymer found in the core part of a cell. DNA is made up of several nucleotides in the form of double helix and it is linked with the transmission of genetic information. Each spiral strand consist of sugar phosphate as backbone and bases are connected to a complementary strand by hydrogen bonding between paired bases Adenine(A), thymine(T), guanine(G) and cytosine(C). Adenine and thymine are connected by two hydrogen bonds while guanine and cytosine are connected by three. In its primitive stage, DNA cryptography is shown to be very effective. Currently, several DNA computing algorithms are proposed for cryptanalysis and steganography problems, and they are very powerful in these areas. The concept of DNA computing combined with fields of cryptography and steganography brings a new hope for powerful, or unbreakable, algorithms [7-9].

There are two complementary chains in the structure of DNA. Each nucleotide in DNA has a sugar component joined to a phosphate group at one point on the sugar, and to a nitrogen containing base attached at another point. The chains in DNA have the phosphate of one nucleotide linked to the sugar of the next nucleotide to form a strand of alternating sugars and phosphates with dangling nitrogenous bases as shown below in Fig 1.

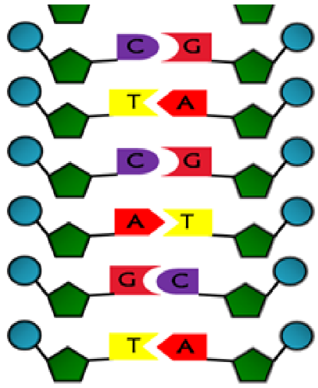

Fig 1: Combination of Nucleotide bases in Strands.

DNA contains two such chains, twisted around each other to form a double-stranded helix with the bases on the inside. Every A on one chain forms weak bonds with a T on the other strand, and every C on a strand bonds weakly to a G on the opposite chain. The two strands, held together weakly by the pairing of A with T, and G with C, are thus complementary, and the sequence in one can be deduced from the other's sequence. The basic DNA structure is shown below in Fig 2 [10].

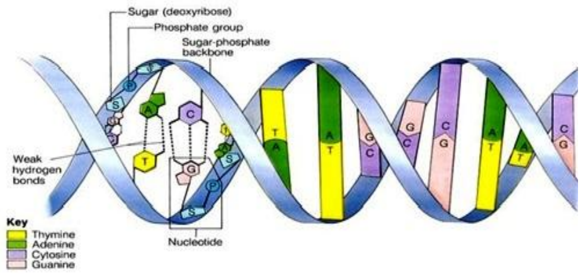

Fig 2: Basic DNA Structure.

These complimentary strands have codons as fundamental building blocks. Codons are basically triplets of nucleotide bases. Table 1 below shows codons forming DNA sequences in two complimentary strands.
As can be seen DNA nucleotide bases are existing in form of codons and are complimentary to each other that is A-T, G- C are complimentary to each other. We can use these codons for encoding and decoding of the data.

Table 1: Complimentary DNA Strands.

AGG	CTC	AAG	TCC	TAG	.,..
TCC	GAG	TTC	AGG	ATC	....

1.2 Laplace Transform

Let, F(t) be a function for t>0. Then L{F(t)} = f(s) = $\int_0^\infty e^{-st}.F(t)\,dt$ is called Laplace Transform of f(t), where the parameter s is positive real. [11]

Theorem 1: $L(e^{at}) = \frac{1}{s-a}$

Proof: $L(e^{at}) = \int_0^\infty e^{-st}.e^{at}dt$

$= Lt_{X\to\infty} \int_0^x e^{-st}.e^{at}dt$

$= Lt_{X\to\infty} \left[\frac{e^{(a-s)t}}{a-s}\right]_0^x$

$= \frac{1}{a-s} \underset{X\to\infty}{Lt} \left\{\frac{1}{e^{(s-a)x}} - 1\right\}$

$= \frac{1}{a-s}(0-1)$

$= \frac{1}{s-a}$

Theorem 2: Let, L{F(t)}=f(s), then $L\{t^n f(t)\} = (-1)^n.\frac{d^n}{ds^n} f(s)$ where n is positive integer

Proof: we have f(s) =

L{F(t)}$= L\{F(t) = \int_0^\infty e^{-st} F(t)dt$

Differentiating both side w.r.t. 's' we get,

$\frac{df}{ds} = \frac{d}{ds}\int_0^\infty e^{-st}.F(t)dt$

$= \int_0^\infty e^{-st}.F(t)dt$

$= \int_0^\infty \frac{\partial}{\partial s}[e^{-st}F(t)dt]$

$= \int_0^\infty -t.e^{-st}F(t)dt$

$= -\int_0^\infty e^{-st}\{tF(t)\}dt$

$= -L\{tF(t)\}$

$\therefore\ L\{tF(t)\} = -\frac{df}{ds}$ or $L\{tf(t)\} = -\frac{d}{ds}[L\{F(t)\}]$ (1)

Hence $L\{t.tF(t)\} = -\frac{d}{ds}L\{tF(t)\}$

i.e. $L\{t^2F(t)\} = (-1)^2\frac{d}{ds}\left(\frac{df}{ds}\right)$ By (1)

$= (-1)^2\frac{d^2f}{ds^2}$

So the result is true for n= 1, 2

Let us assume that the result is true for n=m, then

$L\{t^m F(t)\} = (-1)^m \frac{d^m f}{ds^m}$

Or, $\int_0^\infty e^{-st}.t^m F(t)dt = (-1)^m \frac{d^m f}{ds^m}$

Differentiating both sides with respect to 's' we get

$\frac{d}{ds}\int_0^\infty e^{-st}.t^m F(t)dt = (-1)^m \frac{d^{m+1}f}{ds^{m+1}}$

or, $\frac{d}{ds}\int_0^\infty e^{-st}.t^m F(t)]dt = (-1)^m \frac{d^{m+1}f}{ds^{m+1}}$

or, $\frac{d}{ds}\int_0^\infty -te^{-st}.t^m F(t)]dt = (-1)^m \frac{d^{m+1}f}{ds^{m+1}}$

or, $\frac{d}{ds}\int_0^\infty -e^{-st}t^{m+1}F(t)]dt = (-1)^m \frac{d^{m+1}f}{ds^{m+1}}$

$\therefore L\{t^{m+1}F(t)\} = (-1)^{m+1}\frac{d^{m+1}f}{ds^{m+1}}$

Which shows that the theorem is true for n = m+1.

Hence by mathematical induction, the theorem is true for all positive integer n.

Theorem 2: $L\{t^n e^{at}\} = \frac{n!}{(S-a)at}$

Proof: We have $L\{e^{at}\} = \frac{1}{(S-a)}$ [By Theorem I]

Therefore $L\{t^n e^{at}\} = (-1)^n \frac{dn}{dsn}\frac{1}{(S-a)}$

$= (-1)^n \frac{(-1)^n n!}{(S-a)^{n+1}}$

$= \frac{n!}{(S-a)^{n+1}}$

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[12], AES[13].Conclusions are drawn in section 6.

2. PROPOSED ALGORITM

The following algorithm provides an insight into the proposed DSWLT scheme. The sender converts the original message or plain text into cipher text using the following steps.

2.1 Method of Encryption

BEGIN

Step 1: Select the message, M, to be sent, and convert into an 8 bit Extended ASCII code, M_{bin}.

Step 2: Convert M_{bin} into DNA codes, say M_{dna} using the following convention: A=00, T=01, G=10, C=11 where A, T, G, C are DNA base pairs.

Step 3: DNA coded text is converted into Integer coded text, M_{Int} using the Lookup table mapping numeric value to base nucleotide as given in table 4.

Step 4: Each integer in M_{Int} is used as the coefficients of the Laplace transform of the function f(t) = $\boldsymbol{Gte^t}$ i.e. L{f(t)}= $\mathbf{L}\left\{\sum_{n=0}^{\infty} \mathbf{G_n}\frac{\mathbf{t^{n+1}}}{\mathbf{n!}}\right\}$, where Gn ≥ 0 ∀ n ≥ 8.

Step 5: The coefficients of $\mathbf{L}\{\sum_{n=0}^{\infty} \mathbf{G_n} \frac{t^{n+1}}{n!}\} = \sum_{n=0}^{\infty} \frac{c_n}{s^{n+2}}$ **i. e. C**n $\forall$ **n** ≥ 0 are taken M_{Lap} is constructed by storing $\mathbf{C_n}$ mod 128.

Step 6: Each integer of M_{Lap} is converted to its corresponding ASCII values (7 bit binary equivalents) and on them cumulative XOR as shown in Fig. 3 is performed. The results are stored in M_{XOR}.

Step 7: The ASCII equivalent of the values in M_{XOR} are stored as the cipher text, C.

END

2.2 Method of Decryption

BEGIN

Step 1: The cipher text C is converted ti its corresponding ASCII values (7 bit binary equivalent), C_{ASC}.

Step 2: Cumulative XOR operation is performed and resultant binary streams are converted back to their equivalent decimal form, C_{Lap} to get the coefficients of the Laplace transform back.

Step 3: Inverse Laplace transform is applied over C_{Lap} producing the coefficients which are considered as the Integer codes corresponding to DNA bases, C_{Int}.

Step 4: The Integer codes in CInt are mapped back with DNA bases using table 4 which produces M_{dna}.

Step 5: The DNA bases in M_{dna} are mapped back to their binary codes using table 3 to for M_{Bin}.

Step 6: The binary streams M_{Bin} in are converted to their corresponding ASCII values and hence producing the original message or the plain text M.

END

3. AN EXAMPLE

3.1 Method of Encryption

To illustrate the algorithm, let us consider a two letters word "Go". The ASCII values of "G" and "o" are 71 (01000111) and 111 (01101111) respectively. Corresponding binary bit representation of that word is shown in table 2

Table 2: Binary representation of ASCII coded "Go"

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

The binary stream is coded using DNA coding using the DNA codes as shown in table 3

Table 3: Lookup table mapping 2 bit binary stream to base nucleotide

2 Bit binary stream	00	01	10	11
Base nucleotide	A	C	G	T

The DNA coded text becomes CACTCGTT and Integer coding is applied on this using table 4

Table 4: Lookup table mapping numeric value to base nucleotide

A -10	C - 20	G -30	T - 40

And the corresponding integer coded text is shown in table 5

Table 5. Integer coded text

20	10	20	40	20	30	40	40

Laplace transform is applied taking the above integer codes as following. We consider the standard expansion

$$e^{rt} = \sum_{n=0}^{\infty} \frac{(rt)^n}{n!} = 1 + \frac{rt}{1!} + \frac{r^2t^2}{2!} + \ldots.. + \frac{r^nt^n}{n!} + \ldots\ldots..,$$

where r is a constant (2)

And

$$te^{rt} = \sum_{n=0}^{\infty} \frac{r^nt^{n+1}}{n!} = t + \frac{rt^2}{1!} + \frac{r^2t^3}{2!} + \ldots.. + \frac{r^nt^{n+1}}{n!} + \ldots,$$

where r is a constant (3)

Let $G_0 = 20$, $G_1 = 10$, $G_2 = 20$, $G_3 = 40$, $G_4 = 20$, $G_5 = 30$, $G_6 = 40$, $G_7 = 40$ and $G_8 \geq 0 \; \forall \; n \geq 8$. Let us consider,

$$f(t) = Gte^{2t}$$

$$= t.\left[G_0.1 + G_1.\frac{2t}{1!} + G_2.\frac{2^2t^2}{2!} + G_3.\frac{2^3t^3}{3!} + G_4.\frac{2^4t^4}{4!} + G_5.\frac{2^5t^5}{5!} + G_6.\frac{2^6t^6}{6!} + G_7.\frac{2^7t^7}{7!}\right]$$

$$= 20.t + 10.\frac{2t^2}{1!} + 20.\frac{2^2t^3}{2!} + 40.\frac{2^3t^4}{3!} + 20.\frac{2^4t^5}{4!} + 30.\frac{2^5t^6}{5!} + 40.\frac{2^6t^7}{6!} + 40.\frac{2^7t^8}{7!}$$

$$= \sum_{n=0}^{\infty} G_n . \frac{2^nt^{n+1}}{n!} \text{ where } G_n \geq 0 \; \forall \; n \geq 8$$

Taking Laplace transform on both sides

$$L\{f(t)\} = L\{G.te^{2t}\}$$

$$= L\left\{t.\left[G_0.1 + G_1.\frac{2t}{1!} + G_2.\frac{2^2t^2}{2!} + G_3.\frac{2^3t^3}{3!} + G_4.\frac{2^4t^4}{4!} + G_5.\frac{2^5t^5}{5!} + G_6.\frac{2^6t^6}{6!} + G_7.\frac{2^7t^7}{7!}\right]\right\}$$

$$= \frac{20}{s^2} + \frac{10.(2)}{1!}.\frac{2!}{s^3} + \frac{20.(2^2)}{2!}.\frac{3!}{s^4} + \frac{40.(2^3)}{3!}.\frac{4!}{s^5} + \frac{20.(2^4)}{4!}.\frac{5!}{s^6} + \frac{30.(2^5)}{5!}.\frac{6!}{s^7} + \frac{40.(2^6)}{6!}.\frac{7!}{s^8} + \frac{40.(2^7)}{7!}.\frac{8!}{s^9}$$

$$= \frac{20}{s^2} + \frac{40}{s^3} + \frac{240}{s^3} + \frac{1280}{s^4} + \frac{1600}{s^6} + \frac{5760}{s^7} + \frac{17920}{s^8} + \frac{40960}{s^9}$$

Now we take modulo 128 on 20, 40, 240, 1280, 1600, 5760, 17920, 40960 which produces 20, 40, 112, 0, 64, 0, 0, 0.

We convert each of these integers to their corresponding ASCII values (7 bit binary) and then perform cumulative XOR operation iteratively until a single bit is found. The mechanism is explained taking the binary equivalent of 20 i.e. 0010100 on Fig 3.

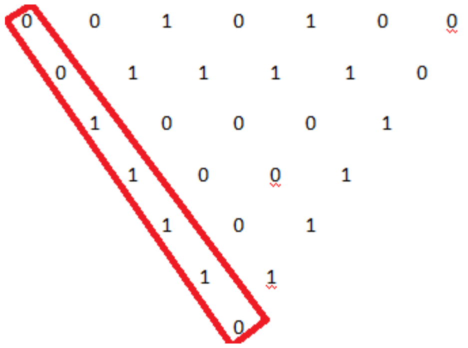

Fig 3: Cumulative XOR Operation on Binary Equivalent of 20 and MSB collection

Thus the encrypted cipher character corresponding to 20 in binary is 0011110. We will perform similar operation on the other values produced before and the collection of their corresponding cipher character would be the cipher text as shown below in table 6.

Table 6: Binary equivalent of Laplace coefficients and their corresponding cipher character

Laplace Coefficients	Binary Equivalents	MSBs of Cumulative XOR	Cipher Text (ASCII)	Cipher Text (Character)
20	0010100	0011110	30	RS
40	0101000	0100010	34	“
112	1110000	1001100	76	L
0	0000000	0000000	0	NULL
64	0101000	1111111	127	DEL
0	0000000	0000000	0	NULL
0	0000000	0000000	0	NULL
0	0000000	0000000	0	NULL

So the cipher text corresponding to “Go” is “"L ☐’

3.2 Method of Decryption

The ASCII values of the cipher text characters are 30, 34, 76, 0, 127, 0, 0, 0. They are converted to their 7 bit binary equivalents and on them cumulative XOR operation as shown in Fig are performed again which produces the binary equivalents of Laplace coefficients. They are converted to their decimal equivalent and inverse Laplace Transform is applied as follows.

Now we consider

$$G.\frac{1}{(s-a)^2} = \frac{20}{s^2} + \frac{40}{s^3} + \frac{240}{s^4} + \frac{1280}{s^5} + \frac{1600}{s^6} + \frac{5760}{s^7} + \frac{17920}{s^8} + \frac{40960}{s^9}$$

$$= \sum_{n=0}^{\infty} \frac{q_i}{s^{n+2}}$$

Taking inverse transform we get

$$= 20.t + 10.\frac{2t^2}{1!} + 20.\frac{2^2t^3}{2!} + 40.\frac{2^3t^4}{3!} + 20.\frac{2^4t^5}{4!} + 30.\frac{2^5t^6}{5!} + 40.\frac{2^6t^7}{6!} + 40.\frac{2^7t^8}{7!}$$

$$= \{t.\left[G_0.1 + G_1.\frac{2t}{1!} + G_2.\frac{2^2t^2}{2!} + G_3.\frac{2^3t^3}{3!} + G_4.\frac{2^4t^4}{4!} + G_5.\frac{2^5t^5}{5!} + G_6.\frac{2^6t^6}{6!} + G_7.\frac{2^7t^7}{7!}\right]\}$$

Here we have $G_0 = 20$, $G_1 = 10$, $G_2 = 20$, $G_3 = 40$, $G_4 = 20$, $G_5 = 30$, $G_6 = 40$, $G_7 = 40$ and $G_n \geq 0 \ \forall \ n \geq 8$.

So the integer codes for DNA nucleotides are 20, 10, 20, 40, 20, 30, 40, 40 which are converted to their corresponding bases using table 4.

So the DNA coded text becomes CACTCGTT and using table 3the binary stream of plain text becomes 0100011101101111.

The bit stream is decomposed into corresponding ASCII equivalent which produces the ASCII values of “G” and “o” are 71 (01000111) and 111 (01101111) respectively. Thus the plain text “Go” is recovered.

4. TESTING AND ANALYSIS

To ensure the security level of a cryptographic algorithm many efforts have been made. Among of them avalanche, bit ratio, non-homogeneity, frequency distribution, time complexity are frequently used in practice. The non-homogeneity test is a technique to test non-homogeneity of the source and encrypted file. In order to accomplish this Monobit test and Serial Test has been performed [14]. In the frequency distribution graph of source and encrypted file by proposed algorithm will be displayed. If the characters in the encrypted file are evenly distributed, it will make the cryptanalysis more difficult. The time complexity indicates how efficiently the proposed algorithm will encrypt the plain text and decrypt from encrypted text.

4.1 Monobit Test

The goal of this test is to determine whether the frequency of 0's and 1's in bit sequences in the cipher text generated by the DSWLT are approximately same. Let n_0 and n_1 denote the number of 0's and 1's in bit sequences respectively. We calculate χ_2 by using the formula.

$\chi^2 = \frac{(n_0 - n_1)^2}{n}$, which approximately follow a χ_2 distribution with one degree of freedom. The computed results are shown in Table 7.

Table 7. Results for Monobit Test and Serial Test

File name	Calculated value		Critical value at 0.05	
	Monobit Test	Serial Test	Monobit Test	Serial Test
Sample1.txt	1.322896	2.4636	3.8415	5.9915
Sample2.txt	0.591039	2.9658	3.8415	5.9915
Sample3.txt	1.141737	2.9826	3.8415	5.9915
Sample4.txt	1.551627	3.2310	3.8415	5.9915
Sample5.txt	1.109129	3.1226	3.8415	5.9915

The calculated values of χ_2 are less in compared to the critical value of χ_2 at $\alpha = 0.05$ (5% level of significance) and 1df (one degree of freedom). It means that these bit sequences pass the Monobit test and can be said to be satisfactorily random with respect to this test.

4.2 Serial Test

The goal of this test is to determine whether the number of occurrence of pairs 00, 01, 10 and 11 in the bit streams in the cipher text generated by DSWLT is approximately same.

Let n_{00} , n_{01} , n_{10} , n_{11} denote the number of occurrence of pairs 00, 01, 10 and 11 respectively in the bit sequences. We calculate χ_2 by using the formula

$$\chi^2 = \frac{4}{n-1}\left(n_{00}^2 + n_{10}^2 + n_{01}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 - n_1^2\right) + 1$$

and the computed values are found to follow approximately the χ_2 distribution with 2 degrees of freedom. The results are shown in Table 7. The calculated values of χ_2 are less than critical value of χ_2 at $\alpha = 0.05$ (5% level of significance) and 2df (two degrees of freedom). It means that bit sequences pass the serial test and are satisfactorily random with respect to this test.

4.3 Frequency Test

Frequency distribution graph of source and encrypted file by proposed algorithm will be displayed. If the characters in the encrypted file are evenly distributed, it will make the cryptanalysis more difficult. Fig. 4(a), 4(b), 4(c) shows the frequency distribution of characters in cipher text for TDES, AES and proposed DSWLT. From the following observations it may be concluded that the proposed DSWLT provides well enough security.

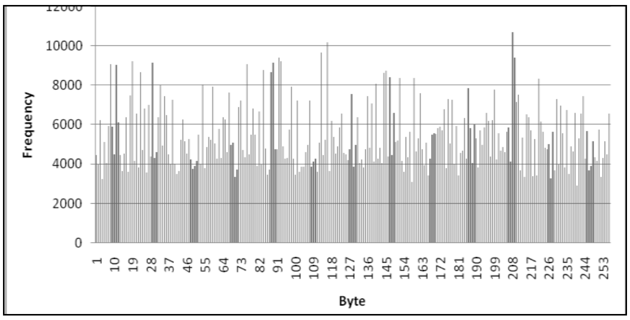

Fig 4(c): Frequency Distribution of characters in cipher files with different file size using TDES

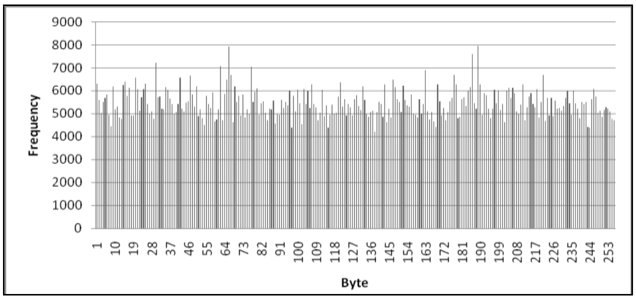

Fig 4(c): Frequency Distribution of characters in cipher files with different file size using AES

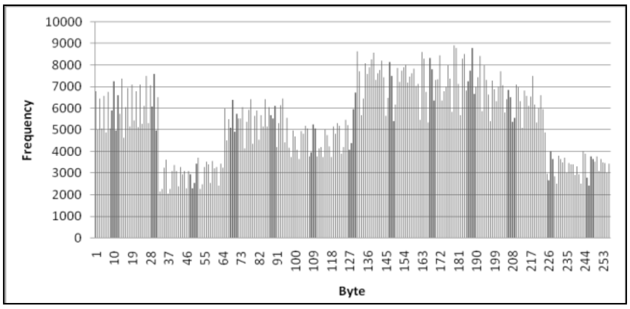

Fig 4(c): Frequency Distribution of characters in cipher files with different file size using DSWLT

4.4 Encryption and Decryption Time

Encryption and Decryption time with respect to different file sizes have been presented in table 8 and table 9 accompanied by corresponding graph as presented in Fig 5(a) and Fig 5(b). It is revealed that proposed DSWLT provides similar encryption and decryption time with TDES but slightly greater than AES. Encryption and Decryption time has been calculated in Hsec which is defined as 100 s of a second.

Table 8: File size v/s Encryption (in Hsecs)

Filename	File Size (in Bytes)	Encryption Time		
		DSWLT	AES	TDES
Sample1.txt	73	~0	~0	~0
Sample2.txt	1220	5	3.5	4.5
Sample3.txt	2190	11	8.7	9.89
Sample4.txt	3470	17	12.8	17
Sample5.txt	4210	22	19.3	21.1

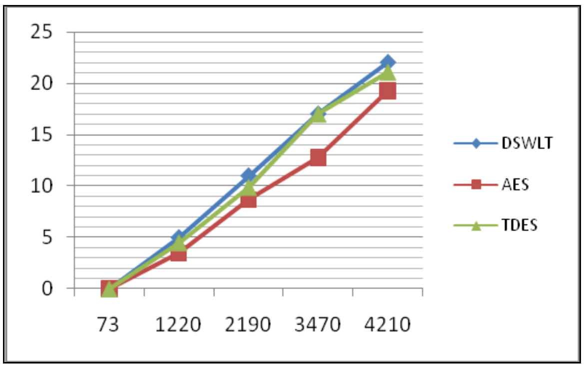

Fig 5(a): Encryption Time (sec) vs. File Size (bytes)

Table 9: File size v/s Decryption (in Hsecs)

Filename	File Size (in Bytes)	Decryption Time		
		DSWLT	AES	TDES
Cipher1.txt	293	0.15	0.08	0.12
Cipher2.txt	4890	7	4.2	5.6
Cipher3.txt	8750	12	8.3	11
Cipher4.txt	13800	17	14	15.4
Cipher5.txt	16800	23	19.6	20.8

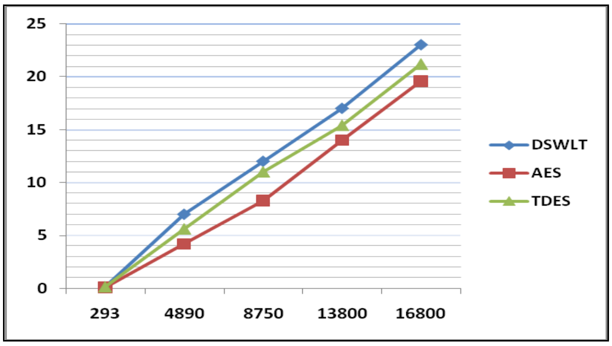

Fig 5(b): Decryption Time (sec) vs. File Size (bytes)

4.5 Encrypted and Decrypted File Size Comparison

We have generated cipher text for various plain texts with sizes ranging from 73 bytes to 4210 bytes and the corresponding cipher texts generated are also of different sizes from 293 bytes to 16800 bytes which has been presented in Fig 6.

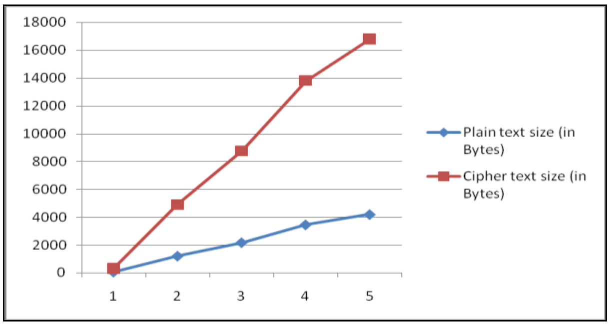

Fig 6: Original File size v/s Cipher File size

5. CONCLUSION AND FUTURE SCOPE

DNA cryptography is the future of the information security. Its complexity and randomness provides a great uncertainty which makes encoding of data in DNA format better than other mechanism of cryptography.On the basis of the observed experimental results, it can be said that 'DSWLT' is extremely efficient and a sufficiently strong cryptographic algorithm that provides a superior level of security. The proposed algorithm is a simple, straight forward but intrinsically strong and compact approach to cryptography using the essence of genetic operations. It provides the same or sometimes even better level of security using minimal time complexity.

6. ACKNOWLEDGMENTS

The authors express a deep sense of gratitude to the Department of Computer Science, Barrackpore Rastraguru Surendranath College, Kolkata-700 120, West Bengal, India for providing necessary support for the work and their family members for being constant inspiration and motivation for pursuing such works.

7. REFERENCES

[1] J. D. Watson, F. H. C. Crick, "A structure for deoxy ribose nucleic acid", Nature, vol. 25, pp. 737-738, 1953.

[2] G. Z. Cui, L. M. Qin, Y. F Wang and X. C. Zhang, "Information Security Technology Based on DNA Computing", IEEE International Workshop on Anti counterfeiting Security, pp. 288–291, 2007

[3] G. Z. Cui, "New Direction of Data Storage: DNA Molecular Storage Technology," Computer Engineering and Applications, vol. 42, pp. 29–32, 2006.

[4] Xing Wang, Qiang Zhang ,"DNA computing-based cryptography", Fourth International Conference on Bio-Inspired Computing, 2009.

[5] L. M. Adleman, "Molecular computation of solution to combinatorial problems", Science, vol. 266, pp. 1021-1024, November 1994.

[6] C. Taylor, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots", Nature, vol. 399, pp. 533-534, 1999.

[7] R. J. Lipton," Using DNA to Solve NP-Complete problems," Science, vol. 268, pp. 542 545, 1995.

[8] G. Z. Cui, "New Direction of Data Storage: DNA Molecular Storage Technology," Computer Engineering and Applications, vol. 42, pp. 29–32, 2006.

[9] Sherif T. Amin, Magdy Saeb, Salah El-Gindi, "A DNA based Implementation of YAEA Encryption Algorithm," IASTED International Conference on Computational Intelligence,2006

[10] Pankaj Rakheja, "Integrating DNA Computing in International Data Encryption Algorithm (IDEA)", International Journal of Computer Applications, pp 1 – 6, Volume 26, No.3, July 2011

[11] Murray R Spiegel, "Schaum's Outline of Laplace Transforms", Mcgraw-hill, 1965

[12] "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.

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

[14] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, San Vo, "A statistical Test Suite for Random and pseudorandom Number Generators for Cryptographic Applications", April 2010, National Institute of Standards and Technology.

8. AUTHORS PROFILE

Sukalyan Som is presently working as an Assistant Professor in Computer Science in Barrackpore Rastraguru Surendranath College, West Bengal, India. He is having over 5 years of teaching experience. He has received his B.Sc. in Statistics from University of Calcutta, West Bengal India and Masters in Computer Application from West Bengal University of Technology, West Bengal, India. His research interest includes Cryptography, Steganography, Image Processing, Computational Geometry etc.

Moumita Som was a student of the department of Computer Science in Barrackpore Rastraguru Surendranath College, West Bengal, India. She has completed her B. Sc. Computer Science Honours under West Bengal State University in 2012. Her Research interest includes Computer Graphics, Analysis of Algorithms, Data Structures, Cryptography and Network Security.