Journal of Engineering and Applied Sciences 12 (Special Issue 8): 8309-8316, 2017
ISSN: 1816-949X


# A Novel Neuro-Symmetric Block Cipher

[1]Abul Hasnat, [1]Dibyendu Barman and [2]Satyendra Nath Mandal
[1]Department of Computer Science and Engineering,
Government College of Engineering and Textile Technology Berhampore, West Bengal, India
[2]Department of Information Technology, Kalyani Govternment Engineering College,
West Bengal, India

**Abstract:** In symmetric key encryption, no matter how strong the encryption technique is there is always a great concern about the trust worthiness of the 'key-exchange' process and the information may be compromised once the key is exposed to intruder. This study proposes a neuro-symmetric block cipher where the key is not shared but a "weight vector" and key input file are exchanged. "Weight vector" is the updated weights of a trained artificial neural network using key input file as input and key target file as output. Turn and mix functions are used to generate 12 different keys at twelve different rounds using the key target file. During encryption, turn, shift and mix functions and XOR operations are applied between the key of first round with plain text. This process is repeated eleven rounds more on intermediate cipher text and key of the respective round to produce the cipher text. An intruder cannot decrypt the cipher text having only key-input file or the weight matrix or both until and unless he knows all of the key-input file, weight vector and structure of ANN (it is not exchanged over network) at the same time. Experimental result shows that the proposed approach is more secure in terms of cryptanalysis. For performance analysis, all statistical tests suggested by NIST and FIPS PUB-140-1 test battery have been applied on the cipher text produced by proposed algorithm and the algorithm shows reasonable good response. Finally, a comparison study is given between the popular symmetric key algorithms and the proposed algorithm. Experimental results shows that the proposed approach is more robust compared to the conventional symmetric key encryption algorithms.

**Key words:** Artificial neural network, Chi-square distribution, NIST test, shift, mix, turn, statistical test, symmetric key

## INTRODUCTION

Cryptography is an art of hiding information (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011; Hasnat *et al*., 2016a, 2015a; Sidi *et al*., 2017; Ali and Sagheer, 2017). A lot of research has been carried out in the field of cryptography and there are many encryption algorithms reported in literature used for secure data transmission. The Advanced Encryption Standard (AES) (Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011; Hasnat *et al*., 2016a, 2015a; Sidi *et al*., 2017; Ali and Sagheer, 2017) is one such algorithm and it is adopted as a standard for encryption by National Institute of Standards and Technology's (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hasnat *et al*., 2016b; Sidi *et al*., 2017). The AES use 128 bits block and it may use 192 and 256 bits. AES is considered as one of the best symmetric key encryption algorithm in literature (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011; Hasnat *et al*., 2016a; Sidi *et al*., 2017; Ali and Sagheer, 2017). But the security is breached once the key is disclosed. There is always a high risk involved in the key-exchange or key-distribution process of any symmetric key encryption (Hasnat *et al*., 2015a; Ali and Sagheer, 2017) model.

In this study, a novel symmetric encryption algorithm is proposed where key-input file and weight vector is shared but the actual key/key-target file is not exchanged. The "weight vector" is the weights of a trained Artificial Neural Network (ANN) (Haykin, 2001; Ghosh *et al*., 2014; Hasnat *et al*., 2016b; Alpaydin, 2010) using key-input file

**Corresponding Author:** Abul Hasnat, Department of Computer Science and Engineering,
Government College of Engineering and Textile Technology Berhampore, West Bengal, India

as input and actual key/key-target file as output. The receiver has to reconstruct the actual key using ANN with the help of the key-input file and weight matrix. One cannot decrypt the cipher text having only the key-input file or the weight matrix or both until and unless intruder knows the key-input file, weight vector and structure of ANN (it is not exchanged over network) at the same time. Therefore, this approach is more robust compared to conventional symmetric key encryption.

**Artificial neural network and terminologies**

**Artificial Neural Network (ANN):** ANN (Haykin, 2001; Ghosh *et al*., 2014; Hasnat *et al*., 2016a; Alpaydin, 2010), a machine learning approach (Hasnat *et al*., 2015a, 2017a, b; Cherif *et al*., 2008) is a parallel distributed processor made up of simple Processing Elements (PE) (Haykin, 2001; Alpaydin, 2010) called neuron which has a natural propensity for storing experimental knowledge (Haykin, 2001; Alpaydin, 2010) and making it available for future use. Each neuron in ANN receives a number of inputs. An activation function is applied to these inputs which results in activation level of neuron. Knowledge about the learning task is given in the form of examples called training examples (vectors). An artificial neural network is specified by neuron model: the information processing unit of the NN an architecture: a set of neurons and links (each link has a weight) connecting neurons and A learning algorithm: used during training of neural network for modifying the weights in order to model a particular learning task correctly on the training examples.

**Key generation terminologies**

**Key-target file:** This file contains the information which is used to encrypt the plain-text. Its elements are used as the target nodes in the neural network. Any type of file can be used as the key-target file. The first 16 bytes (128) bits are used as encrypted key.

**Key-input file:** This file consists of the key which does not take part in the actual encryption procedure but its elements are used as the input to train the neural network. It is used during test of the neural network (during decryption) also and thus needs to be transmitted to the receiver. Any type of file can be used as the key-input file. The first 16 bytes (128) bits are used.

**Initial weight:** This is the matrix of random weights which are used during training procedure (back-propagation) of ANN.

**Weight vector:** This matrix is generated after completion of the training procedure (back-propagation) of ANN. It is the final weights which are to be used at the receiver side where ANN generates the key-target file with help key-input file (during decryption) and thus it is transmitted to the receiver.

**NIST and FIPS PUB-140-1 test battery**

**Frequency distribution diagram:** The frequency distribution diagram (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011) represents the occurrence of individual characters in a test. If the diagram contains equal width indicates that the occurrences of characters are same and gives no information about the individual character.

**Floating point frequency:** The floating point frequency (Stallings, 2006; Mondal *et al*., 2011) indicates the different characters present in any given 64-character long segment in a document. The standard values on different types of documents are for bmp image usually lies between 5 and 20 for text varies in the range between 18 and 30. For exe files, it is between 30 and 40 code sections and in the data section around 20.

**Entropy analysis:** The entropy (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011) of a document gives an index of its information content. The information content of a message M[i] is defined by Eq. 1:

$$M[i] = \log_2\left(\frac{1}{p[i]}\right) = -\log_2 (p[i]) \quad (1)$$

Here, p[i] is the probability of message, M[i] is transmitted by the message source. The document with every character of the character set (0-255), the entropy lies between 0 bit/char and log (256) bit/char = 8 bit/char.

**Chi-square test and degree of freedom:** Chi-square (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011) is a statistical test commonly used to compare observed data with expected data obtained according to a specific hypothesis. The Chi-square is the sum of the squared difference between observed (o) and the expected (e) data (or the deviation, d), divided by the expected data in all possible categories. The degree of freedom is defined the number of characters present in file. The maximum value is 256 as the content of the file is represented by extended ASCII.

**Frequency test:** Let $n_0$ and $n_1$ be the number of zeros or ones in a strings. Test is measured by Eq. 2:

$$X = \frac{(n_0 - n_1)^2}{2} \quad (2)$$

where, X follows a Chi-square distribution with one degree of freedom provided that $n \geq 10$. Let there is a string with length of $n > 10{,}000$. If $n_0 = n_1$ then $x_1 = 0$. It indicates that the greater discrepancy between the observed and expected frequencies of zeros and ones. The value is computed as 3.8415 with 0.05 significance level and one degree of freedom in Chi-square distribution. If $X_1 = 0$ then the sequence passes the test, otherwise fails.

**Poker test:** In poker test (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011), the sub-sequence of string s of length n with element from [0; 1], $0 \leq i \leq n-1$. First it must be transformed through $u_i = d \times s_i$, $0 \leq i \leq n-1$ into whole numbers from the range [0, d-1]. The test statistics used are given by Eq. 3:

$$X = \left(\frac{2^m}{k}\right) \times \left(\sum_{i=0}^{2^m-1} n_i^2\right) - k \quad (3)$$

where by X approaches a Chi-square distribution with $2^m$-1 degrees of freedom.

**Runs test:** This test (Stallings, 2006; Forouzan, 2015; Naor and Shamir, 1994; Shamir, 1979; Shirey, 2007; Earle, 2005; Hadron and Dondeti, 2005; Jon and Arbaugh, 2003; Mondal *et al*., 2011) is used to investigate monotonically rising and falling sub-sequences in a number sequence. A "run-up" of length k means a sub-sequence with $X_{(i+r)} \leq X_{(i+r+1)}$ for all r where $0 \leq r \leq k-1$. For the expected frequency of a run-up of length k in a sequence of length n, the following holds true:

$$E_k = \frac{\left[(k^2 + k + 1) \times (n - k - 1)\right]}{(k+2)!} \quad (4)$$

For a significance level of $\alpha = 0.05$ at $2 \times 3 - 2 = 4$ degrees of freedom there is the critical area above 9.488. With $X_4 = 2.0596 \leq 9.488$, sequence s passes the runs test.

**Serial test:** Let $n_0$ and $n_1$ be the numbers of zeros and ones, respectively in s and let $n_{01}$,$n_{01}$ $n_{01}$ and $n_{01}$ be the relevant numbers of sub-sequences 00, 01, 10, 11 in s. As the sub-sequences overlap, the following applies (with n = bit length of s):

$$n_{00} + n_{01} + n_{10} + n_{11} = n - 1 \quad (5)$$

The test statistics used are given by:

$$X = \left(\frac{4}{n-1}\right) \times \left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \left(\frac{2}{n}\right) \times \left(n_0^2 + n_1^2\right) + 1 \quad (6)$$

where by X approaches a Chi-square distribution with two degrees of freedom, provided that $n \geq 21$.

## MATERIALS AND METHODS

**Proposed algorithm:** The Neuro-Symmetric Block Cipher "NSBC" research in two steps key generation and encryption. Initially two files, the key-input file and key target file are selected randomly. The length of key in input and target file varies from 128-256 bits. Training of ANN starts with random initial weights where input is the key-input file and output is key-target file. The weight matrix of finally trained ANN is sent to the receiver. During encryption process, the plain text is divided into 64 bits blocks. The 128 bits key is passed into key schedule algorithm to produce twelve 64 bits keys. The different keys are produced by turn, shift and mix operation. Finally, XOR operation is applied on plain text block and key is generated in each round to get the cipher text. A single layer feed forward structure with back-propagation (Haykin, 2001; Ghosh *et al*., 2014; Hasnat *et al*., 2016a; Alpaydin, 2010), ANN is used in this case. But it is strongly recommended to use a complex ANN in real life applications.

### Key generation algorithm using ANN at sender side:

**Input:** key-target file say K (16 bytes), key-input file (16 bytes)
**Output:** Weight vector
Step 1: generate two file with random distribution of 128 or 256 bits
Step 2: build a feed forward back propagation multi-layer neural network
Step 3: generate initial weight for network
Step 4: train the network with input, weight and target files
Step 5: when artificial neural network training is complete final weight file will be generated

### Key scheduling algorithm:

**Input:** Key-target file say K (16 bytes)
**Output:** 12 different keys (length 64 bits)
Step 1: for round 1-6, divide K into 2 halves say $K_{First}$ and $K_{Second}$ for first 8 bytes and last 8 bytes, respectively
Step 2: for round 1-3, turn and mix the $K_{First}$ once, twice and thrice, respectively
Step 3: for round 4-6, turn and mix the $K_{Second}$ once, twice and thrice, respectively
Step 4: for round 7-12, divide the key into 2 halves say $K_{odd}$ and $K_{Even}$ (8 odd position bytes and 8 even position bytes)
Step 5: for round 7-9, turn and mix the $K_{Odd}$ once, twice and thrice, respectively
Step 6: for round 10-12, turn and mix the $K_{Even}$ once, twice and thrice, respectively

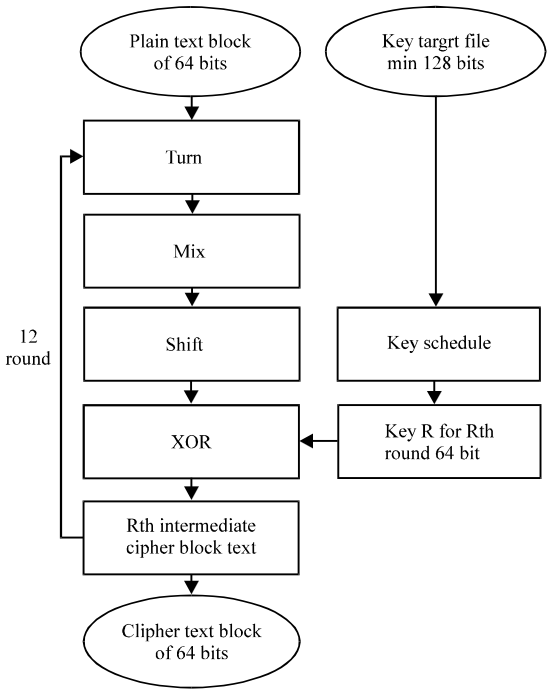


Fig. 1: Plain text (64 bits) encryption flowchart, process is repeated for each 8 byte of plain text

### Encryption algorithm

**Input:** plain text (64 bits), key target file (16 bytes)
**Output:** cipher text (64 bits)

Step 1: divide the plain-text block into 64 bits each (8 bytes)
Step 2: send the contents of the key-target file say K (16 bytes) to the key schedule for obtaining 12 different keys (length 64 bits) for round 1-12
Step 3: encryption procedure (Jon and Arbaugh, 2003) is the plain-text block
  Step 3.1: for round R = 1-12 we have $K_1$-$K_{12}$
  Step 3.1.1: convert P[] bytes to bits and store it in a 8×8 matrix say C[][].
  Step 3.1.2: turn (C[] [], 2). (turning twice to $C_2$[] [])
  Step 3.1.3: Mix ($C_2$[] [])
  Step 3.1.4: RC shift ($C_2$[] [], 4) (Applying Right Circular Shift 4 times).
  Step 3.1.5: convert $C_2$[] [] bits to bytes say C[]
  Step 3.1.6: XOR (C[], reverse ($K_R$)) to get cipher-text block after round R say $C^R$
  Step 3.1.7: If R = 4 or R = 8 then store the respective $C^4$[] and $C^8$[] as intermediate cipher-text blocks
Step 4: key-mixing procedure. We have $C^4$[] and $C^8$[] as intermediate cipher-text blocks (8 bytes each) and the key K (16 bytes). Then, we are mixing to get the key K (16 bytes) for encrypting the next plain-text block.
  Step 4.1: for j = 1-16
  Step 4.1.1: If j< = 8, then XOR (K[j], $C^8$[j])
  Step 4.1.2: If j>8, then XOR (K[j], $C^4$[j])

The encryption process is shown in Fig. 1.

## RESULTS AND DISCUSSION

The proposed neuro-symmetric block cipher is successfully applied on different files of varied types, i.e., .txt, .jpeg, .docx, .pptx, .pdf, .zip, .zar, .exe. For performance analysis, all statistical tests suggested by NIST and FIPS PUB-140-1 test battery have been applied on the cipher text where the proposed algorithm shows reasonable good response in these test. For all tests performed in this research, the key input file and key target file used are debayan.157@gmail.com 7 and1 q2w#E4r5t!9&^5@cjgjk, respectively. Initial weight of ANN is a two dimensional weight matrix of size 16×16 with random values (real number). In this study, a single layer feed forward back-propagation ANN is used where number of neurons (both input and output layer) is sixteen. Learning rate is 1.5, error tolerance is 0.0001. But it is strongly recommended to use a more complex network structure in real applications. The entire plain-text file is divided into blocks of 64 bits each. If the last block is not exactly of 64 bits (8 bytes) then it is made to 64 bits by adding required number of spaces (byte value 32). Sample output of one text file named 'file10.txt' of size 27 KB is shown below.

### Content of the input file or plain-text:

College of Engineering and Management, KOLAGHAT CNI and SSA Laboratory (6th Semester 2006, CSE and EIE)
Experiment: packet filtering firewall (Filename-iptables.txt)
---
---
Feedback:
Please point out mistakes/corrections and suggest which portions should be removed, added, expanded, etc... bkc
Send mail to vu2nil@cse.cemk.ac.in
---------- end of file iptables.txt ----------

**Frequency distribution:** One frequency distribution diagram (all 256 characters) of plain text and encrypted text of file 10.txt are shown in Fig. 2 and Fig. 3, respectively where X coordinate shows the character (range of 0-255) and Y coordinate shows the frequency of the character. The frequency of characters in the plain text of the input file 'file 10. txt' is not evenly distributed (Fig. 2) but the frequency of characters of the encrypted file is evenly distributed (Fig. 3) in the range of 0-255. The occurances is almost same to all characters in Fig. 3 and it ensures that the recovery of plain text from cipher text is more challenging.

**Floating frequency diagram:** The floating frequency specifies number of different characters found in any given 64-character long segment of the document. The floating frequency of a document lies between 18 and 30. In cryptography, the mechanism is mainly used to locate keys amongst large quantities of data (Fig. 4).

**Auto-correlation diagram:** The auto-correlation of a document is an index of the similarity of different sections of the document. It is sometimes possible to research out the key length of an encrypted document from its autocorrelation. The autocorrelation function C[t]

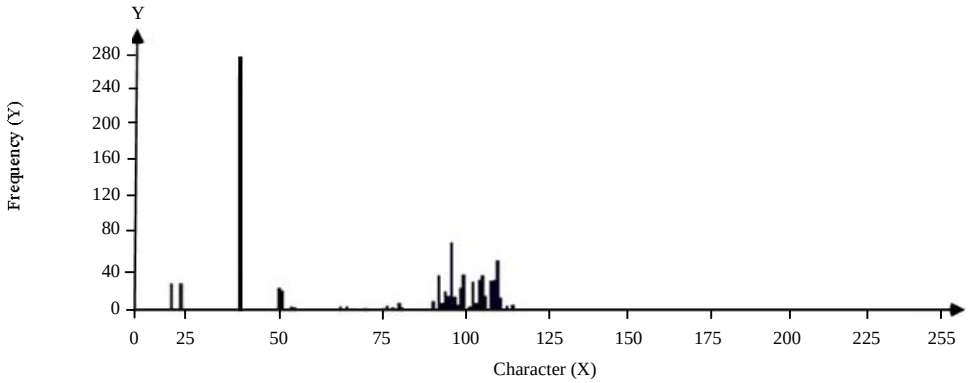


Fig. 2: Frequency distribution of plain-text of file10.txt

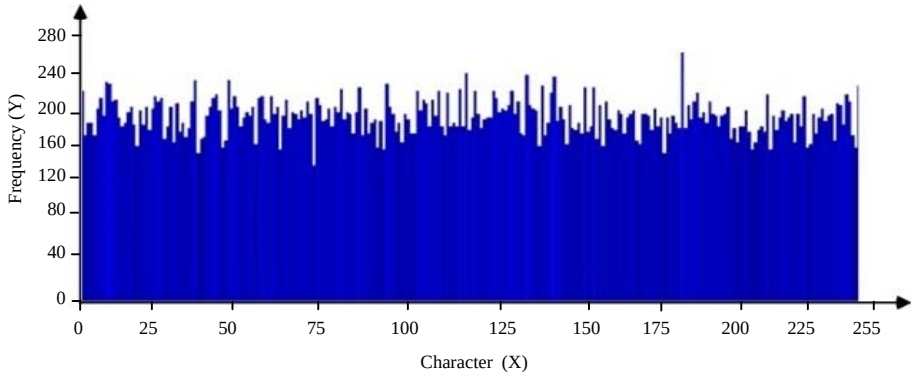


Fig. 3: Frequency distribution of the encrypted file of enfile10.txt

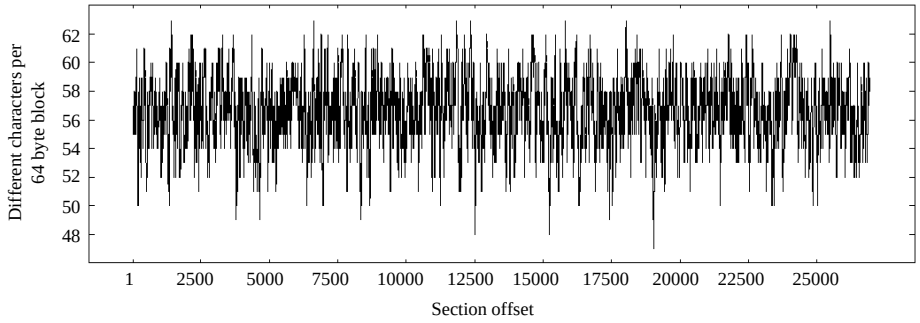


Fig. 4: Floating frequency diagram of enfile10.txt

measures the similarity of sequence (s[i] = s[1] s[2]...) to sequence (s[i+t] = s[1+t] s[2+t]...) which is shifted by t places.

A sequence of length n is examined and the following parameters are defined as A(t) = number of members of sequences (s[i] and (s[i+t]) in the segment under consideration which agree. D (t) = number of members of sequences (s[i] and (s[i+t]) in the segment under consideration which do not agree. The autocorrelation function is given in Eq. 7.

$$C(t) = \frac{A(t)-D(t)}{n} \quad (7)$$

where n is sequences of length, calculated from the number A of agreeing and the number D of non-agreeing. Figure 5 shows auto correlation diagram for enfile10.txt.

**Entropy analysis:** The entropy of a document is an index of its information content. The entropy is measured in bits

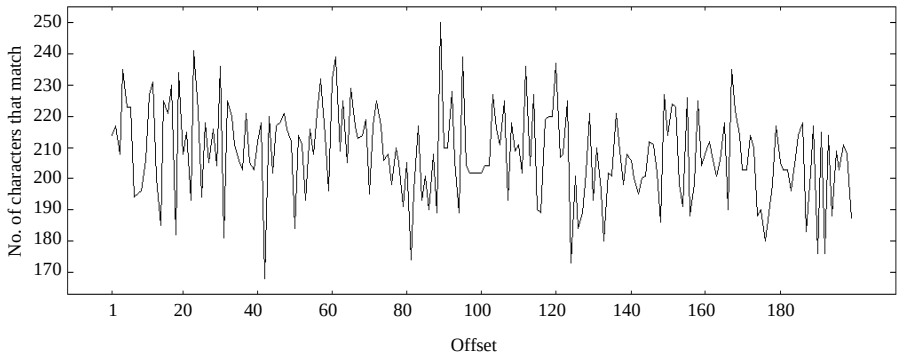


Fig. 5: Auto-correlation diagram of enfile10.txt

Table 1: Fips pub-140-1 test battery

| Test type | Values with significance level 0.05 | Computed test values |
|---|---|---|
| Frequency test | 3.384 | 0.073500 |
| Poker test | 14.070000 | 4.690667 |
| Runs test | 9.488000 | 3.957362 |
| Longest runs test | 34 | 20 |
| Serials test | 5.991000 | 0.094515 |

per character. The maximum value of entropy is indicated that the plain text can not be recovered from encrypted text in statistical attack. The documents which contains every character of the character set (0-255), the entropy lies between 0 bit/char and log (256) bit/char = 8 bit/char. The entropy of encrypted text is computed as 7.992 bit/character. It indicates that the plain text cannot be recovered by statistical test.

**Chi-square test and degree of freedom:** Chi-square is used to check the homogeneity of a document. The low Chi-square value and degree of freedom indicates that the characters in cipher text are not well distributed and few numbers of characters are present in cipher text. In this study, the Chi-square value and degree of freedom have been computed as 49130 and 255, i.e., the encrypted text is totally heterogeneous and maximum number of different types of characters are present in it.

**Fibs PUB-140-1 test battery:** In the FIPS PUB-140-1 battery test, the 'significance level (α)' is set to 0.05. It is shown in Table 1. In the Table 2, a comparison study is given among IDEA, triple-DES, AES and NSBC on in terms of Entropy (E) and Chi-square value (C). All the files were encrypted using the same key. It is clear from the Table 2 that the proposed algorithm produces better result in some cases than AES and in all cases compared to IDEA, triple-DES and NSBC algorithms.

Table 2: Comparison among symmetric key algorithms and proposed algorithm

| File name and size | IDEA | TRIPLE-DES | AES | NSBC |
|---|---|---|---|---|
| File2.txt | C = 6929 | C = 6942 | C = 6949 | C = 6929 |
| 3.68 kb | E = 7.936 | E = 7.921 | E = 7.947 | E = 7.958 |
| File3.txt | C = 15680 | C = 15741 | C = 15742 | C = 15744 |
| 8.356 kb | E = 7.970 | E = 7.974 | E = 7.976 | E = 7.977 |
| File4.txt | C = 51209 | C = 50971 | C = 51084 | C = 51136 |
| 27.452 kb | E = 7.929 | E = 7.935 | E = 7.992 | E = 7.993 |
| File5.txt | C = 24605 | C = 24473 | C = 24602 | C = 24630 |
| 13 kb | E = 7.946 | E = 7.946 | E = 7.987 | E = 7.986 |
| File6.txt | C = 29143 | C = 29087 | C = 29131 | C = 29130 |
| 15.448 kb | E = 7.971 | E = 7.970 | E = 7.988 | E = 7.990 |
| File7.txt | C = 19077 | C = 19003 | C = 19043 | C = 19051 |
| 10.152 kb | E = 7.964 | E = 7.962 | E = 7.981 | E = 7.982 |
| File8.txt | C = 28718 | C = 28600 | C = 28722 | C = 28726 |
| 15.4 kb | E = 7.946 | E = 7.951 | E = 7.988 | E = 7.989 |
| File9.txt | C = 41621 | C = 41581 | C = 41657 | C = 41640 |
| 22.380 kb | E = 7.929 | E = 7.930 | E = 7.991 | E = 7.991 |
| File10.txt | C = 48927 | C = 48893 | C = 49041 | C = 49130 |
| 26.208 kb | E = 7.934 | E = 7.934 | E = 7.993 | E = 7.992 |
| File 11.zip | C = 227023 | C = 226976 | C = 227016 | C = 227048 |
| 165 kb | E = 7.998 | E = 7.998 | E = 7.998 | E = 7.998 |
| File 12.jpeg | C = 155901 | C = 155935 | C = 155898 | C = 155896 |
| 113 kb | E = 7.998 | E = 7.998 | E = 7.998 | E = 7.998 |
| File 13.pptx | C = 155166 | C = 155004 | C = 153838 | C = 154615 |
| 150 kb | E = 7.988(F) | E = 7.987(F) | E = 7.998 | E = 7.998 |
| File 14.pdf | C = 307476 | C = 322012 | C = 305294 | C = 306811 |
| 406 kb | E = 7.999 | E = 7.998 | E = 7.999 | E = 7.999 |
| File15.docx | C = 210667 | C = 208978 | C = 210566 | C = 210519 |
| 152 kb | E = 7.996(F) | E = 7.995(F) | E = 7.998 | E = 7.999 |
| File 16.jar | C = 79399 | C = 79523 | C = 79439 | C = 79573 |
| 48.7 kb | E = 7.992 | E = 7.992 | E = 7.995 | E = 7.996 |
| File 17.exe | C = 64039 | C = 66386 | C = 62811 | C = 61274 |
| 205 kb | E = 7.996(F) | E = 7.997(F) | E = 7.999 | E = 7.999 |

## CONCLUSION

In the proposed neuro-symmetric block cipher, key is generated using back propagation neural network. The algorithm applied on different types of file types, i.e., .txt, .jpeg, .docx, .pptx, .pdf, .zip, .zar, .exe. with varying size. Assessment of the performance of "NSBC" is done using different test suggested by NIST and FIPS PUB-140-1 test battery. Frequency test result shows even distribution of characters in cipher text. The floating point frequency and

auto-correlation diagram shows that there is no repeated pattern in the cipher text, so, the intruder cannot extract original text from cipher text without keys. The value of entropy test is 7.992 (almost 8) which is the symbol of good encryption algorithm. The computed value of degree of freedom and Chi-square test are 255 and 49130, respectively which indicates that all characters are present in cipher text and the high Chi-square value proves the cipher text as heterogeneous in nature. So, the algorithm has passed all FIPS PUB-140-1 test battery. Finally, the comparison study is given with some known symmetric key algorithms based on Chi-square value and entropy. Experimental result shows that the proposed algorithm gives better result in all cases compared to other standard algorithms but in some cases AES performs better. The key distribution is fully secured as the original key is not transmitted but the key is reconstructed using final weight vector, key-input file and the ANN. An intruder is unable to decrypt the cipher text having only key-input file or the weight matrix or both until and unless he knows the key-input file, weight vector and structure of ANN at the same time. Therefore, proposed algorithm is more robust compared to conventional symmetric key encryption algorithms.

## ACKNOWLEDGEMENT

Researcher are thankful to Debayan Das of Kalyani Govt. Engineering College, West Bengal for his contribution in this research.

## REFERENCES

Ali, A.H. and A.M. Sagheer, 2017. Design and implementation of secure chatting application with end to end encryption. J. Eng. Appl. Sci., 12: 156-160.

Alpaydin, E., 2010. Introduction to Machine Learning. 2nd Edn., MIT Press, Cambridge, Massachusetts, ISBN:9780262012430, Pages: 537.

Cherif, A., L. Salhi and M. Talbi, 2008. Gammachirp wavelet and neural network for identification of pathological voices. J. Eng. Applied Sciences, 3: 822-828.

Earle, A.E., 2005. Wireless Security Handbook. Auerbach Publications, Boston, Massachusetts, ISBN:0-8493-3378-4, Pages: 365.

Forouzan, B.A., 2015. Cryptography and Network Security. 3rd Edn., McGraw Hill, New Delhi, India, ISBN:9789339220945,.

Ghosh A., A. Hasnat, S. Halder and S. Das, 2014. A proposed system for cotton yarn defects classification using probabilistic neural network. Proceedings of the International Conference on Recent Advances and Innovations in Engineering (ICRAIE14), May 9-11, 2014, IEEE, Jaipur, India, ISBN:978-1-4799-4039-4, pp: 1-6.

Hadron, T. and L.R. Dondeti, 2005. Security in Wireless LANs and MANs. Artech House Publishers, Norwood, Massachusetts, USA., Pages: 243.

Hasnat, A., A. Ghosh, A. Hoque and S. Halder, 2016b. Pattern classification of cotton yarn neps. Indian J. Fibre Text. Res., 41: 270-277.

Hasnat, A., A. Ghosh, A. Khatun and S. Halder, 2017a. Pattern classification of fabric defects using a probabilistic neural network and its hardware implementation using the field programmable gate array system. Fibres Text. East. Eur., 25: 38-44.

Hasnat, A., A. Ghosh, S. Das and S. Halder, 2015a. Identification of single and double jersey fabrics using proximal support vector machine. Proceedings of the 4th International Conference on Soft Computing for Problem Solving Vol. 335, December 27-29, 2015, Springer, New Delhi, India, pp: 389-401.

Hasnat, A., D. Barman and S.N. Mandal, 2016a. A novel image encryption algorithm using pixel shuffling and pixel intensity reversal. Proceedings of the 2016 International Conference on Emerging Technological Trends (ICETT), October 21-22, 2016, IEEE, Kollam, India, ISBN:978-1-5090-3752-0, pp: 1-6.

Hasnat, A., S. Haider, D. Bhattacharjee and M. Nasipuri, 2015b. A proposed system for gender classification using lower part of face image. Proceedings of the 2015 International Conference on Information Processing (ICIP), December 16-19, 2015, IEEE, Pune, India, ISBN:978-1-4673-7758-4, pp: 581-585.

Hasnat, A., S. Halder, D. Bhattacharjee and M. Nasipuri, 2017b. A proposed grayscale face image colorization system using particle swarm optimization. Intl. J. Virtual Augmented Reality, 1: 72-89.

Haykin, S., 2001. Neural Networks: A Comprehensive Foundation. 2nd Edn., Pearson Education, New Delhi, India, ISBN:9787302049364, Pages: 871.

Jon, E. and W.A. Arbaugh, 2003. Real 802.11 Security: Wi-Fi Protected Access and 802.11i. Addison-Wesley Professional, New York, ISBN-10: 0321136209.

Mondal, U.K., S.N. Mandal, J. PalChoudhury and J.K. Mandal, 2011. Frame based symmetric key cryptography. Intl. J. Adv. Networking Appl., 2: 762-769.

Naor, M. and A. Shamir, 1994. Visual cryptography. Proceedings of the International Workshop on the Theory and Application of Cryptographic Techniques, May 9-12, 1994, Springer, Berlin, Germany, pp: 1-12.

Shamir, A., 1979. How to share a secret. Commun. ACM, 22: 612-613.

Shirey, R., 2007. Internet security glossary. Inf., 2: 1-365.

Sidi, F., A.J. Marzanah, L.S. Affendey, I. Ishak and N.M. Sharef *et al*., 2017. A comparative analysis study on information security threat models: A propose for threat factor profiling. J. Eng. Appl. Sci., 12: 548-554.

Stallings, W., 2006. Cryptography and Network Security. 4th Edn., Prentice Hall, New Jersey, USA., ISBN-13: 9780131873162, Pages: 680.