ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

An Approach towards Designing a CryptoFigureic Confidentiality Algorithm with high PSNR & Time Efficient

Surendra Kumar Sharma[a]*, Ramratan Ahirwal[b]*

[a,b]*Research Scholar, Samrat Ashok Technological Institute, Vidisha (M.P.)*

[a]*Email: surendra107.s@gmail.com*

[b]*Email: ram2004_ahirwal2004@rediffmail.com*

Abstract

SteganoFigurey is the technique of hiding valuable information within other data. While cryptoFigurey makes sure that the data is unreadable to the third party. To fulfill the concern of security, many approaches of steganoFigurey and cryptoFigurey are discussed and developed. In the paper we proposed a work with combined approach of steganoFigurey and cryptoFigurey and is applied to achieve the goal of security with better efficiency as compared to previous methodologies. The information hiding system is mainly designed with the attributes of characters i.e. RGB values. The proposed work presented in this paper implements the five processing steps of our methodology. These steps consists the generation of a pseudo random number with the help of a 128 bit key, generation of four keys to encrypt the message, encryption by symmetric key encryption technique, cipher text hiding by replacing the least significant with the application of modified LSB algorithm, and generation of secured message at the end as output. The proposed algorithm is designed in such a way that before hiding a bit, the bit is optimized first that it increases the PSNR value as compared with previous algorithms without affecting the other parameters.

Keywords: SteganoFigurey; CryptoFigurey; Data Hiding; LSB Method.

1. Introduction

In today's scenario, we all are very much dependent on digital world. We all use internet and many other public networks for communication, information storage, entertainment, business purpose etc.

---

* Corresponding author.

Need of all these are must but with security. The problem with public network or internet is that they are the most unsecure channel to do all the above discussed tasks. Whenever a discussion of security comes, three terms comes in mind: confidentiality, integrity and authentication. Confidentiality ensures that transmitted data or stored data cannot be readable by any other un-authorized person. Integrity ensures that there is no change in data during transmission or storage by intruder or may be by noise. Authentication on the other end ensures that no unauthorized person transmit or gain confidential data.

1.1 SteganoFigurey

This is another way to provide confidentiality over secret text. It hides the secret text behind any other media file, so that no one can guess the presence of secret text. This media file can be text, image or audio, video file. Some times in order to increase the security both algorithms (encryption/decryption and steganoFigurey) get merged.

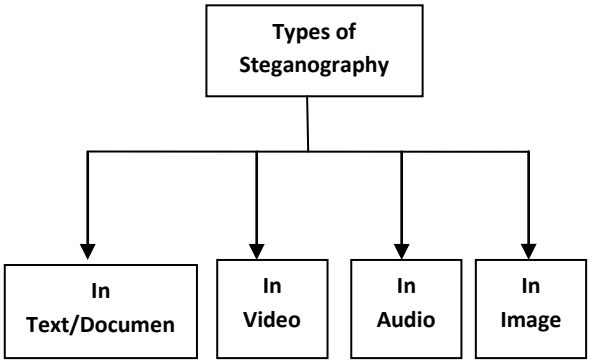

Figure 1: Shows types of steganoFigurey

1.2 CryptoFigurey

The dealing with the shuffling of message to maintain the security is known as cryptoFigurey. The techniques involved in cryptoFigurey are encryption and decryption. Encryption/ Decryption algorithm is an algorithm that shuffles the original secret text in a form that cannot be readable. This shuffling is done with the help of key. It decrypts or rearrange in the original form at receiving end with the same key. If there is a slight change in a sender and a receiver key then it is not possible to regain the original secret text. Fig 1 show the block diagram encryption and decryption process.

1.3 Encryption

The coding of message is done by following a particular algorithm i.e. by symmetric key algorithm. In symmetric key cryptoFigurey, one key is used at sender and receiver side to encrypt and decrypt the message.

1.4 Decryption

The inverse process of encryption is decryption. Here the message is decoded by implementation of algorithm to

recover it in its original form.

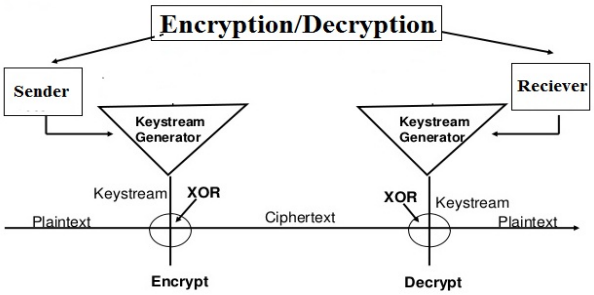

Figure 2: Encryption/decryption at sender side and receiver side

2. Literature Review

Earlier researcher was found that none of the algorithm exactly fulfils the security and computational time requirement. Some research work on timing of an algorithm but in order to improve the timing the compromise in security and some in order to improve security compromises in algorithm timing.

Ray and his colleagues in [3] proposed a new concept in which they first encrypt the secret text and then hide behind the text file. To hide the text behind text file they use two characters which are actually invisible characters having different ASCII value (i.e. 32 and 160). This algorithm after encrypt the secret text convert the resultant cipher text into binary format and then hide each bit behind every spaces of cover text. If bit is 0 replace the blank space by ASCII value 32 and if bit is 1 than replace blank space by character having ASCII value 160.

Tang and his colleagues in [2] presented their own algorithm. According to this algorithm the secret text first converted into hexadecimal format and then hides each hexadecimal character behind cover text character using defined R, G, and B method.

Uddin and his colleagues in [1] presented their algorithm in which secret text first encrypted using DES algorithm and then hide behind cover text file by defined method.

Nath and his colleagues in [4] proposed a new randomization algorithm for generating the randomized key matrix to encrypt plain text many times [4]. This algorithm is dependent on the random text that can be supplied by the user.

Dey and his colleagues in [5] presented a new integrated symmetric-key cryptoFigureic method, named SJA, which is the combination of advanced Caesar Cipher method, TTJSA method, Bit wise Rotation and Reversal method. The authors proposed that the present method will be most suitable for password, SMS or any kind of small message encryption.

Das and his colleagues in [9] presented a new A Challenge in hiding encrypted message in LSB and LSB+1 bit

positions in any cover files: executable files, Microsoft Office files and database files, image files, audio and video files [9].

3. Proposed Work

SteganoFigurey is the process of concealment of a secret message within other media file and cryptoFigurey is the technique of encryption in which a message is encrypted in coded form so that it will be unreadable to the unauthorized entity. The hiding of message is done in such a way that the message will be extracted at the destination. Whereas cryptoFigurey generates a coded version of message i.e. encrypting the message, so the message in an unreadable form will be transferred to the destination and the intended receiver decrypts the message in its original form. The work which has been proposed in this work to conceal the text message uses a combination of steganoFigurey and cryptoFigurey to hide the message with modified LSB method. In this proposed work before concealing the secrete message into another message, the secrete message is encrypted. For this encryption, the algorithm uses four 128 bit keys with pseudo random number and encrypts it. After that this message concealed into cover file using the RGB attributes of each character and send to the recipient. Encryption is performed on a secret message. As the message is made up of characters so the summation of ASCII values of each character is calculated for generation of pseudo random number. The encryption is performed on binary message with the help of a pseudo random number and key.

3.1 Pseudo Random Number Generation

A key of 128 bit is selected randomly and the mod operation is taken out between the key and sum of ASCII values that will generate a pseudo random number. Let Sum of ASCII value=Y and length of key is 16.

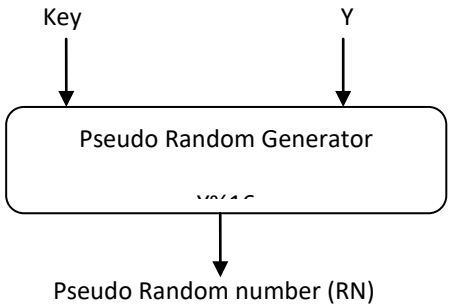

Figure 3: Pseudo Random Number Generation

3.2 Key Generation

In the proposed work we are using four keys instead of one key for encryption. These keys are generated by using 128 bit key and pseudo random number obtained above. Now let A, B, C and D are four variables defined to generate keys. First we assigned the assumed 128 bit key to the variable A. Then the circular left rotation Operation is carried out on A by pseudo random number and assigned the value to B. Now we perform the XOR operation between A and B to get the value of C. In the last the value of D is generated in the same manner as B

is calculated i.e. by performing circular left rotation on C as shown in Figure 3. Now the values of variables A, B, C and D are used with pseudo random number RN to generate four keys as follows: The mod is performed on RN and 4. Then four keys K1, K2, K3 and K4 are calculated. If Mod (RN, 4) is equal to zero then K1 value is set to the A and value of K2 is treated as B and so on. When the value of Mod (RN, 4) is equal to one, then K1 is set by the same value as B value. The value of K2, K3 and K4 are also obtained by putting C, D and A respectively. The calculation of K keys for values 2 and 3 i.e. result of Mod (RN, 4) is obtained from the respective manner as calculated in above values shown in Figure 4

3.2.1 Key Generation Algorithm

```
RN = GetRandom(A);

A = con_text_to_Bin(A);

B = leftrotate(A, RN);

C = xor(A, B);

D = leftrotate(C, RN);
```

```
else if (RN % 4 == 1)

{

K1=B; K2=C; K3=D; K4=A;

}

else if (RN % 4 == 2)

{
```

3.3 Encryption Process

Firstly to carry out the process of encryption, the message has to be converted into its binary format and the resultant sequences of bits are divided into equal size of chunks.

The size of each chunk is 128 bit, and if the size of last chunk is not equal to 128 bit then we do padding. The whole encryption process is divided into two rounds.

For the first round we perform circular left rotation over chunk by RN bits then with result of it we perform XOR operation with key K1.

Then again perform circular left rotation on result by RN bits and divide the result into left and right halves.

Next we perform swapping between halves and then XOR operation is performed on both halves after that we concatenates the result with right half. At last we perform XOR operation with key K2 and get the result of first round with intermediate PT.

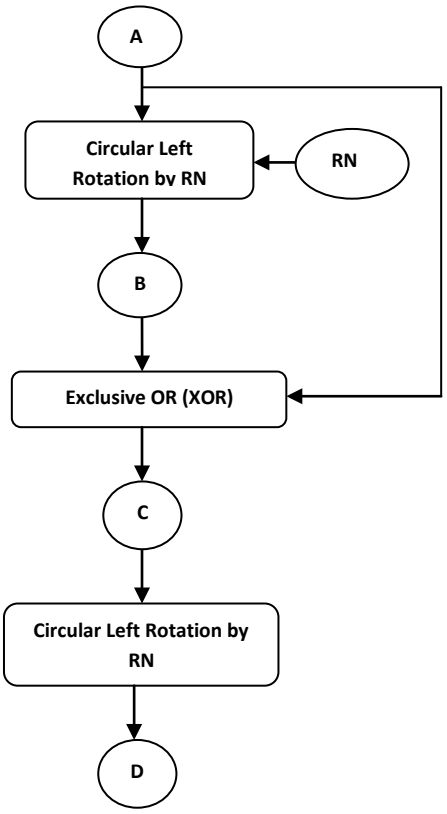

Figure 4: Variable generation

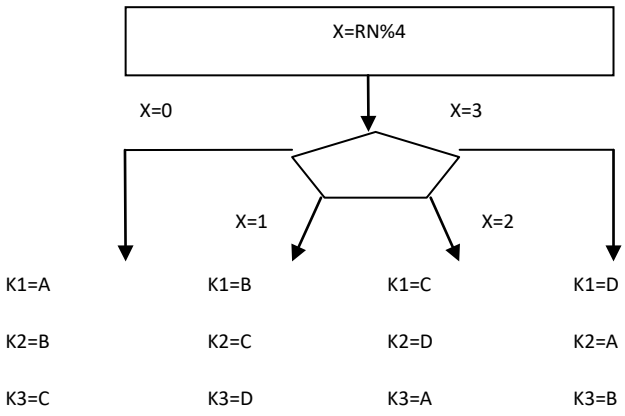

Figure 5: Key generation

3.3.1 Encryption Algorithm

The same process is repeated for the second round over intermediate PT with keys K3 and K4.

And find the final encrypted chunks PT. The whole process is shown in the flow chart Fig 6

```
for (int i = 0 to i < PlainText.Length)
{
PT =con_bin(PlainText.Substring(i, 16));
PT = leftrotate(PT, RN);
PT = xor(PT, K0);
PT = leftrotate(PT, RN);
PTL = PT.Substring(64, 64);
PTR = PT.Substring(0, 64);
PTL = xor(PTL, PTR);
PT = PTL + PTR;
PT = xor(PT, K1);
PT = leftrotate(PT, RN);
PT = xor(PT, K2);
PT = leftrotate(PT, RN);
PTL = PT.Substring(64, 64);
PTR = PT.Substring(0, 64);
PTL = xor(PTL, PTR);
PT = PTL + PTR;
PT = xor(PT, K3);
CipherText += con_to_Char(PT);
}
```

3.4 Hiding process

After performing the encryption the cipher text is hided. Here actually a cover file hides the secret cipher text within it As we know that the color of character depends on its attributes R,G and B.

We assume that cover text file is written in black characters hence color component R, G, B of each character is 0. Now, first calculate the number of 1's in cipher text, if it is more than 50% of total bits than invert all the bits else all bits remain same.

The concept behind inverting the bit is that, by default it is assume the cover file is written in black characters which mean that value of least significant bit of R, G and B component is 0. Hence, if the number of zeros in cipher text is more than changes in stego file is minimum shown in Figure 7.

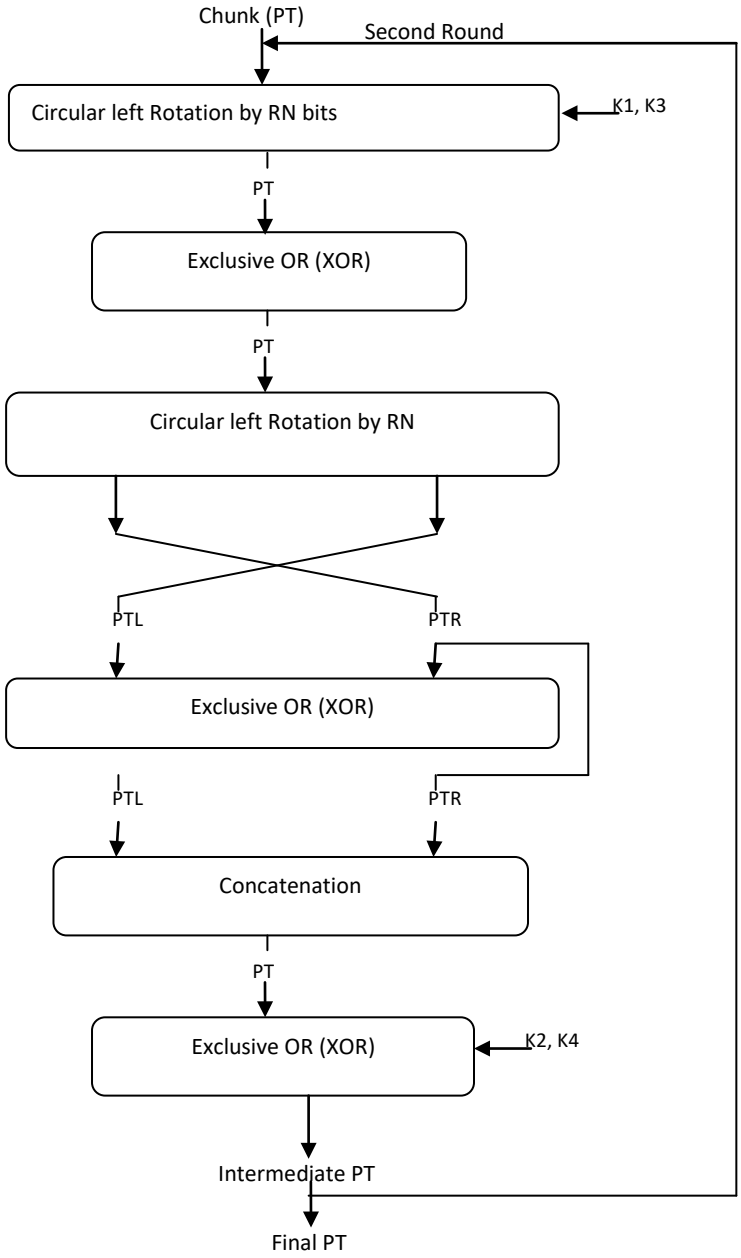

Figure 6: Encryption block

3.4.1 Hiding Algorithm

1. Take cover text file in black character i.e. value of RGB equal to 0.
2. Calculate number of 1's in cipher text
   - I. If number of 1's >50% than invert all the bit
   - II. If number of 1's <50% than all the bit are written same
   - III. Now, behind every RGB attribute of cover file character, hide each bit by just replacing least significant bit of component by cipher text bit

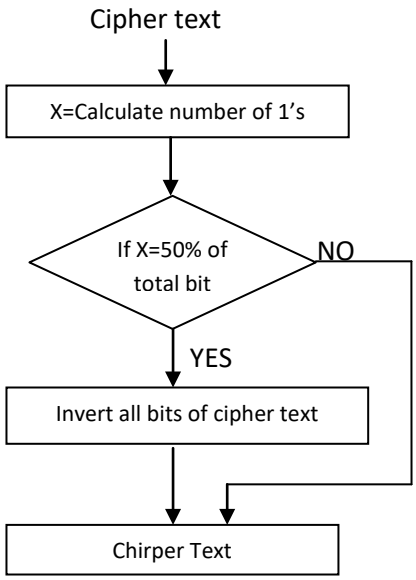

Figure 7: Bit Analysis

4. Performance Analysis

The proposed scheme is compared with Developing an Efficient Solution to Information Hiding through Text SteganoFigurey along with CryptoFigurey, Design And Implementation Of Information Hiding System Based On RGB, and A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm and. We used Dotnet framework to implement the above discussed algorithms using C# language. To compare the results of proposed algorithm with the existing algorithm authors keep the system configuration same for all. The presented results is calculated in Intel core I5 processor 2.40 GHz, 4GB RAM, windows-7 home basic service pack 1.

Number of parameters is used to compare and evaluate the performance of existing and proposed algorithms.

4.1 Performance Analysis of Encryption/ Decryption Algorithm

Designing of new algorithm is not enough, it is compulsory to evaluate the performance of proposed algorithm against other existing algorithms.

4.1.1 Timing Analysis

One of the important parameter to evaluate the performance of any algorithm is its timing analysis. An algorithm must be time efficient if it want to stand in competition in today's digital world. Timing comparison of DES algorithm with proposed algorithm is shown Table 1.

Table 1: Comparison of Encryption Time of Proposed Algorithm with DES on Various File Size in seconds

File Size in KB	**Algorithm**	
	Execution Time in Second	
	Proposed Algorithm	DES
5 KB	0.040	0.290
10 KB	0.090	0.720
20 KB	0.140	2.150

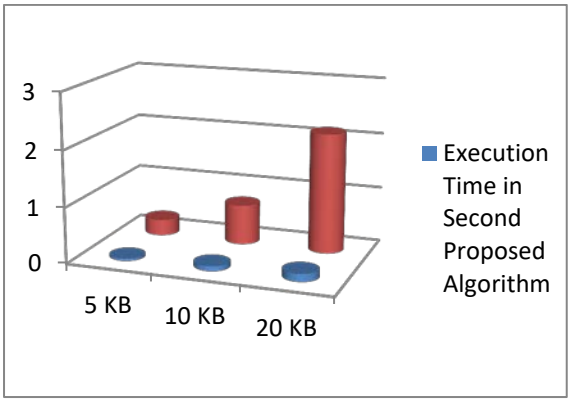

Figure 8: Comparison of Encryption Time of Proposed Algorithm with DES on Various File Size in seconds

Table 1 and Figure 8 clearly shows that proposed algorithm is highly time efficient than standard DES encryption algorithm.

4.1.2 Avalanche Effect

Calculating time efficient is not enough to calculate the performance of any algorithm. It is equally required to check the robustness of an algorithm. Robustness of any algorithm can be measured by avalanche effect.

According to avalanche effect, change of a single bit in a key change the cipher text fifty percent. This is an idle condition, an algorithm which is closed to idle condition consider more robust than other.

Table 2: Avalanche Effect of Proposed Algorithm

Key	**Proposed Algorithm**	**DES**
Key-1	49.63%	49.41%
Key-2		

Table 2 clearly shows that both the algorithms have almost same avalanche effect, which shows that both the algorithm are enough robust against any attack.

4.1.3 Key Analysis

All the security we gain in algorithm is of no use if encryption key is not enough strong. An algorithm having low key length have always chances of attack on it. Proposed algorithm uses 128 bit key which means it need 2^{128} combination to guess the key.

4.2 Performance Analysis of SteganoFigurey Algorithm

SteganoFigurey algorithm adds the security on proposed encryption/ decryption algorithm such that any attack can be done if its presence is known. SteganoFigurey hide the data behind cover text file so that no one can guess the presence of secret information. Again there are number of steganoFigurey algorithms discussed in Design and Implementation of Information Hiding System Based on RGB and A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm. Proposed steganoFigurey is designed in such a way that it is better than all existing algorithms.

4.2.1 Cover File Size

Size of cover file should be minimum. As the size of cover file is large, it requires more time to transmit or more space to store. Proposed algorithm requires cover file 8/3 times than secret file. Whereas Paper [3] require blank space equal to number of bits in secret text. Paper [2] require cover file 2 times more than secret file size. Standard LSB method required cover file size 8/3 times the secret file.

4.2.2 Peak signal to noise ratio (PSNR value)

Calculating PSNR value is important to evaluate the performance of steganoFigurey algorithm. An algorithm having high PSNR value consider better than other having low PSNR value. PSNR value detects the distortion

in cover file causes due to hiding of secret text. Table 3 shows the comparative analysis of PSNR value

Table 3: PSNR value comparison of proposed steganoFigurey algorithm

	Proposed Algorithm	**Paper [3]**	**Paper[2]**	**Standard LSB**	**Paper [1]**
PSNR Value	65.70%	19.13%	62.46%	62.94%	9.25%

Again, It is clearly seen from Figure 9 or Table 3 that proposed algorithm have high PSNR value, it means proposed algorithm have less distortion than any other existing algorithm. Paper 2 is modified LSB method which requires less cover file size but at the same end it has low PSNR value than proposed algorithm. Proposed algorithm modified the standard LSB but in such a way that it improves PSNR value without compromise in file size.

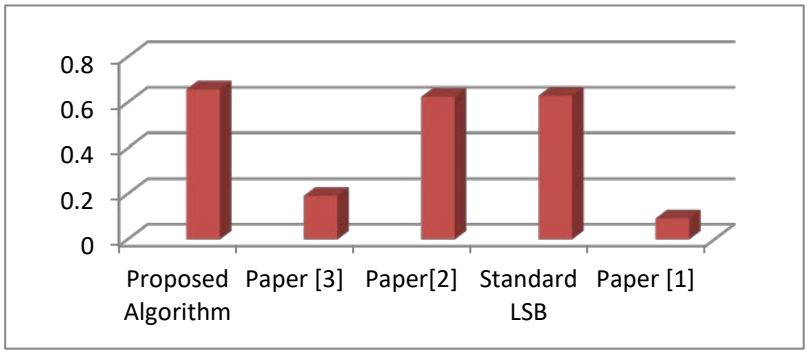

Figure 9: PSNR value comparison of proposed steganoFigurey Algorithm

5. Conclusions

This paper provides an efficient hiding technique for data and efficient security algorithm. The security of proposed algorithm enhanced because of the reason that it firstly encrypts the message through symmetric key encryption technique and then it applies the hiding on cipher text by modified LSB algorithm. Proposed algorithm has better results than previous work.

The results obtained in paper shows that security is greater as compared with the problem raised in previous paper. Proposed algorithm is designed in such a way that it improves some parameters of existing algorithms without compromising the other parameters. As the proposed algorithm is time efficient it is best suitable for Ad hoc network or live communication.

References

[1]. Md. Palash Uddin, Mousumi Saha, Syeda Jannatul Ferdousi, Masud Ibn Afjal, Md. Abu Marjan, "Developing an Efficient Solution to Information Hiding through Text SteganoFigurey along with CryptoFigurey" The 9th International Forum on Strategic Technology (IFOST), October 21-23, 2014, Cox's Bazar, Bangladesh, IEEE

[2]. Xing Tang, Mingsong Chen, "Design And Implementation Of Information Hiding System Based On RGB", 2013 IEEE.

[3]. Rishav Ray, Jeeyan Sanyal, Debanjan Das, Asoke Nath, "A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm", 2012 International Conference on Communication Systems and Network Technologies, IEEE.

[4]. Symmetric Key CryptoFigurey using Random Key generator : Asoke Nath, Saima Ghosh, Meheboob Alam Mallik: "Proceedings of International conference on security and management(SAM'10" held at Las Vegas, USA Jul 12-15, 2010), P-Vol-2, 239-244(2010).

[5]. An Integrated Symmetric key CryptoFigurey Algorithm using Generalised modified Vernam Cipher method and DJSA method: DJMNA symmetric key algorithm : Debanjan Das, Joyshree Nath, Megholova Mukherjee, Neha Choudhary, Asoke Nath: Communicated for publication in IEEE International conference WICT 2011 to be held at Mumbai Dec 11-14, 2011.

[6]. Data Hiding and Retrieval : Asoke Nath, Sankar Das, Amlan Chakraborty, published in IEEE "Proceedings of International Conference on Computational Intelligence and Communication Networks(CICN 2010)" held from 26-28 NOV' 2010 at Bhopal.

[7]. Advanced SteganoFigureic approach for hiding encrypted secret message in LSB, LSB+1, LSB+2, LSB+3 bits in non standard cover files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, International Journal of Computer Applications, Vol- 14, No. 7, Page-31-35, Feb (2011).

[8]. Advanced SteganoFigurey Algorithm using encrypted secret message : Joyshree Nath and Asoke Nath, International Journal of Computer Science and Applications, Vol-2, No. 3, Page- 19-24, Mar (2010).

[9]. A Challenge in hiding encrypted message in LSB and LSB+1 bit positions in any cover files: executable files, Microsoft Office files and database files, image files, audio and video files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath : JGRCS, Vol-2, No. 4, Page- 180-185, Apr (2011).

[10]. New Data Hiding Algorithm in MATLAB using Encrypted secret message: Agniswar Dutta, Abhirup Kumar Sen, Sankar Das, Shalabh Agarwal and Asoke Nath : Proceedings of IEEE CSNT- 2011 held at SMVDU (Jammu), 03-06 Jun, 2011, Page 262-267.