I. J. Computer Network and Information Security, **2018, 5, 60-69**
Published Online May 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.05.07

Cryptography Based on RGB Color Channels using ANNs

Sanjay Kumar Pal
Department of Computer Science and Applications, NSHM College of Management & Technology, Kolkata, 700053, India
E-mail: sarbojay@gmail.com

Sumeet Anand
Department of Computer Science and Applications, NSHM College of Management & Technology, Kolkata, 700053, India
E-mail: anandsumeet57@gmail.com

Received: 22 February 2018; Accepted: 18 April 2018; Published: 08 May 2018

Abstract—Information is by far the most precious thing in almost every field. Everything we do in the present-day world generated some data and most of the data are vulnerable to unwanted threats. The organizations and agencies are becoming more and more dependent on their digitized information systems. Also, the general public is slowly getting cyber- conscious and thus they also fear for the leak and tampering of their secured information. Today's information systems are under the constant threats of manipulation and overriding by various criminal organizations. Thus, the information in today's world is kept under the password authentication. These passwords are a combination of a string of alphanumeric and special characters. Also, the key used to encrypt the information are exposed to either both or one of the parties. To overcome this vulnerability, an encryption technique is proposed where the key will be generated and transmitted using TPM and the final encrypted text will be stored in the image format by segregating the text data into the 3-channelled image, i.e., RGB.

Index Terms—Artificial Neural Networks, connectionist systems, Image Encryption, Cryptography, Decryption.

I. Introduction

There have been various methods devised for secured transmission of information over the internet. One of them is encryption of data before it is transmitted to the server so that even when there is an unauthorized capturing of encrypted data, the actual data mustn't be available to the unauthorized channel. They mustn't be able to deduce the original data from those encrypted texts. The conventional method is more vulnerable to these cyber-attacks such as phishing, brute force, and dictionary attacks.

The method that came into existence to tackle this problem was biometric scans. These scans include voice pattern matching, retina scan, face detection, etc. The major problem with these techniques was the cost of set-up. Not all the organizations were able to put on the expenses and obviously, it wasn't possible for an individual user to get a biometric system at it challenges feasibility. Also, these scans were not always accurate and many times they failed to provide the required portability.

Nowadays, there is a significant increase in the use of one time passwords or OPT. Although, it is a very much secure form to the conventional cyber- attacks, and that is the reason why banks use them for various verifications, but it was not possible for the general people or all the organizations to use OTP for the cost at which it is obtained.

Also, for most of the encryption and decryption algorithms, there is at least one or one pair of keys which is used to encrypt and decrypt the text. In public key encryption techniques, a pair of keys is generated and one of them is used for the encryption and the other is used for decryption; the two keys are provided to sender and receiver, respectively. Whereas, in private key encryption both of the methods of encryption as well as decryption used the same private key. In both of these two methods of encryption and decryption, the keys are exposed to the sender and receiver and there is a chance that an attacker can get hold of the keys by any means necessary.

For trying to tackle these problems, we are going to propose a method of encryption and decryption based on the 3-channeled image. Here, the key generation and distribution are done with the help of a Tree Parity Machine or TPM. The random key is generated when the two TPMs synchronize and then a key is transmitted in between them which is not exposed to any system outside the two synchronized systems. The key is then used by the system to encrypt the text using the proposed algorithm and then by separating and putting the text to the three channels or an RGB image using the algorithm.

II. Terminologies

Before we go into the details of the subject matter we must get familiar with certain technical terms which will be referred throughout the document frequently. A brief explanation of these terms will help to understand the reader about the subject matter without falling into the pit of doubt.

A. *Artificial Neural Networks*

Artificial Neural Networks are a kind of computing system which is inspired by the way the Biological Neural Networks function. The most important feature of an ANN is the learning capabilities that it offers. Unlike conventional computing systems, they are not programmed for any specific task, rather they learn the things according to the learning experiments done upon them and after a few iterations, they become capable to realize patterns. This system was designed to help the computer systems to solve the problems in the same way that a human mind would do, by learning.

B. *Tree Parity Machine*

A Tree Parity Machine or TPM is a multilayer feed-forward neural network. A feed-forward neural network is an ANN in which the connections between the units do not form a loop [1].

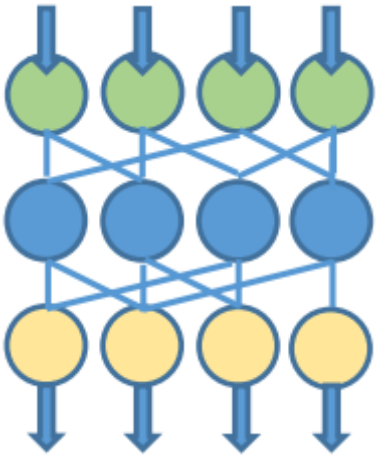

Fig.1. Tree Parity Machine

It is one of the first and simplest types of artificial neural networks and the information moves in only one direction, i.e., forward. The data moves from the input notes to the hidden nodes and finally from the hidden nodes to the output nodes. There are no cycles or loops in this network [1].

A TPM is a special type of feed-forward multi-layered ANN where there is 1 output neuron, K hidden neurons, and K*N input neurons. The input to the networks are able to take 3 values:

$$x_{ij} = \{-1,0,1\}$$

The weights between input and hidden neurons are in the following range:

$$wij = \{-L, \dots, 0, \dots + L\}$$

The output value of each hidden neuron is calculated as a sum of all multiplications of input neurons and these weights:

$$\sigma_{i=} \operatorname{sgn}\left(\sum_{j=1}^{N} w_{ij} x_{ij}\right)$$

Signum function returns -1,0 or 1:

$$\operatorname{sgn}(x) = \begin{Bmatrix} -1 \; if \; x < 1 \\ 0 \; if \; x = 0 \\ 1 \; if \; x > 1 \end{Bmatrix}$$

If the scalar product is 0, the output of the hidden neuron is mapped to -1 in order to ensure a binary output value. The output of the neural network is then computed as the multiplication of all values produced by hidden elements:

$$\tau = \prod_{i=1}^{K} \sigma$$

The output of the tree parity machine is binary.

C. *Cryptography*

Cryptography is the study and practice of secure transmission of messages from the adversaries. The main work of cryptography is to keep the original data secured by manipulating it in such a way that no meaningful data can be derived from it until the manipulation is reversed. In information security, it is about constructing protocols and algorithms that will keep the third parties from reading private messages between the sender and receiver. In general, cryptography is the art and science of converting a meaningful text to utter nonsense to prevent the third party readability. It can only be transformed back to its original state by the means of a key.

D. *Public Key Cryptography*

In public key cryptosystems, a pair of keys is used one of which is with the sender while the other one is with the receiver. The sender uses this key to generate the cipher while the receiver decodes the cipher with the other pair of the key. These keys can't be generated from each other. It is also called Asymmetric Key cryptography [3].

E. *Private Key Cryptography*

In private key cryptosystems, the same key is used to encrypt as well as decrypt the text. The key is with both the sender as well as the receiver. Symmetric Key cryptography [3].

F. *Plaintext*

The private message which is to be sent to the receiver is called plain text. The algorithm is applied to it to

encrypt it.

G. *Cipher Text*

The encrypted text which is obtained after the encryption algorithm is applied on the plaintext is called ciphertext.

H. *KEY*

The key is the random string of bits used to scramble and unscramble the text data. Encryption keys are designed with algorithms intended to ensure that every key is unpredictable and unique.

III. Using Private Key Cryptography to Send Messages Securely

In private key cryptosystems, the same key is used to both encrypt the plaintext as well as decrypt the ciphertext. The keys most of the type is same but sometimes there can be a small transformation between the two key [4]. The key is usually shared between two or more parties which they can use to communicate with each other using a medium for maintaining a secure and private information link [5]. The major drawback of this system is that the key is available with various parties so the intruder can try to get the hold of a key from any of the parties by any means necessary to authenticate into the private link. If it comes to that then the public key encryption system is far more secured just because the keys of this system can't be generated from one another [6].

But the private key cryptosystems can be made much more secure when the key is new every time the connection is made and also the key is not available to the users but is within the system itself. It means that there will be no physical exchange of key between the users but the system will itself synchronize and exchange a mutual key. A similar approach is found in neural cryptography.

IV. Neural Cryptography

In neural cryptography, two or more machines are synchronized and then there is an exchange of a mutual key. The neural key exchange is done when two or more TPMs are synchronized.

The protocol to synchronize two or more TPMs are as follows:

Each party (A and B) uses its own tree parity machine. Synchronization of the tree parity machines is achieved in these steps

a. Initialize random weight values
b. Execute these steps until the full synchronization is achieved

1. Generate random input vector X
2. Compute the values of the hidden neurons
3. Compute the value of the output neuron
4. Compare the values of both tree parity machines

i. Outputs are different: go to b.1
ii. Outputs are same: one of the suitable learning rules is applied to the weights

After the full synchronization is achieved (the weights w_{ij} of both tree parity machines are same), A and B can use their weights as keys. This method is known as a bidirectional learning.

One of the following learning rules [7] can be used for the synchronization:

- Hebbian learning rule:
- Anti- Hebbian learning rule:
- Random Walk:

V. RGB Color Model

The RGB color model is an additive color model in which the three primary colors, i.e., Red, Green, and Blue can be added together to get a wide array of colors. One common application of the RGB color model is the display of colors on a cathode ray tube (CRT), liquid crystal display (LCD), plasma display, or organic light emitting diode (OLED) display such as a television, a computer's monitor, or a large scale screen.

In digital media, the smallest unit of a picture is called a picture element or a Pixel. A pixel is a combination of the three colors to generate a new color according to the specifications provided. During digital image processing, each pixel can be represented in the computer memory or interface hardware (for example, a graphics card) as binary values for the red, green, and blue color components. When properly managed, these values are converted into intensities or voltages via gamma correction to correct the inherent nonlinearity of some devices, such that the intended intensities are reproduced on the display.

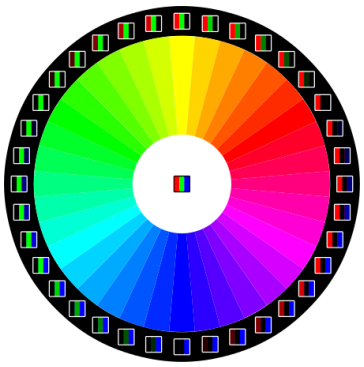

Fig.2. The RGB- Color Wheel

In the most basic RGB scheme, each color can be assigned with a value of 8 different bits, thus making it a 24-bit RGB model, hence, it is called RGB24. The values of these bits can be in a range of 0-255. Now, there can

be 256 values of each R, G, and B; so, a total of 256^3 colors can be possible which is equal to 1,67,77,216 colors.

VI. Proposed Algorithm

In this proposed model of cryptography, the whole process of encrypting the text can be divided into three steps. These steps include:

- Neural Key Generation.
- Making the key to the exact length as of the Plain Text.
- Obtaining ciphertext using the key
- Converting the encrypted cipher to an RGB color image, pixel by pixel.

Finally, for the decryption, the second machine will use the same key and extract the RGB values of each pixel and then reverse the encryption process to obtain the original text. Again, when the next time a connection is established a new key will be generated and the process will go on. Let us look into the process of encryption and decryption in detail.

A. Encryption

The process of encryption consists of three phases:

1) Neural Key Generation

For a neural key generation, we will use TPMs and synchronize them. For synchronizing we will use the following algorithm:

Each party (A and B) uses its own tree parity machine. Synchronization of the tree parity machines is achieved in these steps:

a. Initialize random weight values
b. Execute these steps until the full synchronization is achieved

1. Generate random input vector X
2. Compute the values of the hidden neurons
3. Compute the value of the output neuron
4. Compare the values of both tree parity machines.

   a. Outputs are different: go to b.1
   b. Outputs are same: one of the suitable learning rules is applied to the weights

After a full synchronization is achieved, the TPMs can use the weights as the keys and/or according to the weights they can generate the key with a given set of printable characters. After the synchronization, we will use the following code snippet to generate an 8-bit key:

```
public String makeKey(){
    StringBuilder key=new StringBuilder();
    int keySize=(int)(ABC.length()/(L*2+1));
    int keyLength=(int)(K*N /keySize);
    for(int i=1;i<keyLength;i++){
      int k=1;
      for(int j=(i-1)*keySize;j<i*keySize;j++){
        k+=w[j]+L;
      }
      key.append(ABC.charAt(k));
    }
    return key.toString();
  }
  static final String
ABC="ABCDEFGHIJKLMNOPQRSTUVWXYZ_0123
456789abcdefghijklmnopqrstuvwxyz";
```

1. The key is generated in the printable characters using the characters in the string ABC.
2. The key size is determined by the formula:

(length of(ABC))/(L*2+1), here L is range of weights of TPM

3. The key length is determined by the formula:

(K*N)/keySize, here K is no of hidden neuron and N is no of Input neuron in TPM

4. They key is generated using the loop:

```
for(int i=1;i<keyLength;i++){
int k=1;
for(int j=(i-1)*keySize;j<i*keySize;j++){
k+=w[j]+L;
}
key.append(ABC.charAt(k));

}
```

Let the generated key be **abcdefgh**.

2) Making the key to the exact length as of the Plain Text.

The key generated is of the size of the 8 characters. If the Input string is smaller than the 8 characters, then we need to downsize the key else the key has to be increased up to the number of input characters. The following code is used:

```
public static String getKey(String key, int len) {
    int length = key.length();
    if (length < len) {
      StringBuilder outKey = new
StringBuilder(key.toString());
      int i = 0;
      while (outKey.length() < len) {
        outKey.append(outKey.charAt(i));
        i++;
      }
      return outKey.toString();
    } else {
```

```
        return (key.substring(0, len));
    }
}
```

Let the plain text **PUZZLING**.

3) Obtaining ciphertext using the key

Take an array CharValue[ ] and store the ASCII value of each character of input into it:
For the input – "ABCDEFGH", the ASCII values are

P-CharValue[0]- 80
U- CharValue[1]-85
Z- CharValue[2]-90
Z- CharValue[3]-90
L- CharValue[4]-76
I- CharValue[5]-70
N- CharValue[6]-78
G- CharValue[7]-71

Take another array CharMod and a variable minValue
Take the minimum value from the set of ASCII values and assign it t minValue
Here, minValue=71
Modulo the values of each charValue with minValue and store the respective values in charMod[ ].
Thus,

charMod[0] = charVaue[0] % 71 = 9
charMod[1] = charVaue[1]% 71 = 14
charMod[2] = charVaue[2] % 71 = 19
charMod[3] = charVaue[3] % 71 = 19
charMod[4] = charVaue[4] % 71 = 5
charMod[5] = charVaue[5] % 71 = 2
charMod[6] = charVaue[6] %71 = 7
charMod[7] = charVaue[7]% 71 = 0

Since the String length is 8 and key is 8 characters, we will not do any appending,
The original key was: abcdefgh

Now, we will take another array KeyValue[ ] and store the ASCII values of the key in it and then we will take another array and store the Modulo of each character by the minKeyValue; which is the smallest ASCII value from the set of values from the KeyValue[ ].
So,

a. KeyValue[0]= 97
b. KeyValue[1]= 98
c. KeyValue[2] =99
d. KeyValue[3] =100
e. KeyValue[4]= 101
f. KeyValue[5] =102
g. KeyValue[6]= 103
h. KeyValue[7]= 104

Now, minKeyValue = 97

Thus,

KeyMod[0] = KeyValue[0] % 97 = 0
KeyMod[0] = KeyValue[1] %98 = 1
KeyMod[0] = KeyValue[2] %99 = 2
KeyMod[0] = KeyValue[3] %100 = 2
KeyMod[0] = KeyValue[4] %101= 4
KeyMod[0] = KeyValue[5] %102 = 5
KeyMod[0] = KeyValue[6] %103 = 6
KeyMod[0] = KeyValue[7] %104 = 7

Now, let us take another array EncKey[ ] which will act as the final key for encryption:

EncKey[i] = CharMod[i] + KeyMod[i]

Thus, we have

EncKey[0] = CharMod[0] + KeyMod[0] = 9+0= 9
EncKey[1] = CharMod[1] + KeyMod[1] = 14+1= 15
EncKey[2] = CharMod[2] + KeyMod[2] = 19+2= 21
EncKey[3] = CharMod[3] + KeyMod[3] = 19+3= 22
EncKey[4] = CharMod[4] + KeyMod[4] = 5+4 = 9
EncKey[5] = CharMod[5] + KeyMod[5] = 2+5 = 7
EncKey[6] = CharMod[6] + KeyMod[6] = 7+6 = 13
EncKey[7] = CharMod[7] + KeyMod[7] = 0+7= 13

Now, to get the ciphertext out of each character, we will add minValue(32) to each element of EncKey[ ].
Thus,

CipherText = EncKey + minValue

Therefore,

CiphetText[0] = EncKey[0] + minValue = 9+71 = 80
CiphetText[1] = EncKey[1] + minValue = 5+71 = 86
CiphetText[2] = EncKey[2] + minValue = 21 +71 = 92
CiphetText[3] = EncKey[3] + minValue = 22+71 = 93
CiphetText[4] = EncKey[4] + minValue = 9+71 = 80
CiphetText[5] = EncKey[5] + minValue = 7+71 = 78
CiphetText[6] = EncKey[6] + minValue = 13+71 = 84
CiphetText[7] = EncKey[7] + minValue = 13+71 = 84

4) Converting the ciphertext to RGB color image, pixel by pixel

We will take 3 arrays,

R[ ], G[ ], B[ ]

Here, RGB are the components of a pixel,

R = Red
G = Green
B = Blue

A pixel is composed of the value of each element ranging from 0-255.

The obtained CipherText[] will be divided into groups of three and if in the last group we have elements less than three, we will assign them with 0 to make it a group of three too.

- The first element of every group will be assigned to the array R[].
- The second element of every group will be assigned to the array G[].
- The third element of every group will be assigned to the array B[].

This will take place in this loop:

```
for(int y = 0; y < height; y++){
    for(int x = 0; x < width; x++){
      int r = R[]; //red
      int g = G[]; //green
      int b = B[]; //blue

      int p = (r<<16) | (g<<8) | b; //pixel

      img.setRGB(x, y, p);
    }
```

The CipherText = {80, 86, 92, 93, 80, 78, 84, 84}
Let us divide into groups of three:

{80, 86, 92}, {,93, 80, 78}, {84, 84, 0}

Since the last group was lacking an element so we put a zero there.
Now, each of these group will constitute a pixel.
Let us see the results pictorially.

Pixel [1] = {80, 86, 92}
Pixel [2] = {93, 80, 78}
Pixel [3] = {84, 84, 0}

[P, U, Z] [Z, L, I] [N, G]

Fig.3. The Image form of the String

This is the encrypted message that can now be sent to the receiver.

B. Decryption

For decryption, firstly we will extract the RGB values of each pixel.
The pixel data of each pixel of image can be extracted from the following code snippet:

```
for (int i = 0; i < width; i++)
    {
      for (int j = 0; j < height; j++)
      {
        int pixel = image.getRGB(i, j);
        String idata=(getARGBPixelData(pixel));

        System.out.print(idata);
        if(pix_num<total_pix)       //To delete the line
that generates at end of file
        {
          System.out.print("");
        }
        pix_num++;
      }
```

From there, all the color elements can be backtracked and we can store the ASCII value of ciphertext into an array DecValue[].
We have the value of EncKey[].
We will subtract the value of EncKey[] from the DecValue[].
Thus,

difference[i] = DecValue[i] – EncKey[i]

If the values generated are same in each case the,
We will regenate plaintext[] by adding CharMod[] and difference[ ].
For this case,

difference[0]= DecValue[0] - EncKey[0]= 80 – 9 = 71
difference[1] =DecValue[1] - EncKey[1]= 86 -15= 71
difference[2] =DecValue[2] - EncKey[1]= 92 -21 = 71
difference[3] =DecValue[3] - EncKey[1]= 93 -22 = 71
difference[4] =DecValue[4] - EncKey[1]= 80 - 9= 71
difference[5] =DecValue[5] - EncKey[1]= 78 -7 = 71
difference[6] =DecValue[6] - EncKey[1]= 84 -13 = 71
difference[7] =DecValue[7] - EncKey[1]= 84 -13 = 71

Since the differences of the elements are all equal, add the difference to CharMod[].

So, plaintext[i] = CharMod[i] + difference

Here,

plaintext[0] = CharMod[0] + difference = 9 + 71 = 80
plaintext[1] = CharMod[1] + difference = 14+ 71 = 85
plaintext[2] = CharMod[2] + difference = 19+ 71 = 90
plaintext[3] = CharMod[3] + difference = 19+ 71 = 90
plaintext[4] = CharMod[4] + difference = 5+ 71 =76
plaintext[5] = CharMod[5] + difference = 2 + 71 =73
plaintext[6] = CharMod[6] + difference = 7 + 71 = 78
plaintext[7] = CharMod[7] + difference = 0 + 71 = 71

ASCII values of plaintext[i] are the original letters,

ASCII Text

80	P
85	U
90	Z
90	Z
76	L
73	I
78	N
71	G

So, the text was PUZZLING.

We can directly print the text value in a text file using the following code snippet:

```
int width = image.getWidth();
    int height = image.getHeight();
    int pix_num=1;
    int total_pix=width*height;

    System.out.println("Image Dimension: Height-" + height + ", Width-" + width);
    System.out.println("Total Pixels: " + (height * width));
        for (int i = 0; i < width; i++)
    {
       for (int j = 0; j < height; j++)
       {
          int pixel = image.getRGB(i, j);
          String idata=(getARGBPixelData(pixel));

          System.out.print(idata);
          if(pix_num<total_pix)       //To delete the line that generates at end of file
          {
             System.out.print("");
          }
          pix_num++;
       }
    }
```

Thus, we get out decrypted data.

VII. Analysis

A very important part of an algorithm is the complexity. In a system, there are more than one components and complexity is the study of how well these components can interact with each other. The two most important types of complexity check in a computer system are space and time complexities. The space complexity of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input. Similarly, Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.

The memory of a computer is really cheap these days so the matter of discussion is Time complexity. Time and space complexity depends on lots of things like hardware, operating system, processors, etc. However, we don't consider any of these factors while analyzing the algorithm. We will only consider the execution time of an algorithm.

A. Key Generation

The key generation process consists of a nested loop of 2-levels. The outer loop runs upon key length whereas the inner loop runs upon key size. Let the key length be n and key size be m, therefore, the complexity of key generation is $O(m^n)$.

B. Encryption of Plain Text to Cipher Text

For the encryption of plain text into cipher text, only one loop of a for is required which has an upper limit of the plaintext size. If the size of plaintext is ‘s’ then the complexity of plain text to cipher text conversion is O(s).

C. Conversion of Cipher Text to Image

To convert the ciphertext into the RGB channelled image, a nested for loop will be required where the outer loop will be for the height of the image while the inner loop will be for the width of the image in pixels. If we have an input text of N characters, we will use the nearest upper limit of the perfect square root of N.

If x = sqrt(N), then the complexity of this loop will be $O(n^2)$.

We have analyzed the algorithm to convert the string to bitmap image format using the Java programming language. The reason for picking up bitmap image format are:

i. The bitmap image format is device independent.
ii. BMP format files are uncompressed bitmapped images.
iii. BMP formatted images have a higher resolution.
iv. It is possible to edit each individual pixel.
v. It is a lossless format of image format, i.e., all the original data can be recovered when the file is uncompressed.

Table 1. Time of Text to Image And Back

Length of String	Text to Image time (ns)	Image to Text time (ns)
1	159384653	189294731
10	152979555	174990358
20	162071585	172735582
30	162159096	166732930
40	149728120	173981812
50	120798496	180638100
60	146368644	197898669
70	123981326	134095405
80	122119273	124859144
90	116294887	122127376
100	121587181	129467551
200	135970964	128009023
400	121170149	168596604
800	145433026	158379349
1000	153354451	157318405

The results for the conversion of string to image and again image to string is displayed in the table below. The time has been expressed in the terms of nanoseconds (ns).

VIII. Related Works

There have been various and various types of work on image-based encryption techniques.

A lot of techniques are there where graphical methods are not required, they use text-based encryption only and are successful. They range from random number generation [18] for keys, using cosmos law [19] to compound key generation. But many algorithms and approaches have been defined where the concept of colour channels have been used.

In the year 2012, Color Coded Cryptography was published in which the author proposed a color-coding scheme that can be used for data encryption which represents text in the form of colored blocks by grouping together binary bits and assigning them colors along with Huffman encoding scheme which is used for lossless text compression [8].

In the year 2012, Graph Coloring Approach for information hiding was published in which the author proposed a graph coloring based watermarking system and also analyzed its credibility. It was a constrained-based watermarking technique and it's of a theoretical framework's layout of watermarking techniques for intellectual property protection (IPP) [16] [7].

The year 2012 also witness the publication of a paper by Satyendra Nath Mandal,Subhankar Dutta,Ritam Sarkar titled- "Block Based Symmetry Key Visual Cryptography". It followed the approach of visual cryptography [24]. Visual cryptography has a unique computation free decoding and the results are human understandable.

In 2013, Image Encryption based on the RGB PIXEL Transposition and Shuffling proposed a technique to transpose and reshuffle the RGB values which had a great security value [9]. Later, in the same year another paper that also dealt with RGB PIXEL Transposition and Shuffling was published thus, taking image based encryption over the top again [21] . In the same year, another paper was published titled- A Review of Image Encryption Technique based on Hyper Image Encryption Algorithm that proposed block-based image encryption and Hyper Image encryption techniques [10]. Another paper was published which dealt with more or less with the same approach. It was titled- New Image Encryption Techniques Based On Combination of Block Displacement and Block Cipher Technique and it used to a 128-bit key. The security was really high because of the same and there were no chances of floating point errors [11].

In the same year, A first approach on an RGB encryption was published and it dealt with image encryption using TSRMAC associated with DWT. A formula for matrix affine cipher on an RGB image was proposed where extracting keys and correct arrangement of RMAC parameters were mandatory [12].

In 2015, An enhanced technique of colour image encryption based on random matrix key encoding was proposed in the paper titled- Encryption- Decryption RGB colour image using matrix multiplication. The image was separated into the RGB channels and each channel was encrypted using a technique called double random matrix key encoding then three new coding image matrices were constructed [13].

In the same year, Color Code Based Authentication and Encryption was published that stated a new method to convert the text string into a set of color codes using the randomly generated color maps [14]. Another paper titled RGB Based Secret Sharing Scheme in Color Visual Cryptography proposed a method for images with 256 colors which are converted to 16 standard RGB colors format. It generates shares without compromising the resolution. The Floyd – Steinberg dithering algorithm is used to manipulate the 256 color code image to reduce it to 16 standard colors code image. The proposed method employs (2, 2) XOR-Based visual cryptography method is also used to generate shares. Decryption procedure enables secret image sharing and stacking [15].

In 2016, Nisar Ahmed, Hafiz Muhammad Shahzad Asif, Gulshan Saleem undertook a performance based evaluation of various image-based encryption techniques and the results were published [23].

IX. Future Scopes

There is a great scope of image-based encryption in the near future and especially with ANN based systems, it will be highly beneficial to emulate a real-world scenario of processing the information hiding the way it is meant to do naturally. The only reason why the computers are hackable is that it has no natural perception but with the rise of ANNs, the high perception level of computers systems is on the way. This perception can be used with a wide range of procedures to make the computer system more and more secure. Also, this encryption technique can be used in smart cards as it can be easily applied by using a programming language which has the capability to manipulate image files such as C, C++, Java, Python, etc. It can be highly anticipated by the simple yet effective process.

As different and unique methods of encryptions have been coming up, such as the real-time audio encryption [22], the image based cryptography can also unlock the various other modes of securing data and information.

X. Conclusion

This paper dealt with an ANN-based color coded cryptography. The ANN was used as a Tree Parity Machine and it implemented the synchronization of the systems of the sender and the receiver. When the systems were synchronized, it generated a key which was transmitted to both of the systems. This key was later manipulated to encrypt the text into an image form of RGB channels. Moreover, the paper threw lights on the

various aspects of Artificial Neural Networks, Cryptography, and RGB color space. It discussed various proposed methods of image-based cryptography.

REFERENCES

[1] Zell, Andreas (1994). Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.). Addison-Wesley. p. 73. ISBN 3-89319-554-8.

[2] Diffie, W.; Hellman, M. (1976). "New directions in cryptography" (PDF). IEEE Transactions on Information Theory. 22 (6): 644–654. doi:10.1109/TIT.1976.1055638.

[3] Matt Valeriote, ―Public Key Cryptography‖, McMaster University, October 2014.

[4] Kartit, Zaid (February 2016). "Applying Encryption Algorithms for Data Security in Cloud Storage, Kartit, et. al". Advances in ubiquitous networking: proceedings of UNet15: 147.

[5] Delfs, Hans & Knebl, Helmut (2007). "Symmetric-key encryption". Introduction to cryptography: principles and applications. Springer. ISBN 9783540492436.

[6] Mullen, Gary & Mummert, Carl (2007). Finite fields and applications. American Mathematical Society. p. 112. ISBN 9780821844182.

[7] Kumar Pal, Sanjay & Sen Sarma, Samar. (2012). Hiding Information Using the Graph Colouring Technique. International Journal of Applied Research on Information Technology and Computing. 3. 172. 10.5958/j.0975-8070.3.3.017.

[8] Aditya Gaitonde, "Color Coded Cryptography", International Journal of Scientific & Engineering Research, Volume 3, Issue 7, July-2012 1 ISSN 2229-5518

[9] Q. A. Keste, "Image Encryption based on the RGB PIXEL Transposition and Shuffling," I. J. Computer Network and Information Security, 7, in MECS (http://www.mecs-press.org/), DOI: 10.5815/ijcnis.2013.07.05, pp.43-50, Published Online June 2013

[10] P. Junwale, R. M. Annapurna, and G. Sobha, "A Review on Image Encryption Technique based on Hyper Image Encryption Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 11, pp. 614-618, November – 2013.

[11] K. Kushwah, S. Shibu, "New Image Encryption Technique Based On Combination of Block Displacement and Block Cipher Technique," International Journal of Computer Science and Information Technologies, vol. 4, no. 1, pp. 61 – 65, 2013.

[12] Manish Kumar, D.C. Mishra, R.K. Sharma, "A first approach on an RGB image encryption", http://web.iitd.ac.in/~rksharma/Research%20Publications/Journal/Deep_OLE.pdf

[13] M.AL-Laham, Mohamad. "Encryption-Decryption RGB Color Image Using Matrix Multiplication." International Journal of Computer Science and Information Technology 7.5 109–119. Web.

[14] Rajesh N, Sushmashree S, Varshini V, Bhavani N B, Pradeep D, "Color Code Based Authentication And Encryption", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015.

[15] M.Karolin, Dr.T.Meyyapan, "RGB Based Secret Sharing Scheme in Color Visual Cryptography". International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, July 2015.

[16] Sanjay Kumar Pal, Samar Sen Sarma, Graph Coloring Approach for Hiding of Information, Procedia Technology, Volume 4, 2012, Pages 272-277, ISSN 2212-0173, https://doi.org/10.1016/j.protcy.2012.05.042. (http://www.sciencedirect.com/science/article/pii/S2212017312003210)

[17] Singh, Ajit; Nandal, Aarti (May 2013). "Neural Cryptography for Secret Key Exchange and Encryption with AES" (PDF). International Journal of Advanced Research in Computer Science and Software Engineering. 3 (5): 376–381. ISSN 2277-128X.

[18] Sanjay Kumar Pal, Suman De,"An Encryption Technique based upon Encoded Multiplier with Controlled Generation of Random Numbers", IJCNIS, vol.7, no.10, pp.50-57, 2015.DOI: 10.5815/ijcnis.2015.10.06

[19] Sanjay Kr. Pal, Nupur Chakraborty,"Application of Cosmos's law of Merge and Split for Data Encryption", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.5, pp.11-20, 2017.DOI: 10.5815/ijcnis.2017.05.02

[20] Ahmad Gaeini, "Comparing Some Pseudo-Random Number Generators and Cryptography Algorithms Using a General Evaluation Pattern", I.J. Information Technology and Computer Science, 2016, 9, 25-31 Published Online September 2016 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2016.09.04

[21] Quist-Aphetsi Kester,"Image Encryption based on the RGB PIXEL Transposition and Shuffling", IJCNIS, vol.5, no.7, pp.43-50, 2013. DOI: 10.5815/ijcnis.2013.07.05

[22] M.I.Khalil,"Real-Time Encryption / Decryption of Audio Signal", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.2, pp.25-31, 2016.DOI: 10.5815/ijcnis.2016.02.03

[23] Nisar Ahmed, Hafiz Muhammad Shahzad Asif, Gulshan Saleem,"A Benchmark for Performance Evaluation and Security Assessment of Image Encryption Schemes", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.12, pp.18-29, 2016.DOI: 10.5815/ijcnis.2016.12.03

[24] Satyendra Nath Mandal, Subhankar Dutta, Ritam Sarkar,"Block-Based Symmetry Key Visual Cryptography", IJCNIS, vol.4, no.9, pp.10-19, 2012.

Authors' Profiles

Sanjay Kr. Pal is Faculty in the Department of Computer science, NSHM College of Management and Technology, Kolkata. He has an MCA, M.Tech.(IT) and has already presented his Doctoral Public Seminar. He has 24 years of experience shared between 11 years in Industry and 13 years in Teaching. He has a published book on Graph theory, ―Allurement of Some Graph Algorithms and more than 50 research papers in different International and National Journals.

Sumeet Anand is a final year student of Bachelor's in Computer Applications from NSHM College of Management & Technology, Kolkata appearing for his final Semester Examinations. His basic interest includes Artificial Neural Networks, Information Security, and Cloud Computing.

How to cite this paper: Sanjay Kumar Pal, Sumeet Anand,"Cryptography Based on RGB Color Channels using ANNs", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.5, pp.60-69, 2018.DOI: 10.5815/ijcnis.2018.05.07

