ISSN:2320-0790

An International Journal of Advanced Computer Technology

Image Encryption Using Hash Function and Entropy Function

Prof. M. Hanmandlu[1], Gajanand Meena[2], Narendra Vishwakarma[3]

[1,2]Department of Electrical Engineering, IIT, Delhi
[3]Dept. Of Computer Engineering, NMIMS MPSTME, Shirpur

Abstract: In this paper, two way security is purposed; in which the first part of the algorithm presents the image encryption based on MD5 Hash function and in the second part we use the entropy function. The distinct characteristics of the algorithm are : high security, high sensitivity and high speed that can be applied for encryption of gray-level and color images. The hash function performs the partial encryption. The Entropy function generates a key that helps jumble the information. Both security and performance aspects of the proposed algorithm are analyzed and satisfactory results are obtained.

Keywords: Image Encryption, Image Decryption, Cryptography, Entropy, Information measure

I. INTRODUCTION

Development of computer networks permits us an easy access to digital images. Therefore much research has been done on curtailing the access to digital images by illegal users. Usually, encryption is one of ensuring high security and is used to protect the data during the transmission, for example the data is being transferred via networks (e.g. the Internet, e- commerce, cell phones, and wireless intercom systems).

The most popular encryption technique is the Data Encryption Standard (DES), which is a type of symmetric key algorithm. This is a cipher that operates on 64-bit blocks of data, using a 56-bit key. However to meet the high security needs, a more powerful encryption algorithm is required. Hence Advanced Encryption Standard (AES) is proposed by NIST, which involves a 128-bit block size and supports 128, 192, and 256 bit keys. Cryptosystem must stress upon the secrecy of the key rather than the algorithm. The cipher text must appear to be random and difficult to comprehend. It must also be able to resist the attacks of cryptanalysts and hackers.

Encryption deals with generating codes for a particular set of data, which can be transmitted. In this process, one may make use of Shannon entropy function [2], which describes the properties of long sequences of symbols, and applies the results to a number of basic problems in coding theory and data transmission. Later the definition of entropy was extended to the field of information theory. The probabilistic interpretation of entropy in the context of statistical mechanics is attributed to Boltzmann. Here we have used the recently proposed information theoretic entropy function [3] to replace the Shannon entropy function.

In this paper, we begin with the Hash Function (MD5) algorithm and develop the Hash key and apply the OR operation with image and get the partially encrypted image. We then apply the new entropy function to jumble up the encrypted data. As part of the entropy function, we utilize the ratios of the moments to provide the parameters of the function. Earlier cryptographic systems use a common key at both the encoding and decoding end, and the security of the key is mandatory at both sender and receiver ends. Now a gray scale image is a matrix of size m x n. Here the pixel values range from 0-255. So the sequence is not binary any more. In our approach, simply knowing the key is not enough for a hacker because the arrangement of 8-bit plane is also necessary for decryption. The procedure of transmission of encrypted data and keys is shown in Fig.1.

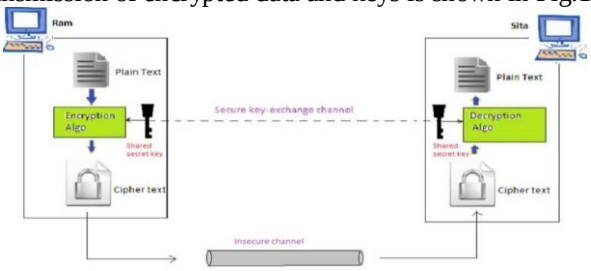

Figure 1: Procedure of transmission of information in an insecure network

The paper is organized as follows: Section 2 discusses the Hash function. The proposed Hanman-Anirban entropy function is described in Section 3. Demonstration of the procedure for the image encoding and decoding is relegated to Section 4. The statistical analysis of the encrypted and decrypted image data for robustness of the technique is presented in Section 5. The conclusions are given in Section 7.

II. HASH FUNCTIONS

A hash function (h) converts an input from an arbitrary large domain (x) into an output in a fixed small range (the hash value y=h(x), often a subset of integers).
Hash function should satisfy three features:

1) Works one way.
2) Output does not reveal information about the input.
3) Hard to find collisions.

MD5 is a cryptography algorithm that gives a 128-bit long output for any arbitrary length input information. The digest also called \Hash" or finger print" of MD5. It was designed in 1991 by Ronald-Rivest.

2.1 The Steps of MD 5

Step 1: Divide the input information into blocks of (512 bit each)
Step 2: Add 64 bits at the last block to get the information about the input length.
Step 3: If the last block's length is less then 512, then add some extra bit's at the end of it. Step 4: Divide each block into 16 words (32 bit each).

2.2 MD 5 Functions

We have 5 functions in MD5. These are elaborated now:

1) Buffer : This is made up of 4 words (32 bit each).
   Word A: 01 23 45 67
   Word B: 89 ab cd ef
   Word C: fe dc ba 98
   Word D: 76 54 32 10

2) Table : It has 64 elements. The Element number i is indicated by ki. These elements are computed using the "sin" function.
   $Ki = abs(Sin(i + 1))* 2^{32}$
3) Auxiliary functions : The logical operators used in the construction of 4 auxiliary functions are (AND, OR, NOT and XOR). The four auxiliary functions are as follows:

F(X,Y,Z) = (X AND Y) OR (NOT(X) AND Z)
G(X,Y,Z) = (X AND Z) OR (Y AND NOT(Z))
H(X,Y,Z) = (X XOR Y XOR Z)
I(X,Y,Z) = (Y XOR (X OR NOT(Z))

4) Processing the buffers : The four buffers (A, B, C and D) are combined with the input words via four auxiliary functions (F, G, H and I). This involves four rounds each having 16 basic operations. Fig. 2 depicts the function F applied to the four buffers (A, B, C and D) with message m_i and constant k_i. The " <<< s" indicates the left shift.

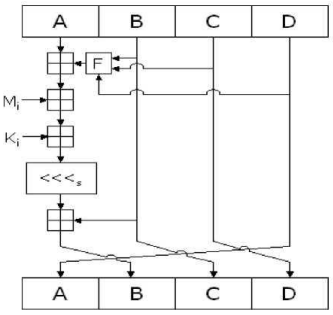

Figure 2: MD5 operation

5) The Output : After performing all rounds, the buffer's (A, B, C and D) would have the Hash key of the initial input

Ex:

1. MD5("The quick brown fox jumps over the lazy dog") =
2. MD5("The quick brown fox jumps over the lazy dog.") = e4d909c290d0fb1ca068ffaddf22cbd0
3. MD5("")
4. MD5("lena.jpg")

1A7590A6F40F7ADC75A9B99BC3E6B6

Figure 3: Hash key example

III. HANMAN-ANIRBAN ENTROPY FUNCTION

Most of the entropy functions are not suitable for representing the information in fuzzy set. These include Shannon, Renyi and Pal and Pal entropy functions. These entropy functions are generalized by a introducing a polynomial in the exponential gain function in [3] resulting in Hanman-Anirban entropy function. The Pal and Pal entropy function is a special case of the Hanman-Anirban entropy function [3], which is given by:

$$H = X<k> pk\ e^{(ap}k^3 + bpk^2 + cpk + d)$$
$$\text{Subject to the condition } £<_k> p_k = 1$$

The values of a, b, c and d are constant but for our algorithm we generate the values of a, b, c and d from the message itself. Now we have to normalize the entropy HN to get the key so that we can jumble the information sequence.

Ex: For Binary Digits :

Let the binary sequence be '11100101001' and we have to jumble it by using Hanman- Anirban entropy function. So first we have to calculate the probability of 0 (p0) and 1 (p_1). Let $p_0 = 0.45$ and $p_1 = 0.55$ now we substitute the values of p_0 and p_1 in Eq. (2) and calculate the final entropy. After normalizing we get a number that is our key for jumbling, Let our key be 4 so the final jumbled sequence (encoded) is '01010011110' . Now for decoding again we calculate the probability of 0 (p0) and 1 (p1) in the jumbled sequence and substitute the values of probabilities in Eq. (2) and get the key back. After applying the reverse jumbling we get the original message as ('11100101001'). Each row of a bit plane (shown in Fig. 5) behaves like a binary sequence. The same procedure is applied on each row of every bit plane (total 24 planes for color image).

IV. PROCEDURE OF ENCRYPTION AND DECRYPTION

After applying MD5 algorithm we get 128- bit long key (shown in Fig. 3). Now we organize the Hash key into 6x6 matrix (K) because the value of hash key varies from 0-15 so we can represent it using 4 bits (0000-1111). By using K we can generate 4- bit matrix (k1, k2, k3 and k4) shown in Fig. 4.

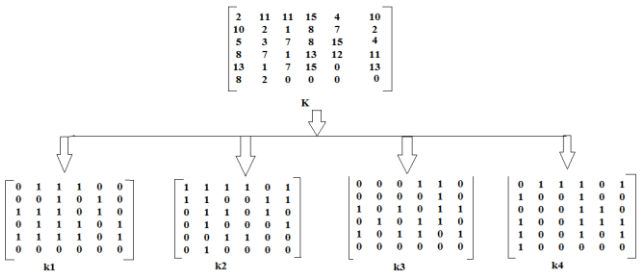

Figure 4: Hash Matrix

In an color image there are three planes (RGB), each plane having 8 bit planes (See Fig.5) because the value of each pixel varies from 0 to 255.

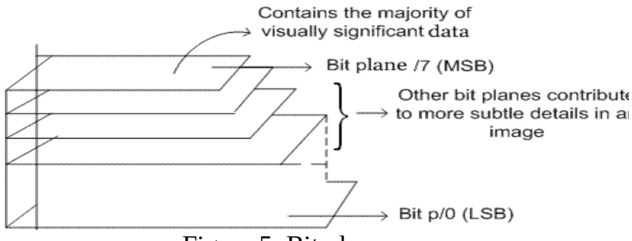

Figure 5: Bit planes

Now we divide each bit plane into 6x6 blocks (c0, c1, c3 … and c(m*n/36-1)).

I(initial image) = (C0, C1, C3 … and C(m*n/36-1))…...(3)
For each block we perform the XOR operation with key (k1, k2, k3 and k4). This leads to:

C0 = XOR (k4, XOR (k3, XOR (k2, XOR(C0 ,k1))))...(4)
After performing that operation for each bit plane (total 24) we get the encrypted image shown in Fig. 6

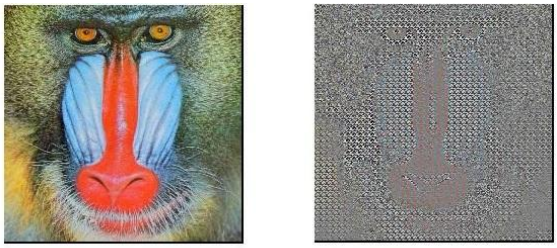

Figure 6: Partially encrypted image

4.2 Part 2: First we apply the entropy function to each row of every binary plane and get the different jumble number for each row of every bit plane. Suppose that the jumbling number is k, and then we circular shift all the numbers from 1 to k, then from 2 to k+1, then from 3 to k+2 and so on till we reach n, where n is the length of the sequence. This whole process can further be iterated by p times.

At the decryption end, the data is decrypted and the data bits are extracted from the sequence. The same jumbled value is evaluated and un-jumbling is performed on the data sequence. We now extend the concept to a binary image of size m × n. Each of these 'm.n' entries either have a value '0' or '1'. This simply implies that we have m (rows) different sequences of length n (columns). The idea here is to apply the above algorithm to the first row, second row, third row and so on. Now for each row we get a transformed (jumbled) sequence. The overall jumbled matrix is the encrypted image data. It is also observed that the jumbled values of each row of binary data vary each time use the entropy function that distorts the image accordingly. The rows and columns jumbling of image are expressed as:

$$X'row,k = Xrow, j+\alpha r \pmod n \quad \text{……..(5)}$$

$$Xp,column = Xi+\beta s \pmod m, \text{ colum…….(6)}$$

Where αr and βs are integer parts of entropy corresponding to rth row and sth column, respectively and X is the original position of pixel and X′ is the new pixel position after jumbling.

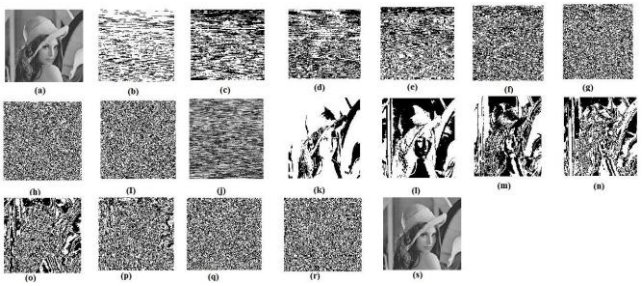

Figure 7: Encryption and decryption results: (a) Original image ; (b) Encrypted image at 1st level; (c) Encrypted image at 2nd level; (d) Encrypted image at 3rd level; (e) Encrypted image at 4th level; (f) Encrypted image at 5th level;(g) Encrypted image at 6th level; (h) Encrypted image at 7th level; (i)Encrypted image at 8th level; (j) Completely encrypted image; (k) Decrypted image at 1st level; (l) Decrypted image at 2nd level; (m) Decrypted image at 3rd level; (n) Decrypted image at 4th level; (o) Decrypted image at 5th level; (p) Decrypted image at 6th level; (q) Decrypted image at 7th level;(r) Decrypted image at 8th level; (s) Completely Decrypted image

V. Statistical Analysis

5.1 Histogram analysis

Among various benefits of the histogram is that, it shows the shape of the distribution of data. For color image the histogram is calculated for red, green and blue planes. Completely encrypted image histogram is given in Fig. 8(b) which is different from the original (shown in Fig. 8(a)), thus indicating the encrypted is secure.

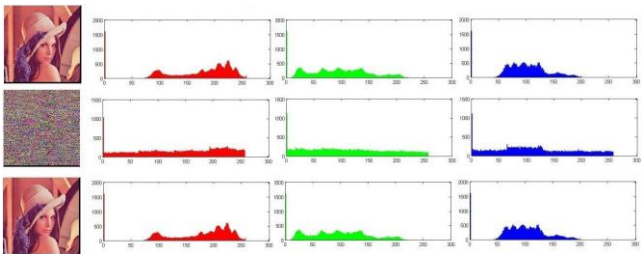

Figure 8: (a) Histogram of an original image (b) Histogram of an encrypted image
(c) Histogram of completely decrypted image

5.2 The Error Measures

We use here : Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Correlation analysis. The Mean Square Error (MSE) between the reconstructed image data and original image data is computed from:

MSE (I, I') = 1/m*n £My=1 £Nx=1 [| I (x,y) - I' (x,y) |]2 ……..(7)

where x and y are the pixels of an image

The Peak Signal Noise Ratio (PSNR) is defined as

PSNR (I, I') = 20.log10 (MAX) - 10.log10 (MSE)……..(8)

Here, MAXI is the maximum possible pixel value of the image. More generally, when samples are represented using linear PCM with B bits per sample, MAX_I is 2^B-1.

Table 1: PSNR and MSE analysis between original, encrypted image

S.no	MSE	PSNR
1.	2.347* 104	91.83

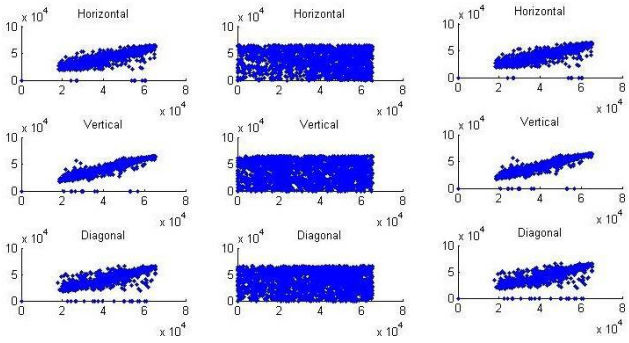

Figure 9: Correlation of an original, encrypted image and decrypted image

VI. Results Of Implementation

Case Study 1 (Peppers Image):

In Fig. 10 the histograms of the encrypted image and original image are nearly uniform and significantly different. Hence Fig. 10 also not provided any clue to employ any statistical analysis attack on the encrypted image.

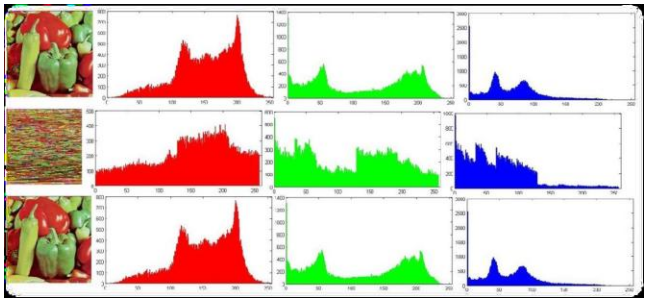

Figure 10: (a) Histogram of an original image (b) Histogram of an encrypted image (c)Histogram of completely decrypted image

Fig. 11 shows the results of correlation analysis. Fig. 11 show significant reduction in relevance of adjacent elements in horizontal, vertical and diagonal elements in image before and after encryption.

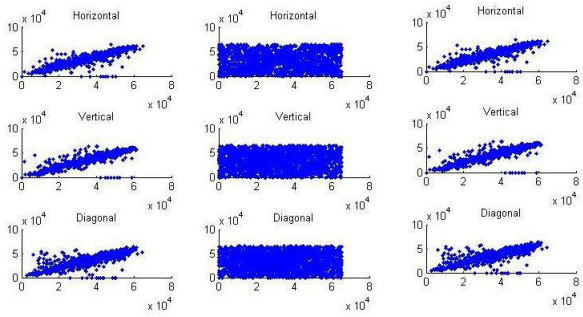

Figure 11: Correlation of an original, encrypted image and decrypted image

Table 2: PSNR and MSE analysis between original, encrypted image

S.no.	MSE	PSNR
1.	2.3251 * 104	91.78

Case Study 2 (baboon Image):

In Fig. 12 the histograms of the encrypted image and original image are nearly uniform and significantly different. Hence Fig. 12 also not provided any clue to employ any statistical analysis attack on the encrypted image.

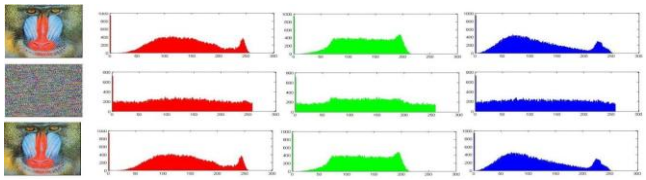

Figure 12: (a) Histogram of an original image (b) Histogram of an encrypted image (c) Histogram of completely decrypted image

Fig. 13 shows the results of correlation analysis. Fig. 13 show significant reduction in relevance of adjacent elements in horizontal, vertical and diagonal elements in image before and after encryption.

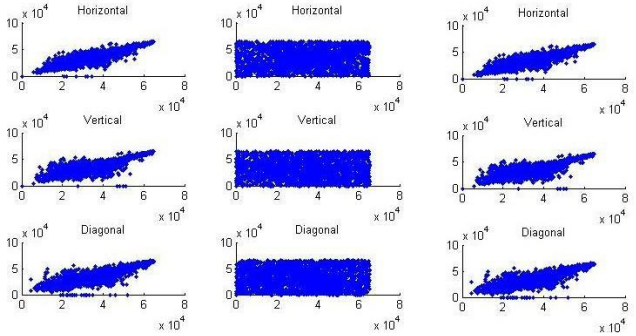

Figure 13: Correlation of an original, encrypted image and decrypted image

Table 3: PSNR and MSE analysis between original, encrypted image

S.no.	MSE	PSNR
1.	2.9176 * 10$_4$	92.78

VII. CONCLUSION

The proposed algorithm gives a very secure code. Furthermore, using the new entropy [3] function, we have jumbled up the digits and the image results. The parameters of this function can be either fixed at by the user's discretion or computed from the message bits thus making easier the job of a user. This key generation becomes more complex for the attacker thus increasing the security of the algorithm. However, the efficiency of the algorithm shows greater dependence on the data sequence. For some data, the algorithm may behave as the conventional skew map. In other cases, it may introduce rigorous jumbling and may produce a seemingly random.

REFERENCES

[1] K H Rosen, Elementary Number Theory and Its Applications, Pearson,New York 1984.

[2] Shannon, A Mathematical Theory of Communication, Bell System Technical Journal, 27 (1948) 379-423.

[3] Madasu Hanmandlu, Anirban Das, Content-based Image Retrival by Information theoretic Measure, Defence Science Journal, 61 (2011) 415-430.

[4] J J Rissanen and G G Langdon, Arithmetic Coding, IBM Journal of Research and Development, 23 (1979) 146-162.

[5] Gwo-Ruey Yu, Dept. of Electr. Eng., I-Shou, "Secure communication using chaotic Synchronization and international data encryption algo-rithm", Proceedings of American Control Conference, 5 (2004) 4319-4324.

[6] Nithin Nagaraj, Prabhakar G Vaidya, Kishor G Bhat, Joint Entropy Coding and Encryption using Robust Chaos, Front for the arXiv,nlin.CD/0608051, 2006.

[7] Y. L Bel'skii, and A. S. Dimitriev, Information transmission determin-istic chaos Radiotekh, Electron, 38 (1993) 1310-1215.

[8] Ajeesh P. Kurian, Sadasivan Puthusserypady, Secure Digital Communi-cation using Chaotic Symbolic Dynamics, Turk J Electrical Engineering,14 (2006).

[9] Ritika Goel, Madasu Hanmandlu, Cryptography Using Information Theoretic Measure, National Conference on Current Trends in IT, Kru-jpal University, (2007) 247-252.

[10] Ch Samson, V U K Sastry, An RGB Image Encryption Supported by Wavelet-based Lossless Compression. IJACSA, 3 (2012) 36-41.

[11] H T Panduranga, S K Naveen Kumar, Advanced Partial Image Encryp-tion using Two- Stage Hill

Cipher Technique. International Journal of Computer Applications, 60 (2012) 14-19.

[12] V. Gopalakrishnan, T. Purusothaman, Enhancing security through compressing, randomized encryption and authentication. ICGST-CNIR Journal, 5 (2006).

[13] Benyamin Norouzi,Seyed Mohammad Seyedzadeh, Sattar Mirza-kuchaki,Mohammad Reza Mosavi, A novel image encryption based on Hash function with only two-round diffusion process, Springer Verlag Berlin Heidelberg, Multimedia System 20(1) (2014) 45-64 .