IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 3, May 2014.
www.ijiset.com
ISSN 2348 – 7968

Image Encryption Using Block-Based Transformation Algorithm

Brijesh Kumar Singh[1], Nidhi Sharma[2], Neha Singla[3], Nikita Sharma[4], Neetu Choudhary[5]

CSE Department, JECRC UDML College of Engineering, Jaipur, India

Abstract—Encryption is used to securely transmit data in open networks. Most of the available encryption algorithms are mainly used for textual data and may not be suitable for multimedia data such as images. In this paper, we introduce a block-based transformation algorithm based on the combination of image transformation and a well known encryption and decryption algorithm called Blowfish. The original image was divided into blocks, which were rearranged into a transformed image using a transformation algorithm, and then the transformed image was encrypted using the Blowfish algorithm. The results showed that the correlation between image elements was significantly decreased by using the proposed technique. The results alsoshow that increasing the number of blocks by using smallerblock sizes resulted in a lower correlation and higher entropy.

Index Terms—Image correlation, Image encryption, Imageentropy, Permutation.

I. INTRODUCTION

Technology has changed the world dramatically. Last few decades have witnessed, some great events, especially in the field of digital communication[1]. Internet has shrunk the whole world. Now it's very easy to access the information as well as data anywhere. To prevent the data from unauthorized access, encryption technique is widely used. Encryption method try to convert original data to another data that is hard to understand and only the intended recipient will be able to understand it. Most of the available encryption algorithms are mainly used for textual data. But due to large data size and real timeconstrains, algorithms that are good for textual data may notbe suitable for multimedia data [2]-[4].So, we introduce a block-based transformation algorithm which tries to reduce the correlation between the neighbor pixels value to ensure the security of digital images.

In most of the natural images, the values of theneighboring pixels are strongly correlated(i.e. the value ofany given pixel can be reasonably predicted from the valuesof its neighbors) [5]-[7]. In order to dissipate the highcorrelation among pixels and increase the entropy value, wepropose a transformation algorithm that divides the imageinto blocks and then shuffles their positions before it passesthem to the Blowfish encryption algorithm. By using thecorrelation and entropy as a measure of security, this processresults in a lower correlation and a higher entropy value whencompared to using the Blowfish algorithm alone, and thusimproving the security level of the encrypted images. Thereare two main keys to increase the entropy; the variable secretkey of the transformation process and the variable secret keyof the Blowfish algorithm.

II. THE PROPOSED TECHNIQUE

A. Description of the Transformation Algorithm

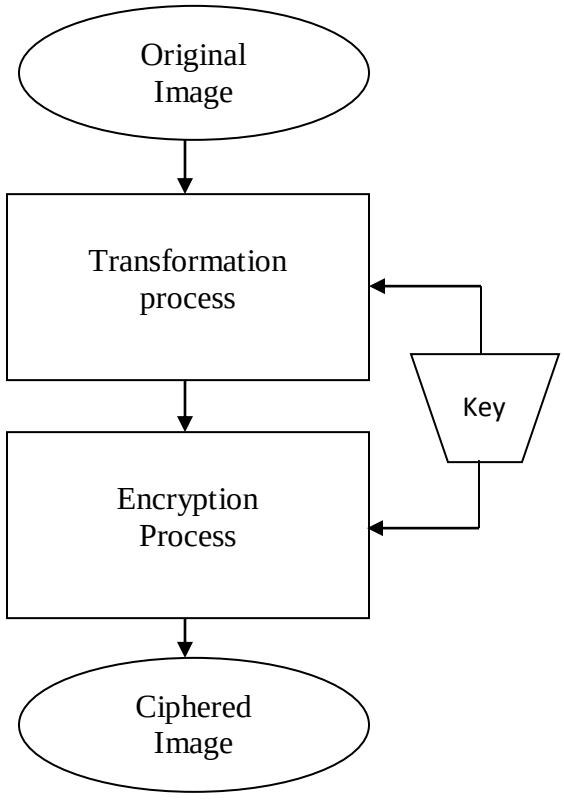
The transformation technique works as follows: theoriginal image is divided into a random number of blocksthat are then shuffled within the image. The generated (ortransformed) image is then fed to the Blowfish encryptionalgorithm. The main idea that is used in this transformation algorithm is that an image can be viewed as anarrangement of blocks. The intelligible information presentin an image is due to the correlation among the imageelements in a given arrangement. This perceivableinformation can bereduced by decreasing the correlationamong the image elements using certain transformation techniques.

The secret key of this approach is used to determine theseed. The seed plays a main role in

ISSN 2348 – 7968

building thetransformation table, which is then used to generate thetransformed image with different random number of blocksizes. The transformation process refers to the operation ofdividing and replacing an arrangement of the original image.

The image can be decomposed into blocks; each onecontains a specific number of pixels. The blocks aretransformed into new locations. For better transformationthe block size should be small, because fewer pixels keeptheir neighbors. In this case, the correlation will bedecreased and thus it becomes difficult to predict the valueof any given pixel from the values of its neighbors. At thereceiver side, the original image can be obtained by theinverse transformation of the blocks. A general block
diagram of the transformation method is shown below.



The transformation algorithm is presented below. It generates a transformation table that will be used then to build a newly transformed image.

ALGORITHM
CREATE_TRANSFORMATION_TABLE

1: Select Image(Bmp format) to be encryption from data store
2: Insert key of 256 bits
3: Calculate Image Pixels Value
Horizontal Value of Pixel = PixelWidth/10
Vertical Value of Pixel = PixelHeight/10
4: Select a Random Function to Calculate Final value forHorizontal and Vertical Pixels

- HorizontalPixel= Select Random Value betweenHorizontal Value of Pixel and PixelWidth
- VerticalPixel= Select Random Value between VerticalValue of Pixel and PixelHeight

5: Select a Variable No-Of-Pixel to store Multiple Value ofHorizontalPixel and VerticalPixel
No-Of-Pixel=HorizontalPixel*VerticalPixel
6: Using Hash Function (Here I am using SHA-1) I amgenerating a Seed Value. This SHA-1 will apply on 256 bitsSelected Key
Seed = SHA-1(Above Selected KEY)
7: Divide Seed into two Part equally Seed-1 and Seed-2

- Seed-1 = First Half of Seed
- Seed-2=Second Half of Seed

8: If Seed-1 is Greater Then Seed-2 Then We Will Selectanother Variable SeedValue and assign any numeric valuebetween 0 to 4 (Randomly Chooseable) Otherwise Value of SeedValue Variable vary between 5 to 9 (Randomlychooseable).
9: If Variable SeedValue is Equal Between 0 to 4 thencalculate new seed value (Here we are working on ASCIIvalue of seed).

- Seed = Seed + (Seed-1 Mod 2) + 1

Otherwise

- Seed = Seed + (Seed-2 Mod 2) + 1

10: Repeat Process 8 to 9 till No-Of-Pixel/2
11: Final Output of Step 10 will represent
END
CREATE_TRANSFORMATION_TABLE

ISSN 2348 – 7968

ALGORITHM
PERFORM_TRANSFORMATION

1: For I = 0 to No-Of-Pixel -1
1.1: Get the new location of block I from the transformation table
1.2: Set block I in its new location
END PERFORM_TRANSFORMATION
Output: Transformed Image.

A. Description of the Blowfish Algorithm

Blowfishis a 64-bit symmetric block cipher that can be effectively used for encryption andsafeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making itideal for securing data.Blowfish Algorithmis a *Feistel Network*,iterating a simple encryption function 16times.

The Blowfish algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion consists of generating the initial contents of one array(the P array), namely, eighteen 32-bit sub keys, and four arrays(the S boxes),each of size 256 by 32 bits,froma key of at most 448 bits .

Data encryption occurs via a 16-round Feistel network. Each round consists of a keydependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

The Blowfish algorithm is presented below. It generates an encrypted image.

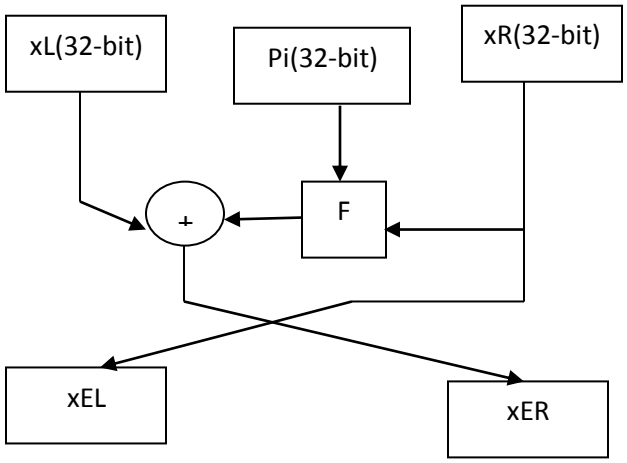
Algorithm

Note: Pi used in this algorithm is the subkey generated from 448-bit key.F function is basically a process of applying substitution and the permutation process.

1: Take the transformed image generated in the transformation process.
2: Divide the image into blocks containing 64 bits in each block.
3:For each block,do

- Divide block into two 32-bit halves: xL, xR.
- Then, for i = 1 to 16:

xEL = xR
xER = [F(xR) and F(Pi)] XOR xL
Swap xL and xR



After the sixteenth round, swap xEL and xER again to undo the last swap.
Then, xER = xER XOR P17 and xEL = xEL XOR P18.
Finally, recombine xEL and xER to get the encrypted block
4:Repeat the step 3 for each block and finally we get encrypted image.

III. EXPERIMENTS

Results are evaluated on the following parameters.

(1) **NORMALISED CROSS CORELATION-** Normalized Cross Correlation (NCC) has been commonly used as a metric evaluate the degree of similarity or dissimilarity between two compared images. The main advantage of the normalized cross correlation over the cross correlation is that it is less sensitive to linear

ISSN 2348 – 7968

changes in the amplitude of illumination in the two compared images.

$$C_{AB} = \frac{\frac{1}{r*c}\sum_{i}^{r}\sum_{j}^{c}(A_{i,j}-\bar{A})(B_{i,j}-\bar{B})}{\sqrt{\frac{1}{r*c}\sum_{i}^{r}\sum_{j}^{c}(A_{i,j}-\bar{A})^2 \frac{1}{r*c}\sum_{i}^{r}\sum_{j}^{c}(B_{i,j}-\bar{B})^2}}$$

Ai,j and Bi,j are the pixels in the ith row and jth column of A and B respectively and r, c represent the no. of rows and columns in the image

(4)ENTROPY-

In information theory, entropy is a measure of the uncertainty in a random variable. In this context, the term usually refers to the entropy, which quantifies the expected value of the information contained in a message. Entropy is typically measured in bits, nats or bans. Shannon entropy is the average unpredictability in a random variable, which is equivalent to its information content.

$$h = -\Sigma(p_i log_2 p_i)$$

Where pi is the frequency of intensity level i in the image. The maximum h an 8–bit image can attain is 8.

In order to evaluate the impact of thenumber of blocks on the correlation and entropy, threedifferent cases were tested. The number of blocks and theblock sizes for each case are shown in Table I.

Table I Different Cases to Test the Impact of the Number of Blocks on the Correlation and Entropy

Case Number	Number of blocks	Block size
1	30×30	10 pixels × 10 pixels
2	60×60	5 pixels × 5 pixels
3	100×100	3 pixels × 3 pixels

Each case produces three output images; (a) a ciphered image using the Blowfish algorithm, (b) a transformed imageusing the proposed algorithm, and (c) a ciphered image usingthe proposed algorithm followed by the Blowfish algorithm.For the rest of this paper, we use image A, image B, image C,and image D to denote the original image, the ciphered imageusing the Blowfish algorithm, the transformed image, and theciphered image using the proposed algorithm followed by theBlowfish algorithm respectively.

Case 1: The image is decomposed into 10 pixels × 10 pixels blocks. Fig.1. shows the resulted images.

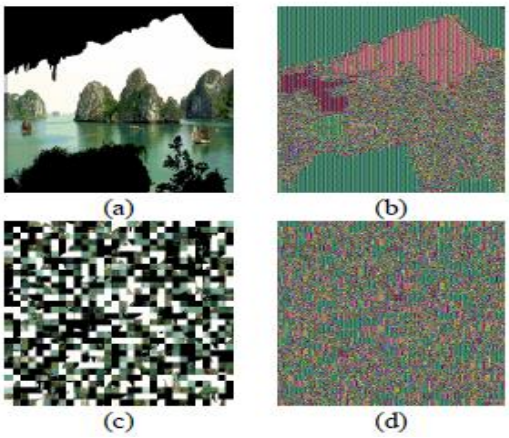


Fig.1. Results of encryption by using 10 pixels × 10 pixelsblocks. (a) Original image. (b) Encrypted image usingBlowfish. (c) Transformed image. (d) Encrypted image using transformation followed by the Blowfish algorithm.

The correlation and entropy results of this case aresummarized in Table I.

Table I: Results of Correlation and Entropy values of Case 1.

Measurement		A	B	C	D
Correlation	Horizontal	0.933	0.035	0.843	0.034
	Vertical	0.936	0.107	0.844	0.038
	Diagonal	0.919	0.032	0.748	0.013
	Opposite Diagonal	0.916	0.034	0.745	0.009
	Average	0.926	0.052	0.795	0.023
Entropy value		2.431	4.799	2.431	5.231

ISSN 2348 – 7968

Case 2: The image is decomposed into 5 pixels × 5 pixels blocks. Fig.2. shows the resulted images.

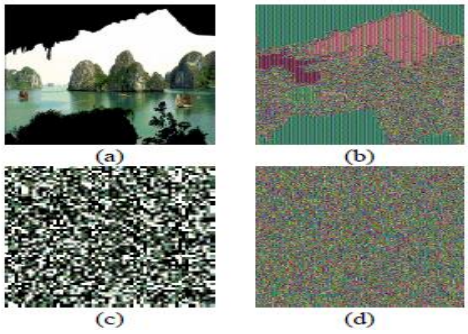


Fig.2. Results of encryption by using 10 pixels × 10 pixelsblocks.
(a) Original image.
(b) Encrypted image usingBlowfish.
(c) Transformed image.
(d) Encrypted image using transformation followed by the Blowfish algorithm.

The correlation and entropy results of this case aresummarized in Table II.

Table II: Results of Correlation and Entropy values of Case 2.

Measurement		A	B	C	D
Correlation	Horizontal	0.9325	0.0346	0.7469	0.0245
	Vertical	0.9362	0.1073	0.7497	0.0067
	Diagonal	0.9186	0.0321	0.5881	0
	Opposite Diagonal	0.9156	0.0337	0.5877	0.0056
	Average	0.9257	0.0519	0.6681	0.0092
Entropy value		2.431	2.4305	4.799	2.4305

Case 3: The image is decomposed into 3 pixels × 3 pixels blocks. Fig.3. shows the resulted images.

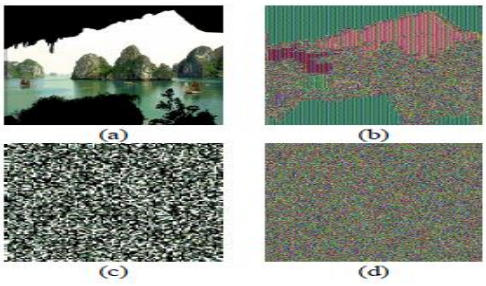


Fig.3. Results of encryption by using 10 pixels × 10 pixelsblocks. (a) Original image. (b) Encrypted image usingBlowfish. (c) Transformed image. (d) Encrypted image using transformation followed by the Blowfish algorithm.
The correlation and entropy results of this case aresummarized in Table III.

Table III: Results of Correlation and Entropy values of Case 3.

Measurement		A	B	C	D
Correlation	Horizontal	0.9325	0.0346	0.6289	0.0129
	Vertical	0.9362	0.1073	0.6265	0.0034
	Diagonal	0.9186	0.0321	0.4145	0.0014
	Opposite Diagonal	0.9156	0.0337	0.4146	0.0049
	Average	0.9257	0.0519	0.5211	0.0056
Entropy value		2.4305	4.799	2.4305	5.5281

IV. RESULTS AND DISCUSSION

The above cases show that using the transformation algorithmfollowed by the Blowfish algorithm resulted in a lowercorrelation and a higher entropy compared to using theBlowfish alone. Dividing the image into a larger number ofblocks made the performance even better. The resultsshowed that the correlation was reduced even further and theentropy was increased as blocks size is decreased.

Table IV summarizes the results of the different cases.

Measurement	Number of blocks	A	B	C	D
Average correlation	30 × 30	0.9257	0.0519	0.7952	0.0234
	60 × 60			0.6681	0.0092
	100 ×100			0.5211	0.0056
Entropy	30 × 30	2.4305	4.799	2.4305	5.2305
	60 × 60				5.4737
	100 ×100				5.5281

ISSN 2348 – 7968

V. CONCLUSION

In this paper a simple and strong method has beenproposed for image security using a combination of block-basedimage transformation and encryption techniques. The cases showed that the correlation was decreased when theproposed algorithm was applied to them before the Blowfishalgorithm. Experimental results of the proposed technique showed that an inverse relationship exists between numberof blocks and correlation, and a direct relationship betweennumber of blocks and entropy. When compared to manycommonly used algorithms, the proposed algorithm resultedin the best performance; the lowest correlation and thehighest entropy.

VI. REFERENCES

[1] W. Lee, T. Chen and C. Chieh Lee, "Improvement of anencryption scheme for binary images," PakistanJournal of Information and Technology. Vol. 2,no. 2, 2003, pp. 191-200. http://www.ansinet.org/

[2] M. V. Droogenbroech, R. Benedett, "Techniques for aselective encryption of uncompressed and compressedimages," In ACIVS'02, Ghent, Belgium. Proceedingsof Advanced Concepts for Intelligent Vision Systems,2002.

[3] S. Changgui, B. Bharat, "An efficient MPEG videoencryption algor i thm, " Proc e edings of thesymposium on reliable distributed systems, IEEEcomputer society Press, 1998, pp. 381-386.

[4] S. Fong, P.B. Ray, and S. Singh, "Improving thelightweight video encryption algorithm," proceedingof iasted international conference, single processing,pattern recognition and application, 2002, pp. 25-28.

[5] S. P. Nana'vati., P. K. panigrahi. "Wavelets:applications toimage compression-I,". joined of the scientific andengineering computing, vol. 9, no. 3, 2004, pp. 4- 10.

[6] c. Ratael, gonzales, e. Richard, and woods, "Digitalimage processing," 2nd ed, Prentice hall, 2002.

[7] AL. Vitali, A. Borneo, M. Fumagalli and R. Rinaldo, "Video over IP using standardcompatible multiple description coding, " Journal of Zhejiang