Image Encryption then Compression System for Optimal Computational Resources

Vijayadarshini .H[1]
PG Student Department of CSE
Godutai Engineering College for
Women Gulbarga

Professor Shivleela Patil[2]
Professor &Head of Department of
CSE Godutai Engineering College
for WomenGulbarga

***Abstract*-The present work proposes an efficient image encryption technique. Image encryption is done using prediction error clustering and random permutation to ensure a high level of security to the image data. By arithmetic coding it ensures effective compression on the encrypted images. Lossless compression is considered so that no loss of information will be tolerated.**

***Index* Terms-Encryption, Decryption, Compression of encrypted image, Decompression.**

I.INTRODUCTION

The advent of internet in the recent days has become very high and the amount of data transfer that takes place through internet has increased. Data being transmitted can be highly confidential and secure in many cases. There are possibilities that this data can be attacked by any attackers or unauthorized users. Hence providing security to data is the most important task.

Usually when a redundant data is transmitted, first compresses the data to remove the redundancy and then encrypt it which may achieve compression efficiency but do not provide security to data but in our proposed method reversing the order first encrypting the data and then compressing it, so as to provide security to the data. In compression then encryption data will be known to the network operator but in encryption then compression data is hided from the network operator to maintain privacy of the data.

In Compression then encryption (CTE), image compression has been performed before the image encryption hence it uses much of computational resources. But by conducting image encryption prior to image compression it can provide high level security to the data and also it provide optimal computational resources, because compression is performed by the channel provider, where channel provider can use incentives and compress the data, typically channel provider has abundant computational resources and channel provider has an overriding enthusiasm for packing all the system activity to amplify the system use it therefore much desirable if compression task is conducted by the channel provider.

II.RELATED WORK

In [1]: portrayed that the LOCO-I (Low complexity lossless Compression for Images) is the calculation at the center of the new ISO/ITU standard for lossless and close lossless compression of persistent tone image, JPEGLS. It is considered as a "low unpredictability projection" of the widespread connection displaying ideal model, coordinating its demonstrating unit to a basic coding unit. By consolidating straightforwardness with the compression capability of connection models, the calculation “appreciates the best of both universes." It is in view of a basic altered setting model, which approaches the ability of the more intricate general procedures for catching high -arrange conditions. The model is tuned for effective execution in conjunction with a more distant family of Golomb –sort codes, which are adaptively picked, and an implanted letters in order.

In [2]proposed setting is based on context adaptive lossless image codec. CALIC gets higher lossless compression of proceeds with tone image. This high coding productivity is proficient with generally low time and space complexities. CALIC encodes and deciphers image in the raster output request with single go through the image. CALIC puts overwhelming accentuation on the image information displaying; a one of a kind element of the CALIC is the utilization of vast number of demonstrating setting to the nonlinear indicator and makes it versatile to the changing source insights. The non-straight indicator adjusts through a blunder criticism system, in this adjustment process, CALIC just gauges the desire of the forecast slip condition on countless as opposed to assessing substantial number of restrictive mistake probabilities.

In [3]He has proposed on compressing of encrypted data allow us to packing of scrambled information permit us to productively pack encoded images and exhibited the consequences of compacting a scrambled parallel images without access to the source measurements at the decoder. We demonstrated how the 2-D source model permits more prominent compression picks up than the 1-D source model. This work actually proposes an expansion to dim scale and other bigger letter set images. A first approach is to split a image up into a progression of bit-planes where every bit-plane speaks to all the bits of equivalent essentialness in the double development of the pixel values. Image structure is ordinarily exceptionally packed in the most critical bit-planes however. Therefore, little compression increase is accessible with this methodology. Precise image models are important to have the capacity to accomplish noteworthy increases when packing encoded information.

In [4] Presented the approach of [3] to the prediction error domain to achieve better lossless compression operations on the encrypted grayscale/color images to the Compression of scrambled information is conceivable by utilizing dispersed source coding. Consider the encryption, trailed by lossless compression of dark scale and shading images. We proposed encryption on the expectation slips rather than specifically applying on the images and utilization circulated source coding for compacting the figure writings. The reenactment results demonstrate that by utilizing the proposed system similar compression picks up, with compression proportions fluctuating from 1.5 to 2.5 can be accomplished not withstanding encryption.

In [5] He has proposed Lossy compression and iterative recreation for encrypted images. This work proposes a novel plan for lossy compression of an encrypted image with flexible compression ratio. A pseudorandom stage is utilized to scramble a unique image, and the scrambled information are effectively packed via tossing the exorbitantly harsh and fine data of coefficients created from orthogonal change. Subsequent to accepting the compacted information, with the guide of spatial connection in common image, a recipient can reproduce the chief substance of the first image by iteratively upgrading the estimations of coefficients. Along these lines, the higher the compression proportion and the smoother the first image, the better the nature of the recreated image.

In [6] proposed Scalable coding of encrypted images, at the beneficiary side got bit plane data serves as the side data to encourage the estimation of image edge data in this manner making the image reproduction more exact. The more bit planes are transmitted, the higher nature of the reproduced image. The trial results demonstrate that our proposed plan accomplishes vastly improved execution than the current lossy compression plan for pixel-quality scrambled images, furthermore accomplishes comparable execution as the best in class lossy compression on the pixel stage based encoded image.

In[7]proposed several methods for lossless compression of encrypted grayscale/color images. The best hypothetical results are acquired by changing color image in an approximated YCbCr area. As to spatial decorrelation, dealing with the forecast lapse gives vastly improved results, however the xor-based calculation may prompt an intriguing speculation of the proposed plan towards lossy compression. It is just expected to delete the lower scrambled bit planes to lessen the bit rate while keeping the nature of the reproduced image adequate (this is impractical when we work with the expectation lapse following such a blunder is processed before part the picture into bit planes).

III.METHODOLOGY

Encryption is the process which is used to increase the safety of information by converting it to another form which is difficult to understand. Its main aim is to make the communication between the users highly secure.

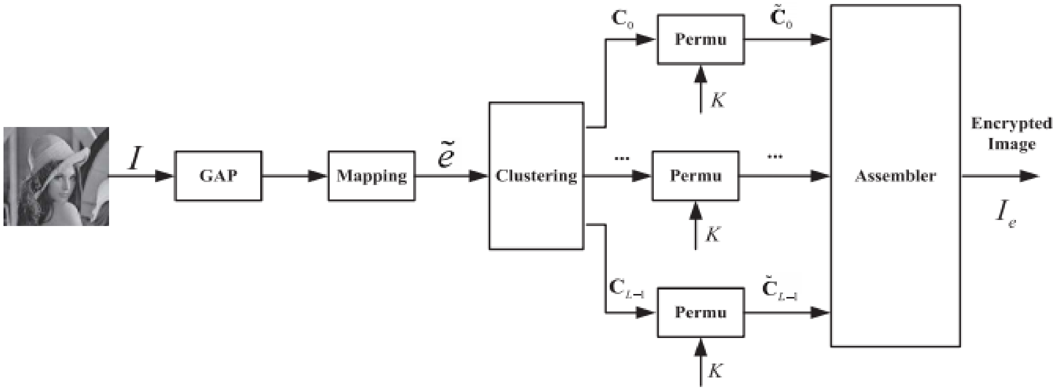


Figure 1: System Architecture of Image Encryption

Encryption process uses keys to encrypt and decrypt the data. The encryption algorithm that is used to encrypt an image works on the prediction errors of the image rather than directly on the original pixel values. The system architecture of the image encryption is depicted in Figure 1.

Step 1: Gradient Adjust Prediction (GAP)

For any image, gradient is the change in the intensity or color in a given direction. Image gradients are useful to extract the information regarding edges from an image. Based on the gradient of the pixel we can decide whether the pixel has a horizontal edge or a vertical edge.

Step2: Mapping

The prediction errors obtained are in the range [-255,255].These values are then mapped into the range [0,255] and the mapped prediction errors are denoted as $\tilde{e}_{i,j}$ as shown in figure 1.The prediction error associated with $I_{i,j}$ can be computed by

$$e_{i,j} = I_{i,j} - \tilde{I}_{i,j} \qquad (1)$$

Step 3: Clustering

Clustering is a process in which an item is grouped into different clusters such that items in a single cluster are more like one another than those in the other group. Hence, a "cluster" is a collection of items that are "similar" among them and "dissimilar" to the items of other clusters.

Step 4: Circular Shifting

Circular shifting is an operation, which will shift the position of the elements of the arrays by the amount of shifts specified. Circular shifting is nothing but an operation of random permutation and secret key k vectors will control the shifting operations of permutated clusters. For example, let us take an array A= [1 2 3 4 5 6]. Now if we want to shift the array by 2 positions circularly then the resulting array which is obtained is A'= [5 6 1 2 3 4]. Here it can be seen that every element in the array has shifted 2 positions to the right. The element 1 has come at the 3rd position, 2 has come at the 4th position and so on. The last element will be coming to the 2nd position as it is a circular shift operation. At the first shift, the last element will be moved to the 1st position and during the 2nd shift it will come to the 2nd position. In this way circular shift operation will be taking place in arrays.

Step 5: Assembler

The assembler will concatenate all the permuted clusters $C_k(\tilde{c}_0.\tilde{c}_1,....,\tilde{c}_{1-1})$ and finally produces encrypted image.

$$I_e = \tilde{c}_0, \tilde{c}_1, ..., \tilde{c}_{L-1}$$

IV.IMPLEMENTATION

Implementation of proposed algorithms and techniques, methods, are described as follows. The following things will show the details about the algorithm and implementation.

4.1 Algorithm for the Image Encryption via prediction error clustering and random permutation

The algorithm for Encryption via prediction error clustering and random Permutation can be summarized as follow:

Step 1: Calculate the mapped prediction errors $\tilde{e}_{i,j}$ of the whole image by using the GAP algorithm.

Step 2: Segregate the prediction errors into L different groups C_k, for $0 \leq k \leq L-1$ and each C_k is formed by concatenating all the prediction errors in a raster scan order.

Step 3: Reshape the prediction errors in each Ck into a 2 D block having 4 columns and [| Ck/4 |] rows where [| Ck/4 |] denotes the number of prediction errors in Ck.

Step 4: Perform two key driven cyclical rotation operations and read out the data in a raster scan order to obtain the permuted cluster.

Step 5: Concatenate all the clusters using an assembler to obtain the final encrypted image. In this way by performing all these steps we obtain the encrypted image.

The exact algorithm used to predict the pixel values is given in the form of a pseudo code as below:

These gradient values are used to predict the values of the pixel. If g_h has a greater value than g_v , it means that horizontal variation is more and N will be picked as the predicted value of X. On the other hand, if g_v is much greater than g_h it implies that there is a larger amount of vertical variation and W will be taken as the predicted value. If the difference values are smaller, then the predicted valued will be the calculated mean of the neighboring pixels.

```
If $g_v - g_h > 80$, P = W; Sharp horizontal edge
else if
$g_v - g_h < -80$, P= N; Sharp vertical edge
else
 P= (W + N) / 2 + (NE –NW) / 4;
If $g_v - g_h > 32$, P = (P + W) / 2; Horizontal edge
else if
$g_v - g_h > 8$, P = (3P + W) / 4; Weak horizontal edge
else
if $g_v - g_h < -32$, P = (P + N) / 2; Vertical edge
else if
$g_v - g_h < -8$, P = (3P + N) / 4; Weak vertical edge
```

4.1.1 Prediction Error Calculation

Using the GAP algorithm, for every pixel $I_{i,j}$ we can obtain the predicted value $\tilde{I}_{i,j}$. From the predicted values that are obtained prediction errors are calculated using the formula in Equation.

$$e_{i,j} = I_{i,j} = \tilde{I}_{i,j}$$

Here $e_{i,j}$ are the prediction error values obtained for each of the pixel location.

4.1.2 Mapping

The prediction errors obtained are in the range [-255, 255]. These values are then mapped into the range [0,255] and the mapped prediction errors are denoted as $\tilde{e}_{i,j}$. The prediction errors are not considered as a whole to perform further operations on them rather they are divided into different clusters. The clustering operation is explained in the following sections.

4.1.3 Cluster Formation

Clustering operations on the prediction errors that are obtained is performed. The prediction errors obtained will be segregated into various clusters and further operations will be performed on the clusters that will be obtained. The selection of number of clusters required should be able to make a balance between the security and the encryption complexity of the system. Larger number of clusters helps in providing a higher level of security to the system. However, it incurs higher complexity of encryption. Hence, the number of clusters required should be chosen appropriately. The steps that are used to perform clustering on the prediction errors can be explained as below:

1. The prediction errors obtained are arranged into a vector of single column.
2. Euclidean distance between each pair of the pixel is measured and stored which is the similarity measure that is used to divide the prediction errors into different groups. The Euclidean distance gives the length of the line segment connecting two points. For a pair of pixels a(x, y) and b(x_1, y_1) Euclidean distance is given by equation.

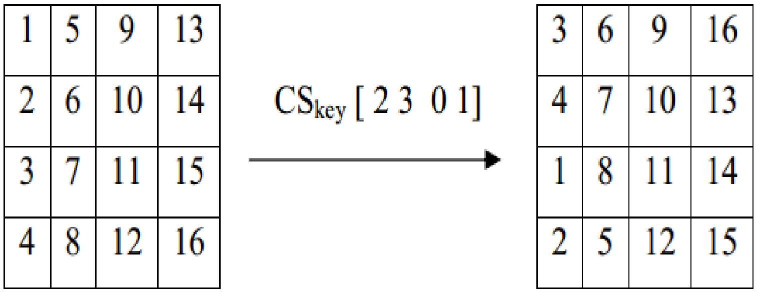
$$E(a,b)^2 = (x-x_1)^2 + (y-y_1)^2$$

3. Number of clusters that are required is specified. Here take the number of clusters to be eight. Based on the number of clusters required prediction errors will be placed into that cluster.

By performing the above clustering operations, we have grouped the prediction errors into different clusters. After the prediction errors are grouped into clusters they are grouped into a 2D block having four columns and four rows denotes the number of prediction errors in the cluster $C_{k.}$ The value of k can be 0, 1 2... and it tells the cluster number. Circular shift operations are then performed on these 2D blocks of data.

4.1.4 Circular Shifts

Circular shift operations will be performed on the 2D blocks of prediction errors that were obtained. To perform the circular shift operation the amount of places that has to be shifted has to be specified. The keys specify this. In encryption of the data here there are 2 keys specified known as column shift key $CS_{key}$and row shift key RS_{key} to perform operations first on the columns of the 2D block and then on the rows of the 2D block.For example, lets us consider a 2D block with a column shift key given by CS_{key}= [2 3 0 1]. The column shift operation on this block of data can be shown in Figure 4.1

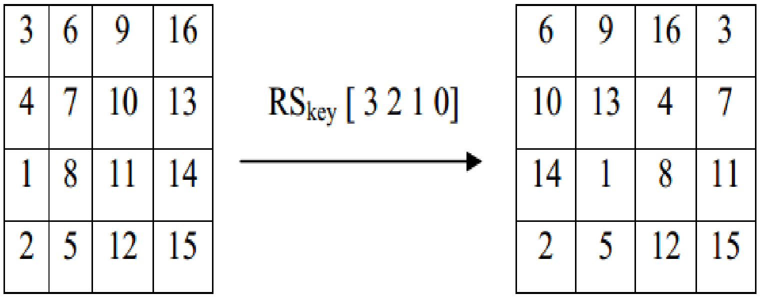


1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

3	6	9	16
4	7	10	13
1	8	11	14
2	5	12	15

Figure 4.1: Column shift on 2D block

Here the shift will be performed on each of the pixels and the location of the pixels will change. There will be no change in the intensity value of the pixels. Only their position will be changed. The key given here will specify the number of positions that have to be shifted in that particular column. Based on this the cyclic shift will take place. The key given [2 3 0 1] will specify that there should be 2 position shift in 1st column, 3 positions shift in 2nd column and so on. After the column shift operation is performed, a row shift operation is performed on the 2D block, which is shifted by column. The row shift operation on this block of data can be as shown in Figure 4.2 below.



3	6	9	16
4	7	10	13
1	8	11	14
2	5	12	15

6	9	16	3
10	13	4	7
14	1	8	11
2	5	12	15

Figure 4.2 Row shift on 2D block

The row shift and the column shift operation will be performed on the 2D block of data as described above. This operation is performed on each of the 2D block generated. The key will be generated randomly and the key is different for every 2D block. Hence, this provides a higher level of security to the system. Then finally all the permuted clusters are concatenated together and a final encrypted image is Obtained, the encrypted image will then be compressed using arithmetic coding.

4.2Lossless Image Compression via AC

After encryption Step involved for compression are as follows:

- De-Assemble to clusters
- Arithmetic coding
- Compressed bit stream

The encrypted image is then compressed using arithmetic coding procedure which reduces the amount of bits in the encrypted image. By performing compression of the information the measure of information that can be transmitted can be expanded to a higher sum, which is generated into bit streams More the compression value the high number of bits is reduced and compression efficiency is high.

4.3Decompression and decryption

The Decompression and Decryption system can be summarized as follow:

- Compression Bit stream
- De-Assemble to clusters
- Arithmetic De-coding
- Perform Reverse Row wise cyclic shifts and reverse column wise cyclic shifts using permutation keys used during encryption
- Assemble the clustered

prediction errors

- Add prediction value with prediction error
- Reconstruction image

At the receiver side, we perform de-arithmetic coding and de-permutation in which reverse cyclic shift operation is performed to get the original cluster and by adding prediction value with prediction error to get back reconstructed image.

V. RESULT AND DISSCUSSION

In this section, the security of image encryption and the compression performance on the encrypted data are evaluated experimentally.

The security of our proposed image encryption and the compression performance on the encrypted data are evaluated experimentally. In Figure 5.1 illustrate original and encrypted images, which we can see that our encryption approach is effective in destroying the semantic meaning of the images, original image is encrypted and which converted into unreadable format. Due to prediction error domain based image encryption can be attain more level of security, where pixel locations are shuffled repeatedly but not pixel vales.

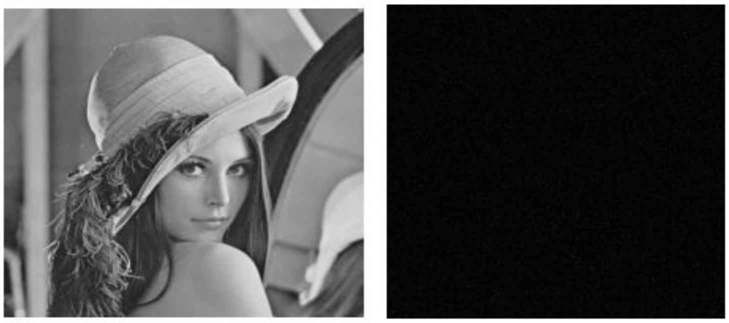


Figure 5.1: (a) original image (b) Encrypted image

Benchmarks in image data compression are the compression ratio and PSNR (Peak Signal to Noise Ratio). The compression ratio is used to measure the ability of data compression by comparing the size of the image being compressed to the size of the original image. The greater the compression ratio means the better the compression efficiency on the encrypted data.

In Figure 5.2 we also compare the rate-PSNR performance of our compression method with JPEG 2000 and the method in [6]. For bit rates above 2 bpp, our method achieves even higher PSNR values than JPEG 2000. The gain in PSNR over JPEG 2000 can be significant for high bit rates. For instance, for the image Lena, the gain is more than 2 dB. As bit rate drops, the PSNR gain over JPEG 2000 decreases. When the bit rate is below 2 bpp, the PSNR gain over JPEG 2000 diminishes and starts to become negative. It can also be seen that the PSNR gain of our method over the one in [6] is quite remarkable. When the bit rate is around 2.50 bpp, the PSNR gain can be over 10 dB for the Lena image. We also notice that the method of [6] seems to suffer from the problem of performance saturation for images with intensive activities such as Harbor, Barbara, and Bridge.

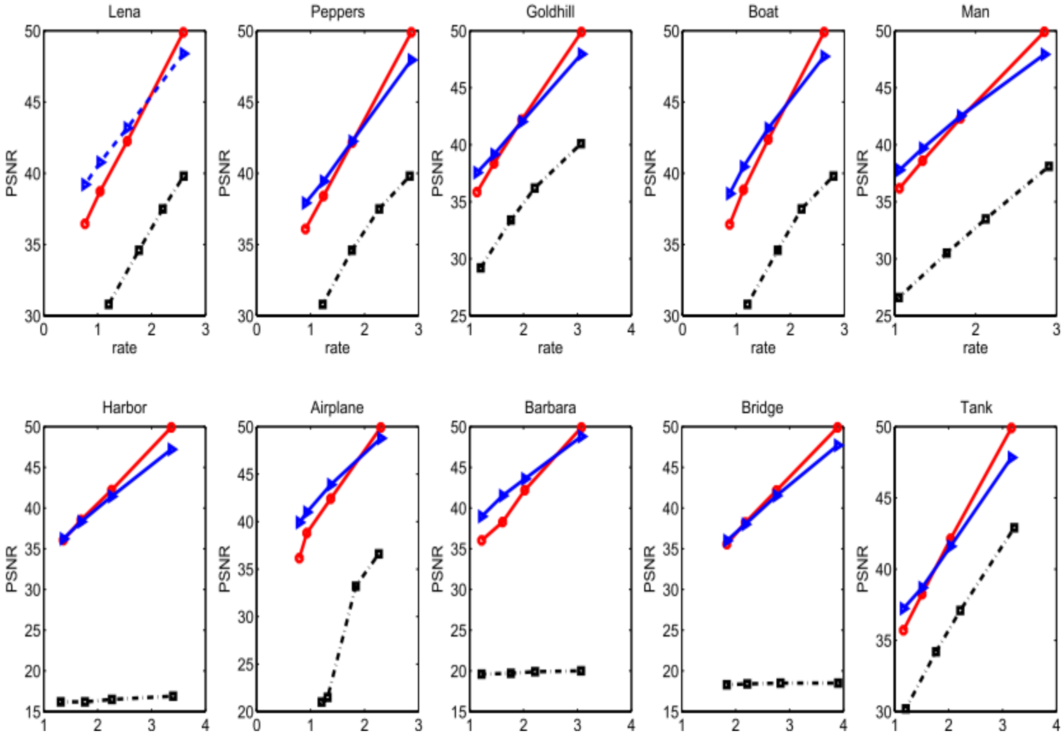


Figure 5.2: Comparison of the rate PSNR performance and bit per pixel of each image.

VI CONCLUSION

The proposed work concludes that image encryption has been achieved via prediction error clustering and random permutation which provides high level of security to the image data. Highly efficient compression of the encrypted image has been achieved by a context-adaptive arithmetic coding approach. Both theoretical and experimental results have shown that reasonably high level of security has been retained and finally Sequential decompression and decryption stage quality of their reconstruction image is guaranteed at the receiver side.

REFERENCES

[1] M. Weinberger, G. Seroussi, and G.Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” IEEE Trans. Image Process., vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[2] X.Wu and N. Memon, “Context-based, adaptive, lossless image coding, ” IEEE Trans. Commun., Vol.45, No. 4, pp.437–444, Apr. 1997.

[3] D.Schonberg, and K. Ramchandran,“On compressing encrypted data,”IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[4] A.Kumar and A.Makur, “Distributed source coding based encryption and lossless compression of gray scale and color images,” in Proc. MMSP, 2008, pp. 760–764.

[5] X. Zhang, “Lossy compression and iterative reconstruction for encrypted image,” IEEE Trans. Inf. Forensics Security, vol. 6, no. 1, pp. 53–58, Mar. 2011.

[6] G.Feng, Y.Ren, and Z.Qian, “Scalable coding of encrypted images,” IEEE Trans. Imag.Process., vol. 21, no. 6, pp. 3108–3114, Jun. 2012.

[7] R.Lazzeretti and M.Barni, “Lossless compression of encrypted greylevel and color images,” in Proc. 16th Eur. Signal Process.Conf., Aug. 2008, pp. 1–5.