Securing an Image Through PEC and Random Permutation

Nikhila P Kumar
M.Tech, IV semester,
Department of Computer science and engineering
Shridevi Institute of Engineering and Technology,
Tumkur.

Mr. Kiran G M B.E., M.Tech., MISTE.,
Assistant Professor,
Department of computer science and engineering
Shridevi Institute of Engineering and Technology,
Tumkur.

***Abstract*-Image encryption is a concept in which an image is protected by an unauthorized access. In most of the applications, image encryption should be done before image compression. This situation leads to a problem of how to design encryption in first stage and compression in second stage using encryption algorithm so that the efficiency of the compressed image should be high and effective. This project presents how to design an image encryption-then-compression (ETC) system in an efficient manner so that both lossless and lossy compressions are to be considered. The proposed image encryption scheme consists of permutation based encryption method in which this method is used over prediction error domain. Since the proposed image encryption scheme works on the prediction error domain, it provides high level of security. Along with the permutation based encryption method, this ETC system design an arithmetic coding (AC) based approach. This leads to efficient compression of the encrypted image which in turns results the proposed system to provide high level of security for an image. To compress the encrypted images we can make use of arithmetic coding approach. Though the compression performance of the encrypted image is bit compromised, it clearly gets advantage compared with the compression of an un-encrypted original image. Along with this, the proposed system comes with better performance compared with the existing systems in terms of security as the image would be required to be encrypted before compressing.**

Key Terms - Encryption-Then-Compression (ETC), Arithmetic coding (AC), Compression of encrypted images.

I INTRODUCTION

In cryptography, encryption is a process in which encoding of messages or information is done in such a way that only authorized person can read it. In encryption scheme, the original message or information is referred as plain text and that plain text is encrypted using an encryption algorithm. The encrypted message or information is referred as cipher text. Cipher text can be read only if it is decrypted. In an encryption scheme usually it uses a pseudo-random encryption key which is generated by an algorithm. Using that key it is possible to encrypt and decrypt a particular message or information. But for a good or well-designed encryption scheme, it requires advanced skills and large computational resources. Thus a message or information is encrypted and only an authorized recipient or person can decrypt the encrypted message or information easily but not by any unauthorized interceptors.

Consider a scenario where being a content owner Alice going to send an image I which should be secured and transmitted in an efficient way to the receiver Bob, through an untrusted channel provider Charlie. This has done as follows. First Alice compresses an image I and the obtained compressed image is T, and then the compressed image T is going to encrypt and obtain an encrypted image E_i. Encryption can be done using a function ES, in which S represents a secret key. This is shown in Fig 1. Charlie simply forwards this encrypted image to Bob. After receiving the image E_i Bob decrypts and decompresses sequentially to get back the image $\hat{I}$.

In some situations, encryption should be done in first stage and compression should be done in second stage. Thus an Encryption-then-Compression system
is designed. In this ETC system, Alice encrypts an image and sends that image to Charlie for compression, where Charlie is an untrusted channel provider. Finally the receiver Bob performs decompression and decryption sequentially to get an uncompressed and unencrypted image using secret key S. This ETC system is illustrated in Fig 2.

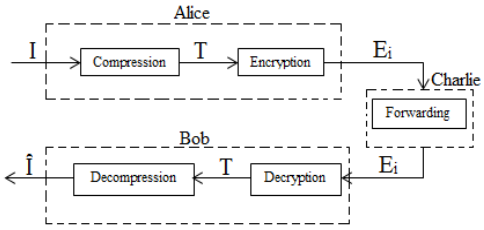

Fig 1: Compression-then-Encryption (CTE) system.

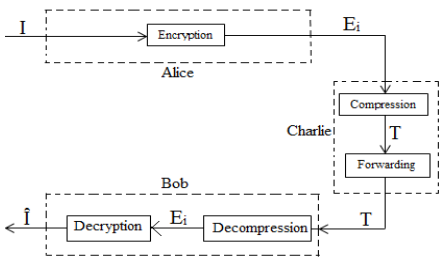

Fig 2: Encryption-then-Compression (ETC) system.

To get the high compression ratios and also the lossy compression for the data what it has been encrypted should also studied [1]-[7].In [1], they have proposed a framework for a scalable lossy coding for the images which are encrypted through multi-resolution. For the linear encryption a mechanism called Compressive Sensing (CS) was
used for the compression of the encrypted images [2]. The remaining part of this paper is as follows. Second section deals with the details of proposed system. Third section deals with the analysis for security and performance. Fourth section deals with the experimental results. And finally in fifth section conclusion will be there.

II PROPOSED SYSTEM

In this second section, three things of proposed ETC system are going to explain. That is, Alice encrypts an image, Charlie compresses the encrypted

2

image, and finally Bob decompresses and decrypts the encrypted and compressed image sequentially.

1. Image encryption through PEC and Random Permutation

In this proposed ETC system, an encryption algorithm is going to design for the purpose of security and better compression for the encrypted image. The image encryption can be done on the prediction error domain. The image encryption is shown in the Fig 3. First most thing is that the predicted error values is going to calculate for each pixels in an image I. For the image I which is going to encrypt, consider for each pixel $I_{a,b}$ first a prediction that is $\bar{I}_{a,b}$ is done using an image predictor GAP, where GAP refers to Gradient Adaptive Prediction [8]. The image predictor GAP is used because it has a good de-correlation capability. The result for this prediction $\bar{I}_{a,b}$ can get back by using context adaptive method [8] and thus the resulted prediction is denoted as $\tilde{I}_{a,b}$. Thus the prediction error can be calculated using the below formula

$$\mathbf{e_{a,b} = I_{a,b} - \tilde{I}_{a,b}} \qquad (1)$$

. Here an energy estimator [8] is used for each pixel location and it is given by,

$$\mathbf{\Delta_{a,b} = d_k + d_l + 2|e_{a-1,b}|} \qquad (2)$$

Where,

$$\mathbf{d_k = |I_{a-1,b} - I_{a-2,b}| + |I_{a,b-1} - I_{a-1,b-1}| + |I_{a,b-1} - I_{a+1,b-1}|}$$
$$\mathbf{d_l = |I_{a-1,b} - I_{a-1,b-1}| + |I_{a,b-1} - I_{a,b-2}| + |I_{a+1,b-1} - I_{a+1,b-2}|} \qquad (3)$$

Thus the cluster design is used for the purpose of security and easy compression of the encrypted data or image. The parameter N selection is needed to balance both security and complexity of the encryption. If N is larger it provides high level of security but it also provides high complexity encryption.
The procedure of image encryption algorithm is shown as follows:

Step 1: The prediction errors which is mapped that is $\tilde{e}_{a,b}$ is going to compute for the whole image I.

Step 2: After computing the prediction errors, it should be divided into N number of clusters and that is denoted as CL_s where s ranges from 0 to N-1.

Step 3: The prediction errors for all clusters CL_s should reshape into four columns and rows as 2-D blocks.

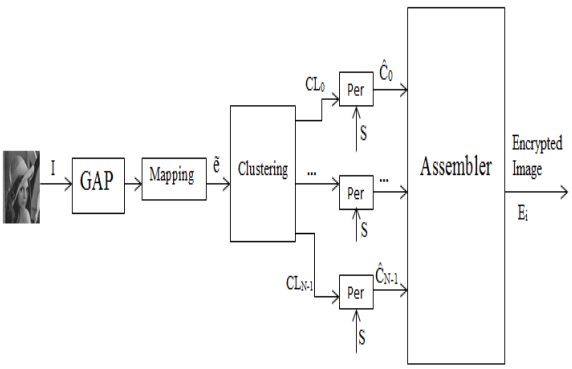

Fig 3. Schematic diagram of image encryption.

Step 4: After reshaping, the cyclic shift should be done for each blocks of prediction error to obtain the clusters which are permuted that is $\hat{C}_s$. This can be done using raster-scan order.

Step 5: Finally assembler is going to concatenate all the clusters which are permuted that is $\hat{C}_s$ and then an encrypted image E_i is obtained.

Step 6: Along with the cluster length, E_i is passed to Charlie. The cluster length is denoted by $|\hat{C}_s|$ where s ranges between 0 and N-2. That is s ranges in $0 \le s \le N-1$. In this way encryption is done using an image encryption algorithm and finally obtained an encrypted image E_i.

2. Compression of Encrypted Image through Arithmetic Coding (AC)

The encrypted image E_i should be compressed. In Fig 4 schematic diagram of the compressing an encrypted image is shown. Compression part is done by Charlie. The de-assembler takes an encrypted bit stream. Then E_i is divided into N segments from $\hat{C}_0$ to $\hat{C}_{N-1}$ as in the way which is done during encryption. Then an arithmetic coding is applied to encode each sequence of the prediction error. Thus $\hat{C}_s$ is converted into binary bit stream after applying

adaptive arithmetic coding. The binary bit stream is denoted as T_s and all the bit streams are generated in parallel. Then an assembler is going to concatenate the entire compressed and encrypted binary bit stream to produce final encrypted and compressed binary bit stream which is denoted as T. The below equation denotes the concatenation of all the compressed and encrypted bit stream

$$\mathbf{T = T_0T_1\ldots.T_{N-1}} \quad (4)$$

3. Decryption and Decompression sequentially

After receiving the encrypted and compressed image that is T, Bob is going to obtain the original image I.

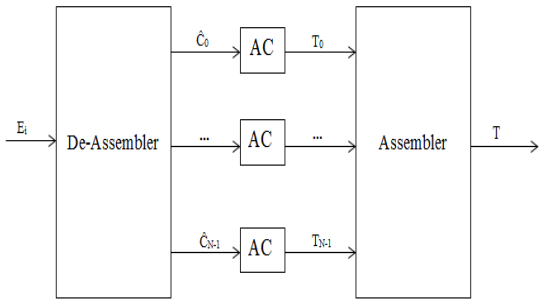

Fig 4: Compressing the encrypted image.

The sequential decompression and decryption is shown in the Fig 6. Initially Bob divides the bit stream T into N number of segments T_s in which it ranges from 0 to N-1. In which each of them is associated with the prediction errors in a cluster. So an adaptive arithmetic decoding is applied to get prediction error sequence which is already permuted that is $\hat{C}_s$. Bob knew the secret key S and hence he performs the de-permutation method or operation with the help of that secret key in order to obtain the original cluster CL_s. For each pixel, pixel value will be there and decoding of those pixel values can be done using raster-scan order. The associated energy error estimator $\Delta_{a,b}$ is calculated for every location (a, b) along with the predicted value. Finally the pixel values can be reconstructed by computing the below formula,

$$\mathbf{\hat{I}_{a,b} = \tilde{I}_{a,b} + e_{a,b}} \quad (5)$$

III SECURITY AND PERFORMANCE ANALYSIS

1. Security Analysis

Remember that the key stream which is controlled the random permutation and that is generated with the help of a stream cipher. Thus the key stream will be different though the encrypted image is same atdifferent times. Hence for the proposed system the cipher text-only attack [10] model is used. Where an attacker can access only the encrypted text that is cipher text and try to recover the plain text that is original image.Though AC model is public and invertible completely, the attacker obtains E_i (encrypted image), where $E_i$is obtained from the concatenation of N clusters, where each cluster consists sequence of prediction sequences. The length of $\hat{C}_s$ is known publicly and hence E_i is partitioned into N number of segments of $\hat{C}_s$where s ranges $0 \leq s$

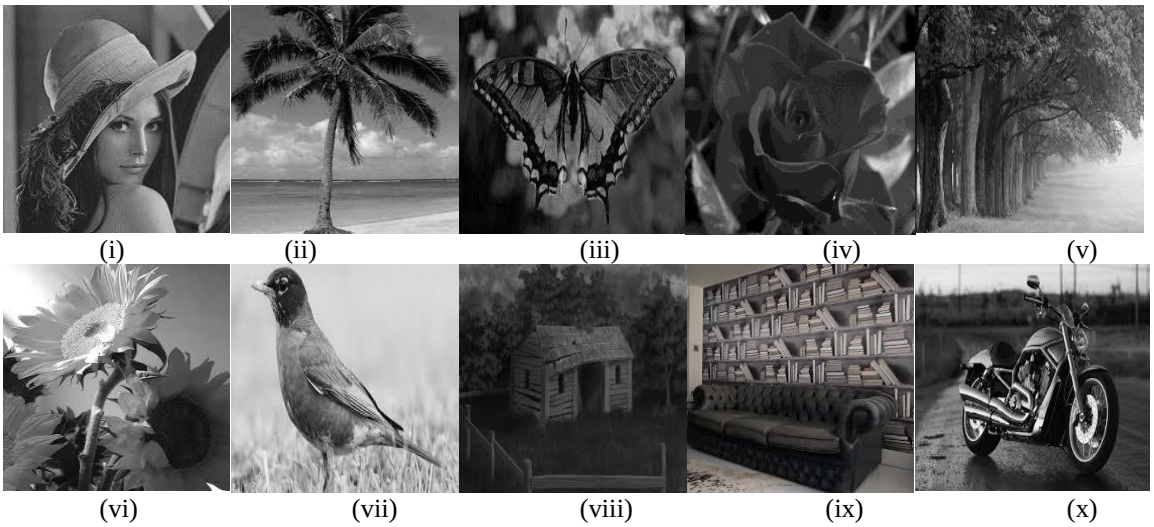

Fig 5:Ten tested images (i) Lena.(ii) Coconut tree. (iii) Butterfly. (iv) Rose. (v) Nature. (vi) Sunflower. (vii) Bird. (viii) House. (ix) Library. (x) Bike.

≤ N-1.The empirical probability mass function for each of $\hat{C}_s$is obtained by calculating

$$P_s(a) = \frac{\#a}{\hat{C}_s} \qquad (6)$$

Here #a is the number of a in each $\hat{C}_s$and a belongs to [0, 255]. Conditional entropy quality is going to calculate and this can be done by doing average of N clusters in order to measure the input image complexity.To get back the original image an attacker can attempt to decode the E_i directly. To decode correctly, attacker is in a need of prediction error $\tilde{e}_{a,b}$ and the predicted value $\tilde{I}_{a,b}$ which is associated. One way to decode E_i is guessing the prediction errors $\tilde{e}_{a,b}$ and that should estimate the cluster index s, by considering the neighboring pixels which are decoded and thus an attacker take one of the element from the cluster CL_s. But the cluster index can only determine with the help of error energy estimator that is $\Delta_{a,b}$, where energy estimatorcan be calculated using the surrounding casual pixels.Thus the predictive coding which is used in proposed
system helps in security purpose. There is another way for an attacker to get back the original image explicitly. If the attacker estimated all the pixels till $I_{a,b}$ correctly, then the cluster index for the pixel which is located at (a, b+1) and the predictions say $\tilde{I}_{a,b+1}$ should be known perfectly, where every casual surroundings are going to decode without any error. The next job of the attacker is to select at least one prediction error from the cluster s. Because of the high spatial correlation the selected prediction error by an attacker makes lot of difference in reconstructed image compared to $I_{a,b}$then attacker easily reject this method. Thus the spatial correlation helps in protecting the original image from the attacker. Thus security can be improved with the help of prediction error clustering and random permutation.

2. Compression Performance

Image predictors say GAP [8] or MED [9] have a strong de-correlation capability and thus a very small amount of inter-dependence exists in the sequence of prediction error. When the traditional predictive coding system is compared with the proposed system where in traditional system compression is going to be done for original image by Charlie where in the proposed system compression is going to be done for permuted one, hence the performance of the compression image matters a lot. Thus it is necessary to consider inter-dependence factor. The inter-dependence left in each CL_s is rather limited, thanks to the superior de-correlation capability of imagepredictor. Hence the compression ratio is verified by the experiments by causing little bit noise after decryption and decompression.

IV EXPERIMENTAL RESULTS

Here the compression performances for encrypted images are experimentally evaluated. Ten images are considered for encryption and compression are

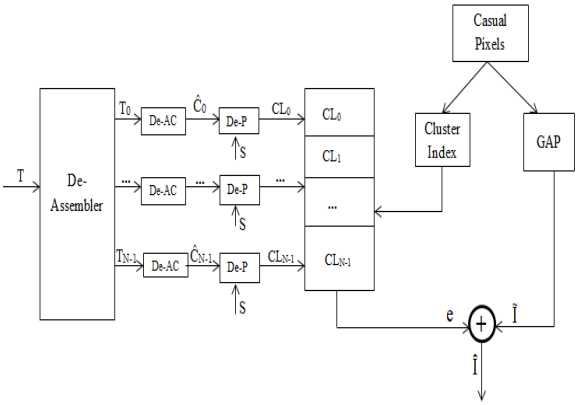

Fig 6: Sequential Decompression and Decryption.

shown in Fig 5. In earlier work the PSNR ranges from 25-30 that means the noise created in those images will be very high and hence the clarity of reconstructed images is little bit less. In Fig 7, the results of PSNR for the reconstructed images are shown by using the PEC and random permutation. In TABLE I, efficiency of the compression for encrypted images are listed for ten images using PEC and random permutation. Thus the permutation-based image encryption approach which is used in proposed system is very useful where the privacy or secrecy of an image is not much required.

V IMPLEMENTATION

The message that is going to be conveyed in this paper is the compression performance would be improved above 75% and the level of security would be high and noise created is less using PEC and random permutation. Hence a personal image can transmit in a good manner.

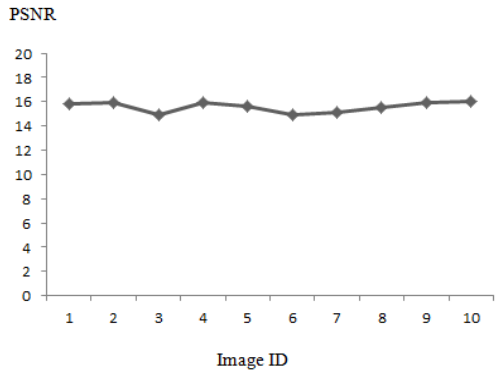

Fig 7: PSNR values for all ten images.

VI CONCLUSION

In this paper, an image Encryption-then-Compression (ETC) system is designed efficiently. In

5

proposed system, image encryption through prediction error clustering and random permutation is done in a well manner. The compression performance for encrypted images using arithmetic coding approach has been achieved. Experimental results are shown. The most

important thing is compression performance of the proposed method is good, where the noise created while reconstructing original unencrypted image is very low. Thus the proposed method achieves high level of security along with the better compression performance.

TABLE I

Images	Compression Ratio (%)	PSNR
Lena	90.856	15.814
Coconut tree	95.337	15.917
Butterfly	90.954	14.951
Rose	85.734	15.922
Nature	95.981	15.601
Sunflower	95.990	14.910
Bird	83.636	15.156
Home	78.834	15.498
Library	93.185	15.974
Bike	93.830	15.982

REFERENCES

[1] X. Zhang, 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.

[2] A.Kumar and A.Makur "Lossy compression of encrypted image by compressing sensing technique," in *Proc. IEEE region 10 conf. TENCON*, jan 2009,pp. 1-6.

[3] X. Zhang, Y. L. Ren, G. R. Feng, and Z. X. Qian,"compressing encrypted image using compressive sensing," in *proc. IEEE 7 thIIHMSP* Oct.2011, pp. 222-225.

[4] X. Zhang, "Lossy compression and iterative recobstruction for encrypted image," *IEEE Trans. Inf. Forensics Security* ,vol. 6, no. 1, pp. 53–58, Mar. 2011.

[5] X. Zhang, G. Sun, L. Shen, and C. Qin, "Compression of Encrypted images with multilayer decomposition," *MultimedTools Appl.*, vol. 78, no. 3, pp. 1–13, Feb. 2013.

[6] D. Schonberg, S. C. Draper, C. Yeo, and K. Ramchandran, " Toward compression of encrypted images and Sequences" *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 749–762, Dec. 2008.

[7] Q. M. Yao, W. J. Zeng, and W. Liu, "Multi-resolution based Hybrid spatiotemporal compression of encrypted videos," in in*Proc. ICASSP*, Apr. 2009, pp. 725–728.

[8] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," *IEEE Trans. Commun.*, vol. 45, no. 4, pp 437-444 , Apr. 1997.

[9] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I Loss-less image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Imag.Process*,vol. 9, no. 8, pp. 1309–1324, Aug. 2000

[10] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook Applied Cryptography*. Cleveland, OH, USA : CRC Press, 1997