DESIGNING AN ADVANCED ETC SYSTEM FOR SECURE IMAGE DATA TRANSMISSION

Shifa M. and Ajmal Mohammed V. M.
ECE Department, M.E.A Engineering College, Perinthalmanna, India
E-Mail: shifamohammed.m@gmail.com

ABSTRACT

In recent years due to the security concerns in the service oriented environments like cloud computing, compression of encrypted data has drawn much attention. A highly secure algorithm is used to encrypt the full image after which lossless compression is done on the encrypted image. To provide reasonably high level of security we couple image encryption scheme operated in prediction error domain with binary permutation. Huffman coding based approach is established for efficiently compressing the encrypted image. Hence a highly efficient image encryption then compression (ETC) scheme considering the lossless compression is designed. Experimental results shows that this encryption then compression scheme provides better performance than existing schemes.

Keywords: image compression, image encryption, lossless compression.

INTRODUCTION

Security of multimedia data has become more important as they are frequently transmitted over open networks. Our military, government, medical fields etc. deals with a large number of confidential images which fallen in to wrong hands may end in catastrophic conditions. Hence encryptions of images are both legal and ethical. The classical way of efficiently and securely transmitting redundant data is to first compress and then encrypt the data. At the receiver side decryption is done followed by decompression. However in some practical scenarios like service oriented environments (e.g., cloud computing), sensoring networks etc., encryption is to be done prior to compression due to its limited computational resources. To maximize the network utilization, the channel provider has an overriding affinity to compress all the network traffic. So it will be much desirable if compression task is carried out by the channel provider who is having plenty of computational resources.

In recent years, processing of secured signals directly in the encrypted domain has gained a great attention. Since no semantic information about the image is available, compression in the encrypted domain was considered to be infeasible. Later in [1-3] signal processing modules working directly on the encrypted data to provide an elegant solution to protect the signals from malicious processing devices were introduced. Coding with side information principles was introduced in [4] which proposed algorithms to compress the encrypted binary images without loss of data. Several methods for lossless compression of encrypted grayscale/color images by applying LDPC codes were proposed in [5]. Better lossless compression performance on the encrypted gray scale/color images were achieved in [6] by applying encryption to prediction errors other than applying it directly to the image. Resolution progressive compression scheme which compress an encrypted image progressively in resolution was discussed in [7]. Recently all these works were extended to efficiently compressing the block cipher encrypted data [8] where randomization prevents identical plain text blocks from being encrypted in to identical cipher text blocks. Higher compression ratios are achieved by lossy a compression which is achieved through decomposition of images in to multiple sub images or layers [9]. Extensions of compression of encrypted videos were also studied later. Linear transformations were used to compress the cipher text produced by the stream cipher [10]. Videos with higher irregular motions were compressed [11], which derives temporal side information from the previous slides and also generates spatial side information by having partial access to the current frame.

Compared with the state-of-the-art lossless image coders that accept unencrypted inputs, the existing systems still fall short in compression performance. The major aim here is the construction of a pair of protocols to overcome any adversaries and various aspects of information security, coupled with compression of images which is almost equally efficient as compressing the unencrypted versions. Lossless compressions of 8-bit grayscale images are considered. Encryption scheme is permutation based approach over the prediction error domain which provides reasonably high level of security. Compression of encrypted image is carried out by Huffman coding.

Section II provides the proposed ETC system. Experimental results are discussed in section III and we conclude in section IV.

PROPOSED ETC SYSTEM

Image Encryption

The encryption algorithm considered here should consider both the security and the ease of compressing the encrypted image. The image encryption scheme operates in the prediction error domain. Schematic diagram of image encryption then compression is given in Figure-1. Consider a transmitter A needs to send a confidential image I_M to receiver B with the help of the channel provide C. As the content owner A being a resource deprived mobile device has no incentive to compress the data before encryption while the channel provider C having plenty of computational resources increases the network utilization by compressing the protected data without any access to the secret keys.

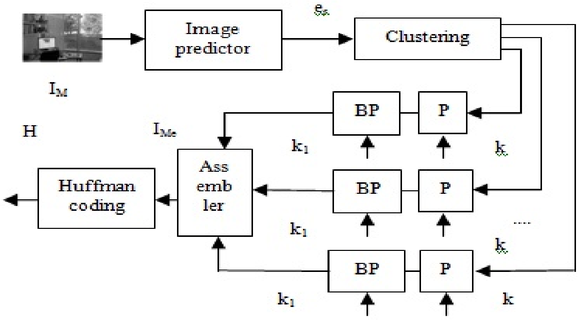

Figure-1. Schematic diagram of compressing an encrypted image.

For each pixel $I_{M(j,k)}$ of the image I_M, a prediction $\tilde{I}_{M(j,k)}$ is to be found. This can be done with the help of an image predictor like GAP, MED etc. But here we use cumulative addition method which is simple and easy to implement while considering other methods. Cumulative addition method is defined as follows. Pixel value $I_{M(1,2)}$ will be replaced with $I_{M(1,1)}$. Similarly $I_{M(1,3)}$ will be replaced with $I_{M(1,2)}$ and it goes on. Coming to second row, elements will be replaced with the corresponding elements of previous row and this will be repeated throughout the image. Finally $I_{M(1,1)}$ will be replaced by a known integer X. Hence $\tilde{I}_{M(j,k)}$ is formed. Prediction error corresponding to $I_{M(j,k)}$ can be estimated using (1).

$$e_{r(j,k)}= I_{M(j,k)} — \tilde{I}_{M(j,k)} \quad (1)$$

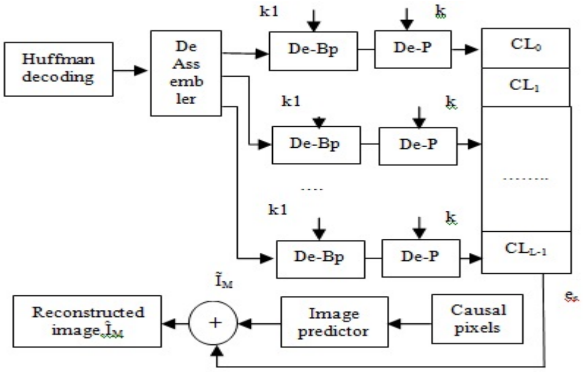

Figure-2. Schematic diagram of recovering the image.

Here 8 bit grayscale images are considered. Hence $e_{r(j,k)}$ can take any value within the range [-255,255]. From (1) it is clear that $e_{r(j,k)}$ must fall in the interval $[-\tilde{I}_{M(j,k)}, 255- \tilde{I}_{M(j,k)}]$ which only contains 256 distinct values. Each of this prediction values can be mapped in between the interval [0,255] by simply taking the absolute value of each of the errors and storing the positions of [-255,-1] in a directory for future use.

Instead of treating all the prediction errors $e_{r(j,k)}$ as whole, divide them in to L distinct clusters. The value of L needs to be carefully selected since L needs to maintain a balance between encryption complexity and security, Larger the value of L, it will be more difficult for the attacker, but increases the complexity of the encryption. Here an optimum value of L=8 is selected. L can either be fixed between transmitter or receiver or can be made publically accessible.

A two key driven cyclical shift is done to each cluster. Let P_k and Q_k be the two secret vectors controlling the column and row shifts for each cluster CL_V, where V is the cluster index. These keys are randomly generated. Which means that key vectors used for the same image at different times will not be same resulting in different encrypted image for the same one. Downward shift is done over column and the same is repeated for the row, i.e. from left to right. Then each permuted prediction errors will be converted from decimal to binary values. BP denotes the binary permutation. Now each cluster is considered as individual groups with zeros and ones. A circular shift using a key k1 is applied to each bits. k1 can be either randomly generated or fixed. After binary permutation, values of prediction errors in each pixel positions will be changed. An assembler is used to concatenate all the clusters together to form an encrypted image.

$$I_{Me} = \hat{CL}_0 \; \hat{CL}_1 \; \hat{CL}_2 .... \hat{CL}_{L-1} \quad (2)$$

Where $\hat{C}L$ is the encrypted cluster. The file size is preserved since the number of prediction errors is equal to that of number of pixels. Each prediction error will be of 8 bits. This encrypted image is passed to the channel provider. If length of each cluster is fixed, then there is no need to send length of the cluster to the channel provider C. Then the overhead induced by sending | $\hat{C}L_V$| , which denotes the length of each cluster can be avoided.

Lossless Compression of Encrypted Images via Huffman Coding

The channel provider does not have access to any of the secret keys. Compression has to be done in the encrypted domain. The major design challenge faced was the compression of encrypted image since no image structure is obtained to enable traditional compression. Less complexity has led to the selection of Huffman coding for lossless compression of encrypted image. Using Huffman coding channel provider C simply compresses the encrypted image I_{Me} into B as shown in Figure-1. Huffman coding is a lossless compression standard in which frequently occurring values are given smaller codes and values that rarely occur are given larger codes. Algorithm works on the estimated frequency of occurrence for each possible value of the source symbol.

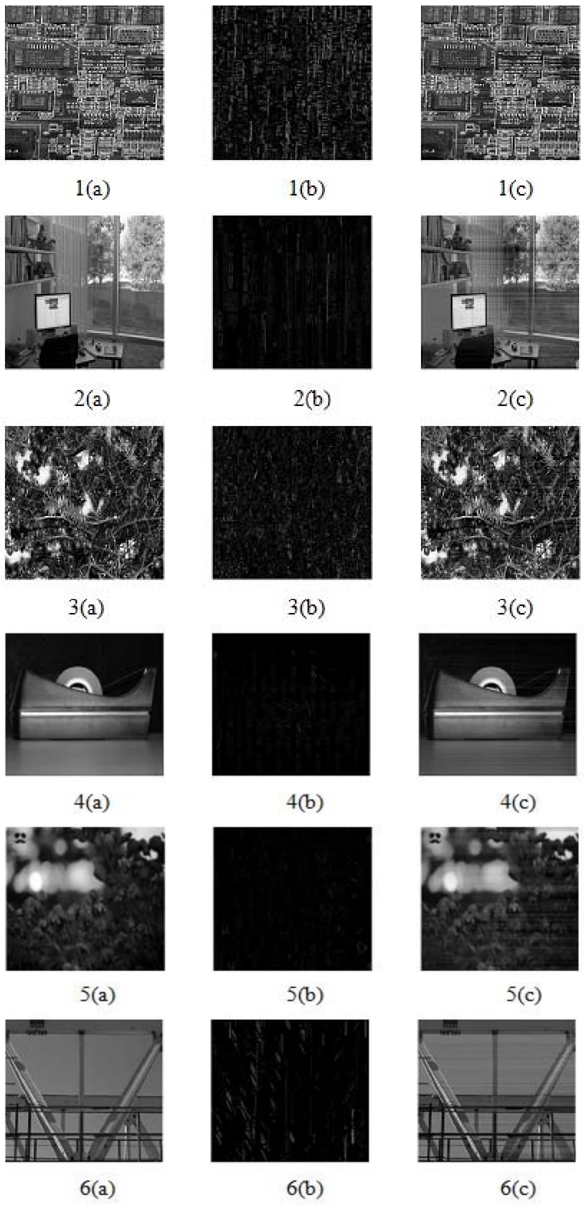

Figure-3. Experimental results of test images (1) board, (2) office, (3)greens, (4) tape, (5) flowers, (6) gantry crane, (a) original and (b) encrypted image (c) reconstructed image.

Huffman coding is easy to implement compared to arithmetic coding. Similar to the encryption process, compression can also be done to clusters but there will be an overhead induced by sending side information like length of each compressed clusters, along with separate dictionaries for distinct clusters.

The algorithmic procedure for performing the proposed ETC system is given as follows:

Step 1: Resize the image into 512*512.

Step 2: Compute all the mapped prediction errors $e_{r(j,k)}$ of the whole image I_M.

Step 3: Divide all the prediction errors into L clusters CL_V, for $0 \leq V \leq L - 1$.

Step 4: Apply two key driven cyclical shift to each prediction error block in a raster scan order .

Step 5: Binary permutation of each prediction error values to obtain the permuted cluster $\hat{C}L_V$.

Step 6: Permuted clusters $\hat{C}L_V$ are concatenated by an assembler to generate final encrypted image I_{Me}.

Step 7: Pass I_{Me} to the channel provider.

Step 8: Huffman coding is then employed to losslessly encode prediction error sequence into a binary bit stream B.

Step 9: Finally compressed form of encrypted image is send to the receiver where sequential decompression and decryption is done.

Joined Decompression and Decryption

The schematic diagram of recovering the image is given in Figure-2. After receiving the compressed encrypted image, receiver B aims at recovering the original image, I_M. First step is the decompression of the received confidential image using Huffman coding. As B knows the cluster length and the number of clusters along with the secret key, it is easy for the receiver to recover the image. With the help of causal pixels from cumulative addition method, it is possible to reconstruct the image. The reconstructed pixel value can be calculated as,

$$\hat{I}_{M(j,k)}=\tilde{I}_{M(j,k)}+e_{r(j,k)} \tag{3}$$

The above discussed lossless compression can be further extended to lossy compression if necessary. But the quantization has to be done by A itself. Since, prediction value $\hat{I}_{M(j,k)}$ is based on original unquantized surrounding pixels which will not be available at the receiver side

EXPERIMENTAL RESULTS

Experimental evaluation of the proposed image encryption then compression system is done using MATLAB. Figure-3 shows the experimental results of the proposed system using 6 test images. It is clear from the figure that the encryption approach followed here is successful in the extirpation of any semantic information of the images.

Statistical information leakage is possible for any ETC system. Through the proposed encryption scheme, it is possible to disrupt this leakage to an extent, which also increases the number of ways in which each cluster can be permuted. Spatial correlation of natural images is another way of recovering the original image by the attacker. Assuming that the attacker somehow correctly calculated the prediction value $\tilde{I}_{M(j,k)}$, then $\tilde{I}_{M(j,k+1)}$ can be easily calculated without much error. Once the attacker knows $\tilde{I}_{M(j,k)}$, then he selects a possible prediction error which is then used to estimate $\tilde{I}_{M(j,k+1)}$. If the reconstructed $\tilde{I}_{M(j,k+1)}$ deviates too much from $\tilde{I}_{M(j,k)}$, attacker rejects the selection. Spatial correlation helps the attacker to reduce the size of the set where the true $e_{r(j,k)}$ lies. By applying binary permutation this correlation is lost to some extent which makes the attacker more difficult to figure out the correct prediction values.

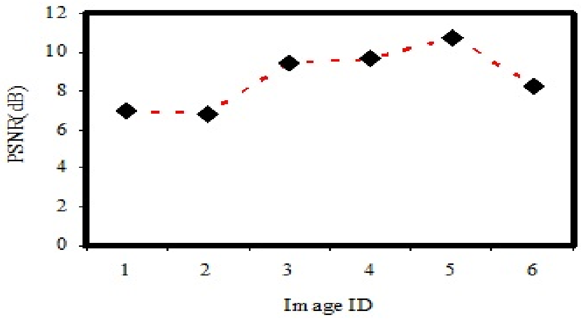

Figure-4. Decoding performance of the test images.

Test images used here are of dimension 512*512.PSNR values of these images are given in Figure-4 with PSNR values in y axis and image ID in x axis. It can be observed that PSNR values are around 10dB which is very low to convey any useful information. From Figure-5 it is clear that compression efficiency of the proposed scheme is greater than existing system. CR refers to compression ratio. Normalized absolute error and cross correlation of the proposed system is discussed in Table-1. For comparison purpose lossless version of the arithmetic coding (AC) is tabulated in the table.

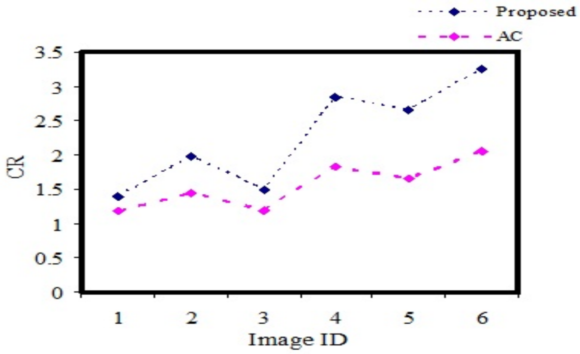

Figure-5. Compression performance of test images.

Table-1. Performance analysis.

Table Head	Normalized absolute error		Normalized cross correlation	
	Proposed	*AC*	*Proposed*	*AC*
1	0.8586	0.8472	0.1536	0.1563
2	0.9407	0.9376	0.0523	0.0560
3	0.8530	0.8411	0.1351	0.1380
4	0.9622	0.9589	0.0272	0.0309
5	0.9600	0.9518	0.0237	0.0268
6	0.9618	0.9566	0.0403	0.0441

CONCLUSIONS

An advanced ETC system for secure image data transmission is designed. Encryption scheme used here is prediction error clustering and random permutation which includes binary permutation of the prediction errors. Lossless compression of the encrypted image is carried out by Huffman coding. Experimental results show that a high level of security along with improved compression efficiency is obtained.

ACKNOWLEDGEMENTS

The authors would like to thank all the faculties of ECE department in MEA Engineering College for all their support during the work. Also authors would like to thank all unknown referees whose papers helped during this work

REFERENCES

[1] T. Bianchi, A. Piva and M. Barni. 2009. “On the implementation of the discrete Fourier transform in the encrypted domain,” IEEE Trans. Inf. Forensics Security, Vol. 4, No. 1, pp. 86–97.

[2] T. Bianchi, A. Piva and M. Barni. 2009. “Encrypted domain DCT based on homomorphic cryptosystems,” EURASIP J. Inf. Security, Article ID 716357.

[3] Z. Erkin, T. Veugen, T. Toft and R. L. Lagendijk. 2012. “Generating private recommendations efficiently using homomorphic encryption and data packing,” IEEE Trans. Inf. Forensics Security, Vol. 7, No. 3, pp. 1053–1066.

[4] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg and K. Ramchandran. 2004. “On compressing encrypted data,” IEEE Trans. Signal Process., Vol. 52, No. 10, pp. 2992–3006.

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

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

[7] W. Liu, W. J. Zeng, L. Dong and Q. M. Yao. 2010. “Efficient compression of encrypted grayscale images,” IEEE Trans. Imag. Process., Vol. 19, No. 4, pp. 1097–1102.

[8] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk and T. Rabin. 2012. “On compression of data encrypted with block ciphers,” IEEE Trans. Inf. Theory, Vol. 58, No. 11, pp. 6989–7001.

[9] X. Zhang, G. Sun, L. Shen and C. Qin. 2013. "Compression of encrypted images with multilayer decomposition," Multimed. Tools Appl., Vol. 78, No. 3, pp. 1–13.

[10] D. Schonberg, S. C. Draper, C. Yeo and K. Ramchandran. 2008. "Toward compression of encrypted images and video sequences," IEEE Trans. Inf. Forensics Security, Vol. 3, No. 4, pp. 749–762.

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

[12] Jiantao Zhou, Xianming Liu, Oscar C. Au and Yuan Yan Tang. 2014. "Designing an Efficient Image Encryption-Then-Compression System via Prediction Error Clustering and Random Permutation", IEEE transactions on information forensics and security, , Vol. 9, No. 1.