Image Encryption-Then-Compression System Operated in the Prediction Error Domain and Random Permutation

Sujatha D[1], Venkatakiran S[2]

[1] M. Tech Scholar, Department of ECE, Sri VenkatesaPerumal College of Engineering & Technology, Puttur, India

[2]Associate Professor, Department of ECE, Sri VenkatesaPerumal College of Engineering & Technology, Puttur, India

Abstract: ***In many practical scenarios, image encryption has to be conducted prior to image compression. This has led to problem of how to design a pair of image encryption and compression algorithms such that compressing the encrypted images can still be efficiently performed. In this paper, we design highly efficient image encryption-then-compression (ETC) system, where both the lossy and lossless compression are considered. The proposed image encryption technique operated in the prediction error domain is shown to be able to provide high level of security. In this paper wealso demonstrate that an arithmetic coding-based approach can be exploited to efficiently compress the encrypted images. The proposed compression approach applied to encrypted images is only slightly worse, in terms of the compression efficiency, than the state-of-the-art loosely/lossless image coders, which take original, unencrypted images as inputs. In contrast, most of existing ETC solutions induce significant penalty on the compression efficiency.***

Keywords: Compression of encrypted image, encrypted domain signal processing

www.ijsr.net

1. Introduction

The transmission and the transfer of images, in free spaces and on lines, are actually still not well protected. The standard techniques of encoding are not appropriate for the particular case of the Images. The best would be to be able to apply asymmetrical systems of encoding so as not to have a key to transfer. Because of the knowledge of the public key, the asymmetrical systems are very expensive in calculation, and thus a protected transfer of images cannotbe envisaged. The symmetrical algorithms impose the transfer of the secret key. The traditional methods of encoding images impose the transfer of the secret key by another channel or another means of Communication. Consider an application scenario in which a content owner Alice wants to securely and efficiently transmit an image I to a recipient Bob, via an untrusted channel provider Charlie. Conventionally, this could be done as follows. Alice first compresses I into B, and then encrypts B into Ie using an encryption function EK (·), where K denotes the secret key as illustrated in Fig.1 (a).The encrypted data I is then assed to Charlie, who simply forwards it to Bob upon receiving I Bob sequentially performs decryption and decompression to get a reconstructed image I. Even though the above Compression then Encryption (CTE) paradigm meets the requirements in manysecure transmission scenarios, the order of applying the compression and encryption needs to be reversed in some other situations.Even though the above *Compression-then-Encryption(CTE)* paradigm meets the requirements in many secure transmissionscenarios, the order of applying the compression and encryptionneeds to be reversed in some other situations. As thecontent owner, Alice is always interested inprotecting theprivacy of the image data through encryption. Nevertheless,Alice has no incentive to compress her data, and hence,will not use her limited computational resources to run acompression algorithm before encrypting the data.

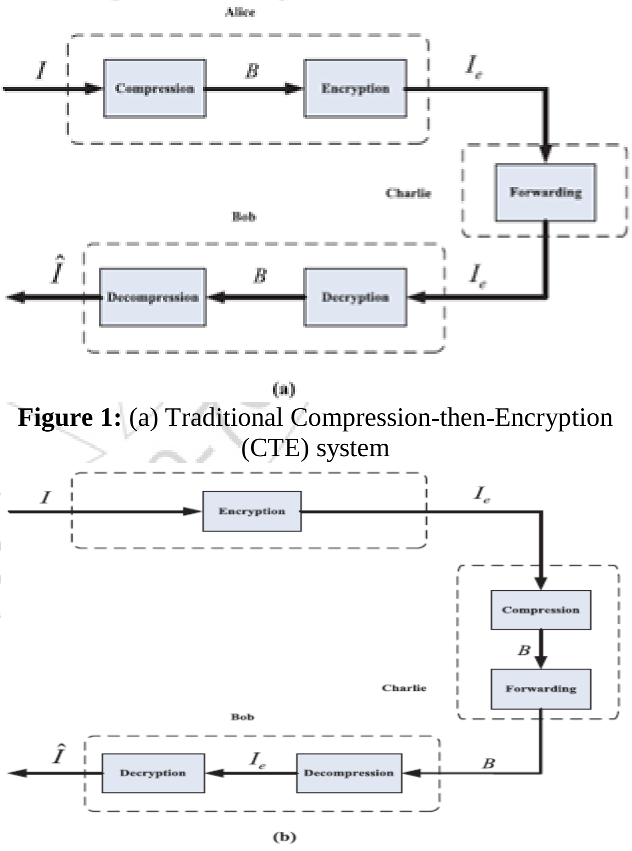

Figure 1: (a) Traditional Compression-then-Encryption (CTE) system

(b) Encryption-then-Compression (ETC) system

This isespecially true when Alice uses a resource-deprived mobiledevice. In contrast, the channel provider Charlie has anoverriding interest in compressing all the network traffic soas to maximize the network utilization. It is therefore muchdesired if the compression task can be delegated by Charlie,who typically has abundant computational resources. A bigchallenge within such *Encryption-then-Compression (ETC)* framework is that compression has to be conducted in theencrypted domain, as Charlie does not access to the secretkey *K*. This type of ETC system is demonstrated in Fig. 1(b). The rest of this paper is organized as follows. Section 2 gives the details of our proposed ETC system. Experimental results are reported in Section 3 to validate our findings. We conclude in Section 4

2. Proposed ETC System Design Process

For Proposed research work has three different modules which will be presented here. We will have the four phases like: Encryption of image, Compression, Decryption. Random permutation and clustering is the new methodology used for image encryption and compression. The phases are

A. Image Encryption

The first phase is the image encryption where the image is split into blocks and these blocks are permutated. Further permutation is applied based on a random number to strengthen the encryption.

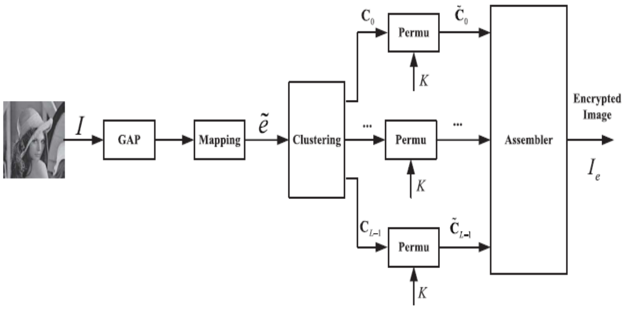

Figure 2: Schematic diagram of image encryption

This phase source image is applied to the image predictor GAP which converts image into pixels. Prediction error is calculated. In the clustering no of pixels grouped together and forms clusters. Once the clusters form then random permutation is applied on each cluster. At the assembler all the clusters combined and we will get encrypted image.

The algorithmic procedure of performing the image encryption is then given as follows:

Step 1:Compute all the mapped prediction errors˜ei,j of the whole image I.

Step 2: Divide all the prediction errors into L clusters Ck, for 0 ≤ k ≤ L − 1, where k is determined by (5), and each Ck is formed by concatenating the mapped prediction errors in a raster scan order

Step 3: Reshape the prediction errors in each Ck into a 2-D block having four columns and ⌈|Ck|/4⌉rows, where |Ck| denotes the number of prediction errors in Ck.

Step 4:Perform two key-driven cyclical shift operations to each resulting prediction error block, and read out the data in raster scan order to obtain the permuted cluster˜C k.

Step 5: The assembler concatenates all the permuted clusters Ck, for 0 ≤ k ≤ L−1, and generates the final encrypted image

$$I_e = \tilde{C}_0\tilde{C}_1 \cdots \tilde{C}_{L-1}$$

in which each prediction error is represented by 8 bits. As the number of prediction errors equals that of the pixels, the file size before and after the encryption preserves.

Step6: Pass Ie to Charlie, together with the length of each cluster ⌈C k|, for 0 ≤ k ≤ L − 2. The values of | C k| enable Charlie to divide Ie into L clusters correctly. In comparison with the file size of the encrypted data, the overhead induced by sending the length ⌈C k| is negligible.

B. Compression Phase

Encrypted image is disassembled and apply arithmetic coding on each cluster. Again the clusters are assembled at the assembler and we will get compressed image.

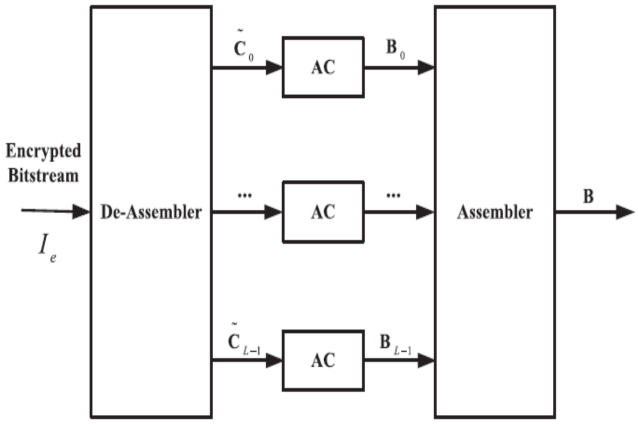

Figure 3: Schematic diagram of compressing the encrypted data

Arithmetic coding

Arithmetic coding is especially suitable for small alphabet (binary sources) with highly s keyed probabilities. Arithmetic coding is very popular in the image and video compression applications.

Consider a half open interval [low, high).Initially, interval is set as [0, 1) and range=high -low =1-0 = 1.Interval is divided into cumulative probabilities of n symbols. For this example, n=3; p (a) =1/2, p (b) =1/4 and p(c) =1/4.We propose an image encryption scheme operated over the prediction and permutation based image encryption method and the efficiency of compressing the encrypted data. The compression of the encrypted file Ie needs to be performed in the encrypted domain, as Charlie does not have access to the secret key K. In Fig. 3, we show the diagram of compression of Ie. Assisted by the side information⌈C k|, for 0 ≤ k ≤ L − 2, a de-

Assembler can be utilized to parse Ie into L segments˜C 0,˜C 1,・・,˜C L−1 in the exactly same way as that done at the encryption stage. An AC is then employed to encode each prediction error sequence˜C k into a binary bit stream Bk. Note that the generation of all Bk can be carried out in a parallel manner to improve the throughput. An assembler concatenates all Bk to produce the final compressed and encrypted bit stream B, namely, B = B0B1 ・・・BL−1

C. Decryption Phase

The receiver takes the help of the key to construct the secret image in the decryption process. The technique proposed is a unique one from the others in a way that the key is generated with valid information about the values used in the encryption process. At receiver receiver must know the prediction value of pixel.The below figure shows the process of sequential decryption and decompression

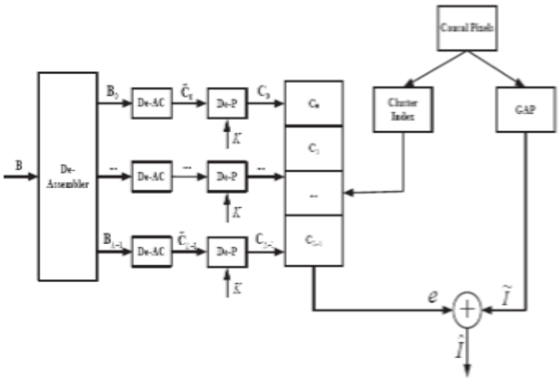

Figure 4: Schematic diagram of sequential decryption and Decompression

3. Simulation Results

The below figure shows the original image, encrypted image and reconstructed image.Table.1 gives the Quality Measures of Image.

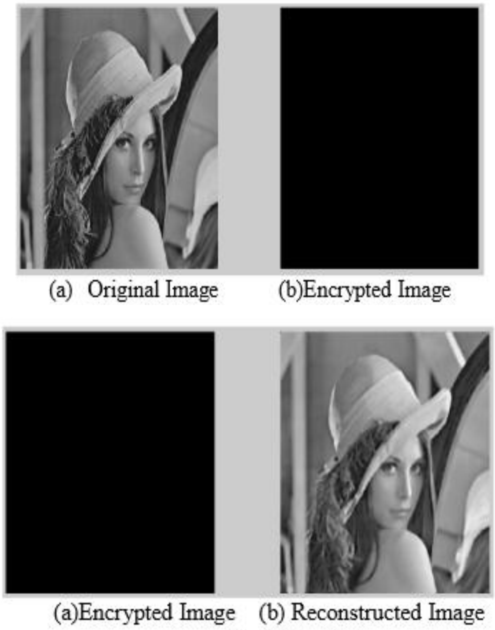

Figure: simulation results

Table 1: Quality measures of Image

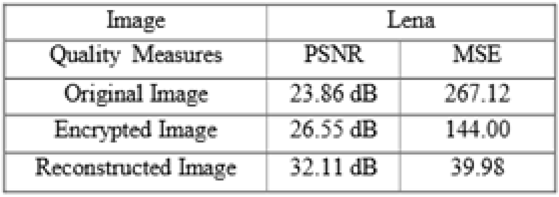

Image	Lena	
Quality Measures	PSNR	MSE
Original Image	23.86 dB	267.12
Encrypted Image	26.55 dB	144.00
Reconstructed Image	32.11 dB	39.98

4. Conclusion

Proposed System is used to design a pair of image encryption and compression technique such that compressing encrypted images. The image encryption has been achieved via random permutation. And compression is achieved by using arithmetic coding where both lossy and lossless compression is considered. The analysis regarding the security of the proposed permutation based image encryption method and the efficiency of compressing the encrypted data. For lossless compression and data hiding optical value transfer method can also be used.

References

[1] J. Zhou, X. Liu, and O. C. Au, “On the design of an efficient encryptionthen-compression system,” in *Proc. ICASSP*, 2013, pp. 2872–2876.

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

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

[4] T. Bianchi, A. Piva, and M. Barni, “Composite signal representation for fast and storage-efficient processing of encrypted signals,” *IEEE Trans.Inf. Forensics Security*, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[5] M. Barni, P. Failla, R. Lazzeretti, A.-R.Sadeghi, and T. Schneider,“Privacy-preserving ECG classification with branching programs and neural networks,” *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2,pp. 452–468, Jun. 2011.

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

[7] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, “On compressing encrypted data,” *IEEE Trans. SignalProcess.*, vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[8] D. Schonberg, S. C. Draper, and K. Ramchandran, “On blind compression of encrypted correlated data approaching the source entropy rate,” in *Proc. 43rd Annu. Allerton Conf.*, 2005, pp. 1–3.

[9] D. Schonberg, S. C. Draper, and K. Ramchandran, “On compression of encrypted images,” in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 269–272.

[10] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, “Oncompression of data encrypted with block ciphers,” *IEEE Trans. Inf.Theory*, vol. 58, no. 11, pp. 6989–7001, Nov. 2012.

Licensed Under Creative Commons Attribution CC BY