An Efficient Compression of Strongly Encrypted Images using Error Prediction, AES and Run Length Coding

Stebin Sunny[1], **Chinju Jacob**[2], **Justin Jose T**[3]

[1]Final Year M. Tech. (Cyber Security), KMP College of Engineering, Perumbavoor, Kerala, India

[2]Assistant Professor, Department of Computer Science and Engineering, KMP College of Engineering, Perumbavoor, Kerala, India

[3]Head, Department of Computer Science and Engineering, KMP College of Engineering, Perumbavoor, Kerala, India

Abstract: ***We implement an efficient compression of strongly encrypted images using error prediction, AES encryption and Run Length Encoding. In normal cases image encryption has to be done before the image compression. In such cases will leads to a problem of how to design a couple of algorithm for image encryption and compression while preserving the quality and security of the image. In order to provide most priority to the security of the image in the Encryption-then-Compression (ETC) system, we implement an efficient error prediction mechanism before the strong encryption schema. Block cipher mode of encryption is used to enhance the security of the image. We also demonstrate run length coding based compression scenario to improve the efficiency of image compression. While testing, we are able to compress an encrypted 640000 bit image to 278656 bits and the results shown that the proposed system is more promising and convenient.***

Keywords: Encryption-then-Compression, Prediction Error, Run Length Coding, Advanced Encryption Standard.

1. Introduction

Consider a situation in which a content owner Alice wants to transmit an image *I* to a recipient Bob in secure and efficient manner, via an untrusted channel provider Charlie. Normally this Alice first compresses *I* into *B*, and then encrypts *B* into *Ie* using an encryption function *EK (B)*, where *K* denotes the secret key as shown in Fig 1. The encrypted data *Ie* is then pass to Charlie, who simply forwards it to Bob. Upon receiving *Ie*, Bob sequentially performs decryption and decompression to get a reconstructed image as I.

Here the content owner Alice wants to do both the compression and encryption mechanism and Bob wants to do both decryption and decompression mechanism which leads both of them has to do those pair of operations in their system side with very reasonable resources. Also Alice and Bob will be responsible for both operation and the complexity of the system will increase. When channel provider Charlie performs common compression and decompression at channel ends, Alice and Bob will concentrate in the encryption and decryption mechanism respectively which will leads more security to the image.

The proposed ETC system is in principle different, and may helpfully complement for the mobile devices. The proposed system is shown in Fig 2. In this system whenever Alice wants to transmit an image I, she encrypts the image using AES algorithm in prediction error domain and transmit through the untrusted channel provider Charlie. The untrusted channel provider Charlie will compress the encrypted image using Run Length Coding before transmit and forward to Bob and decompress the encrypted compressed image before delivering to Bob. Bob needs only to decrypt the encrypted image to get the original image I. Because of this proposed method both Alice and Bob work has reduced with improved performance.

A big challenge within such Encryption-then-Compression (ETC) framework is that compression has to be conducted in the encrypted domain, as Charlie does not access to the secret key *K*.

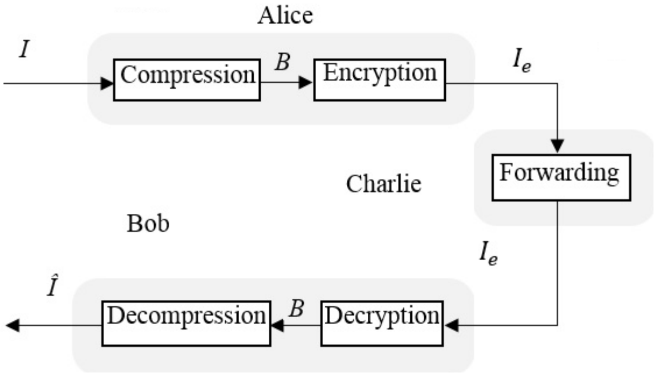


Figure 1: Traditional Compression-then Encryption system

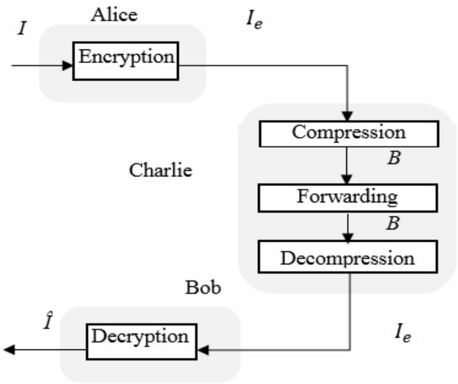


Figure 2: Proposed Encryption-then-Compression system

2. Related Works

Jiantao Zhou et al. [1] “Designing an Efficient Image Encryption-Then-Compression System via Prediction Error Clustering and Random Permutation” suggested a highly efficient image encryption-then-compression (ETC) system. Specifically, they propose a permutation-based image encryption approach conducted over the prediction error domain. A context-adaptive arithmetic coding (AC) is then shown to be able to efficiently compress the encrypted data.

Daniel Schonberg et al. [2] present and analyze an incremental scheme based on exponentially increasing block lengths that is designed to balance the resolution rate of parameter estimation with the redundancy rate of communication.. They build on this work by considering systems that must operate without knowledge of the underlying source statistics, and sources with memory (in particular two-state hidden Markov processes). They presented a protocol for the blind transmission of an encrypted source using a minimal number of bits. The scheme presented is proven to achieve redundancy proportional to the inverse of the square root of the block length, and requires minimal feedback. They investigated the problem of compressing encrypted images when the underlying source statistics is unknown and the sources have memory.

Daniel Schonberg et al. [3] then use a 2-D source model, and develop a scheme to compress encrypted images based on LDPC codes. Coding schemes for secure and efficient communication over noiseless public channels traditionally compress and then encrypt the source data. In some cases reversing the ordering of compression and encryption would be useful, e.g., in enabling the efficient distribution of protected media content. Indeed, not only is it possible to reverse the order, but under some conditions neither security nor compression efficiency need be sacrificed.

Riccardo Lazzeretti et al. [4] investigated the possibility of compressing encrypted grey level and color images, by decomposing them into bit-planes. The feasibility of lossless compression of encrypted images has been recently demonstrated by relying on the analogy with source coding with side information at the decoder. However previous works only addressed the compression of bi-level images, namely sparse black and white images, with asymmetric probabilities of black and white pixels.

Wei Liu et al. [5] proposed a resolution progressive compression scheme which compresses an encrypted image progressively in resolution, such that the decoder can observe a low-resolution version of the image, study local statistics based on it, and use the statistics to decode the next resolution level. Lossless compression of encrypted sources can be achieved through Slepian-Wolf coding.

Xinpeng Zhang et al. [6] proposed a novel scheme of scalable coding for encrypted images. In the encryption phase the original pixel values are masked by a modulo-256 addition with pseudorandom numbers that are derived from a secret key. After decomposing the encrypted data into a downsampled subimage and several data sets with a multiple-resolution construction, an encoder quantizes the subimage and the hadamard coefficients of each data set to reduce the data amount.

Daniel Schonberg et al. [7] presents a framework for compressing encrypted media such as images and videos. Encryption masks the source, rendering traditional compression algorithms ineffective. By conceiving of the problem as one of distributed source coding, it has been shown in previous work that encrypted data is as compressible as unencrypted data.

Qiuming Yao et al. [8] proposes a novel multi-resolution based approach which makes it possible not only to effectively derive the temporal side information from previous frames, but also to generate the spatial side information by having partial access to the current frame. Compression of encrypted data can be viewed as a special case of distributed source coding and can be achieved by applying Slepian-Wolf Coding. However, how to compress the encrypted video efficiently remains a challenging problem especially for those videos with irregular high motion.

Xiaolin Wu et al. [9] proposes an inter-band version of CALIC (Context-based, Adaptive, Lossless Image Codec) which represents one of the best performing, practical and general purpose lossless image coding techniques known today. Inter-band coding techniques are needed for effective compression of multi-spectral images like color images and remotely sensed images.

Marcelo J. Weinberger et al. [10] proposed LOCO-I (LOw COmplexity LOssless COmpression for Images) is the algorithm at the core of the new ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS. It is conceived as a “low complexity projection” of the universal context modeling paradigm, matching its modeling unit to a simple coding unit.

3. Proposed System

Here we present details of four key components of our ETC system by which image encryption conducted by Alice, compression and decompression conducted by Charlie and decryption conducted by Bob.

3.1 Encryption by Error Prediction and AES

We performing image encryption via by calculating error prediction and AES mechanisms. From the perspective of the whole ETC system, the design of the encryption algorithm should simultaneously consider the security and the ease of compressing the encrypted data. To this end, they propose an image encryption scheme operated over the prediction error domain. The schematic diagram of this image encryption method is depicted in Fig. 3

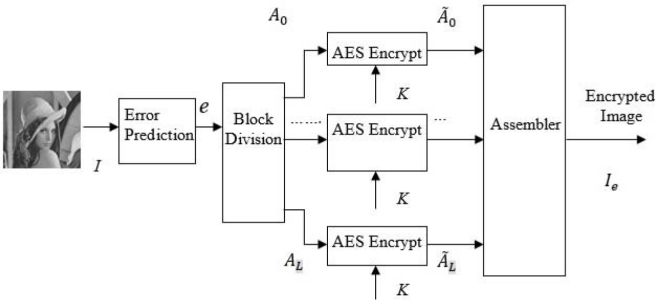


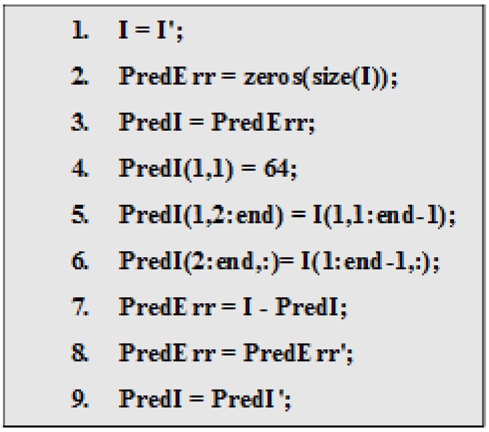
Figure 3: Schematic diagram of image encryption

For each pixel $I_{i,j}$ of the image I to be encrypted, a prediction $\bar{I}_{i,j}$ is first made by using an image predictor. Consequently, the prediction error associated with $I_{i,j}$ can be computed by

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

Prediction error values are in the range [-255,255] and Prediction Image values are in range [0,255]. The code snippet of error calculation is shown in fig 4.

Our proposed image encryption algorithm is performed over the domain of the prediction error $e_{i,j}$. Instead of treating all the prediction errors as a whole, we divide the prediction errors into L blocks based on the concept of AES Encryption (Block Cipher Mode Encryption)[11][12]. The value of L is selected based on the size of the image. Let image size is 100x100 pixel, we divide the image into blocks of 4x4 matrices, then total no of block(L) become 625 if the size of key is 128 bit(16 Byte). Each block is encrypted independently and assembled into encrypted format.



```
I = I';
PredErr = zeros(size(I));
PredI = PredErr;
PredI(1,1) = 64;
PredI(1,2:end) = I(1,1:end-1);
PredI(2:end,:)= I(1:end-1,:);
PredErr = I - PredI;
PredErr = PredErr';
PredI = PredI';
```

Figure 4: Code snippet

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

Step 1: Compute all the mapped prediction errors $e_{i,j}$ of the whole image I.

Step 2: Divide all prediction errors in to L blocks based on the size of image and key size.

Step 3: Perform Block cipher mode AES encryption to each block.

Step 4: The assembler concatenates all the encrypted Blocks $\tilde{A}_L$ and generate final encrypted image. As the number of prediction errors equals that of the pixels, the file size before and after the encryption preserves.

$$I_e = \tilde{\mathbf{A}}_0\tilde{\mathbf{A}}_1 \ldots \tilde{\mathbf{A}}_{L-1} \qquad (1)$$

Step 5: Pass I_e to Charlie.

3.2. Lossless Compression of Encrypted Image Via RLC

Data compression, or called source coding, involves encoding information using fewer bits than the original representation. Data compression is useful since it helps reduce resources usage, which often has some constraints, such as data storage space or transmission capacity. Therefore, many compression techniques are proposed, and one of them is Run Length Coding (RLC). The compression of the encrypted file I_e needs to be performed in the encrypted domain, as Charlie does not have access to the secret key K. In Fig. 5, they show the diagram of lossless compression of I_e.

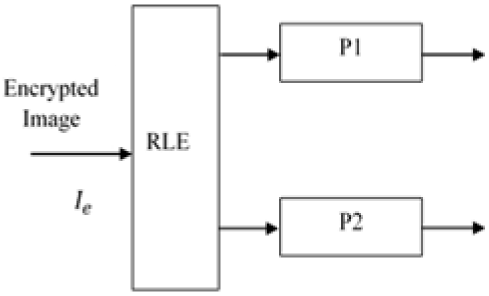


Figure 5: Lossless compression of encrypted image

The encrypted blocks is passed through Run Length Encoding (RLE) function and compressed values are stored in two arrays namely P1 and P2 and the size of these arrays are based on the size of the encrypted image.

3.3. Lossless Decompression

Upon receiving the compressed arrays P1 and P2 of encrypted image, Bob aims to recover the original image I . The schematic diagram demonstrating the procedure of decompression is provided in Fig. 6.

The compressed arrays P1 and P2 is passed through Run Length Decoding (RLE) function and corresponding decompressed values are stored in a 2D array which is the encrypted image.

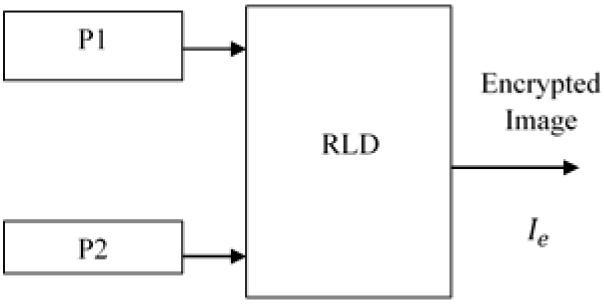


Figure 6: Lossless decompression

3.4. Decryption and Image Reconstruction

Upon receiving Decompressed 2D array is again divided into L blocks as done earlier procedure and AES decryption function is applied to each block using secret key K by Bob. From that the reconstructed pixel value can then be computed by

$$I_{i,j} = \bar{I}_{i,j} + e_{i,j} \qquad (3)$$

These are shown in Fig 7, and error free decompression and decryption is achieved.

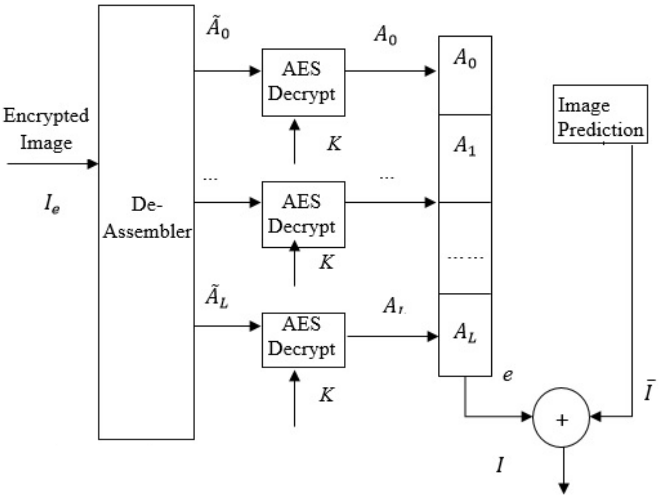


Figure 7: Successful decryption and image reconstruction

4. Result Analysis

In this section, the security of our proposed image encryption and the compression performance on the encrypted data are evaluated experimentally. Fig. 8 illustrates the Lena and Baboon images, together with their encrypted versions, from which we can see that our encryption approach is effective in destroying the semantic meaning of the images. In addition, it can be observed that the encrypted Baboon image looks different than the encrypted Lena image. This is because the Baboon image contains large portion of texture regions that are difficult to compress, resulting in more large-valued prediction errors.

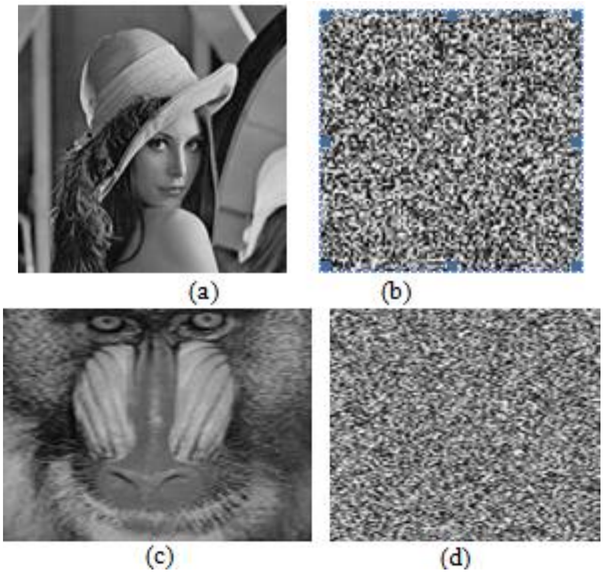


Figure 8: (a) Original Lena; (b) Encrypted Lena; (c) Original Baboon; (d) Encrypted Baboon

Ten images of size 100 × 100 shown in Fig.9 are used as the test set. In Fig. 10, we give the PSNR results of the original image and encrypted image, where x-axis represents the image ID. It can be observed that all the PSNR values are around 39 dB, which is too low to convey any useful semantic information.

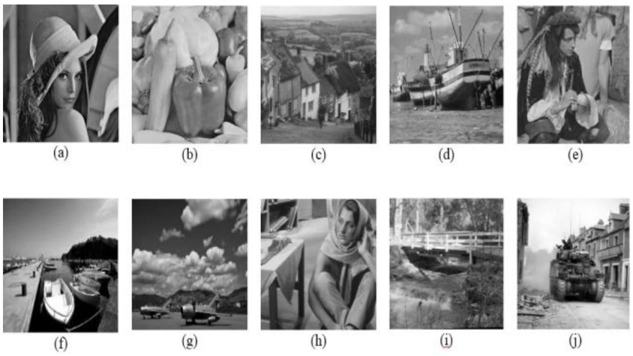


Figure 9: Ten test images

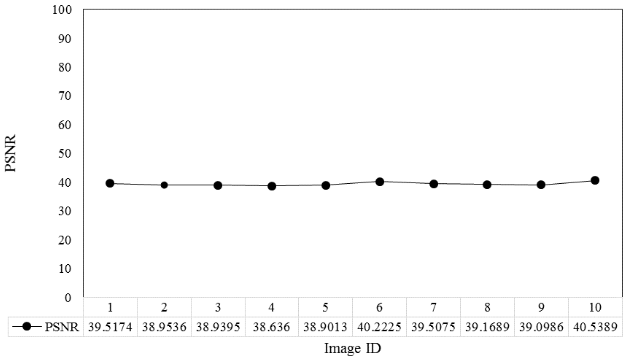


Figure 10: PSNR results of original and encrypted images

An attacker may attempt to decode the encrypted file I_e directly. For correct decoding of $I_{i,\,j}$, the attacker needs to get both the prediction error e and the associated predicted value $\bar{I}_{i,j}$. Next he needs to know AES parameters, which is not possible normally.

Fig.11 shows the PSNR results of the decrypted image and reconstructed image and it can also be observed that all the PSNR values are around 39 dB.

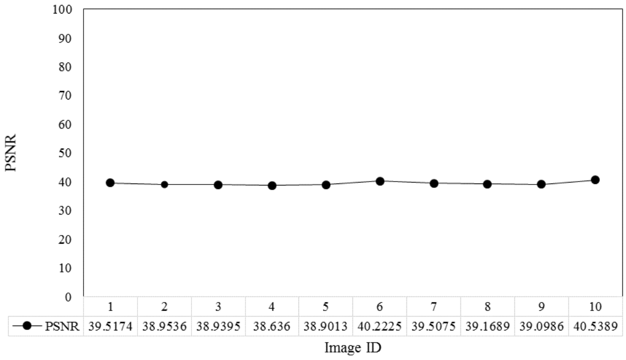


Figure 11: PSNR results of decrypted image and reconstructed image

When we try to calculate the PSNR values of original image and reconstructed image, it can be observed as infinity which means that pixel by pixel values are successfully reconstructed by Bob which is shown in Table 1

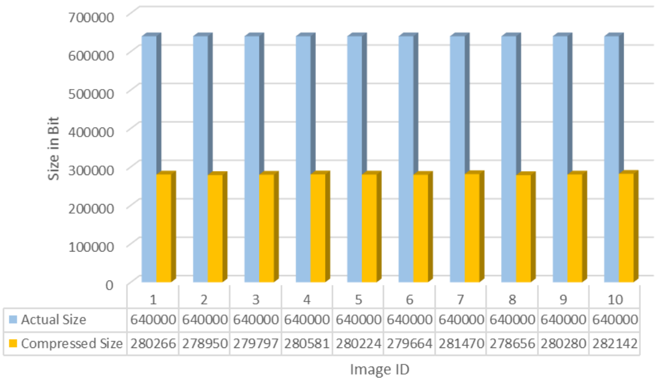
After the encryption process when we analyzed the compression performance of encrypted data through RLC mechanism, which is observed that around 56% of encrypted data is compressed. Fig 12 and Table 2 shows the compression performance of test set images.

Table1: PSNR of original image and reconstructed image

Image ID	*PSNR(OI,RI)*
1	∞
2	∞
3	∞
4	∞
5	∞
6	∞
7	∞
8	∞
9	∞
10	∞

Table 2: Compression rate

ID	*Original Size(Bit)*	*Encrypted Size(Bit)*	*Compressed Size(Bit)*	*Compressed By (%)*	*Compression Rate*
1	640000	640000	280266	56.21	0.4379
2	640000	640000	278950	56.41	0.4358
3	640000	640000	279797	56.28	0.4371
4	640000	640000	280581	56.16	0.4384
5	640000	640000	280224	56.21	0.4378
6	640000	640000	279664	56.30	0.4369
7	640000	640000	281470	56.02	0.4397
8	640000	640000	278656	56.46	0.4354
9	640000	640000	280280	56.21	0.4379
10	640000	640000	282142	55.92	0.4408



	1	2	3	4	5	6	7	8	9	10
■ Actual Size	640000	640000	640000	640000	640000	640000	640000	640000	640000	640000
■ Compressed Size	280266	278950	279797	280581	280224	279664	281470	278656	280280	282142

Figure 12: Compression performance

5. Conclusion

This review make an effort towards comparing some mechanisms and effective tools that can be employed for the compression of encrypted images. Just like any other mechanisms, the systems developed by each researcher is not fully developed and fully error-proof. Compared with previous schemes this paper trying to implement design, development and study of efficient compression of strongly encrypted images. By using strong encryption AES mechanism in error prediction domain security of image is to be improved and because of using simple compression mechanism Run Length Coding, above 50% image compression percentage is achieved.

References

[1] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang "Designing an Efficient Image Encryption-Then-Compression System via Prediction Error Clustering and Random Permutation" in IEEE Trans. Inf. Forensics Security, vol. 9, no. 1, pp. 39–50, Jan. 2014.)

[2] 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.

[3] 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.

[4] 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.

[5] W. Liu, W. J. Zeng, L. Dong, and Q. M. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Imag. Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.

[6] 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

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

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

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

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

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

[12] W. Stallings, Cryptography and Network Security Principles and Practices-Fourth Edition. Prentice Hall, 2005

Author Profile

Stebin Sunny received the B.Tech degree in Information Technology from University Of Calicut in 2011 and currently pursuing final year M. Tech degree in Computer Science and Engineering with specialization in Cyber Security from KMP College of Engineering, Perumbavoor, Kerala, India.

Chinchu Jacob received B.Tech in Computer Science and Engineering and M.Tech in Computer Science and Engineering with specialization in Data Security from Cochin University of Science and Technology, Cochin, Kerala, India in 2010 and 2012 and currently working as assistant professor in KMP College of Engineering Perumbavoor, Kerala, India in Computer Science and Engineering Department.

T.Justin Jose received his B.E and M.Tech degrees in Computer Science and Engineering from Manonmanium Sundaranar University, Tirunelveli in the year 1998 and 2004 respectively. He is currently working as a Professor in the Computer Science and Engineering Department of KMP College of Engineering, Perumbavoor, Kerala, India. He is currently working in the area of image processing and Genetic Algorithms. He is a reviewer of few Indian and International journals. He has authored and co-authored about 15 technical journal papers and conferences.