Simultaneous Encryption and Compression of Digital Images Based on Secure-JPEG Encoding

Saqib Maqbool[1], Nisar Ahmad[1], Aslam Muhammad[1(✉)], and A.M. Martinez Enriquez[2]

[1] Department of Computer Science and Engineering, University of Engineering and Technology Lahore, Lahore, Pakistan
saqib_maqbool2003@yahoo.com, nisarahmedrana@yahoo.com, maslam@uet.edu.pk
[2] Department of CS, CINVESTAV-IPN, Mexico, D.F., Mexico
ammartin@cinvestav.mx

Abstract. Confidentiality and efficient bandwidth utilization requires compression and encryption of digital images. Both of these parameters are necessary for most communication systems. Encryption and compression done separately sometimes result in decreased performance or reduced reconstruction quality. The paper presents a simultaneous encryption and compression scheme for digital images. It modifies standard JPEG compression in a way to encrypt data during compression. The encryption steps are based on a JPEG compressible image encryption scheme. The proposed Secure-JPEG algorithm provides the benefits of encryption along with the ability to provide lossless compression. This scheme results in improved performance and better reconstruction quality than existing schemes utilizing the similar approach.

Keywords: Multimedia security · Encryption · Compression · Simultaneous encryption and compression · JPEG

1 Introduction

The use of digital images and video applications has increased significantly due to availability of inexpensive capturing devices. Data compression is always desired due to limited storage or communication bandwidth. Wireless communication in particular requires low bit rate compression due to power and bandwidth constraints [1]. In contrast, encryption is required to protect the information for illicit use especially in wireless or public networks. Conventionally, compression is performed to reduce the data size and then it is encrypted using a suitable encryption algorithm. The decoder must perform this process in reverse order to obtain the actual data. The time consumed during encryption and decryption is a key tailback in real-time image communication and processing. Moreover, the processing time of compression and decompression of data pose another bottleneck. The computational cost incurred by encryption and decryption of data make it infeasible for many practical applications such as real-time embedded systems [2].

© Springer International Publishing Switzerland 2016
J.F. Martínez-Trinidad et al. (Eds.): MCPR 2016, LNCS 9703, pp. 145–154, 2016.
DOI: 10.1007/978-3-319-39393-3_15

The security of digital images is becoming increasingly important due to rapid evolution of Internet and digital technologies. Moreover, encryption of digital images is different from text data as it possesses high spatial correlation and redundancy. Therefore, traditional encryption schemes such as AES [3], and RSA [4] are not highly appropriate for image or other multimedia data. Degradation of visual content without achieving complete randomness can suffice the purpose of encryption for digital images. Many researchers have proposed encryption algorithms specifically for digital images [5–7]. Moreover, bulk-capacity and high-redundancy of image data require compression along with encryption.

The motivation of this research is acquired from the research of [8] which works on quantized Discrete Cosine Transform (DCT) coefficients which are produced during JPEG compression. In our research, we modify the DCT coefficients during JPEG compression process to produce an encrypted and compressed image which has visual quality nearly same as the original image and encrypted as well. Our Secure-JPEG technique has two benefits. One is that it allows achieving compression and encryption process in a single step. And the second is that it improves the image quality when decompressed and decrypted in the reverse process. Our results have been compared with standard JPEG algorithm and encryption scheme by [9] using Mean Squared Error (MSE), Peak-Signal to Noise Ratio (PSNR), Normalized Correction (NC), and Structural Similarity Index Matrix (SSIM).

The rest of the paper is organized as follows; Sect. 2 contains literature review of some contemporary encryption algorithms. Section 3 provides our proposed algorithm. Section 4 provides the results and discussions whereas the research is concluded in Sect. 5.

2 Literature Review

Several researchers [10–13] have focused only on compression of digital images where security aspects are not considered. Chaotic image encryption algorithms [3–7, 10] are gaining attention due to their inherent sensitivity to initial conditions, pseudo-randomness and ergodicity. They have good confusion and diffusion properties which satisfies the cryptographic requirements. However, these systems encompass security of images only and do not consider compression aspects. Consequently, the need for simultaneous encryption and compression of digital images is a necessary requirement. Several researchers follow this approach in their research and given attention to confidentiality along with data reduction [14–17].

There are two approaches followed by researchers while achieving encryption along with compression of digital images. In the first approach, encryption and compression are done at two different stages [18–20]. These two stages are completely independent of each other and sometime take more time while processing the image at two separate stages. In this scenario the adversary has to focus on cryptanalysis only to break the security of the algorithm without giving any consideration to compression algorithm. In the second approach, compression and encryption of digital images are performed simultaneously in a single stage [16, 21–23]. This combined encryption and compression

result in reduced computational time and more security as the adversary has to consider the compression as well as the encryption algorithm while performing the cryptanalysis of the encryption algorithm.

[8] has presented a shared-key encryption algorithm for JPEG color images. The algorithm operates on DCT coefficients during quantization step. Their process is based on optical encryption by producing two random like shares of 8×8 blocks. Each 8×8 block is passed through the encryption process and random like shares are produced which are then fed to JPEG for further processing. The produced shares are of the same size as the original block so it does not result in increased size. Moreover, the share generation is lossless and the encryption does not add further error in the compressed image. During decryption process these shares are combined to obtain the original DCT-coefficients [8] and the original image is reproduced. They have also provided three extensions to their proposed schemes, one is intended to produce random looking pixel distribution, the second to produce asymmetric shares and the third one is to generate more than two shares. These additional extensions have their own limitations over the original proposed scheme.

In [9] a color image encryption scheme based on orthogonal basis vectors is proposed. The encryption scheme work in two phases: the first one divides the image into 8×8 blocks and then blocks are scrambled by means of Mersenne Twister [24]. This scrambled image is transformed to frequency domain by using DCT. In second phase, a random-number-matrix of the image size is generated using Mersenne Twister. Their proposed algorithms have demonstrated reasonable security but there was spatial correlation in horizontal direction which was explained in terms of orthogonal matrix. Moreover, their proposed scheme introduces intensity change due to grayscale stretching for several times. Although, the scheme was compression friendly and has shown significant compression ratio along with resistance to channel noise the scheme introduced redundant computation by performing DCT during encryption and then during JPEG compression. Moreover, encryption then compression introduces more quantization error as compared to performing the two steps in a single stage. So, it is always preferable to introduce less error by simultaneous encryption and compression.

3 Secure-JPEG

In our proposed Secure-JPEG scheme, a simultaneous compression and encryption algorithm is designed. The algorithm uses the JPEG compression algorithm and introduces encryption during the DCT quantization step. This encrypted sequence is further processed with JPEG compression steps of quantization and entropy coding to obtain an encrypted and compressed image. The security of encryption steps is demonstrated in [9] which is quite reasonable. However, the performance of that algorithm in terms of image quality was not good due to stretching of pixel values to complete grayscale range. The proposed methodology, therefore, provides all the security benefits of [9] algorithm along with its compression and also results in improvement of reproduced image quality.

Figure 1 provides the basic working of our Secure-JPEG scheme. Input plaintext image is fed to the Secure-JPEG algorithm which compresses and encodes the image based on secret key. The protected image data is transmitted on an insecure wireless channel where it is susceptible to different type of attacks. The decoder performs the decoding and finally outputs the decompressed plain image.

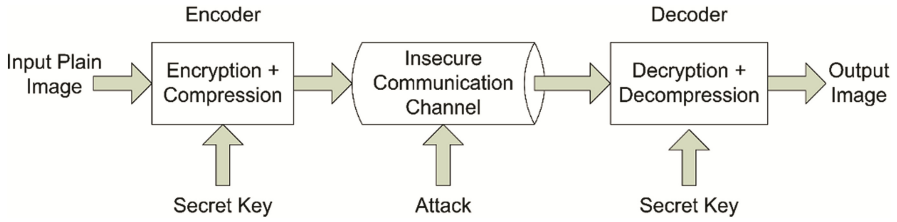


Fig. 1. Secure-JPEG

The lossy part of JPEG compression scheme uses DCT. The input plain image is separated into 8×8 non-overlapping blocks $(\beta_1, \beta_2, \cdots \beta_N)$, where N is equal to the total number of blocks. Zero-padding is used to convert the image matrix into multiple of 8. These blocks are scrambled based on permuted sequence ρ obtained from Pseudo-Random Number Generator (PRNG). We have used Mersenne Twister for PRNG but any cryptographically secure PRNG can be used for this purpose. These 8×8 pixels blocks are transformed using DCT into $(\Psi_1, \Psi_2, \ldots \Psi_N)$. Random matrices $(\gamma_1, \gamma_2, \ldots \gamma_N)$ of 8×8 are generated using the same PRNG separately for each DCT transformed image blocks. These randomly generated matrices are decomposed by SVD into U_i, Σ_i & V_i. The left singular vectors U_i are multiplied with DCT transformed image blocks to obtain encrypted matrix $(€_1, €_2, \cdots €_N) = (\psi_1 \times U_1, \psi_{22} \times U_2, \ldots \psi_N \times U_N)$.

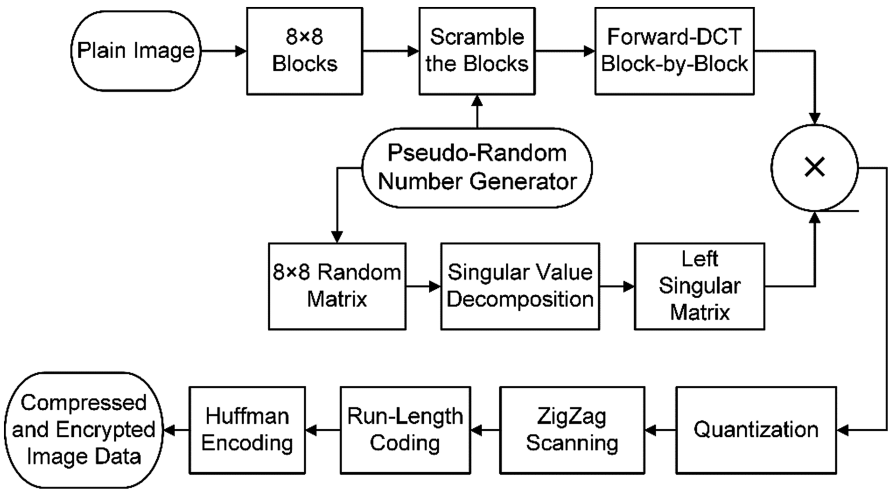


Fig. 2. Encryption process according to Secure-JPEG

This encrypted matrix is then quantized according to JPEG (see Fig. 2). Then run-length coding is performed by reading the 64 elements in zigzag scanning sequence. Then, variable length coding is performed using Huffman's algorithm [25] and compressed image data is obtained for transmission or storage.

The encryption key can be shared by using Diffie–Hellman key exchange or any other suitable algorithm. The decoding algorithm of the proposed scheme follows the same steps in reverse order. There could be two scenarios at decoding end which are provided below. In Scenario-I, the user treats the encoded image data as a JPEG compressed image and performs the decoding according to JPEG algorithm. The algorithm follows Huffman decoding of the compressed data which decodes the variable length code. Then, inverse run-length coding converts this data to its actual sequence. This data is then transformed from a vector sequence to 8×8 blocks. De-quantization retrieves the matrix values before quantization with error introduced by lossy compression factor. The image is formed after backward DCT transformation and combining the 8×8 blocks. The decoded image in this scenario represents the cipher image which is of no use to the intruder if he does not possess the knowledge of secret key and the encryption algorithm. The block diagram of this Scenario-I is shown in Fig. 3.

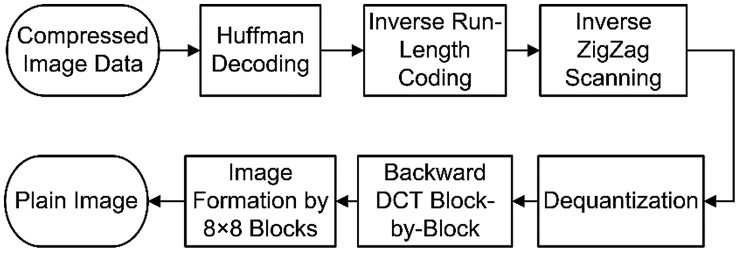


Fig. 3. Scenario-I; decoding according to standard JPEG

In scenario-II; the user possesses the information of the encryption algorithm and attempts to decode the image according to the algorithm. The compressed image data is decoded from variable length coded data using Huffman decoding. This decoded data is then transformed through inverse run-length coding to obtain the original data sequence. This sequence is transformed into 8×8 block through inverse Zigzag Scanning. De-quantization is done to retrieve the data before quantization with an error introduced by lossy compression. Whereas, 8×8 random matrices $\{\gamma_1, \gamma_2, \ldots \gamma_N\}$ are generated from the PRNG according to the secret key. These randomly generated matrices are decomposed using singular value decomposition to U_i, Σ_i & V_i. The left singular matrices are transposed to obtain its inverse as the left singular matrix is an orthogonal matrix and orthogonal matrix has inverse equal to its transpose. The transposed left singular vectors U_i^T are multiplied with de-quantized matrices to obtain decrypted matrices as $(€_1, €_2, \cdots €_N) = (\Psi_1 \times U_1^T, \Psi_2 \times U_2^T, \ldots \Psi_N \times U_N^T)$. These decrypted matrices are backward transformed using DCT to obtain the spatial image. Whereas, permutation sequence is generated same as during encryption. This permutation sequence is used to obtain inverse permutation sequence to perform the inverse

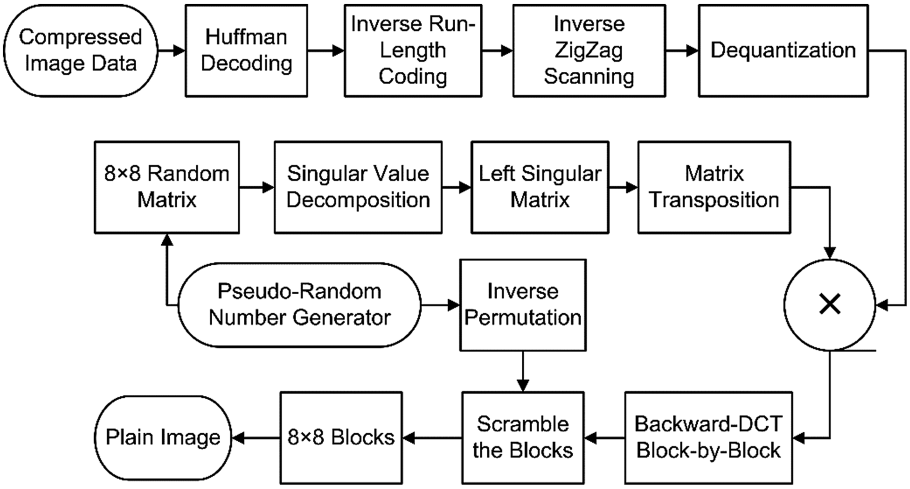
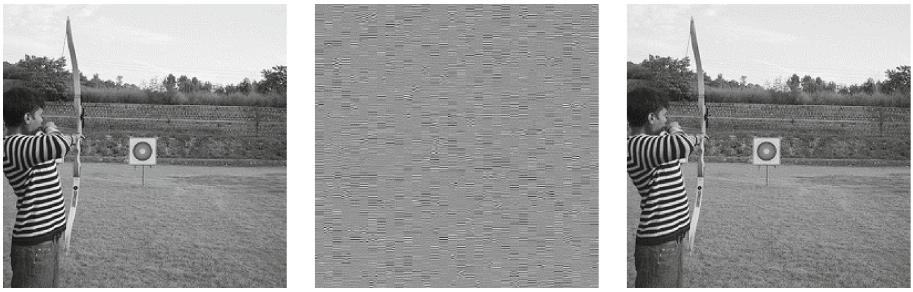


Fig. 4. Scenario-II; decoding according to Secure-JPEG

scrambling of blocks. These blocks are combined to form recovered image with dimensions of original image. The block diagram of this scenario-II is shown in Fig. 4.

4 Experimental Results and Discussions

The proposed scheme has been tested for a set of test images to obtain decoded images produced by standard JPEG and decoded by following Secure-JPEG. Figure 5(a) contains the test image used for encoding. Figure 5(b) contains the decoded image through standard JPEG which only decompress the image and display its output. This image has no visual information for the intruder. Figure 5(c) contains the decoded image through Secure-JPEG which is similar to the input plain image.



(a) (b) (c)

Fig. 5. (a) Original plain image-archer, (b) Decoded image through standard JPEG, (c) Decoded image through Secure-JPEG (Recovered deciphered image)

When the compression ratio is selected as 100 % at the time of encoding the decoded image is exact replica of the input image and only lossless compression is performed. When the image is provided less compression ratio such as for the images displayed in (c) where the compression ratio is selected as 90 % and the decoded image is visually similar but contains some error due to lossy compression during quantization. The change in decoded image due to quantization error is tested by employing Image quality metric [26].

4.1 Recovered Image Quality

Results in Tables 1, 2, and 3 provide the recovered image quality measured by MSE, PSNR, NC, and SSIM. These four metrics compare image quality based on different parameters except PSNR which is based on MSE. Any single matrix is not enough to measure the quality in all cases but they inspect the image similarity with different angles and a combined score of these parameters. Table 2 also provides image quality parameters with images compressed and decompressed with standard JPEG at 90 % quality factor. Table 3 provides the same results for scheme proposed by [9] and compressed by JPEG at 90 % quality factor. It can be seen from comparison of Table 1 and Table 3 that proposed scheme has better reconstruction quality then [9]. Moreover, the proposed scheme has better efficiency as it does not repeat the block formation, forward-DCT and quantization steps. Another significant improvement is reconstruction when it is encoded with 100 % quality factor which was not the same in [9] as they employed intensity scaling and de-scaling.

Table 1. Results of image quality metrics for proposed scheme at 90 % quality

Serial no.	Test images	MSE	PSNR	NC	SSIM
1	Archer	5.6577	37.8415	0.9880	0.9696
2	Cameraman	1.0807	39.8336	0.9981	0.9878
3	Flower	5.0009	38.1881	0.9774	0.9724
4	Glider	7.4606	38.8736	0.9872	0.9709
5	Kodim15	4.8454	38.7250	0.9993	0.9687
6	Lena	8.5593	37.0393	0.9717	0.9547
7	Mandrill	9.0010	35.9914	0.9805	0.9633
8	Peppers	3.0908	37.5445	0.9630	0.9476

Graphical forms are also used to provide comparison of quality parameters to see the reconstruction quality of three approaches. The plot of MSE for the 8 test images shows that the proposed scheme provides lower MSE than [9] followed by JPEG compression at 90 % quality factor. However, JPEG provide lower MSE then the proposed scheme for most of the cases. The graph for the comparison of PSNR shows that the proposed scheme has reasonably higher PSNR than [9].

The plot of the normalized correction shows that the similarity between original and reconstructed images. The proposed scheme has values near 0.98 which are good and

Table 2. Results of image quality metrics for Standard JPEG at 90 % quality

Serial no.	Test images	MSE	PSNR	NC	SSIM
1	Archer	5.3688	37.8027	1.0000	0.9737
2	Cameraman	0.8308	45.9048	0.9999	0.9966
3	Flower	3.9747	39.0987	0.9996	0.9773
4	Glider	4.4526	38.6481	1.0000	0.9750
5	Kodim15	4.9642	38.2276	0.9992	0.9571
6	Lena	4.0011	39.1199	0.9999	0.9672
7	Mandrill	7.3070	36.4888	1.0000	0.9767
8	Peppers	4.8888	38.2444	0.9998	0.9393

Table 3. Results of image quality metrics for [9] + JPEG compression at 90 % quality

Serial no.	Test images	MSE	PSNR	NC	SSIM
1	Archer	6.1243	31.8415	0.8880	0.8796
2	Cameraman	3.1656	32.8336	0.8981	0.8838
3	Flower	4.9373	30.5783	0.9089	0.8879
4	Glider	4.7440	31.0166	0.9186	0.8942
5	Kodim15	9.8341	29.1701	0.8386	0.8605
6	Lena	0.6267	28.7155	0.8937	0.8566
7	Mandrill	8.6986	29.1487	0.9555	0.7821
8	Peppers	5.0673	27.8639	0.8509	0.7450

can be regarded as perceptually similar and are much better than [9]. The graph for SSIM, which is a latest measure of image quality and it is claimed to be a full reference matrix, depicts improvements to MSE and PSNR values. The proposed scheme has higher value of SSIM than [9].

It is evident from the results of Tables 1, 2, and 3 that the proposed Secure-JPEG scheme has demonstrated reconstruction quality which is better than the previous method and the results are comparable to standard JPEG. The proposed Secure-JPEG scheme also provides improved performance due to reduced number of computational steps but the comparison of time consumed during computation on desktop computer cannot be provided as a reference as it is highly dependent on operating system, software environment, and other performance parameters. The Secure-JPEG scheme can be successfully used for encryption and compression of digital images. It also provides the ability to perform lossless operation as well as the lossy operation.

5 Conclusion

In this study, we intended to improve the performance and reconstruction quality of image encrypted by JPEG compressible image encryption scheme. As compression of digital images is vital to efficient bandwidth utilization but doing it separately may result in reduced performance. Moreover, in this scenario it was resulting in more error once while scaling of image pixels during encryption and the second while achieving lossy

compression. Also the scheme was introducing error due to scaling even when the image was being compressed using lossless compression. In our proposed scheme, the advantages achieved are three fold. Firstly, the algorithm is resulting in increased performance due to reduced number of computational steps. Secondly, the error due to compression and encryption was reduced. Thirdly, the scheme resulted in errorless image construction when the compression quality is kept 100 %. Consequently, the algorithm is an improved version with similar security characteristics as of the JPEG compressible image encryption scheme and it can be used for lossless encryption and compression and lossy encryption and compression. In future, the similar mechanism of encryption and compression will be implemented in wavelet domain for JPEG 2000. Moreover, the proposed algorithm will be modified and used for encryption and compression of audio, image sequence, and other multimedia data.

References

1. Lu, Q., et al.: Low-complexity and energy efficient image compression scheme for wireless sensor networks. Comput. Netw. **52**(13), 2594–2603 (2008)
2. Lian, S., Kanellopoulos, D., Ruffo, G.: Recent advances in multimedia information system security. Informatica **33**(1) (2009)
3. Selent, D.: Advanced encryption standard. Rivier Acad. J. **6**(2), 1–14 (2010)
4. Smith, D.R., Palmer, J.T.: Universal fixed messages and the Rivest-Shamir-Adleman cryptosystem. Mathematika **26**(01), 44–52 (1979)
5. Lian, S.: Multimedia Content Encryption: Techniques and Applications. CRC Press, Boca Raton (2008)
6. Furht, B., Socek, D., Eskicioglu, A.M.: Fundamentals of multimedia encryption techniques. In: Multimedia Security Handbook, vol. **4** (2004)
7. Van Droogenbroeck, M., Benedett, R.: Techniques for a selective encryption of uncompressed and compressed images. In: Advanced Concepts for Intelligent Vision Systems (ACIVS) (2002)
8. Sudharsanan, S.: Shared key encryption of JPEG color images. IEEE Trans. Consum. Electron. **51**(4), 1204–1211 (2005)
9. Ahmed, N., et al.: A novel image encryption scheme based on orthogonal vectors. Nucleus **52**(2), 71–78 (2015)
10. Grigoras, V., Grigoras, C.: Chaos encryption method based on large signal modulation in additive nonlinear discrete-time systems. In: Proceedings of the 5th WSEAS International Conference on Non-linear Analysis, Non-linear Systems and Chaos. World Scientific and Engineering Academy and Society (WSEAS) (2006)
11. Philip, M., Das, A.: Survey: image encryption using chaotic cryptography schemes. IJCA, 1–4 (2011). Special Issue on "Computational Science-New Dimensions and Perspectives" NCCSE
12. Wei-bin, C., Xin, Z.: Image encryption algorithm based on Henon chaotic system. In: International Conference on Image Analysis and Signal Processing, 2009. IASP 2009. IEEE (2009)
13. Shum, H.-Y., Kang, S.B., Chan, S.-C.: Survey of image-based representations and compression techniques. IEEE Trans. Circuits Syst. Video Technol. **13**(11), 1020–1037 (2003)

14. Hossein, M., Mahmud, S., Biswas, N.: Image compression and encryption. Int. J. ElectroComput. World Knowl. Interface **1**(3) (2011)
15. Zhou, N., et al.: Image compression and encryption scheme based on 2D compressive sensing and fractional Mellin transform. Opt. Commun. **343**, 10–21 (2015)
16. Alfalou, A., Brosseau, C., Abdallah, N.: Simultaneous compression and encryption of color video images. Opt. Commun. **338**, 371–379 (2015)
17. Tong, X.-J., et al.: A new algorithm of the combination of image compression and encryption technology based on cross chaotic map. Nonlinear Dyn. **72**(1–2), 229–241 (2013)
18. Zhou, J., et al.: Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation. IEEE Trans. Inf. Forensics Secur. **9**(1), 39–50 (2014)
19. Zhou, J., Liu, X., Au, O.C.: On the design of an efficient encryption-then-compression system. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (2013)
20. Bansal, R., Sharma, M.R.: Designing an Efficient Image Encryption-Compression System Using a New Haar Wavelet (2014)
21. Zhu, H., Zhao, C., Zhang, X.: A novel image encryption–compression scheme using hyper-chaos and Chinese remainder theorem. Sig. Process. Image Commun. **28**(6), 670–680 (2013)
22. Aldossari, M., Alfalou, A., Brosseau, C.: Simultaneous compression and encryption of closely resembling images: application to video sequences and polarimetric images. Opt. Express **22**(19), 22349–22368 (2014)
23. Zhou, N., et al.: Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Opt. & Laser Technol. **62**, 152–160 (2014)
24. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform Pseudo-random number generator. ACM Trans. Model. Comput. Simul. (TOMACS) **8**(1), 3–30 (1998)
25. Knuth, D.E.: Dynamic Huffman coding. J. Algorithms **6**(2), 163–180 (1985)
26. Naveed, A., et al.: Performance evaluation and watermark security assessment of digital watermarking techniques. Sci. Int. Lahore **27**(2), 6 (2015)