i-TSS: An Image Encryption Algorithm Based on Transposition, Shuffling and Substitution Using Randomly Generated Bitmap Image

Kanagaraj Narayanasamy(✉) and Padmapriya Arumugam

Department of Computer Science and Engineering, Alagappa University,
Karaikudi 630 003, Tamilnadu, India
kanagaraj.n.in@ieee.org, mailtopadhu@yahoo.co.in

Abstract. In the digitalized era, an enormous amount of digital images are being shared over the different networks and also available in different storage mediums. Internet users enjoy this convenient way of sharing images and at the meantime, they need to face the consequences like chosen plain-text, statistical, differential attacks, and brute-force attack. These attacks and noises create the need of enhancing the image information security. An image encryption algorithm needs to be robust. An image encryption algorithm (*i*-TSS) based on transposition, shuffling, and substitution is presented in this paper, that provides better security to the image. This algorithm is implemented using Java. By assessing the results of image quality metrics, this algorithm proves to be secured and robust against the external attacks.

1 Introduction

Cryptography is one of the best ways to communicate secretly even over the insecure channels [1]. Image encryption means converting the original image to disguised form, just like text encryption. AES, RSA and IDEA [3–5] were widely used text encryption algorithms. These text encryption algorithms can be modified to handle the image data, but the textual data differ from the image data. For instance, if the RGB color model based image's size is 512 × 512 then there would be 786432 numbers of data to be handled. This much of data can be handled by the algorithms which are developed particularly for the image or Multimedia data [2].

Usually, an image encryption system is made up of several components (Fig. 1). As illustrated in the figure, the image is encrypted using an algorithm and a respective key to produce the encrypted/disguised image. Similarly, the encrypted image is decrypted using the key to get the plain image.

According to the usage of the key, the algorithm technique differs. In Symmetric technique, encryption key and decryption key are same; and in Asymmetric technique, encryption key and decryption key differ. It is possible for an adversary to obtain the original image without the respective key by means of cryptanalysis. The proposed algorithm comes under the Symmetric key cryptographic technique.

N. Bjørner et al. (Eds.): ICDCIT 2016, LNCS 9581, pp. 148–156, 2016.
DOI: 10.1007/978-3-319-28034-9_20

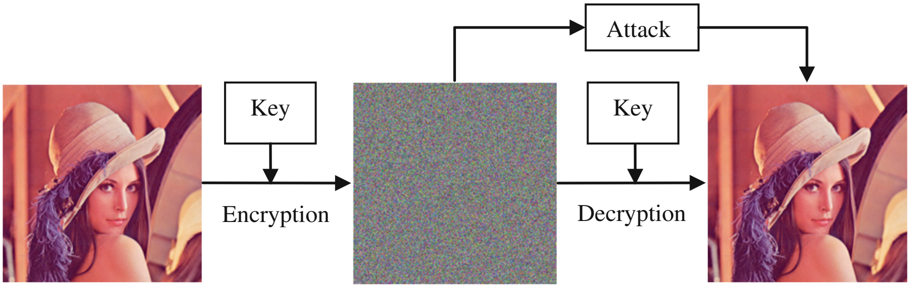


Fig. 1. Process of image encryption and decryption

The security of the image is the primary concern in this paper. The traditional image encryption algorithms such as AES, DES, RSA and the family of ECC based algorithms may not be the best one to choose for image encryption, specifically for speed and applicability in real-time applications. In recent years, several image encryption algorithms [6–12] have proposed. Zang and Liu [6] proposed an image encryption methodology based on permutation – diffusion based image encryption system. The position of the image pixels are shuffled to obtain high plain image sensitivity. The key and plain image decides the key stream in the diffusion step. Lin and Wang [7] proposed an image encryption based on chaos with the Piece Wise Linear (PWL) memristor in Chua's circuit. Diaconu et al. [8], Dascalescu and Boriga [9], and Dalhoum et al. [10] have proposed various image encryption algorithms based on Image scrambling technique. Askar et al. [11] and Zhang et al. [12] have proposed image encryption algorithms based on the chaotic economic map and DNA encoding respectively. Both these algorithms use a chaotic map, but in the [12] logistic chaotic map is used in the shuffling phase, and further DNA coding, and Chebyshev's chaotic map is used in algorithm in different phases. The proposed algorithm is seriously tested using different images and compared with other research works [7, 11, 12] to prove that the proposed algorithm is more effective in resisting attacks.

The rest of the paper is organized as follows. In the second section, the proposed algorithm is explained; in the next section, the experimental study is done; in the fourth section, the results and discussion are done to justify the efficiency of the algorithm; and the last section deals about the conclusion of the proposed algorithm.

2 Proposed Algorithm

The proposed image encryption algorithm *i*-TSS, basically works based on the Transposition, Shuffling, and Substitution processes. Transposition is the process of interchanging pixel's position. During Shuffling, the pixels are scrambled in order to confuse the adversaries. Substitution process is done with the help of a Randomly Generated bitmap image (RGbmp). RGbmp is created with the randomly generated values according to the size of the input image. Those values are assigned to Red, Green, and Blue components to form RGbmp. The transposed and shuffled image is XOR-ed with the RGbmp to obtain the better encrypted image as shown in Fig. 2.

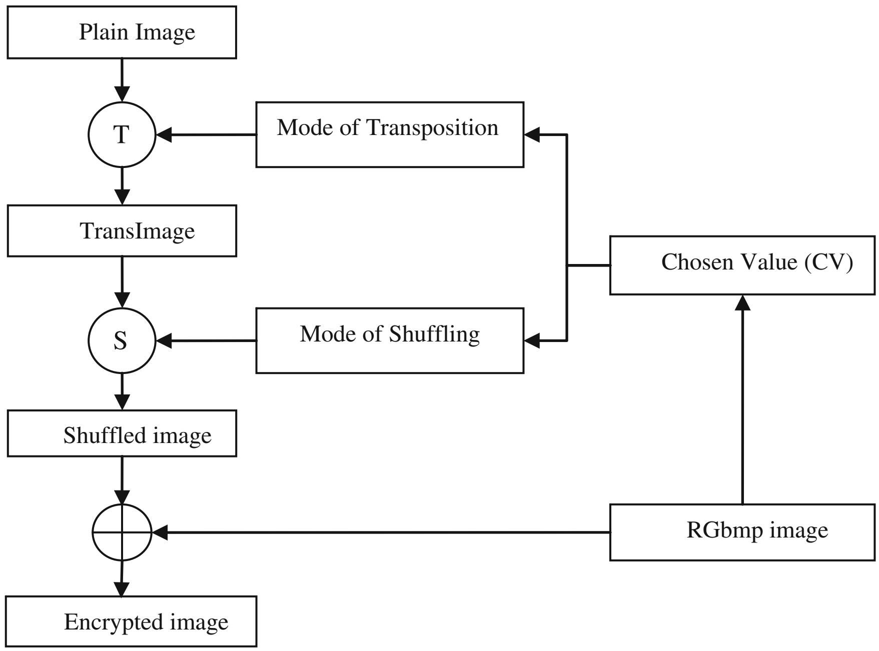


Fig. 2. Overall workflow of Proposed Algorithm (i-TSS)

2.1 Transposition Process

This is the first process in the algorithm. The transposition process starts from an initial position, most probably from the center of the image or nearby position from the center of the image. The initial position changes according to a 'Chosen Value (CV)'. This CV is chosen from the RGbmp with some simplified constraints. The pixels are taken in a spiral manner from the center of the image and repositioned from top to bottom. The CV also decides the number of transposition rounds. This transposition process will totally scramble the plain image.

2.2 Shuffling and Substitution Processes

In the second phase, shuffling the pixel's position is done. The transpositioned-image (TransImage) enters into this stage as input. The TransImage is shuffled either in even or odd order. The order of shuffling is determined based on the chosen value (CV). In the last phase, the Shuffled-image is XOR-ed with Randomly Generated bitmap image (RGbmp). The XOR operator has been used in this phase because it has distinctive properties when compared with other operators [13]. The Red, Green, and Blue values of the shuffled image are XOR-ed with Red, Green, and Blue values of the RGbmp. The resultant images possess better encryption.

3 Experimental Study

Experiments are done on the proposed algorithm (*i*-TSS) to find out the performance, and results are taken into account to assess the robustness and secrecy of the proposed algorithm. The images [14] with various resolutions are considered for the experimental study. The encryption and decryption process of the algorithm is explained in this section.

3.1 Encryption and Decryption Processes

Once the RGbmp is created, a value (CV) is chosen based on some conditions. It will be used to determine the mode of transposition and mode of shuffling. In the transposition process, the plain image is repositioned. The below Table 1 shows the usage of rounds in the process of transposition.

Table 1. Peak Signal-to-Noise Ratio (PSNR) values for a transposition image with various rounds

	Transposition (Round = 1)	Transposition (Round = 45)	Transposition (Round = 150)
PSNR (RGB)	28.5636	28.3612	28.3318

PSNR (RGB band) values for the various transposition images are grouped in the above table. The PSNR (RGB) values clearly reveal the essentiality of the rounds in transposition process. The next phase is shuffling the RGB values. The RGB values are loaded into a one dimensional array and get shuffled in either odd or even order. This single step of the shuffling process makes adversaries to be in the confused state. In the last phase, the randomly generated bitmap image is used to encrypt the shuffled image. The Red, Green and Blue values of the RGbmp are XOR-ed with the respective Red, Green and Blue values of the shuffled image to obtain the encrypted image. The RGbmp is sent as a matrix key file to the receiver end.

In the decryption phase, the RGbmp is recreated using the received matrix key file from the sender. The CV is retrieved from the RGbmp by the same condition used as in the process of encryption. The processes in the encryption work are reversed to get the original image. In the first phase, the RGbmp is used to decrypt the encrypted image using XOR operator. By using the CV, the mode of shuffling and transposition can be found and those findings are used to do the reverse processes of shuffling and transposition.

3.2 Execution Time

The average encryption and decryption speed is determined using Lena image with different sizes varying from 64 × 64 to 1024 × 1024 pixels are 982 ms and 1003 ms respectively on personal computer equipped with an Intel processor (Core i3) with clock speed of 1.7 GHz, 2 GB of RAM and 520 GB of Hard disk capacity.

4 Results and Discussion

The techniques like PSNR [15, 16] and Mean Squared Error (MSE) [17] are the two commonly adopted image quality measures, in which PSNR is actually based on the value of MSE. These two measures are easy to use and have a convenient procedure to implement in mathematical aspect. Mostly all metrics compare the original image with the distorted image and provide the result about the difference or similarity. Later, these metrics are found to be insufficient to assess the quality. New metrics like Structural Similarity Index Measure (SSIM) [18], NPCR (Number of Pixels Change Ratio) [19], Unified Averaged Changed Intensity (UACI) [19] have introduced. PSNR, MSE, NPCR, and UACI metrics have taken into account to assess the encryption work.

4.1 Histogram

Color Histogram is a graphical representation of the colors distribution in an image. If the histogram exhibits the uniform distribution of colors, then the adversaries cannot get any information through statistical attacks [20]. Table 2 illustrates the histograms (RGB) plotted for various original images and histograms (Only Red band) plotted for respective encrypted images. The histograms and the mean values (Table 3) clearly show the uniformity in distribution of colors. This reveals that the encrypted images of *i*-TSS algorithm can easily withstand the statistical attacks.

Table 2. Histogram results for original and respective encrypted images

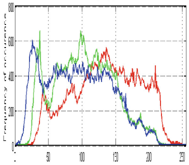
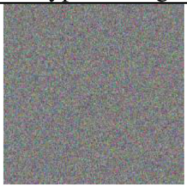
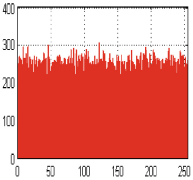
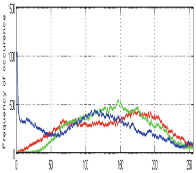
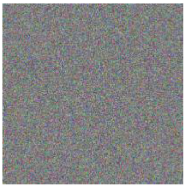
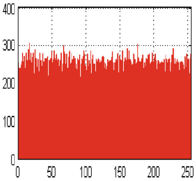
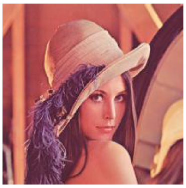
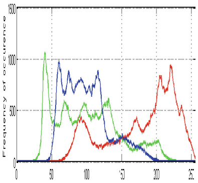
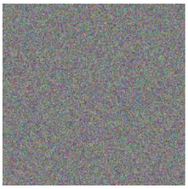
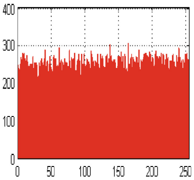
Original Image	Histogram	Encrypted Image	Histogram
 Barbara			
 Goldhill			
 *Lena (Lenna)*			

Table 3. Mean values for original and respective encrypted image

Image name	Mean value (original image)	Mean value (encrypted image)
Barbara	Red: 134.42	Red: 127.59
	Green: 102.04	Green: 127.39
	Blue: 93.41	Blue: 127.24
Goldhill	Red: 137.84	Red: 127.11
	Green: 138.82	Green: 127.36
	Blue: 109.45	Blue: 127.48
Lena	Red: 177.24	Red: 128.32
	Green: 127.92	Green: 127.92
	Blue: 99.17	Blue: 127.13

4.2 PSNR and MSE

PSNR is used to compute the ratio between the maximum possible value of a signal and the power of distorting noise that changes the representation quality [16]. PSNR is expressed in decibel (dB) unit. PSNR is based on the MSE value. MSE is used to calculate the amount of deviation between the original and its disguised image. If the comparing images are identical then the MSE value will be zero and PSNR would be infinity. If the PSNR value is less; then, the quality of the image encryption is better. PSNR and MSE values are calculated between different original image and its encrypted image; and it is tabled in Table 4.

Table 4. PSNR and MSE values between the original and respective encrypted image

Image name	PSNR	MSE
Barbara	20.5113	4.461
Goldhill	19.7036	5.161
Lenna	20.1267	4.822

4.3 NPCR and UACI

In differential attack, an attacker tries to find the plain image by changing a specific pixel in image and traces the differences in the respective output image. A general consideration for all encryption algorithms is that the encrypted image must be different from its original image. This deviation can be measured by means of two criteria: NPCR and UACI. The NPCR is used to measure the rate of change in an encrypted image when a bit is changed in the plain image. The UACI is used to calculate the unified average changing intensity between two encrypted images with a deviation in only one bit in respective plain images. In Table 5, NPCR and UACI values are tabled for different images. In Table 6, NPCR and UACI values of the proposed algorithm are compared with other research results and found to provide better results.

Table 5. Values of NPCR and UACI tests of encrypted images

Image name	NPCR (%)	UACI (%)
Lena	99.62	33.46
Goldhill	99.59	33.46
Barbara	99.60	33.46

Table 6. Comparative results of NPCR and UACI

Measure	[12]	[8]	[9]	[10]	Ours
NPCR (%)	99.7017	99.489	99.431	90.126	99.62
UACI (%)	28.7051	29.006	25.032	NaN	33.46

4.4 Entropy Measure

Entropy is a statistical measure that deals with the randomness of a bundle of data. Theoretically, if the entropy measure of the encrypted images nearly equal to 8 (sh); then the image encryption algorithm is highly robust against entropy attack. From Table 7, it is possible to know justify the leakage of information of the proposed algorithm against entropy attack to be negligible. Further, in Table 8 entropy measure of the *i*-TSS compared with other research results, this shows the betterment in handling entropy attack.

Table 7. Entropy values of the original images and respective encrypted image

Images	Entropy	
	Original image	Encrypted image
Lena	7.6553	7.9989
Goldhill	7.8644	7.9990
Barbara	7.6283	7.9990

Table 8. Comparative results of Entropy measure

Algorithm	Entropy (sh)
[7]	7.9890
[11]	7.9961
[12]	7.9854
Ours	7.9990

5 Conclusion

In this paper a new transposition, shuffling and substitution based cryptosystem has been introduced for image encryption. Improving the randomness in transposition process is the main advantage of the system. The results of a security analysis for three different images show the resistance to chosen plain-text, differential and statistical attacks on the encrypted images. In addition to these, a large key space makes the brute force attack to be impractical; the entropy measure of the proposed algorithm is close to the principle value 8 and the average execution time is 993 ms. Hence, it is suitable for the practical usage in real-time. In future, this work can be combined with chaotic functions to enhance the security of the encryption process.

References

1. Schneier, B.: Applied Cryptography, 2nd edn. Wiley, New York (1996). ISBN 0-471-11709-9
2. Soleymani, A., Ali, Z., Nordin, M.: A survey on principal aspects of secure image transmission. In: Proceedings of World Academy of Science, Engineering and Technology, pp. 247–254 (2012)
3. Stalling, W.: Cryptography and Network Security: Principles and Practice, 6th edn. Prentice Hall, Upper Saddle River (2013). ISBN 978-0133354690
4. Mollin, R.A.: An Introduction to Cryptography. CRC Press, Boca Raton (2006)
5. Vanstone, S.A., Menezes, A.J., Oorschot, P.C.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
6. Zhang, G., Liu, Q.: A novel image encryption method based on total shuffling scheme. Opt. Commun. **284**, 2775–2780 (2011)
7. Lin, Z., Wang, H.: Efficient image encryption using a chaos-based PWL memristor. IETE Tech. Rev. **27**, 318–325 (2010)
8. Diaconu, A.V., Costea, A., Costea, M.A.: Color image scrambling technique based on transposition of pixels between RGB channels using Knight' s moving rules and digital chaotic map. In: Mathematical Problems in Engineering (2014)
9. Dascalescu, A.C., Boriga, R.E.: A novel fast chaos-based algorithm for generating random permutations with high shift factor suitable for image scrambling. Nonlinear Dyn. **74**, 307–318 (2013)
10. Dalhoum, A.L.A., Mahafzah, B.A., Awwad, A.A., Aldamari, I., Ortega, A., Alfonseca, M.: Digital image scrambling using 2D cellular automata. IEEE Trans. Multimedia **19**, 28–36 (2012)

11. Askar, S.S., Karawia, A.A., Alshamrani, A.: Image encryption algorithm based on chaotic economic model. In: Mathematical Problems in Engineering (2015)
12. Zhang, J., Fang, D., Ren, H.: Image encryption algorithm based on DNA encoding and chaotic maps. In: Mathematical Problems in Engineering (2015)
13. http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/BitOp/xor.html. Accessed 10 October 2015
14. http://sipi.usc.edu/database/. Accessed 10 October 2015
15. http://in.mathworks.com/help/images/image-quality-metrics.html and http://in.mathworks.com/help/images/image-quality.html. Accessed 10 October 2015
16. Peak Signal-to-Noise Ratio as an Image Quality Metric: White paper published by National Instruments China (2013)
17. Wang, Z., Bovik, A.C.: Mean squared error: Love it or leave it?—A new look at signal fidelity measures. IEEE Signal Process. Mag. **26**(1), 98–117 (2009)
18. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)
19. Wu, Y., Noonan, J.P., Agaian, S.: NPCR and UACI randomness tests for image encryption. Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. (JSAT), April Edition, 31–38 (2011)
20. Shannon, C.: Communication theory of secrecy systems. Bell Syst. Techn. J. **28**, 656–7151 (1949)