

A Simple Steganography Algorithm Based on Lossless Compression Technique in WSN

R. Vijayarajeswari*, A. Rajivkannan, J. Santhosh

Department of Computer Science & Engineering, KSR College of Engineering, Tiruchengode, India

Email: *vijiraji0505@gmail.com

Received 19 March 2016; accepted 20 April 2016; published 9 June 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Image security has wide applications in data transferring from source to destination system. However, cryptography is a data secrecy technique between sender and receiver, the steganography increases the level of security and acts a protective layer to the hidden information within the source image. In this paper, a compression scheme based image security algorithm for wireless sensor network is proposed to hide a secret color image within the source color image. The main contribution of this paper is to propose a compression scheme which is based on level matrix and integer matrix, and increases the compression level significantly. The performance of the proposed system is evaluated in terms of peak signal to noise ratio (PSNR), mean square error (MSE), number of pixels change rate (NPCR) and unified average changing intensity (UACI). The proposed method achieves 42.65% PSNR, 27.16% MSE, 99.9% NPCR and 30.99% UACI.

Keywords

Image Security, Steganography, Compression, Secret Image, Cryptography

1. Introduction

Nowadays, lots of information is exchanged through the internet. Security of the information plays an important role in internet to protect the data from the attackers or intruders. Many researchers are using cryptography and steganography for data security over the internet medium. Cryptography deals message encryption but the communication is visible but on the other hand, steganography deals with secret message hiding but the communication is not visible [1]. One of the foremost variations of steganography with cryptography is that encoding the traffic, the communications will be secured but people become aware of the existence of message by observing coded information, although they are not able to realize the message in the data. The steganography

*Corresponding author.

technique hides the existence of the message so that intruders cannot detect what communication is going on, thus providing an advanced level of security than cryptography. Both steganography and cryptography systems are used in providing secret communications almost in a similar approach but vary in terms of methods used to break the system [2].

Steganography methods can be divided into spatial (original) domain methods and transform domain methods. The cover image is initially transformed into frequency domain, and in the next level, the embedding of secret messages into the coefficients of transformed cover image is performed. The steganography algorithms are performed based on various levels of security to produce stego images (stg) with high imperceptible [3]. These levels are added to be sure that the difficulties to extract the secret image (S) have been accomplished. One more factor that disputes the security level is the amount of payload capacities in the stego image (stg). Hence, the payload should be calculated cautiously in order to find the maximum number of bits from “S” that can be embedded into a cover image safely and with robustness. **Figure 1** illustrates the steganography process where the secret image is embedded into the source image and the secret image is recovered in the receiver side. The main aim of this paper is to apply the compression scheme of image for wireless sensor networks (WSN).

This paper is organized as 5 sections, where section 1 introduces the basic concepts of steganography and section 2 deals different conventional methodologies for steganography process. Section 3 presents the proposed methodology for image security and lossless compression. Section 4 elaborates the results of the proposed method and discussion about results. Section 5 depicts the conclusion.

2. Literature Survey

Biham *et al.* (2005) [1] have utilized Skipjack algorithm for generating block cipher from the image or data in wireless sensor networks. The 80-bit key was generated by this algorithm, and it was stable from the attackers. The 64-bit information data was encrypted (encoded) using this secret key. The computations of energy consumptions were done to analyze the suitability of this proposed algorithm for wireless sensor networks. Biswas *et al.* (2015) [2] have used Chaotic Map and Genetic Operations for the encryption of data and images. The experiments were conducted using the real-time sensor unit Mica 2 sensor. The number of security mechanisms were analyzed and used in this work to provide confidentiality of the data. A lightweight block cipher algorithm

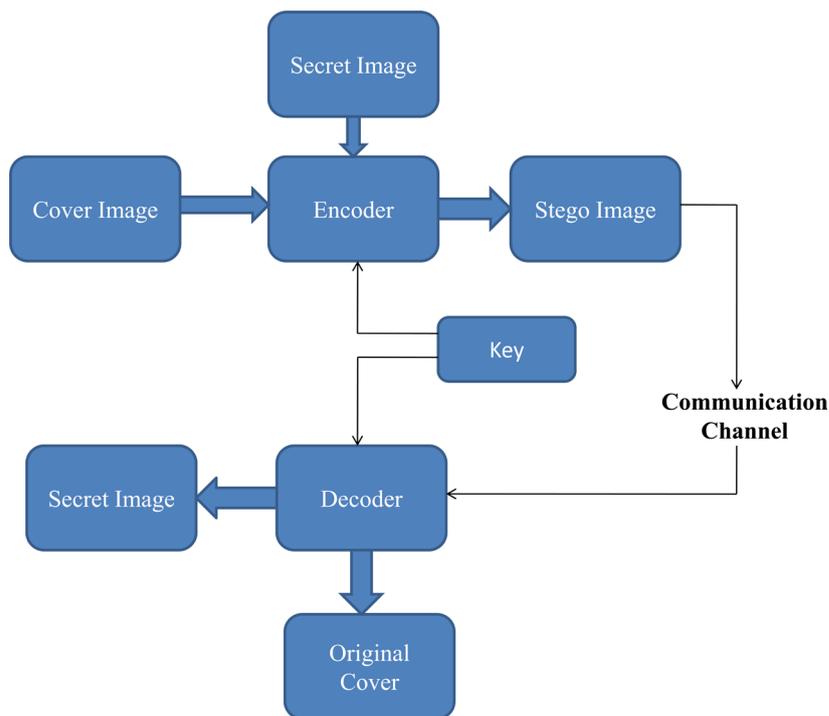


Figure 1. Steganography system.

was proposed to secure the data generated by the sensor unit in wireless sensor networks. The authors achieved NPCR about 99.67%, UACI about 33.42% and information entropy about 7.98 for encrypting and decrypting the lena.jpg image. Bouslimi *et al.* (2012) [3] have proposed a Joint Encryption scheme for securing the medical images. This method integrated the encryption scheme with watermarking technique to increase the security level of the system. The Advanced Encryption Standard (AES) scheme was used as the encryption process. The authors achieved PSNR about 53.94 dB for ultra sound images and 101.99 dB for PET images. Ghrare *et al.* (2009) [4] proposed a matrix based lossless compression technique to compress the medical images. The compression ratio was low due to its lossless technique. Guo *et al.* (2016) [5] have developed biometric based encryption algorithm. Face and finger print were encrypted using this approach and the private key was generated from master secret key. Guo & Le (2010) [6] have developed JPEG double compression based image security algorithm using compression tables. The authors achieved PSNR about 44.1 dB based on two standard compression tables. The main limitation of this paper was that it did not support time cost, and as the method was based on lossy compression, the compression ratio value was less. Lim *et al.* (2013) [7] used biometric discretization method to protect the information against attackers. The Detection Rate Optimized Bit Allocation (DROBA) scheme was applied one of the most effective biometric discretization schemes in this work. The performance of the system was analyzed in terms of false rejection rate and average entropy loss. Ramkumar *et al.* (2014) [8] developed a image security system which imposed a binary image into color image using different quantization tables. This method was not support the imposing a secret color image into source color image. Sahai and Waters (2005) [9] used the fuzzy algorithm for cryptography and private key was generated based on this. This private key decrypt the cipher text using fuzzy logic constructed in encoder side. A private key of identity can decrypt a ciphertext encrypted with another identity, if and only if, the set overlap distance of these two identities. Suzuki *et al.* (2012) [10] developed an encryption algorithm based on TWINE method. The authors utilized the concept of Feistel technique in this work to generate the cipher text from the plain text. The method stated in this work contained 16 branches and 36 rounds. Odai M. Al-Shatanawi *et al.* (2015) [11] proposed Modified Least Significant Bits (MLSB) method to hide the secret image within the source image using the pixels detection in a random manner. The authors were then applied Advanced Encryption Standard (AES) technique on the stego image against from the different attackers. Shuliang Sun (2015) [12] used Canonical Gray Coding (CGC) technique to hide the information using Bit-Plane Complexity Segmentation (BPCS) steganography method. Mandal and Das [13] proposed a steganography method for color images based on difference in pixel values. Secret data was hiding in the component of a pixel in a color image.

The attacks on the secured image were not analyzed by the methodologies stated in conventional techniques [1]-[9]. The conventional methods also not discussed and analyzed the embedding of two color images for security purpose. These limitations are overcome by the methodology proposed in this work. The main contribution of this paper is to provide a novel compression scheme which is based image security algorithm for WSN and it is used to hide a secret color image within the source color image in order to perform both security and compression in WSN applications.

3. Proposed Method

The proposed methodology for image security system based on compression technique is illustrated in **Figure 2**. The main principle of this proposed method is to obtain a secured embedded image with high compression ratio and stable to against attackers as shown in **Figure 2**. The secret color image is block permuted, and the secret key is generated from the block permuted image. This key is used in decoder side to retrieve the secret binary image. The block permuted image is embedded into original source image using block compression table, and the compression technique is applied on this embedded image to produce the compressed patterns.

3.1. Binary Secret Image

The RGB color secret image is initially converted into grey scale image. Each pixel in color image consists of 24 bits resolution and each pixel in grey scale image consists of 8 bit resolution. The grey scale image is further converted into binary secret image based on the dynamic threshold. The dynamic threshold value (T) is determined as,

$$T = [\text{Min}(I) + \text{Max}(I)]/2 \quad (1)$$

where, I represent the grey scale image.

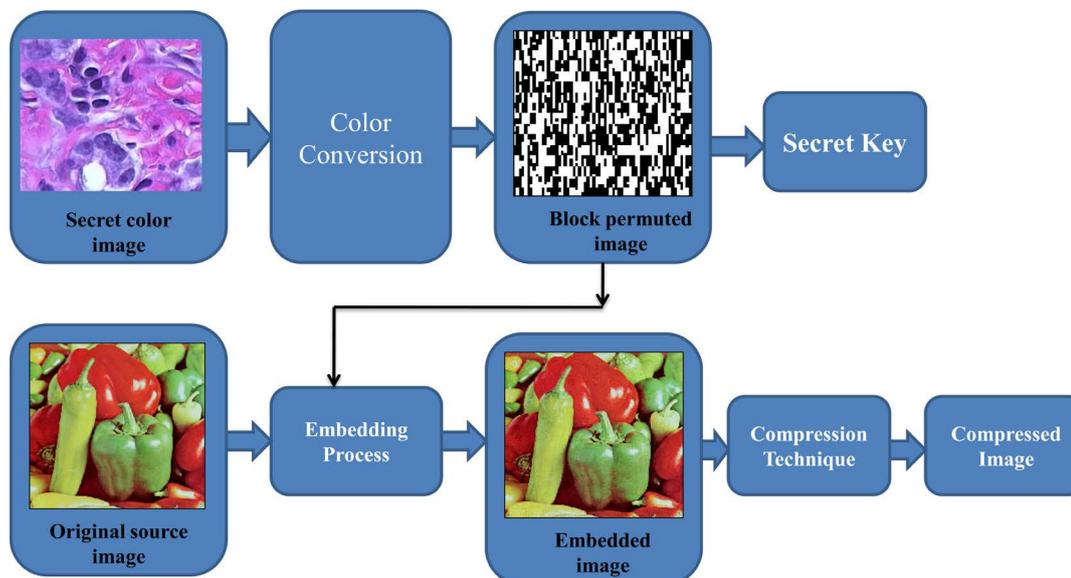


Figure 2. Proposed compression based image security system.

The binary secret image is embedded into the original source image (Ramkumar & Raglend 2014). The width and height of the binary secret image is represented as $X*Y$. Every pixel in the binary secret image is denoted as $P(X, Y)$. The pixel in this image may have the value either zero or one. Zero represents a black pixel, and one represents a white pixel as depicted in Equation (2).

$$P(X, Y) = \begin{cases} 1, & \text{if white pixel} \\ 0, & \text{if black pixel} \end{cases} \quad (2)$$

3.2. Original Source Image

The original source image is denoted as 'S' and this image may be either color or gray scale image. The size of this image is represented as M (width) and N (height). The aspect ratio should be verified before embedding the binary secret image into original source image in order to retain the binary secret image correctly. The aspect ratio (Guo & Le 2010) is defined as,

$$\frac{M}{N} = \frac{X}{Y} \quad (3)$$

In this paper, size of the original source image is 1024×1024 and size of the secret binary image is 128×128 . Then, the aspect ratio is satisfied as,

$$\frac{1024}{1024} = \frac{128}{128} \quad (4)$$

The secret binary image is only embedded into original source image if the aspect ratio is satisfied.

3.3. Block Random Permutation

Every pixel in the binary secret image has either zero or one. All these values are stored in a matrix called "M". The number of rows and columns in the matrix is equal to the size of the binary secret image. The size of the shuffling block is denoted as 'S' and it may have values one to three, which may be determined randomly. The block random permutation algorithm randomizes the matrix M by dividing M into non-overlapping blocks of the size specified by S, and shuffling these blocks. The number of elements in S should match the number of dimensions of M, or S can be a scalar specifying an S-by-S-by-S-by ... block size. "S" should contain positive integers. The size of M in any dimension should be an integer, number of times the specified size of the block in that dimension. The block permuted image is generated by permutation of each pixel block in the secret binary

image with a known order. The order of changing the pixel value in the secret binary image is stored as the permutation secret key. This key is used in the decoder section to de-permute the secret binary image in a reverse order. **Figure 3(a)** shows the binary secret images which have to be embedded into the source image and **Figure 3(b)** shows the block random permuted images of the corresponding binary secret images.

3.4. Embedding Procedure

The secret binary image is embedded into original source image using the following procedure.

Step 1: Determine the randomized factor ‘ r ’ using the size of original source image and secret binary image and it is given as,

$$r = \frac{M}{X} * \frac{N}{Y}. \quad (5)$$

In this paper, the randomized factor is $r = 8 \times 8$.

Step 2: The original source image is split into number of blocks of size ‘ r ’. In this paper, the original source image is split into 16 blocks of size 8×8 (each sub-block size) for the size of original source image 1024×1024 .

Step 3: Each sub-block in original source image is divided with either compression table 1 (**Figure 4**) or compression table 2 (**Figure 5**) (Ramkumar & Raglend 2014) based on the following constraints:

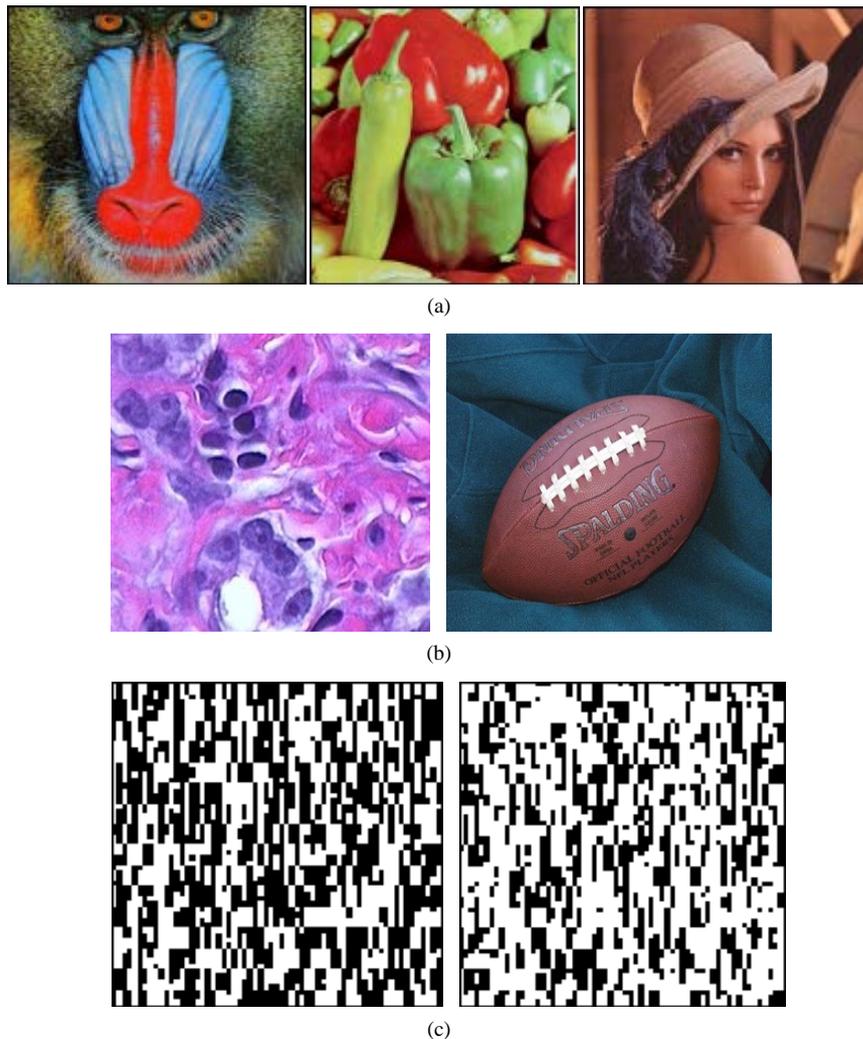


Figure 3. (a) Source color images (babbon, fruits and Lena); (b) Secret binary images “hestain.jpg” and “football.jpg”; and (c) Block random permuted images (after converting binary image).

1	1	2	2	4	4	4	4
2	2	2	2	3	3	3	5
5	2	1	2	3	3	3	3
5	5	5	5	5	1	1	1
2	2	2	2	2	2	3	3
4	4	4	4	2	2	2	2
4	6	6	6	6	7	7	7
7	7	7	1	1	1	1	1

Figure 4. Compression table 1.

1	1	2	2	4	4	4	4
2	2	2	2	3	3	3	5
5	2	1	2	27	22	41	45
6	6	7	7	7	49	51	55
7	7	7	7	55	60	65	67
6	7	6	7	75	75	78	83
5	85	91	95	97	68	76	79
6	91	91	91	96	96	98	99

Figure 5. Compression table 2.

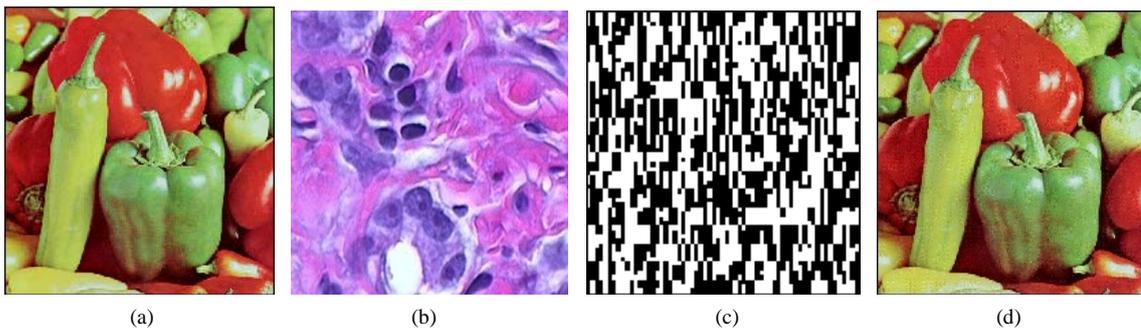


Figure 6. (a) Original source image “fruit.jpg”; (b) Secret color image “hestain.jpg”; (c) Block permuted binary image; (d) Embedded secured image.

$$\text{Secured image} = \begin{cases} \frac{\text{Sub_block}}{QT1}, & \text{if } P(X, Y) = \text{white pixel} \\ \frac{\text{Sub_block}}{QT2}, & \text{if } P(X, Y) = \text{black pixel} \end{cases} \tag{6}$$

The sub-block of original source image is divided by compression table (CT1) if the first pixel of the secret binary image $P(X, Y)$ is one and CT1 is given as **Figure 4**.

The sub-block of original source image is divided by compression table (CT2) if the first pixel of the secret binary image $P(X, Y)$ is zero and CT2 is given as **Figure 5**.

The secured image is an embedded of the secret color image into original source image. This image is an RGB color image and it is further applied to compression technique to compress this embedded-secured image [14]. **Figure 6(a)** shows the original source image, **Figure 6(b)** shows secret binary image, **Figure 6(c)** shows the block permuted binary image and **Figure 6(d)** shows the embedded-secured image.

Step 4: The lossless compression technique (JPEG-LS) is applied on the embedded-secured image to generate the compressed patterns.

JPEG-LS is a lossless/near-lossless compression standard for continuous-tone images [4]. Its official designation is ISO-14495-1/ITU-T.87 [15]. It is a simple and efficient baseline algorithm which consists of two independent and distinct stages called modeling and encoding. JPEG-LS was developed with the aim of providing a low-complexity lossless and near-lossless image compression standard that could offer better compression efficiency than lossless JPEG. It was developed because at the time, the Huffman coding-based JPEG lossless standard and other standards were limited in their compression performance. Total decorrelation cannot be achieved by first order entropy of the prediction residuals employed by these inferior standards.

Step 5: The original secret image is obtained in the receiver section by following the reverse procedure.

4. Results and Discussion

To validate the effectiveness of the proposed system, we test our proposed image security algorithm on the publicly available open access images. In this paper, “fruit.jpg” and “Lena.jpg” are used as the original RGB source images. The size of the original source image is 512×512 and stored in JPEG format. The secret binary images used in this paper are “hestain.jpg” and “football.jpg” images. These images are obtained from MATLAB demo toolbox [15] [16]. The size of secret color image is 128×128 . In order to test the robustness of the proposed method, the test source and secret images should have taken from various indoor and outdoor environments. The proposed algorithm is implemented with MATLAB R2014 on a personal computer with Core i3 2.8 GHz CPU and 3 GB RAM. The performance of the proposed system is evaluated in terms of Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Number of Pixels Change Rate (NPCR) and Unified Average Changing Intensity (UACI) that are described in Biswas *et al.* (2015).

4.1. PSNR

It indicates the quality of the decoded secret binary image with respect to the original secret binary image. It is expressed as,

$$PSNR = 20 \log_{10} \frac{MAX_f}{\sqrt{MSE}} \quad (7)$$

where, MAX_f represents the number of pixels in the secret binary image.

4.2. MSE

It indicates the error rate of the decoded secret binary image with respect to the original secret binary image and it is given as,

$$MSE = \frac{1}{X * Y} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|f(i, j) - g(i, j)\|^2 \quad (8)$$

where, $f(i, j)$ represents the original secret binary image and $g(i, j)$ represents the decoded secret binary image. The width and height of the secret binary image is denoted as X and Y respectively.

4.3. NPCR

It determines the percentage of different pixel numbers between the two images such as original secret binary image and decoded secret binary image. It is expressed as,

$$NPCR = \frac{1}{X * Y} \sum K(i, j). \quad (9)$$

Let $K1(i, j)$ be the original secret binary image and $K2(i, j)$ is the decoded secret binary image.

$$k(i, j) = \begin{cases} 1, & \text{if } k1(i, j) \neq k2(i, j) \\ 0, & \text{else} \end{cases} \quad (10)$$

4.4. UACI

It determines the average intensity of differences between the two images such as original secret binary image and decoded secret binary image. It is expressed as,

$$UACI = \frac{1}{X * Y} \sum_0^{m-1} |k1(i, j) - k2(i, j)|. \tag{11}$$

For a better system, the value of UACI should be low, and NPCR should be high. **Table 1** shows the performance analysis of the proposed system in terms of PSNR, MSE, NPCR and UACI. The proposed system achieves the average PSNR about 54.19 dB, MSE about 28.45, NPCR about 99.90 and UACI about 30.99.

Table 2 shows the Quantitative analysis with different color channels for different source images and secret images, respectively. **Table 3** illustrates the performance comparisons of the proposed system with conventional techniques as Guo *et al.* (2016), Biswas *et al.* (2015) and Guo & Le (2010) in terms of NPCR and UACI.

4.5. Information Entropy

The information entropy is computed for the original source image and its encoded image. Its value should be 8 for gray scale image. The security level of proposed system is high if the value of information entropy is equal to 8. The information entropy is expressed as,

$$EN = \sum_{i=1}^l p(mi) * \log_2 [1/p(mi)] \tag{12}$$

Table 1. Performance analysis of proposed system.

Secret color images	PSNR	MSE	NPCR	UACI
Hestain.jpg	41.65	27.16	99.89	31.82
Football.jpg	42.15	29.75	99.91	30.17
Average.jpg	41.9	28.45	99.90	30.99

Table 2. Quantitative analysis with different color channels.

Source image	Channels	Secret image	PSNR (dB)		
			Chen <i>et al.</i> (2008)	El-Emam <i>et al.</i> (2013)	Proposed method
Lena	R	Hestain	37.97	38.19	41.75
	G		37.87	39.10	40.28
	B		39.78	38.75	41.86
	RGB		38.62	38.28	41.65
Fruit	R	Football	36.16	39.28	41.84
	G		37.17	38.38	40.63
	B		35.62	39.27	42.18
	RGB		36.18	39.65	42.15
Baboon	R	Peppers	36.18	39.65	40.71
	G		37.16	38.19	42.19
	B		34.87	39.55	41.27
	RGB		37.89	39.19	41.95

Table 3. Performance comparisons in terms of NPCR and UACI.

Secret binary images	NPCR	UACI
Proposed system	99.90	30.99
Fuchun Guo <i>et al.</i> [3]	98.19	31.27
Kamanashis Biswas <i>et al.</i> [6]	99.60	33.4
Jing-Ming Guo <i>et al.</i> [5]	97.16	36.16

where, $p(mi)$ represents the probability of the symbol mi and t is the total number of symbols. **Table 4** shows the information entropy for source and embedded image. The average information entropy of embedded image is superior to the information entropy of the source image.

Table 5 shows the average information entropy for both proposed and conventional systems. It can be shown that the information entropy of the proposed system is closer to the value of 8, which indicates the chance of information leakage is negligible, and the proposed system for image security is well secured against any attacks.

4.6. Quantitative Evaluations

The performance of the proposed image security algorithm is analyzed using execution time on various CPU. The execution times of the compressing the “hestain.jpg” secret color image embedded into fruit source image are 3.437 sec, 24.6540 sec and 57.6517 sec at CPU frequencies of 2.4 GHz, 1.8 GHz and 800 MHz, respectively. The execution times of the compressing the “football.jpg” secret color image embedded into fruit source image are 4.78 sec, 26.6137 sec and 61.6016 sec at CPU frequencies of 2.4 GHz, 1.8 GHz, and 800 MHz, respectively. The average encoder execution time is illustrated in **Table 6**.

The execution times of the decoder for “hestain” secret color image are 4.76 sec, 24.6540 sec and 57.6517 sec at CPU frequencies of 2.4 GHz, 1.8 GHz, and 800 MHz, respectively. The execution times of the decoder for “football.jpg” secret color image are 4.76 sec, 24.6540 sec and 57.6517 sec at CPU frequencies of 2.4 GHz, 1.8 GHz, and 800 MHz, respectively. The average decoder execution time to extract the secret binary image from embedded image is illustrated in **Table 7**.

Table 4. Performance analysis in terms information entropy.

Original source images	Information entropy	
	Source image	Embedded image
Lena	7.2131	7.9992
Fruit	7.2587	7.9999
Average	7.2354	7.9995

Table 5. Performance comparisons in terms of information entropy.

Secret binary images	Information entropy
Proposed system	7.9995
Fuchun Guo <i>et al.</i> [3]	7.8914
Kamanashis Biswas <i>et al.</i> [6]	7.9936
Jing-Ming Guo <i>et al.</i> [5]	7.9102

Table 6. Average image security execution time (encoder).

CPU frequency	“Hestain” secret color mage	“Football.jpg” secret color image
	Total Time (s)	Total Time (s)
2.4 GHz	3.437	4.78
1.8 GHz	24.65 40	26.61 37
800 MHz	57.65 17	61.60 16

Table 7. Average image security execution time (decoder).

CPU frequency	Processing time of decoder (sec)	
	“Hestain.jpg” secret image	“Football.jpg” secret image
2.4 GHz	4.27	4.76
1.8 GHz	24.6540	24.6540
800 MHz	57.657	57.667

5. Conclusion

In this paper, compression based image security for WSN is proposed to increase the level of security from the attackers. The secret binary image is embedded into the original source image using compression technique. This compression technique produces binary and integer matrix which increases the compression ratio of the proposed image security system. The proposed system has high information entropy which indicates the strength of the security system and achieves average PSNR about 41.9 dB and average MSE about 28.45. The methodology in this paper achieves 99.9% of number of pixels change rate (NPCR) and 30.99% of the unified average changing intensity (UACI). In future, this work can be extended to embed the secret video or audio in the source video or audio in WSN to increase the level of security to the next higher level.

Acknowledgements

The authors would like to thank their friends and colleagues for their constant support throughout the research.

Conflict of Interests

The authors declare that there is no conflict of interests.

References

- [1] Biham, E., Biryukov, A. and Shamir, A. (2005) Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. *Journal of Cryptology*, **18**, 12-19. <http://dx.doi.org/10.1007/s00145-005-0129-3>
- [2] Biswas, K., Muthukkumarasamy, K. and Singh, K. (2015) An Encryption Scheme Using Chaotic Map and Genetic Operations for Wireless Sensor Networks. *IEEE Sensors Journal*, **15**, 2801-2809.
- [3] Bouslimi, D., Coatrieux, G., Cozic, M. and Roux, C. (2012) A Joint Encryption/Watermarking System for Verifying the Reliability of Medical Images. *IEEE Transactions on Information Technology in Biomedicine*, **16**, 891-899. <http://dx.doi.org/10.1109/TITB.2012.2207730>
- [4] Ghrare, S.E., Ali, M.M.A., Jumari, K. and Ismail, M. (2009) An Efficient Low Complexity Lossless Coding Algorithm for Medical Images. *American Journal of Applied Sciences*, **6**, 1502-1508. <http://dx.doi.org/10.3844/ajassp.2009.1502.1508>
- [5] Guo, F., Susilo, W. and Mu, Y. (2016) Distance-Based Encryption: How to Embed Fuzziness in Biometric-Based Encryption. *IEEE Transactions on Information Forensics and Security*, **11**, 247-257. <http://dx.doi.org/10.1109/TIFS.2015.2489179>
- [6] Guo, J.M. and Le, T.N. (2010) Secret Communication Using JPEG Double Compression. *IEEE Signal Processing Letters*, **17**, 879-882. <http://dx.doi.org/10.1109/LSP.2010.2066110>
- [7] Lim, M.H., Teoh, A.B.J. and Toh, K.-A. (2013) Dynamic Detection-Rate-Based Bit Allocation with Genuine Interval Concealment for Binary Biometric Representation. *IEEE Transactions on Cybernetics*, **43**, 843-857. <http://dx.doi.org/10.1109/TSMCB.2012.2217127>
- [8] Ramkumar, D. and Raglend, I.J. (2014) Performance Analysis of Image Security Based on Encrypted Hybrid Compression. *American Journal of Applied Sciences*, **11**, 1128-1134. <http://dx.doi.org/10.3844/ajassp.2014.1128.1134>
- [9] Sahai, A. and Waters, B. (2005) Fuzzy Identity-Based Encryption. In: Cramer, R., Ed., *Advances in Cryptology—EUROCRYPT 2005*, Springer-Verlag, Heidelberg, Germany, 457-473.
- [10] Suzaki, T., Minematsu, K., Morioka, S. and Kobayashi, E. (2012) TWINE: A Lightweight Block Cipher for Multiple Platforms. In: Knudsen, L.R. and Wu, H.P., Eds., *Selected Areas in Cryptography*, Springer-Verlag, Berlin, Germany, 339-354.
- [11] Al-Shatanawi, O.M. and El Emam, N.N. (2015) A New Image Steganography Algorithm Based on Mlsb Method with Random Pixels Selection. *International Journal of Network Security & Its Applications (IJNSA)*, **7**, No. 2. <http://dx.doi.org/10.5121/ijnsa.2015.7203>
- [12] Sun, S.L. (2015) A New Information Hiding Method Based on Improved BPCS Steganography. *Advances in Multimedia*, **2015**, 1-7. <http://dx.doi.org/10.1155/2015/698492>
- [13] Mandal, J.K. and Das, D. (2012) Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain. *International Journal of Information Sciences and Techniques*, **2**, No. 4. <http://dx.doi.org/10.5121/ijist.2012.2408>
- [14] Chang, C., Lin, C. and Fan, Y. (2008) Lossless Data Hiding for Color Images Based on Block Truncation Coding. *Pattern Recognition*, **41**, 2347-2357. <http://dx.doi.org/10.1016/j.patcog.2007.12.009>

- [15] Chen, J., Chen, T.S., Lin, C., Chen, S.Y. and Lin, J. (2015) A Simple JPEG-LS Compressed Technique for 2DGE Image with ROI Emphasis. *The Imaging Science Journal*, **63**, No. 2. <http://dx.doi.org/10.1179/1743131X14Y.0000000086>
- [16] El-Emam, N. and Al-Zubidy, R. (2013) New Steganography Algorithm to Conceal a Large Amount of Secret Message Using Hybrid Adaptive Neural Networks with Modified Adaptive Genetic Algorithm. *Journal of Systems and Software*, **86**, 1465-1481. <http://dx.doi.org/10.1016/j.jss.2012.12.006>