I. J. Computer Network and Information Security, **2022, 4, 43-59**
Published Online August 2022 in MECS (http://www.mecs-press.org/)
DOI:10.5815/ijcnis.2022.04.04

An Efficient Image Steganography Scheme Using Bit-plane Slicing with Elliptic Curve Cryptography and Wavelet Transform

Ganavi M
CSE/JNNCE/Shivamogga, Karnataka, 577204
E-mail: gaanavi4@jnnce.ac.in

Prabhudeva S
MCA/JNNCE/Shivamogga, Karnataka, 577204
E-mail: pdshirematt@gmail.com

Hemanth Kumar N P
CSE/EPCET/Bengaluru, Karnataka, 560049
E-mail: hemanthnp@gmail.com

Received: 08 January 2022; Revised: 01 March 2022; Accepted: 23 March 2022; Published: 08 August 2022

Abstract: Information security is indispensable in the transmission of multimedia data. While accumulating and distributing such multimedia data, the access of data from a third person is the real security challenging issue. Information hiding plays an important role. Scramble the data before hiding it in carrier media gives enhanced security level for the data. In this paper, bit plane slicing is used to represent an input image with eight planes at bit-level instead of pixel-level. As the least significant bit contains noisy information, only the most significant bit plane can be used to represent an image. At the first level, an input image is processed through the spatial domain. Transform domain techniques are used to process the image at the middle level. Elliptic curve cryptography is used to scramble and descramble the MSB plane image. A logistic chaotic sequence of the input image is added to the most significant bit plane image to generate the final scrambled image. The discrete wavelet transform is used to embed the scrambled image in its high-frequency sub-bands. At the last level, a least significant bit technique, a spatial domain is used to embed the scrambled image in the carrier image. Message integrity is also verified by finding the hash of an input image. The performance of the proposed method is evaluated through various security measures. It gives good results as number of pixel change rate is closer to 100% and unified average changing intensity is 33.46.

Index Terms: Bit-plane Slicing, Elliptic Curve Cryptography, Discrete Wavelet Transform, Least-Significant-Bit, Scramble, Embed.

1. Introduction

With the rapid growth in the usage of digital data, there is a necessity for the protection of sensitive data. So, information security plays an important role not only while it is transmitted also when it is stored. Researchers have tried to handle such information. Various algorithms have been implemented. All these algorithms are implemented with the aim of information security such as confidentiality, integrity, and availability [1]. Sensitive information when it is transmitted, should not be visible and nobody should be able to identify that some information is hidden in it. To overcome such security issues, cryptography [2], and steganography [3] play an important role. Scattering of information present in sensitive data can be done through cryptography. Hide such data within the carrier media can be carried out using steganography.

Maintaining the information confidentiality, integrity and undetectability is requisite to protect from unauthorized persons. These security measures can be achieved by using better cryptographic and steganographic algorithms. Confidentiality of the information can be achieved through cryptographic scrambling algorithms [4]. Hashing algorithms are useful to address message integrity. SHA-256 is one such hashing where it cannot be reversed [2]. It's a one-way function. This can be used on images to achieve message integrity. Even though the attacker gets the hashed message, it is not possible to reverse back to extract the actual message from it. Information undetectability can be achieved through embedding using better transform-based steganography techniques.

Various traditional algorithms such as DES, AES, RSA, and Elliptic curve cryptography (ECC) can be used for scrambling data. Spatial Least-Significant-Bit [5] embedding and transform embedding techniques like DCT [6], and Discrete Wavelet Transform (DWT) [7] can be used for steganography. Among all traditional cryptosystems, ECC consumes less power, memory, and more computational complexity [4]. The key size in ECC of 256-bit is like 3072-bit in RSA key and 10,000 times stronger than a 2048-bit RSA key [8]. So, it requires an insignificant load in the network and computing power. In 1985, the approach of elliptic curves in cryptography was put forward by Mr. Neal Koblitz and Mr. Victor S. Miller, and later this algorithm is started extensively used since 2004. ECC functions by strengthening a distinct set of public and private keys for the scramble and descramble of data over the web [9]. There are other scramble methods like Diffie-Hellman and RSA cryptography. They work upon the construction of keys by making use of larger prime numbers that necessitate enough computational power. To scramble the data, elliptical curve cryptography uses composite and statistically hardy keys. For digital signatures, ECC is utilized in digital signatures utilizing Elliptic Curve DSA (ECDSA key) and in key interchange utilizing Elliptic Curve Diffie-Hellman (ECDH) [4]. ECDSA is utilized over numerous safety systems and is more frequently used in messaging applications and bitcoin security. ECDSA is also applied in Transport Layer Security (TLS), by scrambling the link among web browsers and a web application. Also used in a piece of the SSL standard taking advantage of signing SSL certificates. ECC plays an important role in the field of cryptography. But when it is combined with the steganography technique, the scrambled data can be hidden to obtain another level of security.

Chaos-based algorithms are useful in image encryption. These algorithms provide high speed, sensible computation, and better security. After being presented with the initial value and its parameters it can generate a confusion matrix. This is sensitive to initial conditions. The contingent-like behavior of chaos is productively broadened into encrypted images [10]. As though a low-dimensional chaotic system is uncomplicated to accomplish, its key space is small, and it undergoes severe dynamic degradation. Therefore, combining with other algorithms would greatly reduce the problem of key space and dynamic degradation.

Hiding the secret data in another media can also be carried out using spatial domain techniques [11]. The Least Significant Bit (LSB) embedding is one of the spatial domain techniques. Here, the least significant bits in the pels of the carrier image are changed with a bit of the secret image. The visibility of the image will not be changed if the LSB bits of the pixel in an image is changed. Therefore, the LSB embedding technique is one of the most important steganography techniques in use today [5]. But the LSB embedding is susceptible to steganalysis. So, encrypt the data before embedding it in LSB make the system more secure.

The discrete wavelet transform has been used in many applications like science, engineering, mathematics, and computer science. It is a transform domain technique. DWT finds useful in cryptography, steganography, and compression. When DWT is applied to images it gives a good compression ratio without losing much information, but it requires more processing power. DWT converts data to a transformation domain with four sub-bands [7]. Lower bands contain actual information whereas high-frequency sub-bands contain noise. Therefore, processing only low-frequency bands is helpful in cryptography and compression. High-frequency sub-bands are helpful to embed secret information in steganography. Instead of directly hiding the raw data in the high-frequency sub-bands, scramble it using crypto algorithms and hide it in LSB results to a more secure system. This makes it difficult for any intruder to identify the presence of one piece of information in other media.

The Grayscale image is represented with 8-bits for each pels. These 8-bits in all the pels are grouped as separate planes. The technique used to perform this is known as Bit-plane slicing [12]. Equivalently there will be eight planes in an image. The least significant bits from all the pels are grouped as Bit-plane 1, next bits as Bit-plane 2, and the last most significant bits are grouped as Bit-plane 8. The upper four Most-Significant-Bit (MSB) planes contain all the information of the image. The lower four LSB planes contain noise. So upper four MSB planes are useful in compression and in cryptography where it saves the memory and reduces the computational time for any secret image to be processed. But lower LSB planes are useful in embedding secret or encrypted information using steganography [3]. After embedding information in LSB planes, the carrier media will not be degraded, and the embedded information will not be seen. So, Bit-plane slicing is supportive to achieve high image imperceptibility.

In this paper, an approach with a combination of spatial and transform domain crypto-stego algorithms is proposed. The input secret image is Bit-plane sliced, extracted only MSB planes, scrambled using ECC and chaotic sequence, embedded this in a carrier media using DWT and LSB techniques. SHA-256 is also applied to check message integrity. Rather than processing the entire input image, a process only the useful information that is present in the MSB planes reduces the total processing time of the algorithm. Noisy data present in LSB is filtered out. To make it still more random, the chaotic sequence generated for the input image is added to the ECC sequence before embedding it in the cover media. This makes it difficult for the attacker to descramble until & unless he gets the chaotic sequence which is made available only to the intended receiver. Used transform domain algorithm DWT as it is always best suited for embedding sensitive information. An effort is made to put-together the layer-by-layer security for the information which is to be shared safely to the actual receiver. The paper is organized as follows. Section I gives the Introduction to cryptography and steganography, II is about the contribution of the work., and III is about the literature review. Section IV presents the methodology of the proposed system architecture. Section V comprises the results & performance analysis and section VI discusses the conclusions drawn and future scope.

2. Contribution of this Work

There is a necessity of securing sensitive information in this rapidly growing digital world. The proposed scheme is a combination of the crypto-stego system by keeping in mind the various security issues.

1. Deployed a spatial and transform domain scramble and embed technique. Stronger transform domain algorithms such as ECC and DWT are used at the important stages of this scheme. To check the message's integrity, SHA-256 is also taken into consideration. The chaotic sequence is added to make it difficult for the attacker to identify the encryption scheme.
2. To reduce the computational time, considered only MSB planes of the input plain image by using the bit plane slicing method. The reason for considering only the MSB planes is that all the information of any image is present in the MSB planes when compared to LSB planes as this contains only noise.
3. Various statistical analyses such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Structural Similarity Index Measure (SSIM), Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI) are analyzed to show the effectiveness of the proposed work.

3. Literature Review

A bit-plane slicing scheme in Ref. [12] has been used in hiding the secret image in the carrier RGB color image. Bit-plane slicing is applied to the watermarked secret image. The generated is embedded in the luminance channel of the carrier image. Performance is analyzed using robustness. A steganography algorithm has been proposed in Ref. [11] where a combination of DWT and hidden region is selected in the DWT domain. A steganography method, where DWT is applied to compress an input image and DFT is applied on the carrier to calculate the coefficients in Ref. [4]. These coefficients are used to embed the compressed input data. Five different cases are explained by applying the DWT on input images at two-level & three-level along with the varying threshold value. The best results are generated for case 5, where LSB is used to embed.

The LSB is used to embed the secret data in the carrier in Ref. [13]. Stego-key is randomly selected from Z_{24} and it is used to embed the data in a 24-bit carrier. Three characters from input data are embedded in 24-bits of the carrier. Security analyses are presented in terms of robustness, imperceptibility, and embedding capacity. Reference [6] demonstrates the LSB embedding method based on DCT coefficients of the carrier image. DCT coefficients of the carrier are calculated. If coefficients are less than the threshold values, then MSB of the secret data is inserted in the LSB of that carrier pixel. A steganography method based on a deep neural network has been proposed in Ref. [14]. Input is applied with DCT. The generated DCT coefficients are scrambled using ECC. The scrambled data is embedded into the carrier using SegNet deep network model. This is successful to achieve a PSNR of 40dB and SSIM 0.96, respectively.

An embedding technique has been proposed in Ref. [15] where the data is transformed by DWT, and the LL band is again applied with Singular Value Decomposition (SVD). The transformed data is inserted at HL2 and HH2 bands of the DWT transformed carrier image. This embedded image is subjected to different attacks to check the robustness of the proposed method. Imperceptibility is also checked by measuring PSNR, MSE, and normalized correlation coefficient. A steganography method in Ref. [4] where medical data is encrypted using ECC and later embedded at LSB of the carrier image. The system is implemented using the Java language. Security measures like PSNR and MSE are measured for ECC and RSA. A video steganography method has been proposed in Ref. [14]. The carrier video is segmented into frames. The selected frames are transformed using Fast Discrete Curvelet Transform (FDCT) and binary forms of the secret data are embedded in these coefficients. The stego key is encrypted using ECC. In this method, not only input data but also key is encrypted. An embedding method in Ref. [5] has been proposed where the least significant bit substitution is used to embed the secret data carrier. A pseudo-random number generator is used to generate random pixel locations in the carrier. The random bit position is selected using the 3-3-2 method to hide the secret data.

ECC ElGamal and 3D Lorenz map are used to encrypt the secret data in Ref. [8]. The mapping table is used to convert the pixel to a point equivalent. Security analyses are carried out to check the performance of the proposed method. In this method, MSB planes are extracted, and later DWT & chaotic sequences are generated to get the encrypted version of the input image in Ref. [16]. NPCR and UACI are measured as security analyses. An encryption method along with key management & distribution using the asymmetric RSA algorithm has been proposed in Ref. [7]. The chaotic sequence is generated input image and applied with Schur decomposition to scramble it. This scrambled image is embedded in LH and HH sub-bands after obtaining the DWT of the carrier.

In the literature review, necessary contributions to the cryptography and steganography system are explained. The quality metrics for image security before embedding are acted as very much important factors. Bit-plane slicing is very much required to reduce the total processing time of the input image. ECC, a smaller key size with more security would be the better scrambling algorithm to make the system more robust against the attacks when compared to RSA.

Transform level combined with spatial type steganography techniques can achieve a high embedding capability with less distortion. The henceforth complex computational level at scrambling with transform level steganography is the most well-put forward approach.

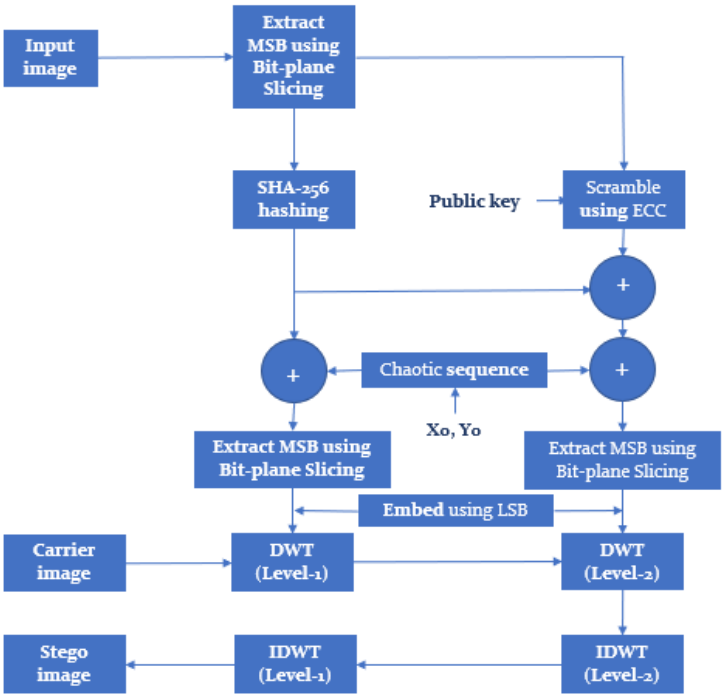

Fig.1. Scramble and Embed at the sender

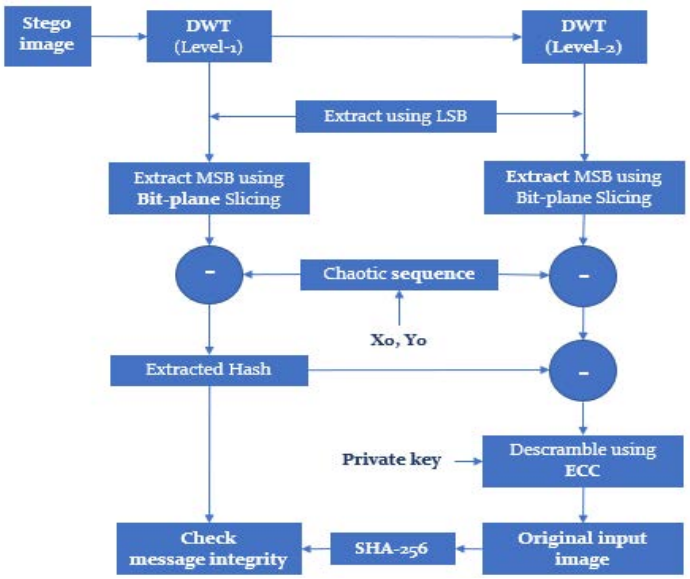

Fig.2. Extract and descramble at the receiver

4. Methodology

In the proposed scheme, the input plain image is processed through three levels. In the first-level and last-level, the spatial domain is considered. The very important stage is the second stage where the transform domain is considered for processing the image. In the first level, Bit-plane slicing is used to extract only the MSB planes of the input image. In the second level, ECC with modifications and the logistic chaotic map is adopted to scramble the input image. Also, generate a hash of the input by using SHA-256 and it is combined with a scrambled image to verify the message integrity. The DWT is applied to the carrier image to get into four sub-bands as LL1, LH1, HL1, and HH1. Again, consider the LL1 sub-band to apply DWT to get 2nd level sub-bands as LL2, LH2, HL2, and HH2. At the last stage, the scrambled image and combined image are embedded in the high-frequency sub-bands of 1st and 2nd level DWT. The

least significant bits at high-frequency bands are considered for the embedding process, thereby generating the secure stego-image. As DWT is applied for a carrier with two-level, benefitted to achieve more PSNR and more embedding capacity. A reverse procedure is applied on the receiver side to extract and embed. The block diagram of the proposed method is depicted in Fig.1. & Fig.2. The mapping process is shown in Fig.3.

Table 1. The notations used in the proposed method

Symbols	Descriptions
Ep (a, b)	Elliptic curve points with a=4, b=20, p=251
G (x1, y1)	Generator with x & y coordinates where x1=0, y1=32
A_priv	Private key at sender A
A_pub	Public key at sender A
(A_pub_x, A_pub_y)	Public key with x & y coordinates
r_rand	Random number
R_rand	Random number after multiplying with generator
(R_rand_x, R_rand_y)	Random number with x & y coordinates
Isecret_image	Input secret image
Pi (x, y)	Input secret image with x & y coordinates
B_priv	Private key at receiver B
B_pub	Public key at receiver B
(B_pub_x, B_pub_y)	Public key with x & y coordinates
Ci (x, y)	Cipher image with x & y coordinates
Cscrambled	Scrambled image
(I_{rm}, I_{gm}, I_{bm})	Red, green, & blue channels of the input image
$I_{rm8}, I_{rm7}, I_{rm6}, I_{rm5}, I_{rm4}, I_{rm3}, I_{rm2}, I_{rm1}$	Eight planes of the input image
I_{msb_im}	MSB plane image generated by combining upper four channels (I_{rm8}, I_{rm7}, I_{rm6}, I_{rm5})
I_{lsb_im}	LSB plane image generated by combining lower four channels (I_{rm4}, I_{rm3}, I_{rm2}, I_{rm1})
I_{scramb}	Scrambled image after applying ECC on MSB image
I_{hash}	Hash image of the MSB image
I_{scramb_hash}	Hash added to scrambled image
I_{chaos}	Chaos generated for the input image
I_{sr_chaos}	Chaos added to scrambled image
I_{ha_chaos}	Chaos added to hash image
I_{cover}	Cover image
[LLc1, LHc1, HLc1, HHc1]	Level-one approximation & detail coefficients matrices generated from cover image
[LLc2, LHc2, HLc2, HHc2]	Level-two approximation & detail coefficients matrices generated from LLc1 matrix
dwt2	2D Discrete wavelet transform
haar	Haar wavelet transform
idwt2	2D Inverse discrete wavelet transform
I_{stego}	Stego image
[LLs1, LHs1, HLs1, HHs1]	Level-one approximation & detail coefficients matrices generated from stego image
[LLs2, LHs2, HLs2, HHs2]	Level-two approximation & detail coefficients matrices generated from LLs1 matrix
I_{hash_calc}	Calculated hash from I_{hash} & I_{masb_im}

4.1. Scramble and descramble Process

Security of input image is provided by scrambling it using the ECC algorithm. The origination of the secret key is based on the prime number and selected global point. The secret key origination differs on the prime number selected for the identical carrier image. The following steps demonstrate the ECC algorithm:

- Consider an elliptic curve *Ep (a, b)* and find all possible points.
- Assign each point on the curve to each pixel value in an input Gray-scale image in the range from 0 to 255, as shown in Fig.3 (a) to (e).
- Choose generator *G (x1, y1)* with large order *n* in *Ep (a, b)*
- At sender A**,** choose private key and compute public key using generator *'G'* as in (1) & (2)

$$A\_priv = x \tag{1}$$

$$A\_pub = X = x * G(x1, y1) = (A\_pub\_x, A\_pub\_y) \tag{2}$$

- Choose another random number *'r_rand'* and multiply with *'G'* as in (3)

$$R\_rand = r\_rand * G(x1, y1) = (R\_pub\_x, R\_pub\_y) \quad (3)$$

- Read the pixel from input secret image '*Isecret_image'* and map to the corresponding point which are generated in step ii. Repeat this process for all the pixels in an input image as shown in Fig.3 & Fig.4.
- Now, all pixels are replaced by *(x, y)* coordinate points as *Pi (x, y)*, shown in Fig.4.
- Also at receiver B, choose a private key and compute the public key using generator '*G'* as in (4) & (5)

$$B\_priv = y \quad (4)$$

$$B\_pub = Y = y * G(x1, y1) = (B\_pub\_x, B\_pub\_y) \quad (5)$$

- Equation (6) is used to compute cipher by considering each pixel of an input secret image *Pi (x, y)* using '*A_priv'* and '*B_pub'* as

$$Ci(x1, y1) = Pi(x, y) + (A\_priv * B\_pub) + (R\_rand) \quad (6)$$

where $i = 0, 1, 2, \ldots\ldots 255$

- The newly generated coordinate in *Ci (x1, y1)* is again mapped to a new pixel value according to the index values as shown in Fig. 3. (b), resulting in a final scrambled image '*Cscrambled'*. This is presented in Fig.3. (d) & Fig.3. (e).
- Now, send '*Cscrambled'* by embedding it in a cover image using DWT.
- At the receiver, descramble the cipher to get back the actual plaintext using as in (7)

$$Pi(x, y) = Ci(x1, y1) - (B\_priv * A\_pub) - (R\_rand) \quad (7)$$

where $i = 0, 1, 2, \ldots\ldots 255$

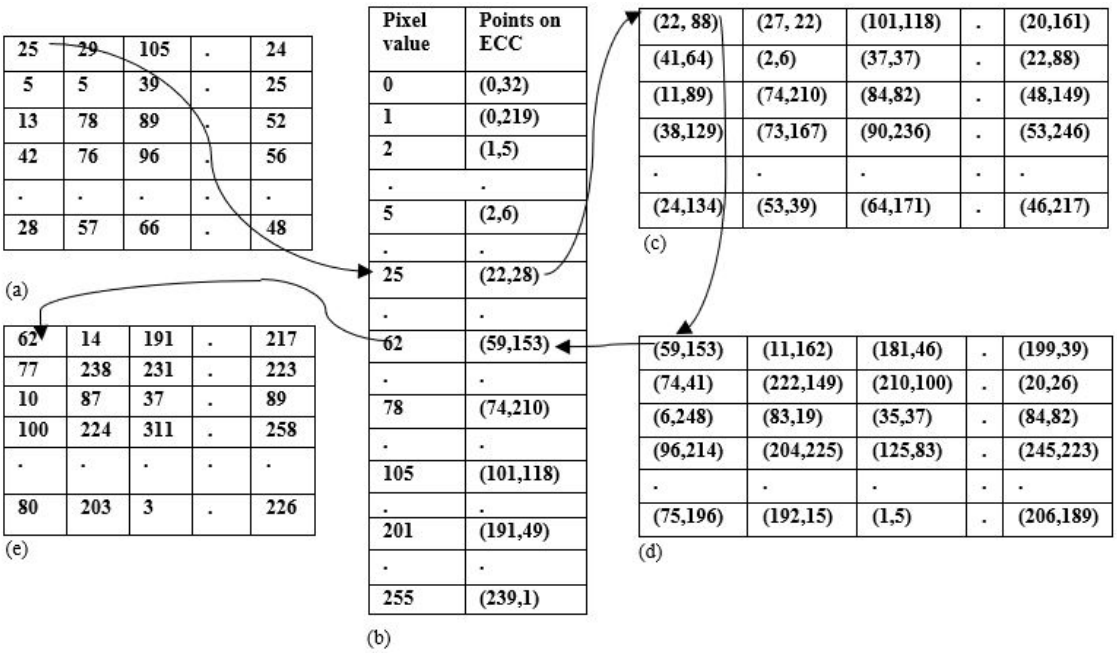

Fig.3. (a). Grayscale image with pixel value, (b). Index values for the elliptic curve Ep (a, b) where a=4, b=20, p=251, G (x1, y1) = (0, 32), (c). Pixel mapped to ECC points according to index values, (d). Encrypted ECC points, & (e). ECC points mapped to pixel according to index values.

4.2. Embed and extract Process

The second level of security for the scrambled image can be provided by embedding it in a cover image. The robustness of the system can be enhanced by fully utilizing the properties of the cover image. The DWT decomposes the cover image into different sub-bands as approximation and details. Approximation includes flat sub-band whereas details include noisy sub-bands. So, it is always a better choice to make use of a noisy sub-band rather than the flat sub-

band to embed a secret message. The DWT is a frequency domain method. This method is more complicated and time-consuming than spatial domain methods. Anyway, DWT is more shielded and invulnerable to noises. The Least Significant Bits (LSB) technique [1] is used to replace the LSB in the noisy sub-band of the cover image with the bits of the encrypted data without damaging its features. This technique is easy to implement and understand. Hashing Algorithms are used to check the integrity of the secret data which must be sent to the recipient. SHA-256 is a patented cryptographic hash function that generates a value of 256 bits in length.

The steps to embed are as follows:

- Read input image.
- Separate R, G, B channels (I_{rm}, I_{gm}, I_{bm}) from the input image.
- Consider R-channel image, I_{rm} of size (N_1 X N_2).
- Apply Bit-plane slicing on R-channel image I_{rm} and generate eight bit-planes as in (8)

$$I_{rm8}I_{rm7}I_{rm6}I_{rm5}I_{rm4}I_{rm3}I_{rm2}I_{rm1} = Bit-planeSlicing\left(I_{rm}\right) \tag{8}$$

- Separate MSB, LSB planes and generate equivalent images using as in (9) & (10)

$$I_{msb\_im} = \left(I_{rm8} * 2^7 + I_{rm7} * 2^6 + I_{rm6} * 2^5 + I_{rm5} * 2^4\right) \tag{9}$$

$$I_{lsb\_im} = \left(I_{rm4} * 2^3 + I_{rm3} * 2^2 + I_{rm2} * 2^1 + I_{rm1} * 2^0\right) \tag{10}$$

- Since the data is present in I_{msb_im} and noise in I_{lsb_im}, take only the MSB plane image I_{msb_im} from (9) for further processing.
- Apply ECC scrambling with modifications on I_{msb_im} as in (11)

$$I_{scramb} = ECC\left(I_{msb\_im}\right) \tag{11}$$

- Find hash I_{hash} of the I_{msb_im} using SHA-256 and add this to I_{scramb} as in (12) & (13)

$$I_{hash} = sha256hasher.ComputeHash\left(I_{msb\_im}\right) \tag{12}$$

$$I_{scramb\_hash} = I_{hash} + I_{scramb} \tag{13}$$

- Generate chaotic sequence I_{chaos} for the input image I_{rm} using initial conditions X_0, Y_0
- Add chaotic sequence I_{chaos} to I_{scramb} and I_{scramb_hash} as in (14) & (15)

$$I_{sr\_chaos} = I_{scramb} + I_{chaos} \tag{14}$$

$$I_{ha\_chaos} = I_{scramb\_hash} + I_{chaos} \tag{15}$$

- Extract MSB planes from I_{sr_chaos} and I_{ha_chaos} generating 1st scrambled and 2nd scrambled images.
- Read the cover image I_{cover}
- Equation (16) represents the decomposition of the cover image using single-level 2D DWT to get approximation coefficients matrix *'LLc1'* and detail coefficients matrices *'LHc1', 'HLc1', and 'HHc1'* as horizontal, vertical, and diagonal, respectively.

$$\left[LLc1,\ LHc1,\ HLc1,\ HHc1\right] = dwt2\ (Icover, 'haar') \tag{16}$$

- Embed 2nd scrambled image 'I_{ha_chaos}' image in '*HHc1*' detail coefficient matrix using LSB method.
- Equation (17) represents the decomposition of approximation coefficients matrix *'LLc1'* using single-level 2D DWT to get approximation coefficients matrix *'LLc2'* and detail coefficients matrices *'LHc2', 'HLc2', and 'HHc2'* as horizontal, vertical, and diagonal, respectively.

$$\left[LLc2,\ LHc2,\ HLc2,\ HHc2\right] = dwt2\ \left(LLc1,\ 'haar'\right) \tag{17}$$

- Embed 1st scrambled image 'I_{sr_chaos}' in *'HHc2'* detail coefficient matrix using LSB method.
- Reconstruct the image *'LLc1'* by using a single level 2D Inverse DWT (IDWT) based on the approximation matrix *'LLc2'* and details matrices *'LHc2', 'HLc2', and 'HHc2'* as horizontal, vertical, and diagonal, respectively as in (18)

$$LLc1 = idwt2\ (LLc2,\ LHc2,\ HLc2,\ HHc2, 'haar') \tag{18}$$

- Reconstruct the image by using a single level 2D Inverse DWT (IDWT) based on the approximation matrix 'LLc1' and details matrices *'LHc1', 'HLc1' and 'HHc1'* as horizontal, vertical, and diagonal, respectively. This stage generates the stego image I_{stego} as in (19)

$$I_{stego} = idwt2\left(LLc1, LHc1, HLc1, HHc1, 'haar'\right) \tag{19}$$

The steps to extract are as follows:

- Consider the stego image I_{stego}.
- Decompose the I_{stego} using single-level 2D DWT to get approximation coefficients matrix *'LLs1'* and detail coefficients matrices *'LHs1', 'HLs1', and 'HHs1'* as horizontal, vertical, and diagonal, respectively as in (20)

$$\left[LLs1, LHs1, HLs1, HHs1\right] = dwt2\left(I_{stego}, 'haar'\right) \tag{20}$$

- Extract 'I_{ha_chaos}' from *'HHs1'* using the LSB method.
- Decompose again the *'LLs1'* matrix using single-level 2D DWT to get approximation coefficients matrix *'LLs2'* and detail coefficients matrices *'LHs2', 'HLs2', and 'HHs2'* as horizontal, vertical, and diagonal respectively, as in (21)

$$\left[LLs2, LHs2, HLs2, HHs2\right] = dwt2\left(LLs1, 'haar'\right) \tag{21}$$

- Extract 'I_{sr_chaos}' the scrambled image from *HHs2* using the LSB method.
- Subtract chaotic sequence from extracted images, as in (22) & (23)

$$I_{scramb} = I_{sr\_chaos} - I_{chaos} \tag{22}$$

$$I_{scramb\_hash} = I_{ha\_chaos} - I_{chaos} \tag{23}$$

- Subtract 'I_{scramb}' from I_{scramb_hash} to get I_{hash}, as in (24)

$$I_{hash} = I_{scramb\_hash} - I_{scramb} \tag{24}$$

- Descramble 'I_{scramb}' using the ECC algorithm to get back the actual input image, as in (25)

$$I_{msb\_im} = ECC\left(I_{scramb}\right) \tag{25}$$

- Find hash of an obtained image I_{msb_im} , as in (26)

$$I_{hash\_calc} = sha256hasher.ComputeHash\left(I_{msb\_im}\right) \tag{26}$$

- Equations (24) and (26) are now used to verify the two hash values 'I_{hash}' and 'I_{hash_calc}' whether they are the same or different. If it is the same, indicates that the sent scrambled image has not been modified by the intruder. Otherwise, it is modified. This process is carried out to check for message integrity.

5. Results and Performance Analysis

The set of input images and cover images used for the proposed work are shown in Fig.4. Input secret and cover images are used from USC-SIPI [17] and KODAK [18] Image datasets. Fig.5 shows the ECC plot based on the points generated for a=4, b=20, p=251. Bit-plane slicing of the input image, its MSB and LSB plane images are shown in Fig.6.

(a), (b), (c) & (d). The input image shown is a San Diego.tiff. Scrambled image generated from ECC & logistic chaotic map is shown in Fig.7. (a) and hashed image added to scrambled is shown in Fig.8. (a). Its equivalent bit-plane slicing, MSB & LSB images are shown in Fig.7. (b), (c) & (d) and Fig.8. (b), (c) & (d) respectively. The cover image, its two-level 2D DWT, stego image, and the descrambled image are shown in Fig.9. (a), (b), (c) & (d). The cover image shown is a Mandril.tiff. Histograms of all these are presented in Fig.10. Histograms of the input secret image, its ECC scrambled image, its hash added to scrambled images are shown in Fig.10. (a), (b) & (c). It is observed that the histogram of the scrambled image indicates the uniform distribution of information, which makes any viewer judge the tonal distribution. This shows more randomness is achieved from the proposed approach. Histograms of the cover image, its embedded image, and descrambled image are shown in Fig.10. (d), (e) & (f). Histograms of the cover & stego image are almost similar. Nobody can identify the presence of a secret image in the cover image. It is very much difficult for an attacker to identify the hidden image and try to extract it. So, this system is more secure.

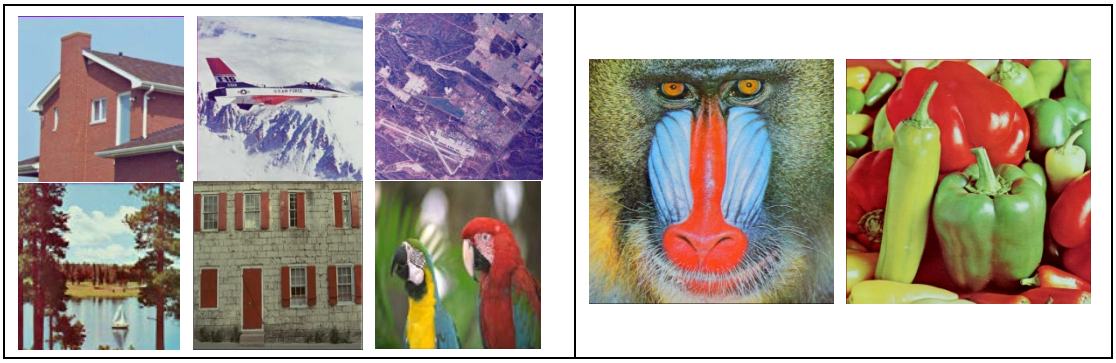

Fig.4. Set of input secret images (House.tiff, Airplane.tiff, San Diego.tiff, Tree.tiff, kodim01.png, kodim23.png) and set of cover images

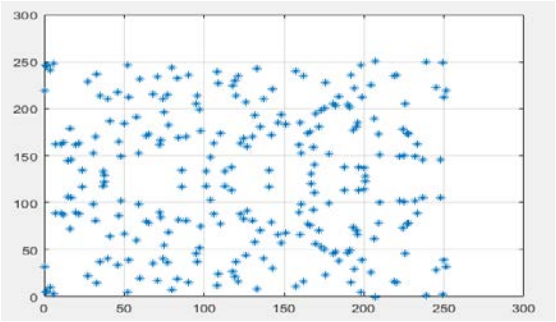

Fig.5. Plot of ECC points for a=4, b=20, p=251

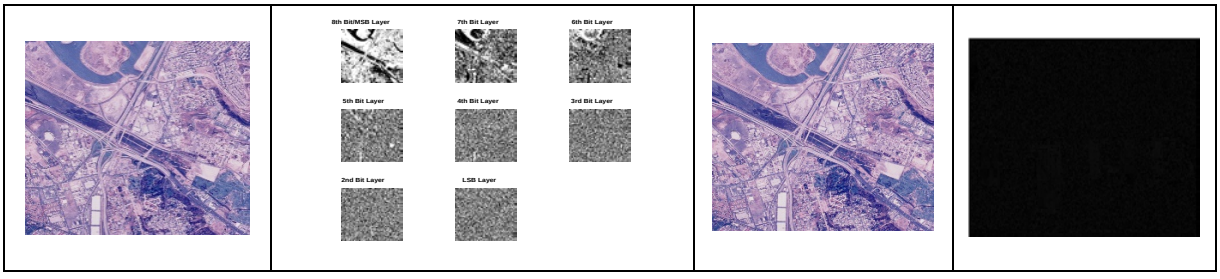

Fig.6. (a). Input Image (San Diego) (b). Bit-plane slicing (c). MSB Image and (d). LSB Image (noisy image)

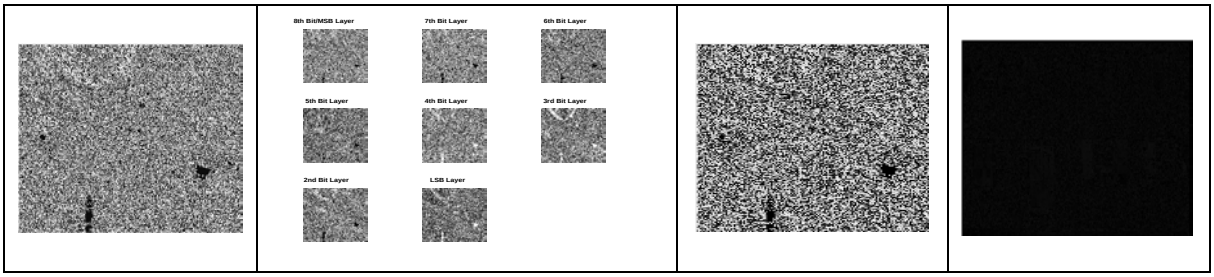

Fig.7. Images of (a). Scrambled 1 (b). It's equivalent Bit-plane slicing (c). MSB Image and (d). LSB Image (noisy image)

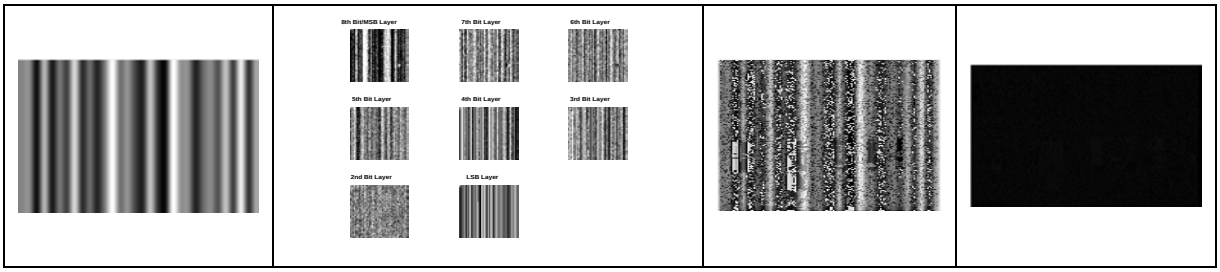

Fig.8. Images of (a). Scrambled 2 (b). It's equivalent Bit-plane slicing (c). MSB Image and (d). LSB Image (noisy image)

Fig.9. (a). Cover image (Mandril.tiff) (b). Two-level 2D DWT (c). Stego Image (d). Descrambled image

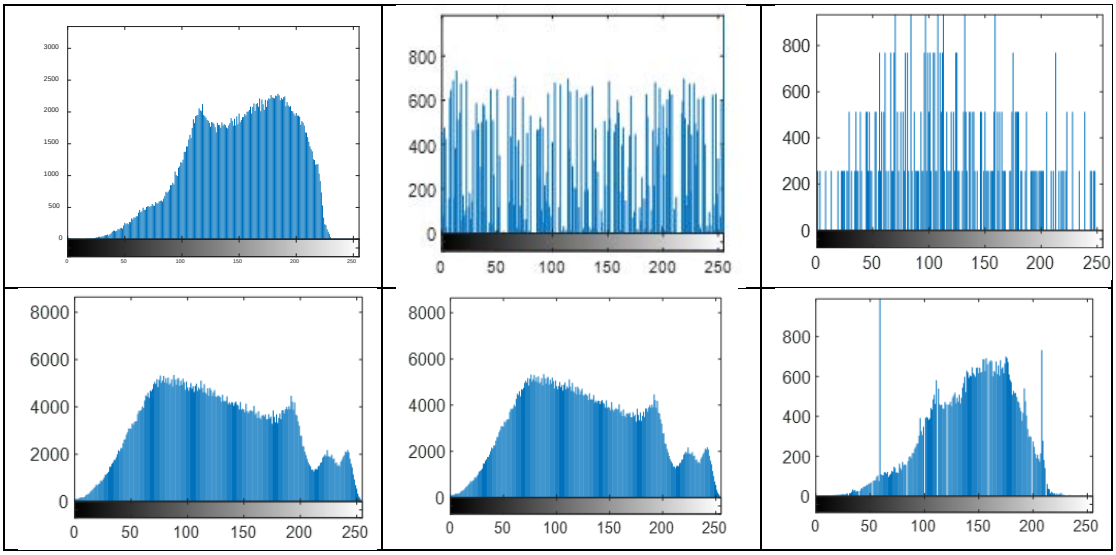

Fig.10. Histograms of (a). Input image, (b). Scrambled 1, (c). Scrambled 2 (d). Cover (e). Stego and (f) Descrambled

5.1. Performance analysis

To evaluate the performance of the proposed algorithm, many analyses such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM), Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI) and Correlation Coefficient are measured. Simulation is carried on MATLAB 9.7 version.

A. MSE, PSNR, and SSIM

MSE is a parameter used to find the signal loss. The nature of the obtained image is better if MSE is lesser. MSE can be calculated by (27)

$$MSE = \frac{1}{M * N} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \left[f(p,q) - g(p,q) \right]^2 \tag{27}$$

Where M, N = total rows & columns in picture, $f(p, q)$ = input picture, $g(p, q)$ = output picture

PSNR is for estimating the imperceptibility of the reconstruction of an image. The higher PSNR value indicates a higher signal to noise ratio, given by (28)

$$PSNR_{dB} = 10\log_{10}\left(\frac{255^2}{\sqrt{MSE}}\right) \tag{28}$$

SSIM is used to indicate the similarity between the cover and stego. The range is from 0 to 1. It is efficient for higher values [7] and it can be given by (29)

$$SSIM = \frac{(2\mu_{ci}\mu_{si} + a_1)(\sigma_{cisi} + a_2)}{(\mu_{ci}^2 + \mu_{si}^2 + a_1)(\sigma_{ci}^2 + \sigma_{si}^2 + a_2)} \tag{29}$$

Where a_1 *and* a_2 are constants. *ci* and *si* are cover & stego images. Then, μ & σ are average & standard deviation.

B. Number of Pixel Change Rate (NPCR), Unified Average Changing Intensity (UACI), and Information Entropy

NPCR is useful in computing the percent of pixel variation in cipher concerning pixel variation in plain [18]. It can be given by (30)

$$NPCR = \frac{1}{M * N}\sum_{i,j} D(i, j) * 100\% \tag{30}$$

UACI determines the average intensity variation among two images [18]. It can be given by (31)

$$UACI = \frac{1}{255 * M_1 * M_2}\sum_{p,q}[IM_1(p,q) - IM_2(p,q)] * 100\% \tag{31}$$

where M_1 *and* M_2 give image size, $IM_1(p,q)$ and $IM_2(p,q)$ is the cipher with pel values before and after image variation.

Entropy is used as a statistical measure of randomness to determine the texture of an image [7]. It can be calculated by using (32)

$$E = \sum_{i=1}^{X-1} I(Y_i)\log_2 I(Y_i) \tag{32}$$

where *X* is the total symbols, and *I(Yi)* is the probability of the existence of the symbol *(Yi)*.

Two major sections of the proposed method are to scramble first and then embed. First, consider the results of the scrambling process. Six different secret images are taken from the datasets USC-SIPI [17], Kodak [18]. These images are used as input for scrambling using ECC and logistic chaotic map along with SHA-256. Performance measures like PSNR, MSE, SSIM, NPCR, and UACI are calculated for scrambled images. This is presented in Table 2. Lesser PSNR & more MSE gives more scrambling of pixels indicating more randomness in the scrambled image. This becomes difficult for any intruder to reverse back the process. For scrambled images, always SSIM should be very less, NPCR should be 100, and UACI ideal value is more than 33.46. The last row indicates the average of values for all the input secret images. For the proposed method, PSNR as 8.7677 (dB), MSE as 8.88e+03, SSIM as 0.0700, NPCR as 100, and UACI as 33.4635 respectively. The values for performance metrics are satisfying w.r.t to their ideal values.

Table 2. Resultant values of PSNR, MSE, SSIM, NPCR and UACI for Scrambled images

Input images	PSNR	MSE	SSIM	NPCR	UACI
House.tiff	9.7466	6.8931e+03	0.2694	100	33.4635
Airplane.tiff	6.9289	1.3188e+04	0.0108	100	33.4635
San Diego.tiff	9.0391	8.1128e+03	0.0083	100	33.4635
Tree.tiff	8.0042	1.0295e+04	0.0260	100	33.4635
kodim01.png	9.3177	7.6086e+03	0.0279	100	33.4635
kodim23.png	9.5697	7.1797e+03	0.0777	100	33.4635
Average	**8.7677**	**8.88e+03**	**0.0700**	**100**	**33.4635**

Table 3. Performance metrics of PSNR, MSE, SSIM, NPCR, UACI, and Information Entropy for Stego images (Cover image: Mandril.tiff)

Input images	PSNR	MSE	SSIM	NPCR	UACI	Information Entropy	
						Cover image	Stego image
House.tiff	84.5347	0.0079	1	0.0229	33.4635	7.7624	7.7624
Airplane.tiff	84.5347	0.0077	1	0.0228	33.4635	7.7624	7.7624
San Diego.tiff	84.3923	0.0069	1	0.0236	33.4635	7.7624	7.7624
Tree.tiff	84.1207	0.0078	1	0.0252	33.4635	7.7624	7.7624
kodim01.png	84.4867	0.0079	1	0.0231	33.4635	7.7624	7.7624
kodim23.png	84.4867	0.0078	1	0.0231	33.4635	7.7624	7.7624
Average	**84.4259**	**0.0077**	**1**	**0.0235**	**33.4635**	**7.7624**	**7.7624**

Table 4. Performance metrics of PSNR, MSE, SSIM, NPCR, UACI, and Information Entropy for Stego images (Cover image: Peppers.tiff)

Input images	PSNR	MSE	SSIM	NPCR	UACI	Information Entropy	
						Cover image	Stego image
House.tiff	88.3951	0.0082	1	0.0094	33.4635	7.6698	7.6698
Airplane.tiff	88.3950	0.0082	1	0.0094	33.4635	7.6698	7.6698
San Diego.tiff	88.0565	0.0082	1	0.0092	33.4635	7.6698	7.6698
Tree.tiff	87.4495	0.0081	1	0.0116	33.4635	7.6698	7.6698
kodim01.png	88.2792	0.0082	1	0.0096	33.4635	7.6698	7.6698
kodim23.png	88.2793	0.0082	1	0.0096	33.4635	7.6698	7.6698
Average	**88.1424**	**0.0081**	**1**	**0.0098**	**33.4635**	**7.6698**	**7.6698**

The second major part of the proposed work is to analyze the results of the embedding process. Carrier image is transformed using DWT and then use LSB to embed, resulting in stego image. For all the six scrambled images from Table 2, two different carrier images used are Mandril.tiff and Peppers.tiff. Results obtained after embedding in these two images are presented in Tables 3 & 4. For stego, higher PSNR, lesser MSE, SSIM as 1, NPCR should be lesser and UACI ideal value is more than 33.46. If these values are achieved, it represents cover image is embedded with minimum distortion. By considering the carrier image as Mandrill. iff as shown in Table 2, the stego image1 resulted with PSNR as 84.4259 (dB), MSE as 0.0077, SSIM as 1, NPCR as 0.0235, UACI as 33.4635 respectively. By considering the carrier image as Peppers.tiff as shown in Table 3, the stego image2 resulted with PSNR as 88.1424 (dB), MSE as 0.0081, SSIM as 1, NPCR as 0.0098, UACI as 33.4635 respectively. After embedding, carrier and stego images are analyzed for information entropy. Information entropy remains the same even after embedding the data. This is also presented in Tables 3 & 4. These values show that the proposed approach is immune to differential attack.

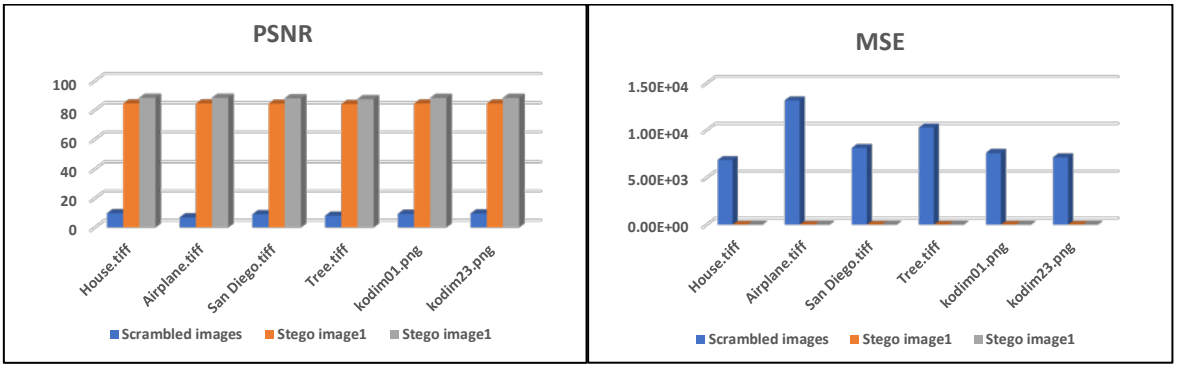

Fig.11. A plot of PSNR and MSE for Scrambled, Stego image1, and Stego image2

The graphs are plotted for scrambled, stego image1, and stego image2 by taking all the six images as input secret images. These are shown in Fig.11. As needed PSNR is lesser for scrambled and higher for stego images. MSE is higher for scrambled and very lesser for stego images.

The performance metrics like PSNR, MSE, SSIM, NPCR, and UACI are also calculated for images of different data sets. The data sets used are USC-SIPI and Kodak. Images of aerials and miscellaneous are used from the USC-SIPI data set. Resultant values for scrambling of images from different data sets are presented in Table 5. The last row indicates the average of values for data sets. For the proposed method, PSNR as 8.9766 (dB), MSE as 8.25e+03, SSIM as 0.0292, NPCR as 100, and UACI as 33.4635 respectively. The values for performance metrics are satisfying w.r.t to their ideal values. These are shown in Tables 5, 6, and 7 for scrambled, stego1, and stego2 images, respectively.

Table 5. Performance metrics of PSNR, MSE, SSIM, NPCR, and UACI of Scrambled images for datasets

Data Sets		Scrambled images				
		PSNR	MSE	SSIM	NPCR	UACI
USC-SIPI	Aerials	8.8022	8.5676e+03	0.0114	100	33.4635
	Miscellaneous	8.6840	8.8038e+03	0.0233	100	33.4635
Kodak		9.4437	7.39e+03	0.0528	100	33.4635
Average		**8.9766**	**8.25e+03**	**0.0292**	**100**	**33.4635**

Scrambled images from Table 5 are embedded in the carrier images Mandril.tiff and Peppers.tiff, which resulted in stego image1 and stego image2. Results obtained after embedding in these two images are presented in Tables 6 & 7. By considering the carrier image as Mandrill.tiff as shown in Table 6, the stego image1 resulted with PSNR as 84.3375(dB), MSE as 0.00785, SSIM as 1, NPCR as 0.0239, and UACI as 33.4635, respectively. By considering the carrier image as Peppers.tiff, the stego image2 resulted with PSNR as 88.2792(dB), MSE as 0.0082, SSIM as 1, NPCR as 0.0106, and UACI as 33.4635 respectively. After embedding, carrier and stego images are analyzed for information entropy. Information entropy remains the same even after embedding the data. This is also presented in Tables 6 & 7. These values show that the proposed approach is immune to differential attack.

Table 6. Performance metrics of PSNR, MSE, SSIM, NPCR, and UACI of Stego images for datasets (Cover image: Mandril.tiff)

Data Sets		Stego images				
		PSNR	MSE	SSIM	NPCR	UACI
USC-SIPI	Aerials	84.5347	0.0079	1	0.0229	33.4635
	Miscellaneous	83.9911	0.0078	1	0.0259	33.4635
Kodak		84.4867	0.00785	1	0.0231	33.4635
Average		**84.3375**	**0.00785**	**1**	**0.0239**	**33.4635**

Table 7. Performance metrics of PSNR, MSE, SSIM, NPCR, and UACI of Stego images for datasets (Cover image: Peppers.tiff)

Data Sets		Stego image				
		PSNR	MSE	SSIM	NPCR	UACI
USC-SIPI	Aerials	88.3950	0.0082	1	0.0094	33.4635
	Miscellaneous	87.0014	0.0081	1	0.0130	33.4635
Kodak		88.2792	0.0082	1	0.0096	33.4635
Average		**87.8919**	**0.0082**	**1**	**0.0106**	**33.4635**

The proposed approach gives less PSNR (8.9766 dB), more MSE (8.25e+03), and SSIM (0.0292) values compared with the other method in [8] for scrambled images. This is presented in Table 8. The plots for PSNR, MSE, and SSIM in comparison with other existing methods are shown in Fig.12.

Table 8. Comparison of PSNR, MSE, SSIM with other existing methods for scrambled images

Other existing methods	PSNR	MSE	SSIM
Dhanesh Kumar et al [5]	8.2857	8.9430e+03	0.007133
Proposed method	**8.9766**	**8.25e+03**	**0.0292**

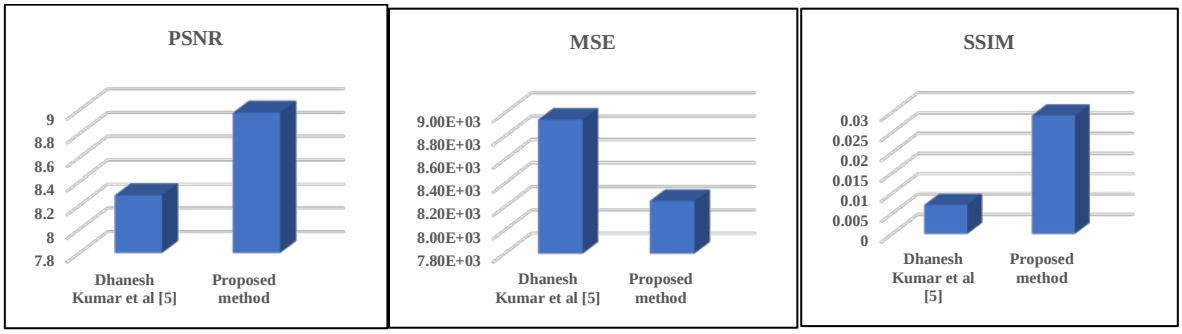

Fig.12. A plot of PSNR, MSE & SSIM for the proposed method when compared to the existing method [5].

The proposed approach gives NPCR as 100 and UACI 33.4635 values compared with other existing methods [7, 2] for scrambled images. These are presented in Table 9. The plots for NPCR & UACI in comparison with other existing methods are shown in Fig.13.

Table 9. Comparison of NPCR & UACI with other existing methods for scrambled images

Other existing methods	NPCR	UACI
Youxia Dong et al [18]	99.6125	33.4776
Abhilash Ashok Bhadke [2]	99.5963	33.4666
Proposed method	**100**	**33.4635**

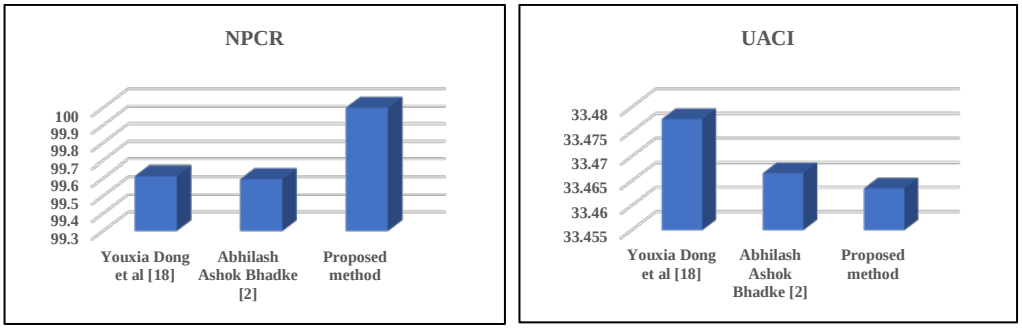

Fig.13. A plot of NPCR & UACI for the proposed method when compared to existing methods [18, 2].

The proposed approach gives more PSNR as 87.8919 dB and less MSE as 0.0082 values compared with other existing methods [4, 1] for stego images. This is presented in Table 10. The plots for PSNR and MSE in comparison with other existing methods are shown in Fig.14.

Table 10. Comparison of PSNR, MSE with other existing methods (Cover image: Peppers.tiff)

Other existing methods	PSNR	MSE
Eshraq S. Hureib and Adnan A. Gutub [6]	74.6214	0.0130
A. Hambouz, et al [1]	87.4905	0.0094
Proposed method	**87.8919**	**0.0082**

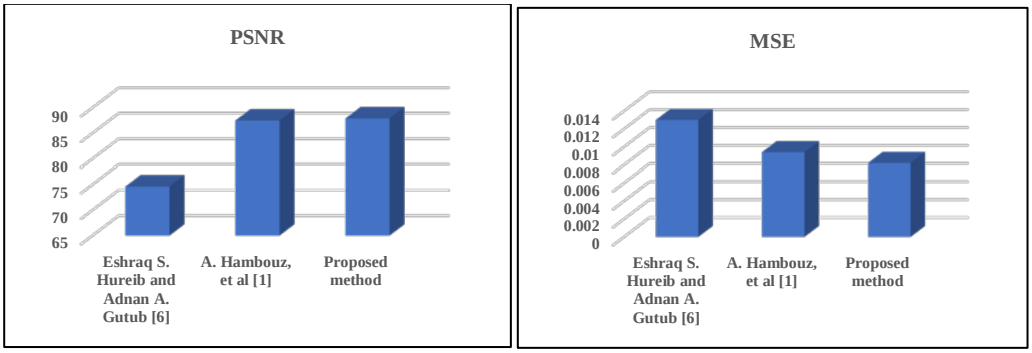

Fig.14. A plot of PSNR & MSE for the proposed method when compared to existing methods [5, 7, 1].

C. Correlation Coefficient

The correlation coefficient characterizes the interrelation among any two variables [7]. If the value of coefficient outreach to zero, then the image has been ciphered with a better system. The correlation coefficient is computed for horizontal, vertical & diagonal positions of input and cipher images. It can be calculated by (33)

$$C_{x1,x2} = \frac{1}{\sigma_{x1} * \sigma_{x2}} Cov(x1, x2) \tag{33}$$

Where $C_{x1, x2}$ is the correlation coefficient, *Cov* is the covariance of variables *x1 & x2*, σ_{x1} & σ_{x2} are standard deviations of x1 & x2, respectively.

The correlation coefficient values for input & scrambled are presented in Table 11 for three different secret images. The horizontal, vertical & diagonal correlation coefficients are computed for R, G, & B channels of input secret images. It is observed that the correlation coefficient is very less for the scrambled image when compared to an input image. The correlation coefficient for input, scrambled, cover, and stego images are presented in Fig.15. (a), (b), (c), & (d). It is observed that the correlation coefficient for the scrambled image is uniformly distributed. The correlation coefficient of

a pixel with the neighboring pixel is very less that indicating more randomness in the scrambled image. Carrier and stego images are almost the same, which indicates undetectability of embedded data is good. So, the proposed method gives a better scrambling and embedding effect with a higher security level.

Table 11. Correlation Coefficient values for input and scrambled images

Input Secret images	Three channels	Input			Scrambled		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
kodim01.png	R	0.9141	0.9239	0.9202	0.1125	0.1324	0.1017
	G	0.8660	0.8600	0.8778	0.0131	0.1022	0.0321
	B	0.9115	0.9066	0.9066	0.1301	0.1120	0.0250
kodim23.png	R	0.9224	0.9229	0.9254	0.3571	0.3449	0.3559
	G	0.8671	0.8615	0.8672	0.1328	0.1323	0.1025
	B	0.9050	0.9105	0.9089	0.2539	0.2251	0.2256
San Diego.tiff	R	0.9266	0.9305	0.9230	0.0264	0.0124	0.0333
	G	0.8680	0.8673	0.8718	0.0011	0.0022	0.0132
	B	0.9032	0.9107	0.9063	0.0321	0.0012	0.0112

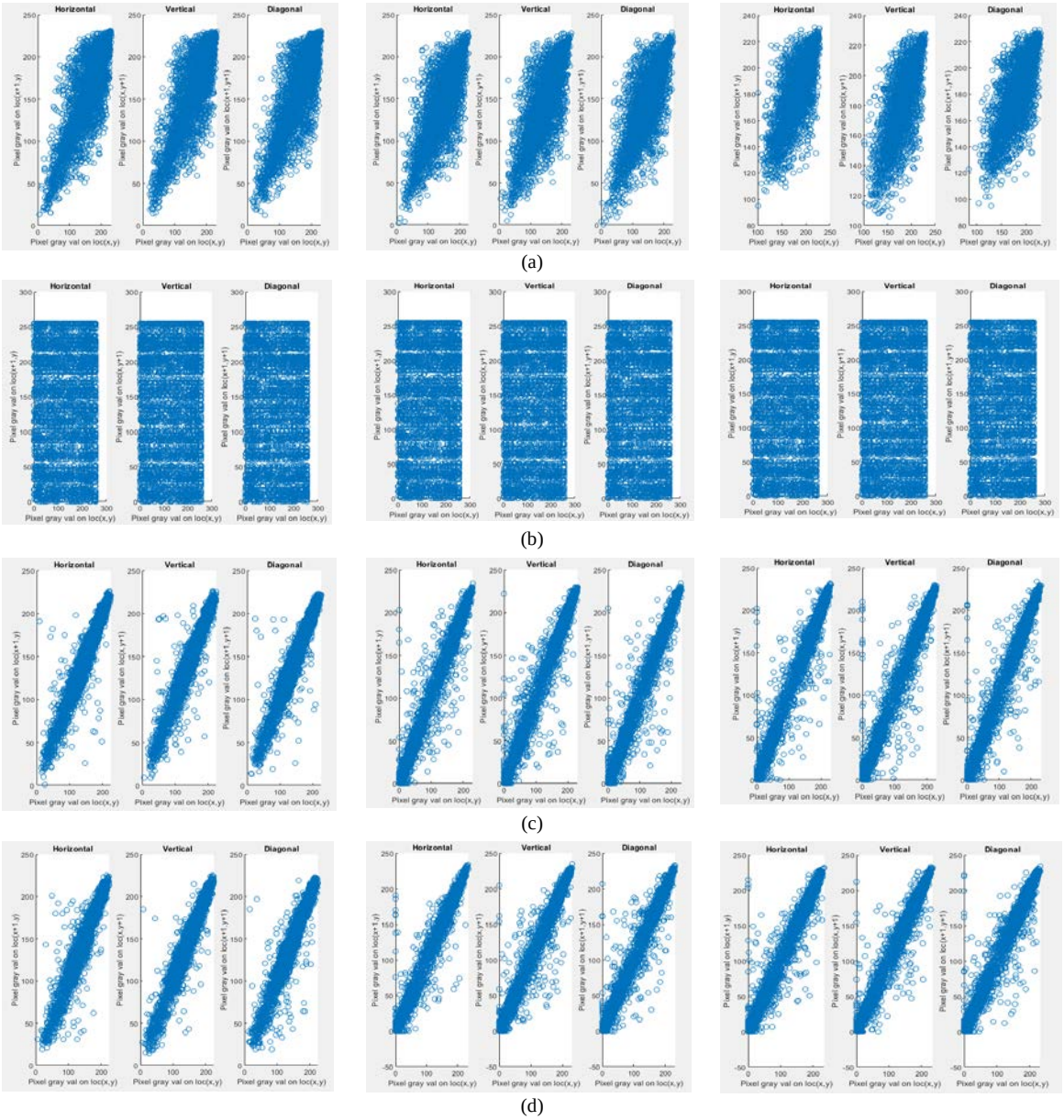

Fig.15. The correlation coefficient for (a). Input secret image (b). Scrambled image (c). Cover image and (d). Stego image for R-, G-, & B-channels equivalently.

5.2. Discussions

The proposed method is analyzed for performance metrics like PSNR, MSE, SSIM, NPCR, UACI, information entropy, and correlation coefficient. This method is successful with good results for all the performance metrics and helpful to achieve research objectives such as confidentiality, integrity & undetectability of data. As bit plane slicing is used for the entire process only MSB planes are utilized. Time taken to process the LSB noisy image is removed. More randomness in the scrambled image results in more confidentiality. Such images cannot be extracted by any intruder unless descramble it with a private key. Using ECC a stronger public and a private key combination is applied in the proposed method. SHA-256 is a one-way function, which cannot be reversed even if it is compromised, thereby helpful to achieve message integrity. A stronger transform domain technique like DWT with LSB resulted in a good undetectable stego image. Instead of placing the raw image as it is in the high-frequency sub-bands, it is processed through hashing & scrambling and then it is embedded in the transform-based DWT. In ahigh-frequency sub-bands, the least significant bit in a byte is used to embed the scrambled data. As data is scrambled, transformed and then it is embedded in LSB, it becomes difficult for any third-party to extract the actual data. Therefore, the proposed method greatly enhances the security of images.

6. Conclusion and Future Scope

An efficient crypto-stego scheme is proposed for images. The proposed scheme uses bit-level scrambling of images by applying the bit-plane slicing method to reduce the scrambling time. To enhance the security of the input plain images, scrambling and embedding have been carried out using more stronger algorithms like ECC, logistic chaotic map, and DWT. A message integrity check is done using a hashing algorithm. The purpose of combining spatial and transform domain techniques is to reduce the processing time. Scrambling is done only on the most significant bit planes of the input, reducing the computational time. DWT and LSB are used in the embedding and extraction process. Scrambled and stego image outputs show that the algorithm gives better statistical measures. Along with scrambled, a hash of an input image is embedded in two different levels of DWT. The effectiveness of the proposed method is analyzed by measuring various security parameters. For stego, more than 75 dB of PSNR, very less MSE, SSIM as 1, NPCR approximately 100, UACI as 33.4635, better information entropy, shows that proposed method gives good imperceptible steganographic system. The proposed method gives good results for the performance metrics PSNR, MSE & SSIM when compared to other methods such as hashing with embedding, symmetric chaos with bit planes, ECC Elgamal with chaos, ECC with embedding, DWT & Schur decomposition of images. A very less correlation coefficient value shows that the ciphering method has resulted in more randomness in the scrambled image. Computational time to process the entire image is reduced as only MSB planes of bit-plane slicing are used in this approach. This approach is secure and suitable for applications of scrambling & hiding of images. In the future, scramble and embed only the region of interest from the input plain image to make it faster and more robust for real-time crypto-steganographic applications.

References

[1] A. Hambouz, Y. Shaheen, A. Manna, M. Al-Fayoumi, and S. Tedmori, “Achieving data integrity and confidentiality using image steganography and hashing techniques,” In 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS), pp. 1–6, IEEE, 2019.

[2] Abhilash Ashok Bhadke, Surender Kannaiyan, and Vipin Kamble. “Symmetric Chaos-Based Image Encryption Technique on Image Bit-Planes using SHA-256”, 2018 Twenty Fourth National Conference on Communications (NCC), pp. 1-6, IEEE, 2018.

[3] Aumreesh Kumar Saxena, Sitesh Sinha, Piyush Shukla," Design and Development of Image Security Technique by Using Cryptography and Steganography: A Combine Approach", International Journal of Image, Graphics and Signal Processing, Vol.10, No.4, pp. 13-21, 2018.

[4] Eshraq S. Hureib, Adnan A. Gutub, “Enhancing medical data security via combining elliptic curve cryptography and image steganography”, International Journal of Computer Science and Network Security (IJCSNS), Vol.20, No.8, pp. 1-8, August 2020.

[5] U. A. Md. Ehsan Ali, Emran Ali, Md. Sohrawordi, Md. Nahid Sultan," A LSB Based Image Steganography Using Random Pixel and Bit Selection for High Payload ", International Journal of Mathematical Sciences and Computing, Vol.7, No.3, pp. 24-31, 2021.

[6] Gaurav Kumar, Rajeev Kumar, " Analysis of Arithmetic and Huffman Compression Techniques by Using DWT-DCT", International Journal of Image, Graphics and Signal Processing, Vol.13, No.4, pp. 63-70, 2021.

[7] Youxia Dong, Xiaoling Huang, Guodong Ye, “Visually Meaningful Image Encryption Scheme Based on DWT and Schur Decomposition”, Security and Communication Networks, pp. 1-16, 2021. DOI:10.1155/2021/6677325

[8] Dhanesh Kumar, Anand B. Joshi, Sonali Singh, Vishnu Narayan Mishra, “Digital color-image encryption scheme based on elliptic curve cryptography ElGamal encryption and 3D Lorenz map”, In AIP Conference Proceedings 2364, pp. 020026-1 to 020026-13, 2021. DOI:10.1063/5.0062877.

[9] Sonali Rout, Ramesh Kumar Mohapatra, “Video Steganography using Curvelet Transform and Elliptic Curve Cryptography”, In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-7.

IEEE, 2020
[10] Somayyeh Jafarali Jassbi, Ashkan Emami Ale Agha, " A New method for Image Encryption Using Chaotic Permutation", International Journal of Image, Graphics and Signal Processing, Vol.12, No.2, pp. 42-49, 2020.
[11] Ping Pan, Zeming Wu, Chen Yang, Bing Zhao, "Double-Matrix Decomposition Image Steganography Scheme Based on Wavelet Transform with Multi-Region Coverage", Entropy, Vol. 24, no. 2, 2022. https://doi.org/ 10.3390/e24020246.
[12] Mustaqim Abrar M., Pal A., Shashriar Sazzad T.M, "Bit Plane Slicing and Quantization-Based Color Image Watermarking in Spatial Domain", In: Uddin M.S., Bansal J.C. (eds) Proceedings of International Joint Conference on Advances in Computational Intelligence. Algorithms for Intelligent Systems. Springer, Singapore, pp.371-383, 2021. https://doi.org/10.1007/978-981-16-0586-4_30.
[13] Priyadharshini A, Umamaheswari R, Jayapandian N, Priyananci S, "Securing Medical Images using Encryption and LSB Steganography", 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), pp. 1-5, 2021. DOI: 10.1109/ICAECT49130.2021.9392396
[14] Xintao Duan, Daidou Guo, Nao Liu, Baoxia Li, Mengxiao Gou, Chuan Qin, "A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network", IEEE Access, pp. 25777-25788, 2020. DOI: 10.1109/ACCESS.2020.2971528
[15] Aree A. Mohammed, Dilman A. Salih, Ari M. Saeed, Mohammed Q. Kheder, "An imperceptible semi-blind image watermarking scheme in DWT-SVD domain using a zigzag embedding technique", Multimedia Tools and Applications, Vol. 79, No. 43, pp.32095-32118, 2020. DOI: 10.1007/s11042-020-09694-9.
[16] Shafique, Arslan, Jameel Ahmed, Mujeeb Ur Rehman, Mohammad Mazyad Hazzazi, "Noise-Resistant Image Encryption Scheme for Medical Images in the Chaos and Wavelet Domain", IEEE Access, Vol. 9, pp.59108-59130, 2021. DOI: 10.1109/ACCESS.2021.3071535
[17] USC-SIPI Image Database Website. http://sipi.usc.edu/database
[18] KODAK Image Dataset Website. http://r0k.us/graphics/kodak

Authors' Profiles

Ganavi M is working as an Assistant Professor in the Department of Computer Science & Engineering (CSE) at Jawaharlal Nehru New College of Engineering (JNNCE), Shivamogga, Karnataka, India. She is pursuing a Ph.D. in Computer Science and Engineering from the Department of CSE, JNNCE, Shivamogga. Her research interests include Cryptography and information security.

Prof. Prabhudeva S is working as a Professor and Director in the Department of Master of Computer Applications (MCA) at Jawaharlal Nehru New College of Engineering (JNNCE), Shivamogga, Karnataka, India. He has received his Ph.D. degree in Reliability Engineering from IIT Bombay, India in 2010. Presently three research scholars are pursuing Ph.D. under his guidance. His research interests include Reliable and Security Modelling. He has published 18 papers in international journals and conferences. He has 17 years of research experience.

Hemanth Kumar N P is working as an Assistant Professor in the Department of Computer Science & Engineering (CSE) at East Point College of Engineering and Technology (EPCET), Bengaluru, Karnataka, India. He is pursuing a Ph.D. in Computer Science and Engineering from the Department of CSE, JNNCE, Shivamogga. His research interests include Big Data and Cloud Computing.

How to cite this paper: Ganavi M, Prabhudeva S, Hemanth Kumar N P, "An Efficient Image Steganography Scheme Using Bit-plane Slicing with Elliptic Curve Cryptography and Wavelet Transform", International Journal of Computer Network and Information Security(IJCNIS), Vol.14, No.4, pp.43-59, 2022. DOI:10.5815/ijcnis.2022.04.04