Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.DOI Number

Joint Crypto-Stego Scheme for Enhanced Image Protection with Nearest-Centroid Clustering

Amna Shifa[1], Muhammad S. Afgan[1], Mamoona N. Asghar[1], Martin Fleury[2], Imran Memon[3], Saima Abdullah[1], and Nadia Rasheed[4]

[1]Department of Computer Science and IT, The Islamia University of Bahawalpur, 63100, Punjab, Pakistan
[2]School of Computer Science and Electronic Engineering, Colchester, CO4 3SQ, United Kingdom
[3]College of Computer Science, Zhejiang University, Hangzhou, Zhejiang, China
[4]Department of Computer Systems Engineering, University College of Engineering & Technology, The Islamia University of Bahawalpur , 63100, Punjab, Pakistan.

Corresponding author: M. N. Asghar (e-mail: mamona.asghar@iub.edu.pk).

This research article is the part of Project (National Research Program for Universities (NRPU-2016)) with No: 6282/Punjab/NRPU/R&D/HEC/2016. We appreciate Higher Education Commission (HEC) of Pakistan for the execution of this security project in The Islamia University of Bahawalpur, Pakistan.

ABSTRACT Owing to the exceptional growth of information exchange over open communication channels within the public Internet, confidential transmission of information has become a vital current concern for organizations and individuals. In the proposed content-protection scheme, the decryption key is embedded in the encrypted image by utilizing machine learning, nearest-centroid clustering classifier, followed by Least Significant Bit matching (LSB-M) in the spatial domain. An image is first encrypted with the Advanced Encryption Standard (AES) algorithm in output feedback (OFB) mode, after which the AES key is embedded into the encrypted image. Preliminary nearest-centroid clustering followed by shuffling the sequence of pixels within the clusters before applying LSB-M makes any attack more complex, as the bits of the key are further dispersed within the encrypted image. In terms of contributions, one contribution is the direct implementation of the proposed security mechanism on color images rather than first converting them into gray tones. Another contribution of the Crypto-Stego method is that, it requires no separate key distribution mechanism to decrypt the information. In addition, a parallel-processing approach is implemented to improve the execution time and the efficiency of the scheme by exploiting system resources. Extensive experiments were performed on RGB images with different resolutions and sizes to confirm the effectiveness of the scheme. The high Structural Similarity (SSIM) index score confirmed that the overall carrier image and stego-image were unaltered by processing. While an average value over the test images of 0.0594 for Mean Squared Error (MSE) confirmed that malicious individuals cannot detect the presence of stego data in the cover image. Moreover, negligible pixel intensity histogram changes also validated the effectiveness of the proposed scheme. An average 77% efficiency and 1.5 times speed-up factor was achieved through parallel processing showed the effectiveness of the joint Crypto-Stego method for image confidentiality.

INDEX TERMS Cryptography, image processing, nearest-centroid clustering, LSB-M, steganography, parallel processing

I.INTRODUCTION

In recent years, the trend towards secret information and private data exchange over the public Internet has been grown enormously and, perhaps not unexpectedly, attracted the attention of all kinds of malicious individuals and organizations desirous of gaining access to that confidential information. Therefore information confidentiality is becoming a basic requirement of both organizations and individuals. Multimedia is widely used for many applications, with information retrieval being common. These retrieval techniques not only require robustness but confidentiality too [1]. To achieve confidentiality, many methods of steganography within images have been proposed, for example in [2]–[4].

However, cryptography is also often employed to preserve confidentiality, including image confidentiality. As is well-known, there are two main forms of encryption, symmetric encryption with the same key used for decryption as for encryption, and asymmetric encryption in which a different public key is employed for encryption, with the private key only available to the intended receiver. While asymmetric encryption avoids the need to distribute the encryption key, as the public key can be used for encryption, it has the serious overhead arising from the need for a Public Key Infrastructure (PKI) to authenticate the public key. In fact, asymmetric encryption is rarely used for encryption itself, other than for encrypting symmetric keys, owing to its high computational overhead. Therefore, in this paper symmetric key encryption is directly employed but the key is hidden within the encrypted image by means of steganography. Steganography allows the transmission of sensitive content by concealing it inside a digital medium such as text, image, audio or video so that only the intended recipient knows of its presence [5].

Recently, a variety of multimedia (image/video) steganographic techniques have been [6]–[11] proposed by researchers. Generally, steganography is categorized into two major types, (i) spatial domain techniques, and (ii) transform domain techniques. In the spatial domain steganography, direct manipulation is normally applied to the pixels of the cover image such as pixel indicator techniques (PIT) [12], pixel value differencing (PVD) technique [13], edge-based techniques [14] and Least Significant Bit (LSB) replacement technique [15]. The most common technique amongst these is LSB replacement. In this method, the least significant bits of the carrier image pixels are replaced with the bits of the secret key to be inserted. Although, these techniques provide higher data capacity they have a higher probability of detection and image processing attacks. For example, LSB replacement has been successfully breached by the Regular Singular (RS) groups attack for bitrates as low as 0.03 [16]. In transform domain steganography, transformed coefficients such as those arising from the discrete cosine transform (DCT)[17], discrete wavelet transform (DWT) [18] and Discrete Fourier transform (DFT) [19] methods are utilized for information hiding to provide more resilience against attack. However, transform domain steganography is inefficient in terms of computational complexity and steganographic capacity. Thus, the spatial domain techniques are highly feasible for the efficient data hiding due to less computational overhead. Therefore, this paper proposes a new, simple, and robust scheme for data confidentiality that still offers the features present in state-of-art methods. Moreover, the techniques are employed to hide the symmetric key of an encrypted image within the encrypted image itself.

Moreover, with the emergence of the Internet of Things (IoT) environment and the need to address the requirements of smart mobile devices (SMDs), because the processing capacities of IoT devices and SMDs are constrained, more efficient and more robust mechanisms are required for data protection and information hiding. Hence, algorithms that provide higher imperceptibility, smaller computational overhead and higher steganographic capacity need to be devised. Consequently, this paper proposes a joint Crypto-Stego scheme utilizing nearest-centroid classification to (also called the Rocchio classifier and often confused with k-means clustering) achieve higher information confidentiality for color images, a scheme without the overhead of prior key distribution. The Crypto-Stego images generated in the scheme, according to results presented in this paper, do not allow the identification of the secret key/information camouflaged in the distorted /encrypted image except, of course, by the sender and receiver. More specifically the goals of the research are:

1. An effective keyless joint cryptographic and steganography scheme for confidential transmission of color (RGB) images without converting them into gray scale.
2. A transparent and robust data hiding mechanism with negligible probability of detection.
3. The results of parallel processing should show the efficiency of the implemented scheme. Results should confirm that there is minimal complexity in producing the Crypto-Stego-image.

The remainder of this paper is organized as follows. Section II reviews prior research in the area of combined steganography and cryptography. Section III outlines the context of this work within steganography and cryptography techniques. Section IV discusses the architecture and operating procedure of the scheme. Experimental results and a performance evaluation are provided in Section V. Evaluation continues in Section VI with a statistical security analysis, followed by Section VII's comparison with the research of others. Concluding remarks are made in Section VIII.

II.RELATED WORK

The goal of information confidentiality provision is to prevent sensitive content from being revealed to unauthorized person during transmission. Many approaches have been proposed to achieve higher imperceptibility, varying in complexity from simple steganography methods to more sophisticated joint steganography and cryptography techniques in some way. This Section reviews the various schemes that have been proposed in the area of information hiding.

A.JOINT STEGANOGRAPHY AND CRYPTOGRAPHY TECHNIQUES FOR INFORMATION CONFIDENTIALITY

Although, the purpose of both methods, cryptography and steganography, is to achieve information confidentiality, each alone can probably not provide sufficient protection against current automatic and highly sophisticated attacks. Therefore, several combined cryptography and stenography schemes have been proposed by researchers to enhance the confidentiality of sensitive information communicated over the public Internet [20]–[24].In [4], Khan et al applied five various security level to achieve minimum detectability with their proposed magic least significant bit substitution method (M-LSB- SM) steganographic technique. In their proposed method the researcher achieve the good PSNR and SSIM, however the computational efficiency of the proposed scheme is not discussed clearly. In [20], Zhou at el. applied the RSA encryption algorithm to encrypt the control message with the improved LSB steganography in which embedding is performed with the control information. In improved LSB algorithm red and blue channels of colored images are used for the information hiding and green channel is used to determine the embedding position of the secret information.

In [25], Sridevi et al., proposed combined steganography and cryptography for secure information transmission by means of the LSB method for data embedding within original image. After that the stego image was encrypted with the Advanced Encryption Standard (AES) algorithm. The drawback of that approach is that the cipher sent to the recipient does not resemble the original image because encryption is performed after embedding. In [26], Song et al., proposed a secure communication protocol using combined steganography and cryptography techniques. In this protocol, data hiding and encryption is accomplished simultaneously. The approach is based on the LSB matching (LSB-M) technique and Boolean functions in stream ciphers for high security of information. However, the weakness of that approach is that it focuses only on gray scale images. In [27], Naraya and Prasad presented two approaches to secure data wherein steganography and cryptography are combined. In the first approach each byte of the secret image is converted into cipher text by using the S-DES algorithm. After that encrypted text is embedded into a cover image by XOR encoding with the 2nd LSB of cover image pixel. In the second approach, the secret image is simply encrypted by using S-DES algorithm and embedded it in the cover image as stated above. In [28], Usha, proposed a three-layered protection scheme by combining encryption and steganography. Double encryption is applied to the hidden text. Firstly, the plain text is encrypted by the Playfair cipher and after that the AES algorithm is applied to further encrypt the information to be hidden. Then the encrypted information is camouflaged in an image by LSB replacement. Although double encryption of the plain text to be hidden in the cover image increases the data security, it also increases the computational overhead. In [29], Joshi and Yadav proposed image steganography combined with cryptography on gray images in the spatial domain. In this method, first the secret message is encrypted by the Vernam cipher to enhance information security and then the ciphered message is embedded in the cover image with the LSB substitution (LSB-S) algorithm. Although the proposal achieves good results in terms of better data hiding capacity, the results are again achieved on gray images only. Another similar approach [30] is proposed by Pawar and Gawande in which the researchers combined steganography and cryptography to Mobile (M)-commence as well as e-commerce. In the method, the AES algorithm with a 128-bit key is used for encryption. Random LSB steganography is utilized to embed the encrypted information so as to make detection more difficult.

B.MACHINE LEARNING BASED APPROCHES FOR INFORMATION HIDING

In addition, some researchers utilize color segmentation, pattern matching and machine learning approaches to steganography. The authors of [31] employed clustering by means of the color pattern matching method, after which the secret message is embedded in a selected cluster. After that the stego image was transmitted over the communication channel. In this technique, clustering was performed on a predefined color palette range and then the data was embedded in the cluster. However, this technique was easily detectable when there is small color range within the image because the choice of available clusters to hide the information is limited. Pillai et al. [11] put forward a hybrid scheme for information hiding and security. In the scheme, LSB-S along with nearest-centroid classification and Data Encryption Standard (DES) encryption techniques are applied to color images. Though the scheme has its merits, the DES algorithm is no longer considered strong enough to resist cryptoanalysis and brute force key search attacks.

The authors of [25–31] combined cryptography with steganography in some way. However, the authors of [24–31] did not apply steganography to the encrypted images, as suggested by this paper. The authors of scheme [25] did apply steganography to encrypted images but they failed to disperse the altered pixels throughout the encrypted image, making an attack more feasible. To our best knowledge, most researchers are focused on the encryption of the secret information and the quality of the stego-image to minimize detection of its existence in the stego-image. They achieved an enhanced level of confidential transmission by encrypting the secret information either before embedding it within the cover image or after embedding it into the cover image but they neglected robustness against attacks. Performing steganography before encryption of the hidden image is less robust. Moreover, encrypted secret messages embedded within plain images are easier to detect through stego-analysis. Therefore, this paper presents an improved joint encryption and steganography approach by integrating the AES encryption algorithm with the LSB matching (LSB-M) technique and nearest-centroid classification to achieve: maximal data confidentiality and a negligible level of detection susceptibility. The proposed scheme is said to be secured against stegoanalytic attacks as the pixels are already dislocated in the encryption process. Because of that, a stegoanlysis algorithm will fail to differentiate between the encrypted image and encrypted-stego image. What is more, if an adversary were to succeed in intercepting the encrypted image, it is highly unlikely that they will detect a secret key embedded in the image even with knowledge of the encryption algorithm.

Moreover, state-of-the-art approaches that have been presented in this Section use sequential processing for the encryption and embedding process. However, parallel processing for encryption and embedding is a promising alternative. Therefore, in this research work parallel encryption and embedding has been performed to enhance the efficiency of the proposed scheme. The performance and efficiency in term of computational complexity and cost is evaluated over sequential processing.

III.CONTEXT

In this Section, three techniques; AES encryption, nearest-centroid classification and LSB matching, utilized in the scheme, are revisited.

A.AES ENCRYPTION

AES encryption is widely utilized because of its security, ease of implementation, defense against threats, flexibility in terms of encryption/decryption and of keying material. AES is a cryptographic technique introduced in 2001 to encode sensitive information (plain text) in order to make is unreadable in a scrambled form (cipher text) without the key. The encrypted information is sent over insecure channels to achieve information confidentiality. AES itself is a symmetric block cipher algorithm that employs the same key for encryption and decryption. It supports 128-, 192- and 256-bit key length followed by 10, 12, and 14 rounds of the encryption process respectively. In each round, four steps are performed. FIGURE 1 shows the AES process that is performed in the following steps: (1) Substitution of bytes using a substitution table or S-box (2) Shifting of row data of the state array by different offsets (3) Mixing the data within each column of the state array (4) Adding a round key (Cipher Key) to the state array. As AES uses a single key with a limited key length this makes it more efficient in terms of computational time, memory utilization for encryption and for decryption as compared to asymmetric key algorithms [32]. Besides, AES is more powerful algorithm which can resist many attacks. Therefore, we selected the AES for encryption in our scheme.

In AES, generally five modes of operations are used for the data block i.e. (i) Electronic Codebook (ECB) (ii) Cipher Block Chaining (CBC) (iii) Cipher Feedback (CFB) (iv) Output Feedback (OFB) and (v) Counter (CTR). In the scheme, AES encryption in Output Feedback (OFB) mode is performed on the cover image. The AES in OFB mode operates as a stream cipher (rather than a block cipher) and any modifications to a plaintext block P_i are reflected in the corresponding ciphered block C_i, where $i = 1, 2, 3\ldots n$ with n the number of plaintext blocks, but other ciphered blocks remain unaffected. Thus, OFB mode provides more transmission error resilience.

In OFB, X_{i-1} is an input block from i-1 stage, which has been AES encrypted using key K_e. Then $X_{i-1}$is again AES encrypted using key Ke to produce Xi. After that X_i and the next plaintext block P_i are XORed together to output encrypted block C_i. For encryption of the following plaintext block, AES encryption with K_e, is again performed on the X_i of the previous stage to produce $X_{i+1}$and then XORing is performed with the plaintext P_{i+1} to output $C_{i+1}$and so on. In OFB mode, the decryption process is identical to encryption process. Moreover, OFB generates different output $C_i$for the identical input P_i because of the random initialization vector IV [32]. Equations (1) and (2) represent the encryption and decryption processes in OFB mode respectively.

$$C_i = P_i\ XOR\ X_i \tag{1}$$

$$P_i = C_i\ XOR\ X_i \tag{2}$$

Where i=1, 2, 3, …….n, for n stages of block encryption, and X_i = {Encrypt ($K_e(X_{i-1})$)}

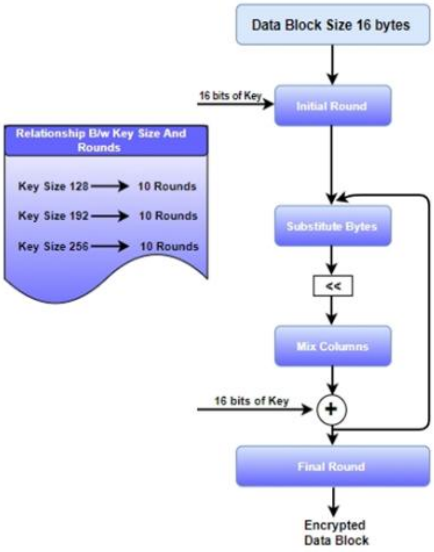


FIGURE 1. **AES symmetric encryption process.**

B.NEAREST-CENTROID CLASSIFICATION

Nearest centroid classification is a classifier frequently employed in image processing, though probably k-means clustering, which clusters or classifies by arithmetic mean rather than proximity to a centroid, is more popular. It acts to group related data items into clusters without any prior knowledge of the dataset. As digital images are an array of numbers represent the intensities of colors at different points ranging from 0 -255 for each color component. Grayscale images require 8-bits to store a pixel while Red-Green-Blue (RGB) images required 24-bits per pixel, i.e. 8 bits for each color component. Therefore, RGB images contain a large number of potential colors and their color variation is not gradual, as is the case for grayscale images. Therefore, in the proposed scheme, clustering is performed on the RGB pixel value intensities to achieve a form of color quantization without affecting visual perception but at the same time saving computational cost and time. Additionally, another purpose of clustering in steganography is to achieve a greater dispersion of the secret message bits in the carrier image.

Clustering or classification is the process of partitioning similar data items into groups determined by attributes such as color values, size, and texture and so on, and a group of data items having similar attributes is named as a cluster. Thus, the data items within a cluster are similar to each other but different from the data items of other clusters [33]. Nearest-centroid classification is an unsupervised clustering algorithm that randomly chooses the cluster center (centroid) from the given data points and compares it, the centroid, with surrounding data points based on their attributes, calculating the squared Euclidean distance. The number of clusters or classifications, k, is sometimes input by the user but for the reasons explained in Section IV, herein, it was auto-generated. The Euclidean distance between two points Xi and Yi can be obtained as follow:

$$d(Xi\,,Yi) = \sqrt{\sum_{i=0}^{n}(Xi - Yi)}(3)$$

where Xi=(r_i,g_i,b_i), Yi=(r_{i+1},g_{i+1},b_{i+1}) and i = 1,2,3,4,…..n for n candidate data points.

The data points similar to the centroid are assigned to the cluster having that centroid. When, for all clusters, all similar data points are assigned to a cluster, a new set of ‘k’ cluster centroids are recomputed and the process is repeated until all data points are allocated to their appropriate clusters. A stopping criterion such as no further changes in classification halts the iterations.

C.LSB MATCHING

The LSB replacement technique is most efficient and conventional method used in image steganography due to high steganography capacity and minimum human perceptible distortions. In this method the least significant bit of some pixels of the cover or carrier image are swapped with bits of the secret information to be hidden. As mentioned already, the RGB color images are represented as an array or matrix of pixels and each pixel represents the intensities of RGB channels. Therefore, the small alteration of each color component by LSB replacement does not affect the change the overall visual perception of an image.

Nevertheless, this method is relatively poor against statistical analysis and robustness, with the result that it can be easily exploited by an adversary. Therefore, the proposed scheme utilizes the LSB-M technique, which is an improved variant of LSB replacement. In this scheme, pixel values of the cover image are increased (+1) or decreased (-1) randomly to match with the message bits to be embedded in order to reduce the asymmetry produced by the conventional LSB replacement method [16]. Hence, LSB-M provides better imperceptibility and is more challenging to stego-analysis aimed at detecting LSB-type hiding.

IV.PROPOSED SCHEME

This section introduces the proposed Crypto-Stego scheme to achieve better information confidentiality over open communication channels. The proposed scheme consists of five procedures: 1) Cover image encryption, 2) Clustering, 3) shuffling, 4) Key embedding, and 5) Information extraction. For the ease of readers, note that four kinds of images are discussed throughout this manuscript: a) cover image (C_i) means a plain original colored image, b) encrypted image(E_c) means an image after selective encryption on the color pixels, c) stego image (S) means a plain original colored image with embedded secret key, and d) carrier image (S') means encrypted image with an embedded secret key. FIGURE 2 shows the basic architecture of the scheme. In the first procedure, the original colored image denoted by C_i, consisting of N number of pixels ($H \times W$), was encrypted by AES encryption so that the pixel values of the cover image were distorted. Equation (4) represents the encryption process turning a cover image into an encrypted image.

$$\text{ENCRYPT: } C_i\,.K_i \rightarrow E_c \qquad (4)$$

Where C is the set of all colored images, for all colored images $C_i \in$ C. K is the set of all secret keys, for all secret keys $K_i \in K$. E_c is the encrypted image.

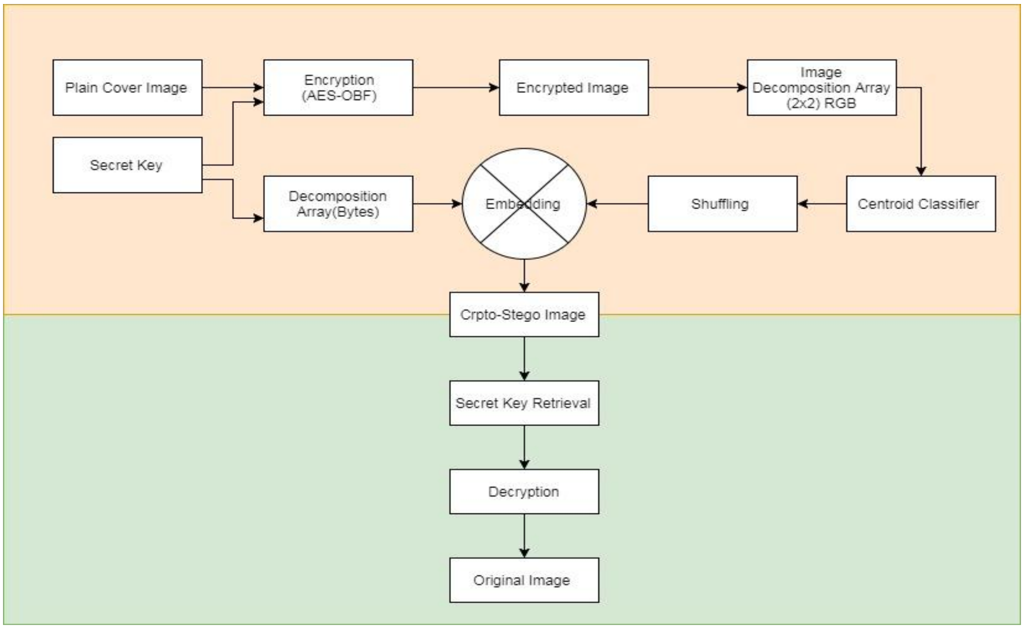
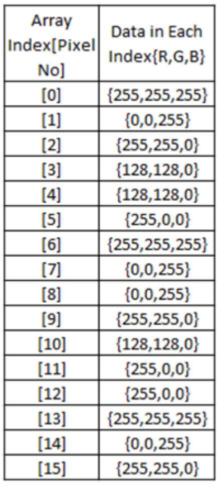


FIGURE 2. Joint Crypto-Stego scheme for enhanced content protection AES symmetric encryption process.

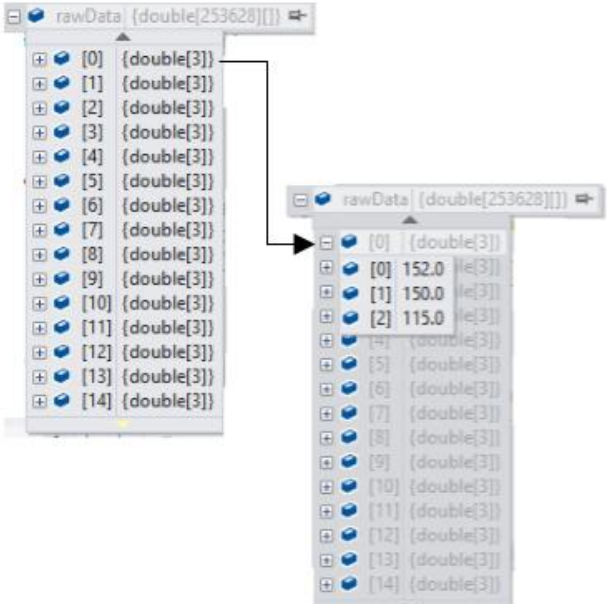
The encryption in OFB mode for error resilience was performed on the cover image $C_i$after firstly converting the image bitmap into a 1D byte array of RGB pixels as shown in FIGURE 3. The initialization vector (IV) for OFB was generated via a Pseudo-Random Number (PRN) generator from a pre-set initial seed value held at both the sender and receiver. AES, with its block size of 16 bytes was performed on the array with a 256-bit size key.



Array Index[Pixel No]	Data in Each Index{R,G,B}
[0]	{255,255,255}
[1]	{0,0,255}
[2]	{255,255,0}
[3]	{128,128,0}
[4]	{128,128,0}
[5]	{255,0,0}
[6]	{255,255,255}
[7]	{0,0,255}
[8]	{0,0,255}
[9]	{255,255,0}
[10]	{128,128,0}
[11]	{255,0,0}
[12]	{255,0,0}
[13]	{255,255,255}
[14]	{0,0,255}
[15]	{255,255,0}

FIGURE 3. 1D byte array of the RGB pixels.

After that the encrypted bytes were stored as a bitmap array to reconstruct the encrypted image. The large key size utilized to achieve the higher data security. The obtained encrypted image denoted by E_c was used for steganography rather than the cover image. Moreover, the bitmap array was converted into a double 2 D array [index]{data}(array within array; Outer array for data index and inner array for RGB) as represented in FIGURE 4.



```
rawData {double[253628][]}
  [0] {double[3]}
  [1] {double[3]}
  [2] {double[3]}
  [3] {double[3]}
  [4] {double[3]}
  [5] {double[3]}
  [6] {double[3]}
  [7] {double[3]}
  [8] {double[3]}
  [9] {double[3]}
  [10] {double[3]}
  [11] {double[3]}
  [12] {double[3]}
  [13] {double[3]}
  [14] {double[3]}

rawData {double[253628][]}
  [0] {double[3]}
    [0] 152.0
    [1] 150.0
    [2] 115.0
  [5] {double[3]}
  [6] {double[3]}
  [7] {double[3]}
  [8] {double[3]}
  [9] {double[3]}
  [10] {double[3]}
  [11] {double[3]}
  [12] {double[3]}
  [13] {double[3]}
  [14] {double[3]}
```

FIGURE 4. Double 2 D array [index] {data}; Outer array for index and inner array for data (RGB values)

In the second procedure, nearest-centroid classification of color intensity values of pixels was performed, aimed at partitioning each pixel of the image into *k* clusters, as shown in FIGURE 5. As normal, centroid-based clustering was performed to get the cluster indices and centroid locations and after that the secret/decryption key the embedded inside the encrypted image *Ec*.

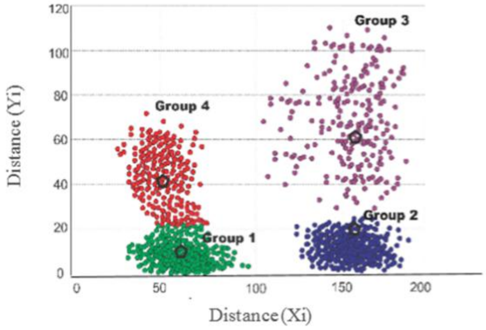
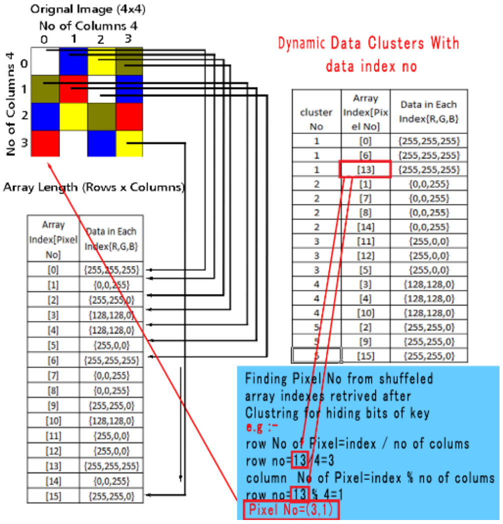


FIGURE 5. Nearest-centroid classification of colored pixels.

Furthermore, when clustering was performed, the array returns the data with their corresponding cluster numbers. Therefore, in the proposed scheme, the array was shuffled to store data point indices with their respective cluster. Indexing allocates the physical co-ordinates of the pixels within the encrypted bitmap and also performs cluster assignment of each pixel without physically re-assigning the pixel positions. FIGURE 6 shows the index retrieval procedure for key hiding. The pixel position for hiding the key obtained is as follows:

Row no. = Index [Pixel no.]/Total no. of Columns (5)

Column no = Index [Pixel no.]% Total no. of columns (6)


Orignal Image (4x4)
No of Columns 4
0
1
2
3
No of Columns 4
0
1
2
3
Array Length (Rows x Columns)
Dynamic Data Clusters With data index no

Array Index[Pixel No]	Data in Each Index{R,G,B}
[0]	{255,255,255}
[1]	{0,0,255}
[2]	{255,255,0}
[3]	{128,128,0}
[4]	{128,128,0}
[5]	{255,0,0}
[6]	{255,255,255}
[7]	{0,0,255}
[8]	{0,0,255}
[9]	{255,255,0}
[10]	{128,128,0}
[11]	{255,0,0}
[12]	{255,0,0}
[13]	{255,255,255}
[14]	{0,0,255}
[15]	{255,255,0}

cluster No	Array Index[Pixel No]	Data in Each Index{R,G,B}
1	[0]	{255,255,255}
1	[6]	{255,255,255}
1	[13]	{255,255,255}
2	[1]	{0,0,255}
2	[7]	{0,0,255}
2	[8]	{0,0,255}
2	[14]	{0,0,255}
3	[11]	{255,0,0}
3	[12]	{255,0,0}
3	[5]	{255,0,0}
4	[3]	{128,128,0}
4	[4]	{128,128,0}
4	[10]	{128,128,0}
5	[2]	{255,255,0}
5	[9]	{255,255,0}
5	[15]	{255,255,0}

Finding Pixel No from shuffeled array indexes retrived after Clustring for hiding bits of key
e.g :-
row No of Pixel=index / no of colums
row no=13/4=3
column No of Pixel=index % no of colums
row no=13 % 4=1
Pixel No=(3,1)

FIGURE 6. Index retrieval process for key embedding.

Lastly, the secret decryption key was subsequently segmented according to the number of clusters, with equal-sized segments. Furthermore, each segment was embedded within a different cluster by means of the stronger LSB-M algorithm, after which the image was transmitted to the receiver. As previously mentioned, LSB-M provides the same steganographic capacity as LSB replacement but reduces the symmetry produced by the conventional LSB technique. Mathematically, embedding can be defined as in (7). EMBD represents the encoding process of the secret key into encrypted image E_c.

$$EMBD: E_c . Ki \rightarrow S' \quad (7)$$

where S' is encrypted image with embedded secret key.

In fact, the secret decryption key was embedded in different clusters uniformly to achieve better information hiding and robustness to attack. Moreover, using pixels with the same attributes to embed the secret key helps in retrieval. The number of clusters k was auto-generated by reference to the image height and width. The total number of colored pixels were calculated in encrypted image E_c by considering three conditions on which the cluster were generated. First we compared the lowest value of R pixels to G and B pixels values. However, if there were not any red pixels having the lowest value, then the lowest value of B pixels were compared with R and G pixels. If both conditions were not in existence then, in the third condition, the pixels having equal R, G, B pixel values were considered. After that the pixel value of validated conditions and the location of pixels were added with R, G and B original values to get a seed random number. That seed random number varied on the basis of image contents, which finally was used to calculate a dynamic cluster number. The pseudocode to calculate dynamic cluster number is given in FIGURE 6. Moreover, the number of clusters was limited to a maximum of eight to reduce the computational complexity. The auto-generation of the cluster number eliminates the need for prior cluster number sharing at the receiving end, when retrieving the hidden data. Although, a greater number of clusters provide better hiding capacity as well as protection against attacks, it incurs high processing time and computational cost that will affect the efficiency of the algorithm.

Pseudo-code of auto-cluster number (k) handling

```
Input: Image
IH=Height of Image;
IW=Width of Image;
int A= IH+ IW;
K=No of clusters;
for (inti=0;i<= A;i++ )
{
If (pixel.R<pixel.G<pixel.B)
{B=Pixel.Location+ pixel.R+ pixel.G+ pixel.B; }
Else
{
        If(pixel.R>pixel.G>pixel.B)
            {B=Pixel.Location+ pixel.R+ pixel.G+ pixel.B;}
Else
            {
             If(pixel.R==pixel.G== pixel.B)
            {B=Pixel.Location+ pixel.R+ pixel.G+ pixel.B; }
            }
}
int C=B/A;
int D=B%A;
if(C>=1 && C<=8)
{
K=C;
Break;
}
Else
{
K=D;
Break;
}}
```

FIGURE 6. Pseudo-code of auto-generation of the number of clusters.

The stepwise process of nearest-centroid base clustering of encrypted pixels and secret-key embedding within the encrypted image is described below:

Step1: Cover image bitmap was converted into byte array of length (bitmaprow*bitmapcolumn) (total no of pixels)*3 (total no. of bytes) in bitmap image.

Step2: Byte array was encrypted with AES (Rijndael cipher) algorithm (by using System.Security.Cryptography (SSC) namespace of .net Framework with AES Algo (Rijndael cipher).

Step 3: The encrypted byte array was stored as a Bitmap array to reconstruct the encrypted image.

Step 4: The bitmap array was converted into a double 2 D array [index]{data}(array within array; Outer array for data index and inner array for data).

Step 5: Passed the double array having data of all the pixels into Nearest-Centroid classification for dynamic/auto clustering based on the encrypted image content.

Step 6: The output returned the data with its cluster no. Therefore, shuffling was performed to convert the output into data points' indexes of their respective clusters such as first cluster, then 2nd cluster, and so on.

Step 7: Pixel (row and column) no. were calculated by using step 4 and then the equal no of bits of symmetric secret key data was embedded in pixels of every clusters.

FIGURE 8 represents the step-wise simulation procedure of the proposed scheme for finding the nearest centroid and implementation of the joint Crypto-Stego scheme.

```
Pseudo-code of Algorithm for Nearest-Centroid Classification
Input: 1) Bitmap of encrypted Image, 2) Numbers of clusters, 3) Text Data.
 Double Data [Image Height * Image Width] [R, G, B].
  Loop from i = 0 to i < Bitmap Height
   Loop from j = 0 to j < Bitmap Width
      Data[count]=[Pixel R, Pixel G, Pixel B]
        Count ++
   End Loop
  End Loop
KmensKmobj =new Kcluster (Numbers of clusters(k))
Kmobj.Compute ( Data[count ])
    Return(int [] cluster Numbers)
intIndex_Data []=[Image Height * Image Width]
intIndex_Data []=int[] cluster Numbers
GetClusterIndexs((Data, Index_Data [], Numbers of clusters, 1)
 Return(int [] Data_point_Index)
List Data_point_Index<int>
Loop from i = 0 to i < Image Height * Image Width
Row_Number= Data_point_Index / Row Width
Column_Number= Data_point_Index % Row Width
Color pixel = Image.GetPixel(Row_Number, Column_Number)
// now, clear the least significant bit  from each pixel element
R = pixel.R - pixel.R % 2;
G = pixel.G - pixel.G % 2;
B = pixel.B - pixel.B % 2;
// for each pixel, pass through its elements (RGB)
Loop from n = 0 to j < 3
IF pixel Element % 8 is equal to 0 THEN
          IF character bits are completed THEN
IF pixel Element - 1 % 3 is less than 2 THEN
bmp.SetPixel(Row_Number , Column_Number ,Color.FromArgb(RGB));
 ENDIF
                          ENDIF
                           ELSE
                             IFcharIndex is greater than text.LengthTHEN
                               all characters hidden
                            ELSE
                             Next Character
                              Switch case for all three elements of pixel
                               pixel Element
                               Case based on pixel Element
                                  Case  is equal to 0
                                     R += charValue % 2;
             charValue /= 2;
                                  Case  is equal to 1
                                      G += charValue % 2;
             charValue /= 2;
                                  Case  is equal to 2
                                       B += charValue % 2;
             charValue /= 2;
                                  End Case
                                     charIndex= charIndex+1
                              ENDIF
bmp.SetPixel(Row_Number , Column_NumberColor.FromArgb(R, G, B));
             End Loop
Output: Stego-Bitmap-Image
```

FIGURE 8. Pseudo-code of nearest centroid classification algorithm.

In the final procedure, inverse steganography was applied to extract the hidden secret key from the carrier image (S') and lastly, the cover image was retrieved using the extracted secret key. In addition, at the receiver, auto-generation of the number of clusters allowed the pixels of the image to be re-assigned to clusters and, as previously mentioned, eliminates, a separate mechanism for cluster number sharing in advance. The hidden key was then extracted by reversing the LSB-M algorithm, after which the image can be decrypted. Because the image was first encrypted before hiding the encryption key, existing tools aimed at finding statistical anomalies within plaintext images are thwarted. Equation (8) and (9) represents key extraction (EXT) and decryption (DECRYPT) process.

$$\text{EXT: } S' \rightarrow K_i\,, E_c \tag{8}$$

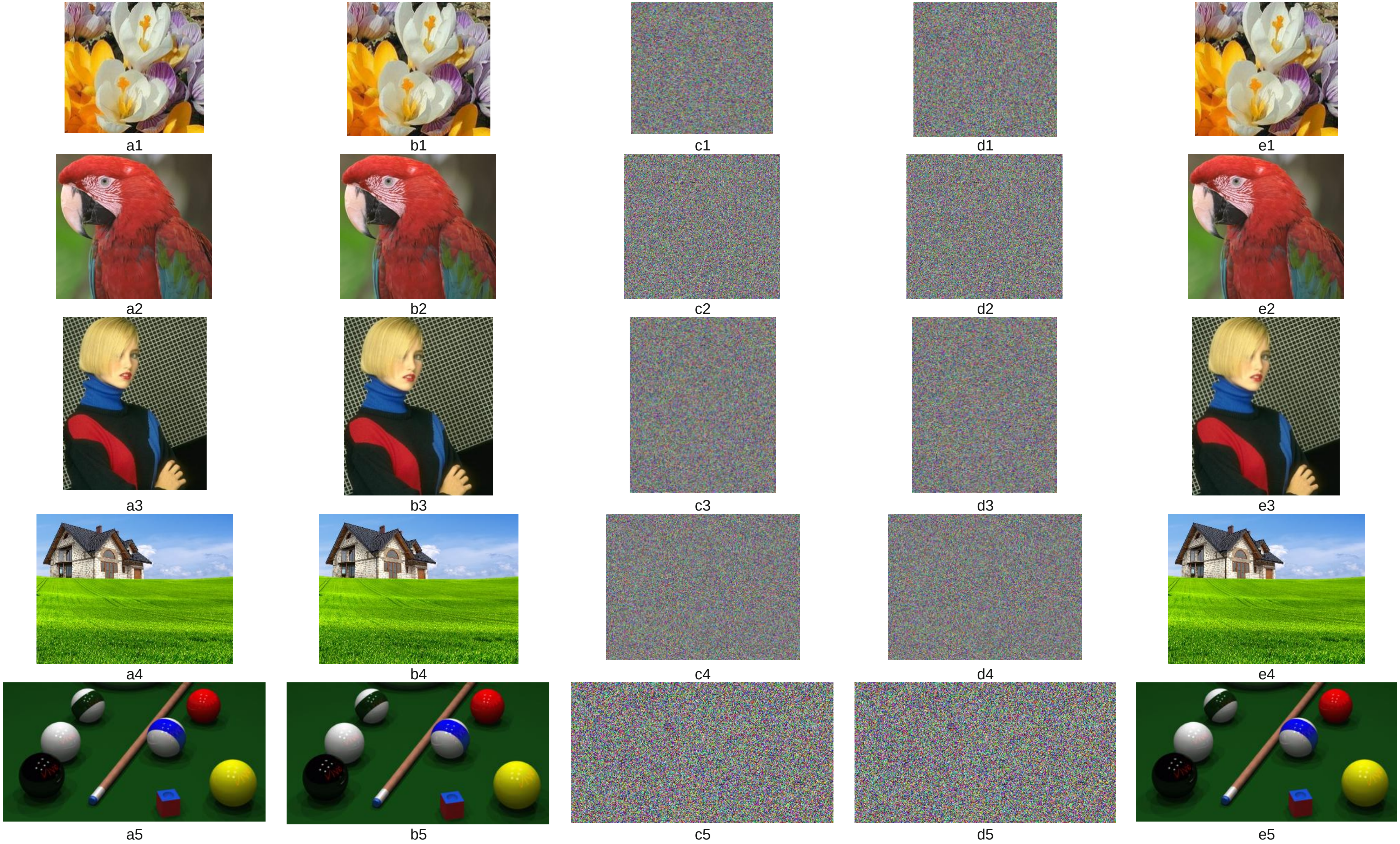
$$\text{DECRYPT: } E_c \rightarrow C_i \tag{9}$$

V. EXPERIMENTAL RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed Crypto-Stego scheme, extensive experiments were performed on 120 images. The 24-bit color images have various resolutions and sizes and were downloaded from the USC-SIPI image database (http://sipi.usc.edu/database/) and BSD500 dataset: (https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html). All experiments were performed on a 64-bit operating system with 1.70 GHz Core i3-4010U processor and 4 GB RAM. The algorithm were developed using the C# programming language in the Visual Studio 12 environment. Parallel processing was performed using the Message Passing Interface (MPI).

In the experiments, as previously mentioned, 256-bit sized secret keys were employed. FIGURE 9 shows the visual results of the scheme on a sample dataset/images (Flowers (303 × 284 pixels), Parrot (370 × 341) ,Girl (321 × 387), Peppers (506 × 425) , Puppy (236 × 213), House (1441 × 1085), Serrano (555 × 629), Pool (379 × 203) , Strelitzia (471 × 351), and Kid (373 × 410)). It can be seen that there are no visible distortions between the cover images (C_i) FIGURE 9 (a1-a9) vs. the stego images (S) FIGURE 9 (b1-b9) and the encrypted images (E_c) FIGURE 9 (c1-c9)) vs. carrier images (S') FIGURE 8 (d1-d9) generated with the proposed algorithm. Notice that the stego images show the effect of embedding the key bits within color images without first encrypting the image. Furthermore, FIGURE 9 (e1-e9) illustrates that, after extracting the secret/decrypt key from the carrier image, a recovered image is the same as the cover image without creating any visible distortion in the recovered image. Hence, changes owing to embedding the secret key in the carrier image generated by the proposed scheme are undetectable by the human visual perception.

Additionally, when the carrier image was decrypted without first extracting the embedded key the correct positions of the altered pixels could not be identified through visual means. This is because by applying AES encryption first the position of the hidden bits are further dispersed within the image after it is decrypted. Thus, by first applying AES-OFB encryption the secret key were further dispersed across the whole image, which will add more complexity to the task of an adversary to detect any steganography in the carrier image and will result in a failure to retrieve the cover image as well.


a1
b1
c1
d1
e1
a2
b2
c2
d2
e2
a3
b3
c3
d3
e3
a4
b4
c4
d4
e4
a5
b5
c5
d5
e5

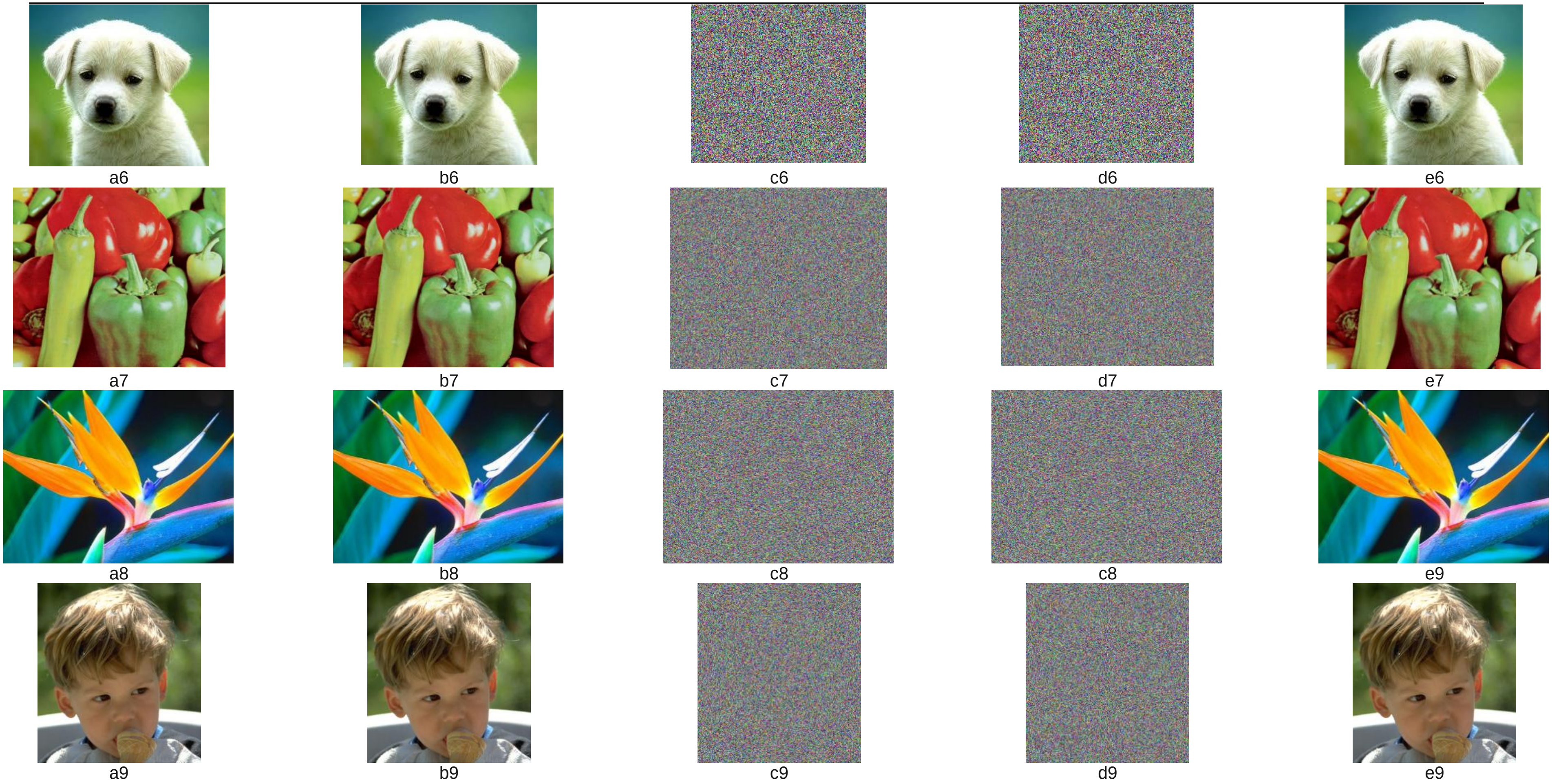
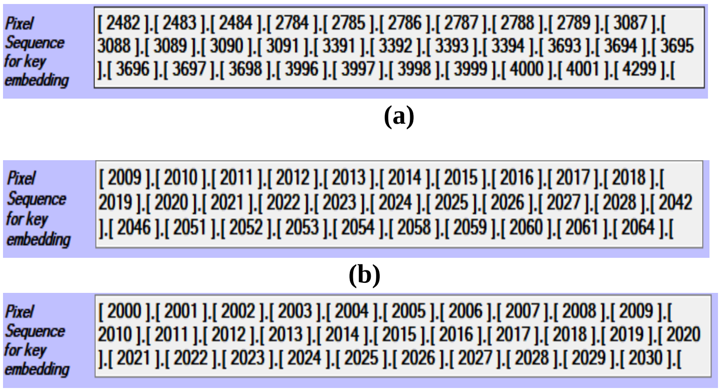


FIGURE 9. Visual results of the proposed scheme for sample dataset (Flowers (303 × 284 pixels), Parrot (370 × 341) , Girl (321 × 387), House (1441 × 1085), Pool (379 × 203), Puppy (236 × 213), Peppers (506 × 425) , Strelitzia (471 × 351) and Kid (373 × 410)) images.(a1– a10) Cover images(C_i). (b1-b10) Steganographic images (S) produced by the proposed algorithm. (c1-c10) Encrypted images(E_c). (d1–d10) Carrier images (S') produced by the proposed scheme. (e1- e10)Extracted images.

A.DYNAMIC EMBEDDING

In the proposed scheme dynamic embedding is employed, rather than available methods of static substitution of least significant bits of pixels within the carrier image. Thus, the embedding capacity of the proposed scheme depends on the number of clusters and the size of the colored image. FIGURE 10 (a,b,c) and FIGURE 11(a,b) show the pixel sequence for embedding secret keys into the stego image and carrier image by using MSB-M followed by nearest-centroid clustering and bit shuffling. It can be observed from FIGURE 10 (a, b, c) that the distribution of secret key bits into the stego-image is uniform within different clusters, while in the carrier images, secret key bits are dispersed within the different clusters (see FIGURE 11 (a, b)), which makes any attempt for detection of a secret message more complex for a stego-analyst. Moreover, the embedding sequence of the secret key bits is different for different image due to its dynamics property (see FIGURE 9(a) and FIGURE 9(b) for a comparison). Thus, as a result of the proposed scheme, an adversary is most unlikely to gain any useful information from the carrier image. Indeed, at the receiver, the same image sequence of pixels is required to retrieve the secret key to decrypt the secret encrypted data.



Pixel Sequence for key embedding

```
[ 2482 ].[ 2483 ].[ 2484 ].[ 2784 ].[ 2785 ].[ 2786 ].[ 2787 ].[ 2788 ].[ 2789 ].[ 3087 ].[
3088 ].[ 3089 ].[ 3090 ].[ 3091 ].[ 3391 ].[ 3392 ].[ 3393 ].[ 3394 ].[ 3693 ].[ 3694 ].[ 3695
].[ 3696 ].[ 3697 ].[ 3698 ].[ 3996 ].[ 3997 ].[ 3998 ].[ 3999 ].[ 4000 ].[ 4001 ].[ 4299 ].[
```

(a)

Pixel Sequence for key embedding

```
[ 2009 ].[ 2010 ].[ 2011 ].[ 2012 ].[ 2013 ].[ 2014 ].[ 2015 ].[ 2016 ].[ 2017 ].[ 2018 ].[
2019 ].[ 2020 ].[ 2021 ].[ 2022 ].[ 2023 ].[ 2024 ].[ 2025 ].[ 2026 ].[ 2027 ].[ 2028 ].[ 2042
].[ 2046 ].[ 2051 ].[ 2052 ].[ 2053 ].[ 2054 ].[ 2058 ].[ 2059 ].[ 2060 ].[ 2061 ].[ 2064 ].[
```

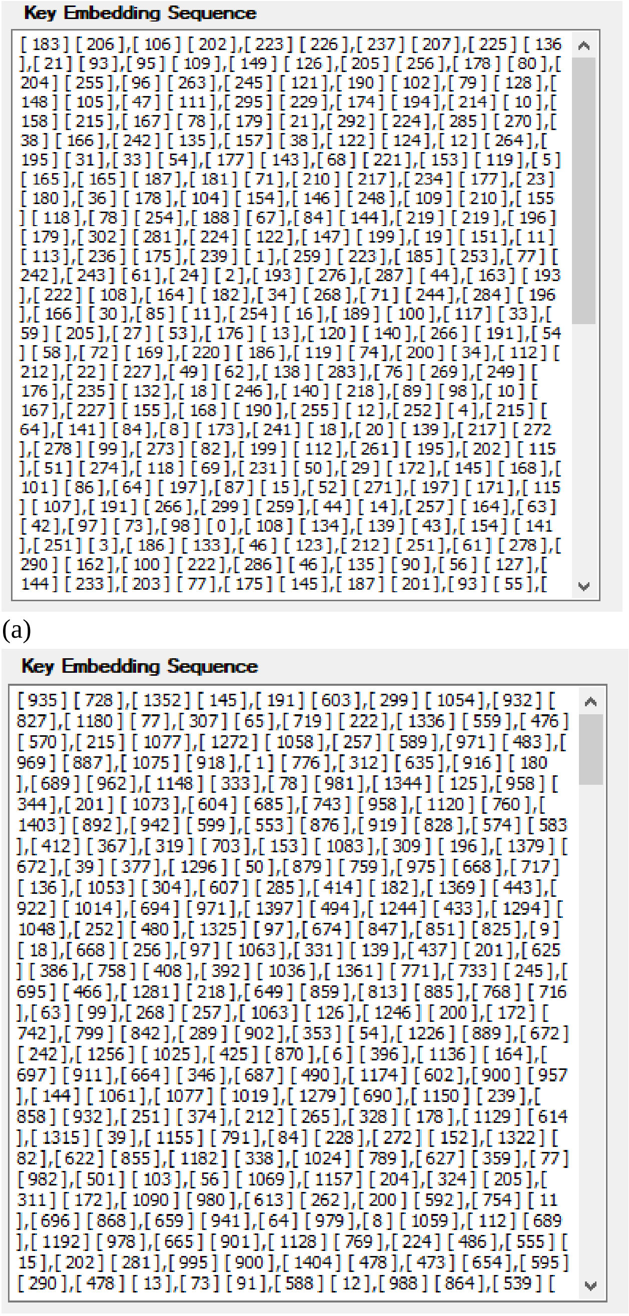
(b)

Pixel Sequence for key embedding

```
[ 2000 ].[ 2001 ].[ 2002 ].[ 2003 ].[ 2004 ].[ 2005 ].[ 2006 ].[ 2007 ].[ 2008 ].[ 2009 ].[
2010 ].[ 2011 ].[ 2012 ].[ 2013 ].[ 2014 ].[ 2015 ].[ 2016 ].[ 2017 ].[ 2018 ].[ 2019 ].[ 2020
].[ 2021 ].[ 2022 ].[ 2023 ].[ 2024 ].[ 2025 ].[ 2026 ].[ 2027 ].[ 2028 ].[ 2029 ].[ 2030 ].[
```

(c)

FIGURE 10. Secret key bit sequence embedding within the stego-images through the proposed scheme: (a) Secret key bit embedding sequence within the Flowers stegoimage, (b) Secret key bit embedding sequence within the Girl stegoimage, and (c) Secret key bit embedding sequence within the House stegoimage.



Key Embedding Sequence

```
[ 183 ] [ 206 ],[ 106 ] [ 202 ],[ 223 ] [ 226 ],[ 237 ] [ 207 ],[ 225 ] [ 136
],[ 21 ] [ 93 ],[ 95 ] [ 109 ],[ 149 ] [ 126 ],[ 205 ] [ 256 ],[ 178 ] [ 80 ],[
204 ] [ 255 ],[ 96 ] [ 263 ],[ 245 ] [ 121 ],[ 190 ] [ 102 ],[ 79 ] [ 128 ],[
148 ] [ 105 ],[ 47 ] [ 111 ],[ 295 ] [ 229 ],[ 174 ] [ 194 ],[ 214 ] [ 10 ],[
158 ] [ 215 ],[ 167 ] [ 78 ],[ 179 ] [ 21 ],[ 292 ] [ 224 ],[ 285 ] [ 270 ],[
38 ] [ 166 ],[ 242 ] [ 135 ],[ 157 ] [ 38 ],[ 122 ] [ 124 ],[ 12 ] [ 264 ],[
195 ] [ 31 ],[ 33 ] [ 54 ],[ 177 ] [ 143 ],[ 68 ] [ 221 ],[ 153 ] [ 119 ],[ 5 ]
[ 165 ],[ 165 ] [ 187 ],[ 181 ] [ 71 ],[ 210 ] [ 217 ],[ 234 ] [ 177 ],[ 23 ]
[ 180 ],[ 36 ] [ 178 ],[ 104 ] [ 154 ],[ 146 ] [ 248 ],[ 109 ] [ 210 ],[ 155
] [ 118 ],[ 78 ] [ 254 ],[ 188 ] [ 67 ],[ 84 ] [ 144 ],[ 219 ] [ 219 ],[ 196 ]
[ 179 ],[ 302 ] [ 281 ],[ 224 ] [ 122 ],[ 147 ] [ 199 ],[ 19 ] [ 151 ],[ 11 ]
[ 113 ],[ 236 ] [ 175 ],[ 239 ] [ 1 ],[ 259 ] [ 223 ],[ 185 ] [ 253 ],[ 77 ] [
242 ],[ 243 ] [ 61 ],[ 24 ] [ 2 ],[ 193 ] [ 276 ],[ 287 ] [ 44 ],[ 163 ] [ 193
],[ 222 ] [ 108 ],[ 164 ] [ 182 ],[ 34 ] [ 268 ],[ 71 ] [ 244 ],[ 284 ] [ 196
],[ 166 ] [ 30 ],[ 85 ] [ 11 ],[ 254 ] [ 16 ],[ 189 ] [ 100 ],[ 117 ] [ 33 ],[
59 ] [ 205 ],[ 27 ] [ 53 ],[ 176 ] [ 13 ],[ 120 ] [ 140 ],[ 266 ] [ 191 ],[ 54
] [ 58 ],[ 72 ] [ 169 ],[ 220 ] [ 186 ],[ 119 ] [ 74 ],[ 200 ] [ 34 ],[ 112 ] [
212 ],[ 22 ] [ 227 ],[ 49 ] [ 62 ],[ 138 ] [ 283 ],[ 76 ] [ 269 ],[ 249 ] [
176 ],[ 235 ] [ 132 ],[ 18 ] [ 246 ],[ 140 ] [ 218 ],[ 89 ] [ 98 ],[ 10 ] [
167 ],[ 227 ] [ 155 ],[ 168 ] [ 190 ],[ 255 ] [ 12 ],[ 252 ] [ 4 ],[ 215 ] [
64 ],[ 141 ] [ 84 ],[ 8 ] [ 173 ],[ 241 ] [ 18 ],[ 20 ] [ 139 ],[ 217 ] [ 272
],[ 278 ] [ 99 ],[ 273 ] [ 82 ],[ 199 ] [ 112 ],[ 261 ] [ 195 ],[ 202 ] [ 115
],[ 51 ] [ 274 ],[ 118 ] [ 69 ],[ 231 ] [ 50 ],[ 29 ] [ 172 ],[ 145 ] [ 168 ],[
101 ] [ 86 ],[ 64 ] [ 197 ],[ 87 ] [ 15 ],[ 52 ] [ 271 ],[ 197 ] [ 171 ],[ 115
] [ 107 ],[ 191 ] [ 266 ],[ 299 ] [ 259 ],[ 44 ] [ 14 ],[ 257 ] [ 164 ],[ 63 ]
[ 42 ],[ 97 ] [ 73 ],[ 98 ] [ 0 ],[ 108 ] [ 134 ],[ 139 ] [ 43 ],[ 154 ] [ 141
],[ 251 ] [ 3 ],[ 186 ] [ 133 ],[ 46 ] [ 123 ],[ 212 ] [ 251 ],[ 61 ] [ 278 ],[
290 ] [ 162 ],[ 100 ] [ 222 ],[ 286 ] [ 46 ],[ 135 ] [ 90 ],[ 56 ] [ 127 ],[
144 ] [ 233 ],[ 203 ] [ 77 ],[ 175 ] [ 145 ],[ 187 ] [ 201 ],[ 93 ] [ 55 ],[
```

(a)

Key Embedding Sequence

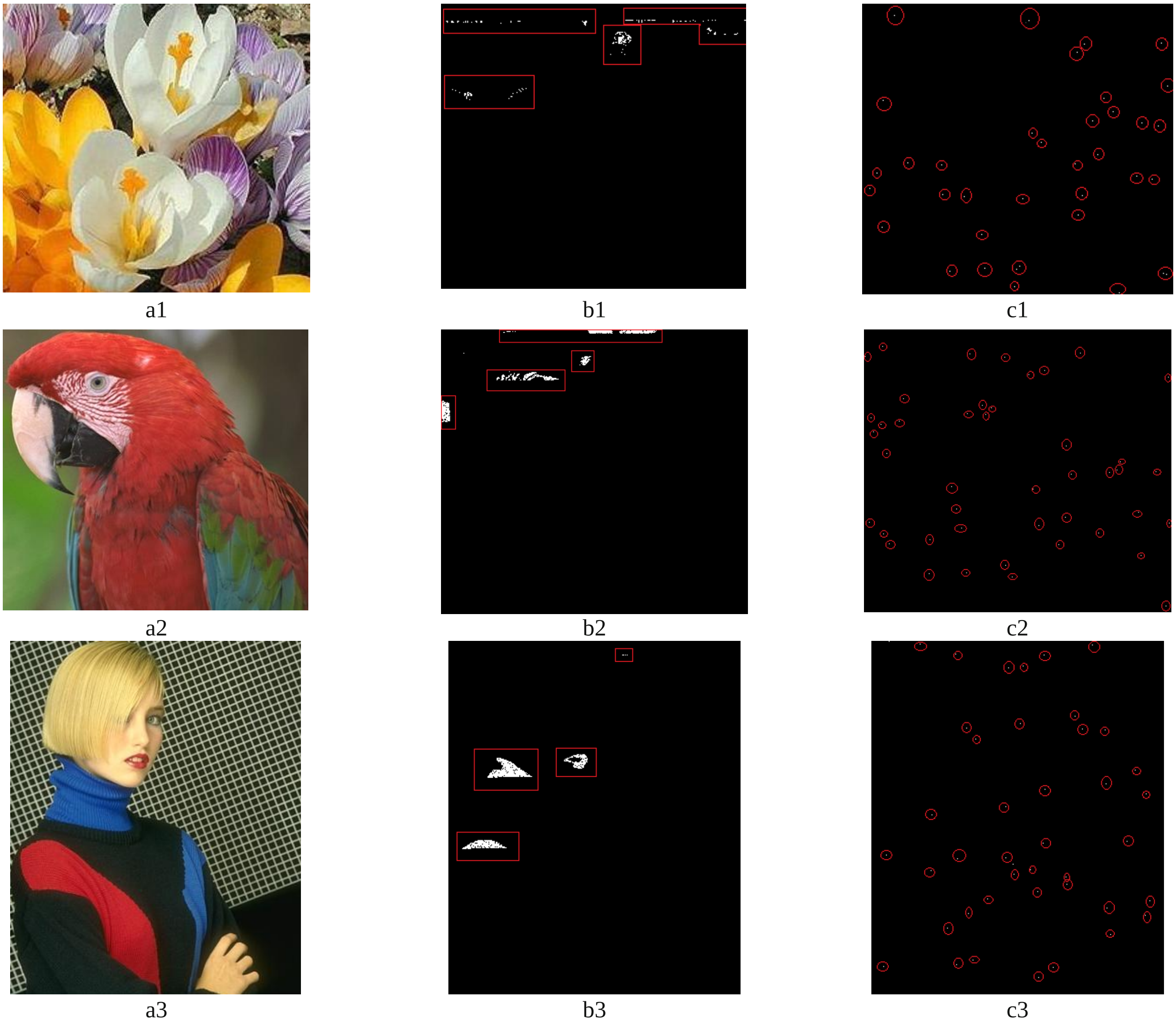
```
[ 935 ] [ 728 ],[ 1352 ] [ 145 ],[ 191 ] [ 603 ],[ 299 ] [ 1054 ],[ 932 ] [
827 ],[ 1180 ] [ 77 ],[ 307 ] [ 65 ],[ 719 ] [ 222 ],[ 1336 ] [ 559 ],[ 476 ]
[ 570 ],[ 215 ] [ 1077 ],[ 1272 ] [ 1058 ],[ 257 ] [ 589 ],[ 971 ] [ 483 ],[
969 ] [ 887 ],[ 1075 ] [ 918 ],[ 1 ] [ 776 ],[ 312 ] [ 635 ],[ 916 ] [ 180
],[ 689 ] [ 962 ],[ 1148 ] [ 333 ],[ 78 ] [ 981 ],[ 1344 ] [ 125 ],[ 958 ] [
344 ],[ 201 ] [ 1073 ],[ 604 ] [ 685 ],[ 743 ] [ 958 ],[ 1120 ] [ 760 ],[
1403 ] [ 892 ],[ 942 ] [ 599 ],[ 553 ] [ 876 ],[ 919 ] [ 828 ],[ 574 ] [ 583
],[ 412 ] [ 367 ],[ 319 ] [ 703 ],[ 153 ] [ 1083 ],[ 309 ] [ 196 ],[ 1379 ] [
672 ],[ 39 ] [ 377 ],[ 1296 ] [ 50 ],[ 879 ] [ 759 ],[ 975 ] [ 668 ],[ 717 ]
[ 136 ],[ 1053 ] [ 304 ],[ 607 ] [ 285 ],[ 414 ] [ 182 ],[ 1369 ] [ 443 ],[
922 ] [ 1014 ],[ 694 ] [ 971 ],[ 1397 ] [ 494 ],[ 1244 ] [ 433 ],[ 1294 ] [
1048 ],[ 252 ] [ 480 ],[ 1325 ] [ 97 ],[ 674 ] [ 847 ],[ 851 ] [ 825 ],[ 9 ]
[ 18 ],[ 668 ] [ 256 ],[ 97 ] [ 1063 ],[ 331 ] [ 139 ],[ 437 ] [ 201 ],[ 625
] [ 386 ],[ 758 ] [ 408 ],[ 392 ] [ 1036 ],[ 1361 ] [ 771 ],[ 733 ] [ 245 ],[
695 ] [ 466 ],[ 1281 ] [ 218 ],[ 649 ] [ 859 ],[ 813 ] [ 885 ],[ 768 ] [ 716
],[ 63 ] [ 99 ],[ 268 ] [ 257 ],[ 1063 ] [ 126 ],[ 1246 ] [ 200 ],[ 172 ] [
742 ],[ 799 ] [ 842 ],[ 289 ] [ 902 ],[ 353 ] [ 54 ],[ 1226 ] [ 889 ],[ 672 ]
[ 242 ],[ 1256 ] [ 1025 ],[ 425 ] [ 870 ],[ 6 ] [ 396 ],[ 1136 ] [ 164 ],[
697 ] [ 911 ],[ 664 ] [ 346 ],[ 687 ] [ 490 ],[ 1174 ] [ 602 ],[ 900 ] [ 957
],[ 144 ] [ 1061 ],[ 1077 ] [ 1019 ],[ 1279 ] [ 690 ],[ 1150 ] [ 239 ],[
858 ] [ 932 ],[ 251 ] [ 374 ],[ 212 ] [ 265 ],[ 328 ] [ 178 ],[ 1129 ] [ 614
],[ 1315 ] [ 39 ],[ 1155 ] [ 791 ],[ 84 ] [ 228 ],[ 272 ] [ 152 ],[ 1322 ] [
82 ],[ 622 ] [ 855 ],[ 1182 ] [ 338 ],[ 1024 ] [ 789 ],[ 627 ] [ 359 ],[ 77 ]
[ 982 ],[ 501 ] [ 103 ],[ 56 ] [ 1069 ],[ 1157 ] [ 204 ],[ 324 ] [ 205 ],[
311 ] [ 172 ],[ 1090 ] [ 980 ],[ 613 ] [ 262 ],[ 200 ] [ 592 ],[ 754 ] [ 11
],[ 696 ] [ 868 ],[ 659 ] [ 941 ],[ 64 ] [ 979 ],[ 8 ] [ 1059 ],[ 112 ] [ 689
],[ 1192 ] [ 978 ],[ 665 ] [ 901 ],[ 1128 ] [ 769 ],[ 224 ] [ 486 ],[ 555 ] [
15 ],[ 202 ] [ 281 ],[ 995 ] [ 900 ],[ 1404 ] [ 478 ],[ 473 ] [ 654 ],[ 595 ]
[ 290 ],[ 478 ] [ 13 ],[ 73 ] [ 91 ],[ 588 ] [ 12 ],[ 988 ] [ 864 ],[ 539 ] [
```

(b)

FIGURE 11. Secret key bit embedding sequence within the carrier image through the proposed scheme (a) Secret key bit embedding sequence within the carrier Flowers image. (b) Secret key bit embedding sequence within the carrier House image.

FIGURE 12 shows comparative results of secret key bits' position within the stego-image and Crypto-Stego image with our proposed algorithm for sample images ((Flowers (303 × 284 pixels), Parrot (370 × 341), Girl (321 × 387)). FIGURE 12 (a1-a3) shows the original RGB colored images followed by the secret key bit positions within the stego-images (FIGURE 12 (b1-b3)) and with the proposed Crypto-Stego-images (FIGURE 12 (c1-c3)) as grayscale images with the proposed nearest-centroid clustering method. The comparison is performed by matching of the carrier image vs. the stego image with an in-house stego analyzer tool. In FIGURE 12 (b1-b3) the position of secret key bits in a stego-image are shown in white within red bounding boxes, while the position of secret key bits in an encrypted image (Crypto-Stego-image) are shown in white encircled in red within FIGURE 12 (c1-c3).

The results confirm that in a Crypto-Stego image, greater dispersion of secret key bits and lower distortion within the carrier has been achieved as compared to the stego images. Thus, the proposed Crypto-Stego scheme attains enhanced confidentiality and imperceptibility compared to the stego-images. Moreover, the secret key bits dispersion due to nearest-centroid clustering followed by AES-OFB encryption makes retrieval of the cover image retrieval a very much complex task for a stego-analyst operating without knowing the exact decryption key.


a1 b1 c1

a2 b2 c2

a3 b3 c3

FIGURE 12. Comparative results of secret key bit positions with the proposed algorithm for sample images of different sizes and resolution (Flowers (303 × 284 pixels), Parrot (370 × 341), Girl (321 × 387)): (a1-a3)Cover images. (b1-b3) Position of secret bits within the stego images, and (c1-c4) Position of secret bits within the carrier images. All positions are marked by red boundaries.

In order to improve the computational cost and efficiency of the proposed scheme parallel processing is applied for the encryption and embedding processes. Table 1 compares the execution time of the proposed Crypto-Stego scheme with standalone processing and parallel processing on the sample images. The computational time is measured in seconds based on a C# implementations on a computer with Intel Core (TM) i-3 4010U CPU processor at 1.70 GHz, as previously mentioned.

The computational time allocation for the encryption process and embedding process separately for sample images. It can be observed that the proposed scheme achieves a higher computational time for the embedding process (compared to sequential processing) as compared to encryption. Therefore, the proposed method does not add much more computational overhead arising from encryption. The results also show that an average (arithmetic mean) speed-up of 1.5 times is achieved with the parallel processing. The speed-up factor is the increase in execution speed with the parallel processing. Mathematically speedup factor is calculated as follows:

$$Speedup(S_p) = (T_s) / (T_p) \qquad (10)$$

where$T_s$is execution time with a single processor and T_p is the execution time with multiple processors. T_p includes communication time as well as computational time. Equation (11) represents the efficiency of the parallel processing, which is directly proportional to the speedup factor and inversely proportional to the number of processor used in the parallel processing. The communication overhead increases as the number of processors increases and, hence, affects the efficiency.

$$\textbf{Efficiency } (\mathbf{E}) = \frac{\boldsymbol{Sp}}{\boldsymbol{N}} * \mathbf{100} \qquad (11)$$

where$N$ is the total number of processors used in the parallel processing. The computational cost of the sequential processing (C_s) is the execution time with a single processor (T_s), while the cost of parallel processing (C_p) is (T_p*N). Therefore, the parallel processing cost (C_p) can be computed as:

$$\mathrm{C_p} = \frac{\mathrm{T_s} * \mathrm{N}}{\mathrm{S_p}} \qquad (12)$$

or by

$$\mathrm{C_p} = \frac{\mathrm{T_s}}{\mathrm{E}} \qquad (13)$$

The results shown in Table 1 also demonstrate that the efficiency of the scheme is improved by an average of 77.5% with parallel processing without affecting the image quality. Moreover, parallel processing is efficient if and only if (iff) the cost of parallel processing $C_p \approx$ Ts. FIGURE 13 shows the efficiency of the proposed scheme through the comparison of sequential time (T_s) and parallel execution cost (C_p). The results show that, the computational complexity of parallel processing increases with the increase of image size and color complexities (color intensities) due to greater communication overhead. The larger image or images with greater color intensities (color variation) incurred larger communication time because of greater scattering of pixels within the encrypted image. Clearly, *House* presents a special case as a result of its much larger number of pixels compared to the other test images.

TABLE 1.
COMPARATIVE SEQUENTIAL AND PARALLEL PROCESSING TIME OF SAMPLE IMAGES FOR THE CRYPTO-STEGO SCHEME.

Image	**Image Size (kb)**	**Encryption Time (s)**		**Embedding Time (s)**		**Total Execution Time (s)**		**Speedup via parallel execution**	**Efficiency via parallel execution in %**
		Sequential	**Parallel**	**Sequential**	**Parallel**	**Sequential (*T_s*)**	**Parallel (*T_p*)**		
Profile	78.3	0.023	0.015	0.38	0.22	0.41	0.24	1.72	86.07
Flower	87	0.036	0.019	0.70	0.39	0.74	0.41	1.80	90.25
Lina	110	0.025	0.016	0.70	0.41	0.73	0.43	1.70	85.28
Puppy	147	0.046	0.025	1.28	0.72	1.32	0.74	1.78	88.97
Pool	226	0.012	0.011	1.54	0.94	1.55	0.95	1.63	81.88
Flowers	252	0.038	0.026	1.17	0.74	1.21	0.76	1.58	78.83
Fruits	262	0.033	0.022	1.38	0.87	1.41	0.89	1.57	78.43
Art	306	0.037	0.025	1.48	0.94	1.52	0.97	1.58	78.78
Statue	356	0.034	0.023	1.59	1.00	1.63	1.02	1.60	79.55
Girl	364	0.055	0.04	1.64	1.04	1.70	1.08	1.57	78.41
Parrot	370	0.098	0.064	5.25	2.84	5.35	2.90	1.84	92.08
Kid	448	0.035	0.035	0.94	0.70	0.98	0.74	1.32	66.06
Strelitzia	485	0.028	0.03	5.97	3.27	6.00	3.30	1.81	90.77
Splash	569	0.041	0.043	2.57	1.78	2.61	1.82	1.42	71.39
Peppers	630	0.09	0.059	1.98	1.40	2.07	1.46	1.41	70.79
Watch	706	0.049	0.044	1.99	1.42	2.05	1.46	1.40	70.03
Caps	743	0.055	0.048	2.01	1.45	2.07	1.49	1.38	69.061
Ship	751	0.086	0.065	2.34	1.64	2.43	1.70	1.42	71.18
Serrano	1024	0.097	0.079	2.79	2.07	2.89	2.15	1.35	67.35
House	4577.28	0.234	0.238	51.38	45.69	51.61	45.92	1.12	56.19
Average (s)		**0.0576**	**0.046**	**4.46**	**3.48**	**4.51**	**3.52**	**1.55**	**77.57**

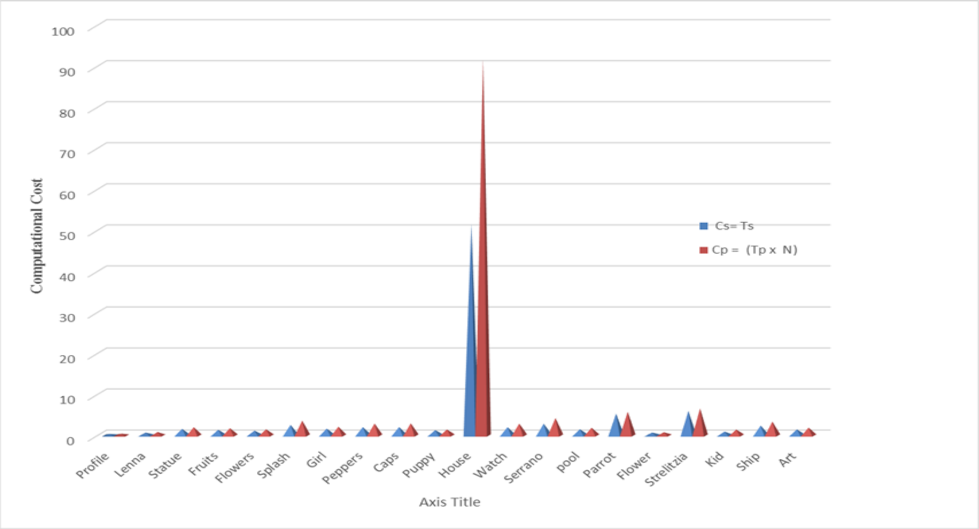


FIGURE 13. **Comparative analysis of sequential and parallel processing in terms of computational cost for the Crypto-Stego scheme.**

VI.STATISTICAL SECURITY ANALYSIS

This Section evaluates the performance and strength of the proposed scheme through statistical analysis by MSE and SSIM, Peak Signal to Noise Ratio (PSNR), and histograms

A.STRUCTURAL SIMILARITY INDEX (SSIM)

SSIM [34] is an important statistical method for measuring the similarity between two images, given that it tries to match the human visual system's response rather than being a pixel-by-pixel objective comparison. Therefore, for assessing the strength of the Crypto-Stego scheme, the structural similarity between the encrypted cover image and Crypto-Stego image was calculated. The value of SSIM index is between [-1, 1] and the resultant value 1 indicate that both images are identical to each-other while a value of zero shows that there is no correlation between two images. SSIM is calculated [33] using the formula of (14).

$$SSIM(a,b) = \frac{(2\mu_a\mu_b+D1)(2\sigma_{ab}+D2)}{(\mu_a^2+\mu_b^2+D1)(\sigma_a^2+\sigma_b^2+D2)} \qquad (14)$$

where (μ_a, μ_b) represents the average intensity value of images (a, b), ($\sigma^2_{a,}$ σ^2_b) represents the variance of images (a, b), and σ_{ab} represents the covariance of images a and b. D1 and D2 are two variables to stabilize the division factor. Table 2 shows the comparative statistical security analysis for sample images based on SSIM and MSE. The value of SSIM = 1 throughout for the comparison between original colored image vs. stego image and for encrypted image vs. Crypto-Stego image shows that as far as SSIM is concerned the steganographic images are identical to the carrier image. Likewise, the same resultant SSIM value 0.019 on average, for original colored image vs. encrypted image and original colored image vs. Crypto-Stego-images demonstrate the proposed scheme is robust against a statistical analysis of this kind to detect hidden information. Thus, the scheme provides much enhanced protection of a hidden decryption key within encrypted images.

B.MEAN SQUARED ERROR (MSE)

Mean Squared Error **(**MSE) is used to show the difference between two images i.e. carrier image and stego-image. MSE outcome is always a non-negative value, with values closer to zero indicating a minimal difference between the cover and stego images. MSE is calculated as [35]:

$$MSE(x,y) = \frac{\sum_{i=1,j=1}^{m.n}(x_{i,j}-y_{i.j})^2}{mn} \quad (15)$$

where (m,n) are an image pixels' dimensions and (x,y) are the pixel coordinates within an image.

Table 2 provides the MSE values for sample images through the proposed scheme. The average, 0.001 value of original colored image vs. stego-image and 0.0594 for encrypted image vs. Crypto-Stego-image shows that the secret key is embedded in such a way that produces negligible 'noise' in the carrier which are undetectableeven with the MSE method. Additionally, an average 4705.73 MSE of original colored image vs. encrypted indicates that pixels are dispersed sufficiently before embedding the secret key within images and the same average 4705.73 value of MSE of cover image vs. Crypto-Stego image indicates the there is no difference between the encrypted and stego-image originals as far as MSE is concerned. Hence, a stego-analyst also cannot extract any useful information from a steganographic image through this method.

TABLE 2.
STATISTICAL SECURITY ANALYSIS OF THE PROPOSED SCHEME THROUGH SSIM AND MSE OF SAMPLE IMAGES

Image	Image Size (Pixels)	Secret Key Size (Bits)	Original colored Image vs. Stego-Image		Original colored Image vs. Encrypted-Image		Original Color image vs. Crypto-Stego Image		Encrypted Carrier Image vs. Crypto-Stego Image	
			SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE
Profile	(139 × 191)	256	1	0.0005	0.018	4302	0.018	4302	1	0.403
Lenna	(256 × 256)	256	1	0.001	0.016	4134	0.016	4134	1	0.027
Flower	(170 × 174)	256	1	0.004	0.009	4174	0.009	4174	1	0.461
Puppy	(236 × 213)	256	1	0.003	0.023	4776	0.023	4776	1	0.0003
Flowers	(303 × 284)	256	1	0.0005	0.026	4336	0.026	4336	1	0.004
Girl	(321 × 387)	256	1	0.001	0.011	6015	0.011	6015	1	0.030
Statue	(345 × 352)	256	1	0.0002	0.016	4554	0.016	4554	1	0.004
Parrot	(370 × 341)	256	1	0.0012	0.025	3578	0.025	3578	1	0.005
Kid	(373 × 410)	256	1	0.002	0.026	3949	0.026	3949	1	0.138
Pool	(379 × 203)	256	1	0.002	0.018	6919	0.018	6919	1	0.017
Ship	(427 × 599)	256	1	0.001	0.022	3303	0.022	3303	1	0.002
Art	(431 × 242)	256	1	0.002	0.016	8847	0.016	8847	1	0.0001
Splash	(451 × 430)	256	1	0.0002	0.024	3438	0.024	3438	1	0.001
Strelitzia	(471 × 351)	256	1	0.0008	0.020	4879	0.020	4879	1	0.0002
Watch	(501 × 481)	256	1	0.0007	0.020	4507	0.020	4507	1	0.006
Peppers	(506 × 425)	256	1	0.0009	0.020	3836	0.020	3836	1	0.003
Fruits	(512 × 512)	256	1	0.0003	0.025	5138	0.025	5138	1	0.001
Serrano	(555 × 629)	256	1	0.0005	0.018	5226	0.018	5226	1	0.008
Caps	(625 × 389)	256	1	0.0008	0.025	3498	0.025	3498	1	0.018
House	(1441×1085)	256	1	0.0000	0.0972	1730	0.0972	1730	1	0.000
Average			**1**	**0.001**	**0.019**	**4705.73**	**0.019**	**4705.73**	**1**	**0.0594**

C.PEAK SIGNAL TO NOISE RATIO (PSNR)

PSNR is a method traditionally used in comparative statistical analysis of images. It measures the quality of images in terms of PSNR (dB). It is calculated by dividing the signal strength by its mean squared error as given below [36].

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}_{\mathrm{r}}^{2}}{\mathrm{MSE}}\right) \qquad (16)$$

whereMAX_r^2 is the squared maximum pixel value that can exist in the image.

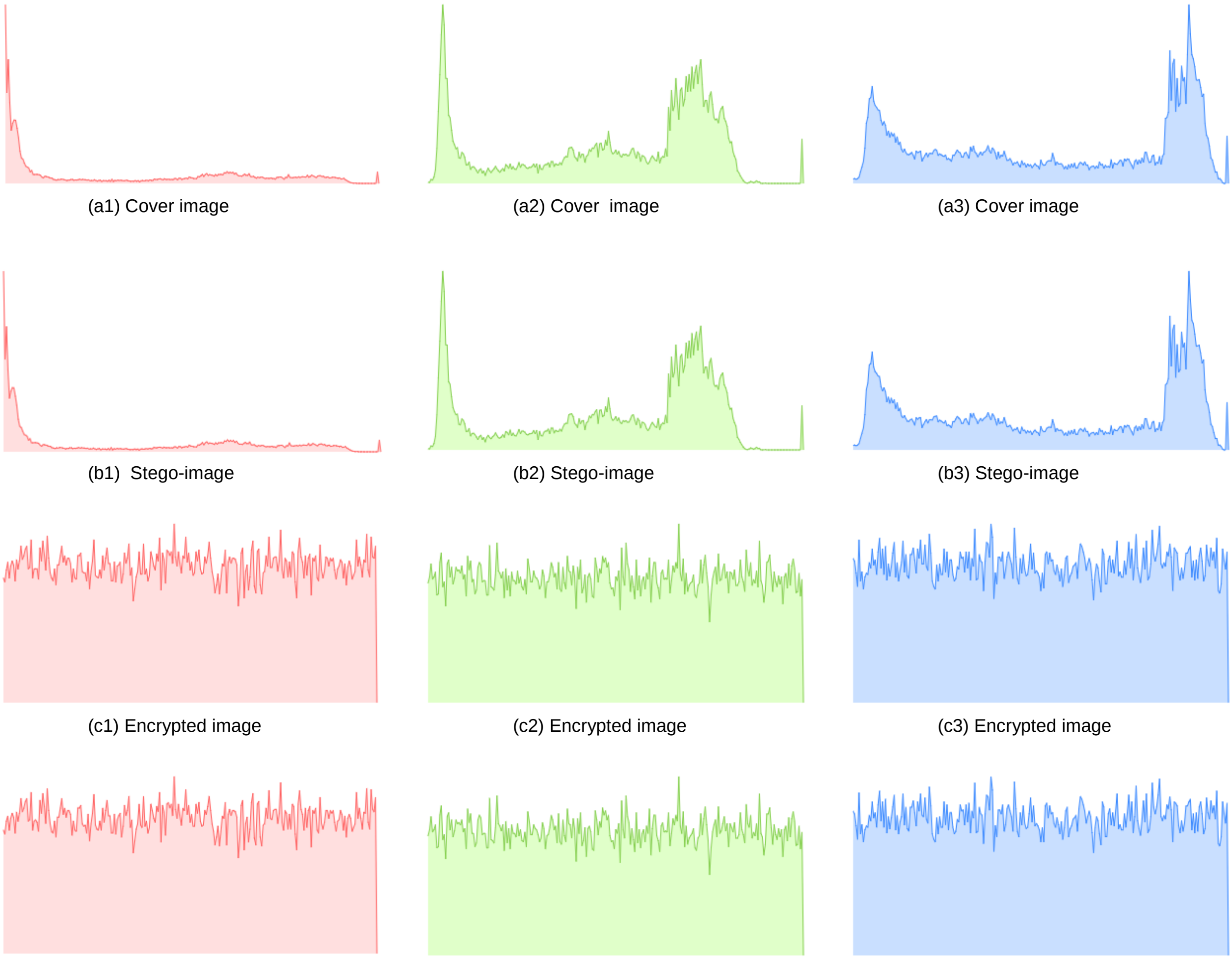
PSNR values, which are on a logarithmic scale, range between [0, infinity]. The ideal value of PSNR is infinity and, therefore, a higher value of PSNR indicates a better stego-image quality in comparison to the cover image. Average PSNR values of original colored image vs. stego-image and original colored image vs. Crypto-Stego image are provided in Table 3. From the results, it can be seen that the proposed scheme has promising results in terms of PSNR with average of 78.19 for stego-images and 69.36 for Crypto-Stego images.

TABLE 3.
PSNR OF COVER IMAGES VS. STEGO-IMAGES AND ENCRYPTED IMAGE VS. CARRIER IMAGES

Image	Size (Pixels)	Secret Key Size (bits)	Cover Image vs. Stego-Image	Encrypted Image vs. Carrier Image
			PNSR	PNSR
Profile	(139 × 191)	256	[Y:80.91,U:80.56,V:80.78] dB	[Y:51.55,U: 55.11,V:55.11]dB
Lenna	(256 × 256)	256	[Y:76.32,U:77.79,V:77.40] dB	[Y:70.27,U:68.94 ,V:65.41]dB
Statue	(345 × 352)	256	[Y:83.74,U:83.25,V:82.74] dB	[Y:72.09,U:61.43 ,V:73.20]dB
Fruits	(512 × 512)	256	[Y:82.24,U:81.75 ,V:81.93] dB	[Y:67.27,U: 61.91,V:69.49]dB
Flowers	(303 × 284)	256	[Y: 80.26,U:80.47 ,V:80.89] dB	[Y:77.53,U:71.28 ,V:70.98]dB
Splash	(451 × 430)	256	[Y:84.93,U:84.44 ,V:84.10] dB	[Y:67.64,U:72.54 ,V:77.45]dB
Girl	(321 × 387)	256	[Y:76.48,U: 76.45,V:75.87] dB	[Y:70.12,U:65.22 ,V:62.82]dB
Peppers	(506 × 425)	256	[Y:78.72,U:78.92 ,V:78.27] dB	[Y:71.39,U:59.80 ,V:72.90]dB
Caps	(625 × 389)	256	[Y:78.60,U:78.83 ,V:78.55] dB	[Y:73.39,U: 72.42,V:65.10]dB
Puppy	(236 × 213)	256	[Y:72.22,U:72.89 ,V:72.72] dB	[Y:62.24,U:55.35 ,V:83.49]dB
House	(1441× 1085)	256	[Y:80.68,U:81.96 ,V:81.79] dB	[Y:84.27,U:78.24 ,V:82.04]dB
Watch	(501 × 481)	256	[Y:79.08,U:79.23 ,V:78.71] dB	[Y:69.90,U:64.36 ,V:65.93]dB
Serrano	(555 × 629)	256	[Y:81.02,U:81.03 ,V:80.75] dB	[Y:68.70,U: 94.38,V:65.29]dB
Pool	(379 × 203)	256	[Y:74.32,U: 74.76,V:74.22] dB	[Y:65.07,U:63.84 ,V:88.45]dB
Parrot	(370 × 341)	256	[Y:76.67,U: 76.94,V:76.20] dB	[Y:68.86,U:81.01,V:55.79]dB
Flower	(170 × 174)	256	[Y:71.66,U: 71.61,V:71.36] dB	[Y:50.97,U:65.66,V:63.43]dB
Strelitzia	(471 × 351)	256	[Y:78.81,U: 79.10,V:78.56] dB	[Y:83.66,U:68.21,V:76.37]dB
Kid	(373 × 410)	256	[Y:74.73,U:75.00 ,V:74.75] dB	[Y: 56.22,U:67.81,V:65.12]dB
Ship	(427 × 599)	256	[Y:77.63,U:78.51 ,V:77.67] dB	[Y:74.58,U:68.02,V:79.64]dB
Art	(431 × 242)	256	[Y:73.86,U:74.05 ,V:73.87] dB	[Y: 87.54,U:90.44,V:89.55]dB
Average			**[Y:78.14, U:78.37,V:78.05]dB**	**[Y:68.00, U:69.24, V:70.82]dB**

C.HISTOGRAM ANALYSIS

A histogram is a technique to graphically represent the color distribution of images. However, histogram analysis is generally used for stegoanalysis to detect hidden data in stego- images. Therefore, if the histogram of original cover images and stego-images are identically then the steganographic images are more resilient against statistical analysis based on histogram exploration. A histogram analysis of the RGB channels of the sample *Profile* image is shown in FIGURE 14. The histogram of the original *Profile* image for R-, G- and B-channels is shown is FIGURE 14 (a1) FIGURE 14 (a2) and FIGURE 14 (a3) respectively. FIGURE 14 (b1, b2, b3) illustrates the histogram of R-, G-, B-channels of the stego-images. Lastly, the histogram of the encrypted images for R-, G- and B-channels is shown in FIGURE 14 (c1, c2, c3) and FIGURE 14 (d1, d2, d3) illustrates the histogram of Crypto-Stego images for the same channels. The similar RGB histogram of the cover image (FIGURE 14 (a1, a2, a3) vs. stego-image (b1, b2, b3) and encrypted image (c1, c2,c3) vs. Crypto-Stego images (d1.d2,d3) with the proposed scheme indicates that the scheme resilient against histogram analysis.



(a1) Cover image

(a2) Cover image

(a3) Cover image

(b1) Stego-image

(b2) Stego-image

(b3) Stego-image

(c1) Encrypted image

(c2) Encrypted image

(c3) Encrypted image

(d1) Crypto-Stego-image

(d2) Crypto-Stego-image

(d3) Crypto-Stego-image

FIGURE 14. RGB Histograms of Profile image for R-, G- and B-channels: (a1, a2, a3) histogram of cover image, (b1, b2, b3) Histogram of stego-images, (c1, c2, c3) histogram of encrypted image, (d1, d2, d3) histogram of carrier image.

VII.COMPARATIVE ANALYSIS

This Section presents comparative analysis of the proposed scheme with existing research. For a statistical comparison, the PSNR, MSE and SSIM of the proposed scheme are compared with the existing approaches.

Table 4 demonstrates the confidentiality performance of the proposed scheme compared to schemes presented in the literature that also employed combined steganography and cryptography. The results show that the proposed scheme provides promising results in terms of efficiency, security, and robustness. The comparative analysis shows that the authors of the suggested scheme in [4] used visual cryptography along with the LSB technique to achieve a low computational complexity. However, their PSNR value is much lower than that of the proposed scheme, which indicates the possibility of the hidden secret key detectability within the carrier image is much higher as compared to our proposed scheme. Though in [20], the researcher achieved good visual quality and PSNR, however, proposed method is more vulnerable against stegoanlysis and hence less effective as compared to our proposed scheme. In [21], the authors achieved the much higher PSNR value of 74.39 by utilizing the LSB status bit for steganography, however, the computational efficiency of the proposed scheme is much low therefore inefficient in term of computational complexity. In [40] the authors applied the secret key based LSB steganography on the colored image by manipulating the RGB components of colored images and. achieved the good PSNR and SSIM of 0.99. However, their computational complexity is high as compared to this proposed scheme. In addition, in this proposed scheme parallel processing is applied and results are evaluated to verify the performance and the efficiency for sequential and parallel processing.

VIII. CONCLUSION

This paper presents the implementation of a crypt-stego scheme for color (RGB) images, representing enhanced content confidentiality. In the proposed solution, the cover image is encrypted with the AES algorithm and then the decipher key is embedded into the encrypted image by utilizing nearest-neighbor clustering and LSB-M technique. Scrambling of the cover image by using AES encryption prior to data hiding makes the identification of stenography more challenging for an adversary and less detectable by stegoanlysis methods. Moreover, as the decipher key is embedded in the carrier image, no separate mechanism is required to send the key to decipher the image, which makes the proposed scheme more efficient. Our experimental results indicate the advantage of our scheme compared to other methods. Furthermore, SSIM, MSE, PSNR and histogram analysis also confirm that our proposed scheme achieves better stego transparency and confidentiality against different statistical attacks. Additionally, with parallel processing, the efficiency of the proposed scheme has been improved considerably.

TABLE 4.
COMPARATIVE ANALYSIS OF THE PROPOSED SCHEME WITH EXISTING SCHEMES IN THE LITERATURE

Proposed Scheme	**Image Size (Pixels)**	**PSNR**	**SSIM**	**MSE**	**Steganography Scheme**	**Embedding Parameters**	**Encryption Scheme**	**Decryption key required**	**Time Complexity**	**Parallel Processing**
(Khan et al. , 2016) [4]	(256×256)	62.67	0.99	x	Magic Least Significant Bit Substitution Method (M-LSB-SM)	Achromatic component (I-plane) of the hue-saturation-intensity (HSI)	Multi-level Encryption (MLE)	Yes	Medium	No
(Abood, 2017)[9]	(256×256)	63.00	x	0.03	Hash-LSB	RGB pixels	RC4 and Pixel Shuffling	Yes	Medium	No
(Zhou et al., 2016)[20]	(256 × 256)	56.51	x	x	Improved LSB algorithm	Red and Blue Components	RSA	Yes	Medium	No
(Islam et al., 2014)[21]	(512 × 512)	74.39	x	0.023	LSB using status bit.	RGB Color component	AES	Yes	High	No
(Tayal et al., 2016) [38]	(256 × 256)	90.52	x	x	Improved Bit Plane Complex Steganography (IBPCS)	Regions Complexity using the chaotic map	Huffman coding + Visual Cryptography	No	Low	No
(Karim et al., 2011)[39]	(256 × 256)	62.10	x	x	DWT-Haar wavelet.	wavelet coefficients	Filter bank cipher	Yes	High	No
(Saraireh, 2013)[40]	(256 × 256)	56.72	0.99	x	Secret key based LSB	RGB matrix Green or Blue components	XOR	Yes	Medium	No
(Chaudhary et al., 2014) [41]	(512 × 512)	x	x	x	Status Bit LSB Substitution	Blue component of RGB	Visual Cryptography + Huffman encoding	No	Medium	No
Proposed scheme	**(256 × 256)**	**68.90**	**1**	**0.027**	**LSB-M**	**RGB color intensities + nearest-centroid clustering**	**AES**	**No**	**Low**	**Yes**

REFERENCES

[1] M. H. Memon, J. P. Li, I. Memon, and Q. A. Arain, "GEO matching regions: multiple regions of interests using content based image retrieval based on relative locations," Multimed. Tools Appl., vol. 76, no. 14, pp. 15377–15411, 2017.

[2] J. He, W. Lan, and S. Tang, "A secure image sharing scheme with high quality stego-images based on steganography," *Multimed. Tools Appl.*, vol. 76, no. 6, pp. 7677–7698, 2017.

[3] A. Sharif, M. Mollaeefar, and M. Nazari, "A novel method for digital image steganography based on a new three-dimensional chaotic map," *Multimed. Tools Appl.*, vol. 76, no. 6, pp. 7849–7867, 2017.

[4] K. Muhammad, M. Sajjad, I. Mehmood, S. Rho, and S. W. Baik, "A novel magic LSB substitution method (M-LSB-SM) using multi-level encryption and achromatic component of an image," *Multimed. Tools Appl.*, vol. 75, no. 22, pp. 14867–14893, 2016.

[5] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3. pp. 727–752, 2010.

[6] W. C. Wu and S. C. Yang, "Enhancing image security and privacy in cloud system using steganography," in *Proc. ofIEEE Int. Conf. on Consumer Electron.*, 2017, pp. 321–322.

[7] Q. Li, X. Liao, G. Chen, and L. Ding, "A novel game-theoretic model for content-Adaptive Image Steganography," in *Proc. of IEEE 37th Int. Conf. on Distributed Comput. Syst. Workshops*, 2017, pp. 232–237.

[8] A. M. Abdullah, "New approaches to encrypt and decrypt data in image using cryptography and steganography algorithm," *Int. J. of Computer Apps.*, vol. 143, no. 4, pp. 11–17, 2016.

[9] M.H. Abood, "An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms," in *Proc. of Ann. Conf. on New Trends in Info. & Commun. Technol. Apps.*, 2017, pp. 86-90.

[10] S. Atawneh, A. Almomani, and P. Sumari, "Steganography in digital images: Common approaches and tools," *IETE Tech. Rev.*, vol. 30, no. 4, p. 344, 2013.

[11] B. Pillai, M. Mounika, P. J. Rao, and P. Sriram, "Image steganography method using k-means clustering and encryption techniques," in *Proc. of Int. Conf. Adv. Comput. Commun. Informatics*, 2016, pp. 1206–1211.

[12] A. A. A. Gutub, "Pixel indicator technique for RGB image steganography," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 1, pp. 56–64, 2010.

[13] N. Zaker, A. Hamzeh, S. D. Katebi, and S. Samavi, "Improving security of pixel value differencing steganographic method," in *Proc. of 3rd Int. Conf. on New Technologies, Mobility and Security*, 2009, pp. 1-4.

[14] S. Saleh Al-Amri and N. Kalyankar, "Image segmentation by using edge detaction," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 3, pp. 804–807, 2010.

[15] B. S. Champakamala, K. Padmini, and D. Radhika, "Least Significant Bit algorithm for image steganography Overview of steganography," *Int. J. Adv. Comput. Technol.*, vol. 3, no. 4, p. 5, 2014.

[16] D. Lerch-Hostalot and D. Megías, "LSB matching steganalysis based on patterns of pixel differences and random embedding," *Comput. Secur.*, vol. 32, pp. 192–206, 2013.

[17] K. Qazanfari and R. Safabakhsh, "High-capacity method for hiding data in the discrete cosine transform domain," *J. Electron. Imaging*, vol. 22, no. 4, 2013.

[18] P.-Y. Chen and H.-J. Lin, "A DWT based approach for image steganography," *Int. J. Appl. Sci. Eng. Int.*, vol. 4, no. 4, pp. 275–290, 2006.

[19] W. Y. Chen, "Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques," *Appl. Math. Comput.*, vol. 196, no. 1, pp. 40–54, 2008.

[20] X. Zhou, W. Gong, W. Fu, and L. Jin, "An improved method for LSB based color image steganography combined with cryptography," in *Proc. of IEEE/ACIS 15th Int. Conf. on Computer and Info. Sci., 2016, pp. 1-4.*

[21] M. R. Islam, A. Siddiqa, M. P. Uddin, A. K. Mandal, and M. D. Hossain, "An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography," in *Int. Conf. on Informatics, Electron. and Vision*, 2014, pp. 1-6.

[22] D. Bloisi and L. Iocchi, "Image based steganography and cryptography," in *Proc. of 2nd Int. Conf. on Computer Vision Theory and Apps.*, 2007, pp. 127–134.

[23] K. Joshi and R. Yadav, "A new LSB-S image steganography method blend with cryptography for secret communication," in *Proc. of 3rd Int. Conf. on Image Info. Process.*, 2016, pp. 86–90.

[24] M. I. S. Reddy and A. P. S. Kumar, "Secured data transmission using wavelet based steganography and cryptography by using AES algorithm," *Procedia Computer Science*, vol. 85, pp. 62–69, 2016.

[25] R. Sridevi, V.L. Parachuri and K.S.S. Rao, "Image steganography combined with cryptography," *Int. J. of Computers & Technol.*, vol. 9, no. 1, pp. 977-984, 2010.

[26] S. Song, J. Zhang, X. Liao, J. Du, and Q. Wen, "A novel secure communication protocol combining steganography and cryptography," in *Procedia Engineering*, 2011, vol. 15, pp. 2767–2772.

[27] S. Narayana and G. Prasad, "Two new approaches for secured image steganography using cryptographic techniques and type conversions," *Signal Image Process. An Int. Journal(SIPIJ)*, vol. 1, no. 2, pp. 60–73, 2010.

[28] S. Usha, G. A. S. Kumar, and K. Boopathybagan, "A secure triple level encryption method using cryptography and steganography," in *Proc. of 2011 Int. Conf. on Computer Sci. and Network Technol.*, 2011, vol. 2, pp. 1017–1020.

[29] K. Joshi and R. Yadav, "A new LSB-S image steganography method blend with Cryptography for secret communication," in *Proc. 2015 3rd Int. Conf. Image Inf. Process.*, pp. 86–90, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2815037, IEEE Access

[30] P. Y. Pawar and S. H. Gawande, “M-Commerce security using random LSB steganography and cryptography,” *Int. J. Mach. Learn. Comput.*, vol. 2, no. 4, pp. 427–430, 2012.

[31] C. Singh and G. Deep, “Cluster based image steganography using pattern matching,” *Int. J. of Emerging Trends & Technology in Computer Sci.*, vol. 2, no. 4, pp. 47–50, 2013.

[32] C. Paar and J. Pelzl, “The Advanced Encryption Standard (AES),” in *Understanding Cryptography,* Berlin, Germany:Springer Verlag, pp. 87–117, 2010.

[33] M. A. Tayal and M. M. Raghuwanshi, “Review on various clustering methods for the image data,” *J. Emerg. Trends Comput. Inf. Sci.*, vol. 2, pp. 34–38, 2010.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, and E. P. Simoncelli, “Image quality assessment: from error measurement to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[35] H. L. Tan, Z. Li, Y. H. Tan, S. Rahardja, and C. Yeo, “A perceptually relevant MSE-based image quality metric,” *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4447–4459, 2013.

[36] M. Ghanbari, “Scope of validity of PSNR in image / video quality assessment,” *Electron. Lett.*, vol. 44, no. 13, pp. 9–10, 2008.

[37] P. Marwaha and P. Marwaha, “Visual cryptographic steganography in images,” in *Proc. of 2nd Int. Conf. Comput. Commun. Netw. Technol.*, 2010, pp. 1-6.

[38] N. Tayal, R. Bansal, S. Dhal, and S. Gupta, “A novel hybrid security mechanism for data communication networks,” *Multimed. Tools Appl.*, vol. 2, no. 3, pp. 1–28, 2016.

[39] S. M. M. Karim, M. S. Rahman, and M. I. Hossain, “A new approach for LSB based image steganography using secret key,” in *Proc. of14th Int. Conf. Comput. Inf. Technol., 2011*, pp. 286–291, .

[40] S. Saraireh, “A secure data communication system using cryptography and steganography,” vol. 5, no. 3, pp. 125–137, 2013.

[41] D. Chaudhary, S.Gupta and M.Kumari, "Novel hybid security mechanism for data communication networks", *Int. J. Information Privacy, Security and Integrity*, vol. 2, n