Multimedia Tools and Applications (2023) 82:33063–33085
https://doi.org/10.1007/s11042-023-14735-0

Securing encrypted image information in audio data

Zeba Shamsi[1] · **Dolendro Singh Laiphrakpam**[1]

Received: 18 March 2022 / Revised: 4 July 2022 / Accepted: 4 February 2023 /
Published online: 6 March 2023

Abstract
Advances in communication technologies have fueled growth in digital data transfer. Images are one of the most often conveyed types of digital information. Crytographic techniques help create cipher but also lure attackers as it indicates secret communication. To overcome, a method for encrypting a secret image and hiding it in audio data is proposed. The Ikeda map is used to create the encrypted image, which is then hidden in the audio's lifting wavelet transform. Various statistical experiments show that the proposed approach conceals the encrypted image while causing minimal changes to the audio. The proposed algorithm shows robustness towards noise addition or random audio crop attack by retrieving a visually perceivable image after the attack. The suggested approach outperforms the current algorithms in terms of imperceptibility and embedding capacity.

Keywords Image encryption · Ikeda map · Audio data · Information hiding · Lifting wavelet transform

1 Introduction

As communication technology improves, data transmission through the internet has a manifold. Cryptographic encryption techniques help in providing confidentiality by generating unintelligible data. However, this unintelligible data indicates some secret communication that lures attackers to perform various attacks to decipher the secret communication. Concealing cryptographic encryption data in some cover data helps reduce attacks. Concealing through embedding on the cover data can be carried out using spatial or frequency domain [13]. The challenges in embedding the information on cover data include high imperceptibility, embedding capacity and robustness. Increasing any of the mentioned three qualities reduces the other two qualities. The three qualities are interrelated [7]. Embedding on the

Dolendro Singh Laiphrakpam contributed equally to this work.

✉ Dolendro Singh Laiphrakpam
ldsingh.cse@gmail.com; ldsingh@cse.nits.ac.in

Zeba Shamsi
zeba_rs@cse.nits.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology Silchar, Silchar 788010, Assam, India

spatial domain has better embedding capacity than the frequency domain, but frequency domain embedding has got better concerning imperceptibility. Various authors have developed various methods to conceal multimedia data like text, audio, and image in some other multimedia data to conceal confidential data. Some of the text hiding methods are given in [6, 10, 15, 18, 19, 30]. Karakus et al. [15] proposed a method to conceal the doctor's comments in a medical image using Genetic Algorithm-Optimum Pixel Similarity. Karakus et al. method is compared with the classical similarity-based method and found to have more data hiding capacity. Ditta et al. [6] proposed a method to conceal personal text data using Arabic text as a carrier. The method hides the secret information using Unicode characters like Zero Width Joiner and Zero Width character. In order to achieve high security, the secret information is encrypted using bit inversion before concealing in the cover medium. Younus et al. [30] proposed a method to secure secret text by hiding it in a cover image. The secret text is compression using the Huffman technique and encrypted using the Vigenere cipher. Random blocks are selected from the cover image using the Knight tour algorithm (KTA) and the compressed, encrypted data are embedded using Exploiting Modification Direction (EMD) technique. Ren et al. [19] proposed a method to hide secret messages using Advanced Audio Coding (ACC) audio as a carrier. The method is designed to reduce the distortion in the carrier audio by combining Huffman codeword histogram, modifying quantized modify discrete cosine transformation and perceptual masking using the psychoacoustic model. Qi et al. [18] inspired by natural data hiding technique, proposed a method to securely transmit confidential data by creating a synthetic haze image that resembles some natural weather condition. The private data is embedded using the HILL cost function. Some demerits are possibilities of false during parameter estimation and binary embedding only. Rahman [10] proposed a method for securely sending confidential information for a nuclear reactor by hiding the information in the middle frequency of a DCT transformed cover image. The private data is converted into a binary sequence and every two bits are hidden in LSB1 and LSB2 of the middle frequency DCT transformed cover image. Since the direct secret data are embedded, the possibility is that an attacker recovers the confidential data if he or she knows that communication is taking place using this method. Basu et al. [3] proposed a method to hide the secret image into another cover image. A differential evolution optimization algorithm is deployed to hide the secret data constructively.

As an image is one of the most communicated data, various authors have developed methods for concealing the secret image using various multimedia covers such as images, audio or videos. Some image hiding methods are given in [1, 8, 9, 11, 17, 22, 23, 25, 28, 31, 32]. Thanki et al. [23] proposed a method to hide a secret image using another cover image. The cover image's Ridgelet coefficient and wavelet coefficient are obtained by applying Finite Ridgelet Transform and single-level DWD, respectively. The secret image is scrambled using Arnold's transform and embedded in the LL subband of the cover image. The drawback of the method is that Arnold's transformation takes a lot of execution time. Valander et al. [25] proposed a method to hide a secret image using a cover image using Integer Wavelet Transform (IWT). The secret image is encrypted using a modified Logistic chaotic map to increase the security of the secret image. Banik et al. [11] proposed a robust binary image hiding technique using audio as the carrier. The image undergoes Arnold's scrambling operation and is embedded using the cocktail party effect at the LH subband of the Discrete Wavelet Transform of the audio carrier. El-Latif et al. [1] proposed a method to hide an image half the size of the cover image using an S-box created from quantum walks. The S-box determines the position of embedding the secret image data to the cover image. Mukherjee et al. [17] proposed a multi-bit embedding method that can embed 2-6 bits with

an image as a carrier or 6-13 bits with audio as a carrier. Embedding is done using pixel value differencing (PVD) on the spatial domain. Zhang [32] proposed a data hiding technique using an image as a cover. The cover image undergoes Haar IWT to avoid truncation error. The data are hidden in the edges of the LL sub-band based on multidirectional line encoding. Wahab et al. [26] proposed a method for hiding secret data on a cover image. The secret data is first encrypted using RSA (Rivest Shamir Adleman) encryption scheme and compressed using Huffman code. The encoded data is hidden using the LSB embedding technique on the LH, HL, and HH sub-band for DFT decomposed cover image. Abdulhammed et al. proposed a novel method to hide secret data into a cover image. The secret image data is stored in the edges of a cover image identified using a strong edges detection algorithm (SEDA). The position for embedding is computed using a random sequence generated using the Chirikov map. Yu et al. [31] proposed a reversible data hiding technique using audio as a carrier. The personal data is converted into novenary digits, embedded into the audio (single channel) carrier using a magic matrix, and converted into dual-channel stego-audio. The proposed method has got better signal-to-noise-ratio (SNR), and objective difference grade (ODG) than that of the method given by Xiang et al. [28]. Shafi et al. [22] proposed a method to hide data using audio as cover based on amplitude differencing. The method uses two audio covers of similar size to embed the data. The amplitude difference from the two cover audio with multiple indexes ranging from 0 to 255 (each at an interval of 16) is used to generate the stego-audio, where 4 bits of the secret data are embedded sequentially. The proposed method has good imperceptibility, but the maximum embedding ratio is just 12.5% of the cover audio and low robustness. El-Khamy et al. [8] proposed a method to hide image data in the audio signal. The image is encrypted using a random sequence generated from the Logistic map with XOR operation to increase security. The audio signal undergoes a two-level Cohen-Daubechies-Feauveau Integer to Integer lifting wavelet transform. The binary bits 0 or 1 of the encrypted image are hidden using a threshold technique in the second detail sub-band coefficients. El-Khamy et al. [9] proposed a method to hide image data into the audio signal using sample comparison with Discrete Wavelet Transform (DWT). The original image is first encrypted using the RSA encryption technique and binary data is generated from the cipher image. The position of embedding is based on a pseudo-random number generated. The cover audio is decomposed using DWD. The cipher bits are embedded by comparing the sample of DWD with a threshold value and making necessary changes to the coefficient of DWD if required based on the threshold. The method is robust to noise attacks, but it has a low embedding capacity.

Cipher images lure attackers to perform cryptanalysis as it indicates secret communication. Concealing the cipher image into a cover image while maintaining high imperceptibility, robustness and high embedding capacity are of research importance. Motivated by the various data hiding works, a method is proposed to conceal secret image information in audio data. The proposed method aims to improve the embedding capacity while maintaining high imperceptibility and robustness. The contribution of the paper includes:

1. The secret image is converted to non-intelligible data by scrambling and encrypting using Ikeda chaotic map. The initial conditions for the Ikeda map are secretly shared using Elliptic Curve Cryptography (ECC) cryptosystem.
2. To avoid a known-plaintext attack, the initial conditions for the Ikeda map depend on the Secure Hash Algorithm (SHA3-384) bits applied to the secret image.
3. To avoid suspicious communication taking place to an attacker, the secret image's intelligible data is concealed in the Lifting Wavelet Transform (LWT) of the carrier audio,

maintaining a good embedding capacity with significantly less compromise on the carrier audio quality.

4. To decrease the computation involved in random number generation using the Ikeda map, a base conversion operation is applied so that each loop generates 12 random values.

The following is the organization of this paper. In Section 3, the proposed scheme is described, which includes key exchange and keys generation for the Ikeda map, generating the scrambled-encrypted image and concealing the encrypted image into audio data along with reversing the process to get back the secret image. Section 4 shows the simulation results and the analyses of the proposed method. The conclusion is given in Section 5.

2 Ikeda map

Kensuke Ikeda pioneered the development of the Ikeda map [14]. The Ikeda map is a discrete time dynamic system given by:

$$\begin{aligned} x_{k+1} &= \alpha + \beta(x_k Cos[t] + y_k Sin[t]); \\ y_{k+1} &= \beta(x_k Sin[t] - y_k Cos[t]); \end{aligned} \tag{1}$$

where $t = c - \frac{d}{1+x_k^2+y_k^2}$, $\alpha \in (0, 8)$, $\beta \in (0.6, 1)$, $c \in (2, 5)$, $d \in (35, 50)$

Figure 1 shows the attractor, bifurcation diagram and Lyapunov characteristic exponents of the Ikeda map.

Equation 1 is iterated and plotted to depict the attractor as shown in Fig. 1a. The bifurcation plot with $\alpha \in (0, 8)$ with a step size of 0.005 and $\beta = 0.6$ is shown in Fig. 1b. A bifurcation diagram helps in visually examining the chaotic behavior. From Fig. 1b, it is seen that the chaotic behavior is more ergodicity for $\alpha > 2.3$. Lyapunov exponent measures the exponential separation for two minuscule close orbits with respect to variation in the control parameter. A positive Lyapunov exponent indicates that the system is chaotic. The Lyapunov exponent plot given in Fig. 1c denotes that the Ikeda system is chaotic. In order to generate a chaotic sequence for image scrambling and cipher image generation, the Ikeda map is used in the proposed scheme as it poses the desired chaotic behavior.

3 Proposed scheme

3.1 Key exchange and keys generation for Ikeda map

1 Get the 384 bits hash value (h) by applying (SHA3-384) on the input image.

2 Apply elliptic curve point multiplication operation between the hash value h and the generator G of a finite field elliptic curve given by $\varepsilon_\rho : y^2 \equiv x^3 + ax + b \bmod \rho$ to get an elliptic coordinate $hG(hG_x, hG_y)$. $hG(hG_x, hG_y)$ is secretly shared using ECC to the other communicating party [21].

3 hG_x and hG_y are converted into binary bits each of 384 bits.

4 Two sets of initial parameters for the Ikeda map are computed from the values of the bits from Step [3] as follows. For image scrambling, the binary bits from hG_x are used and for image encryption, the binary bits from hG_y are used. Where,
$x_0 = [b_1 ... b_{64}]_2/2^{64}$;
$y_0 = [b_{65} ... b_{128}]_2/2^{64}$;

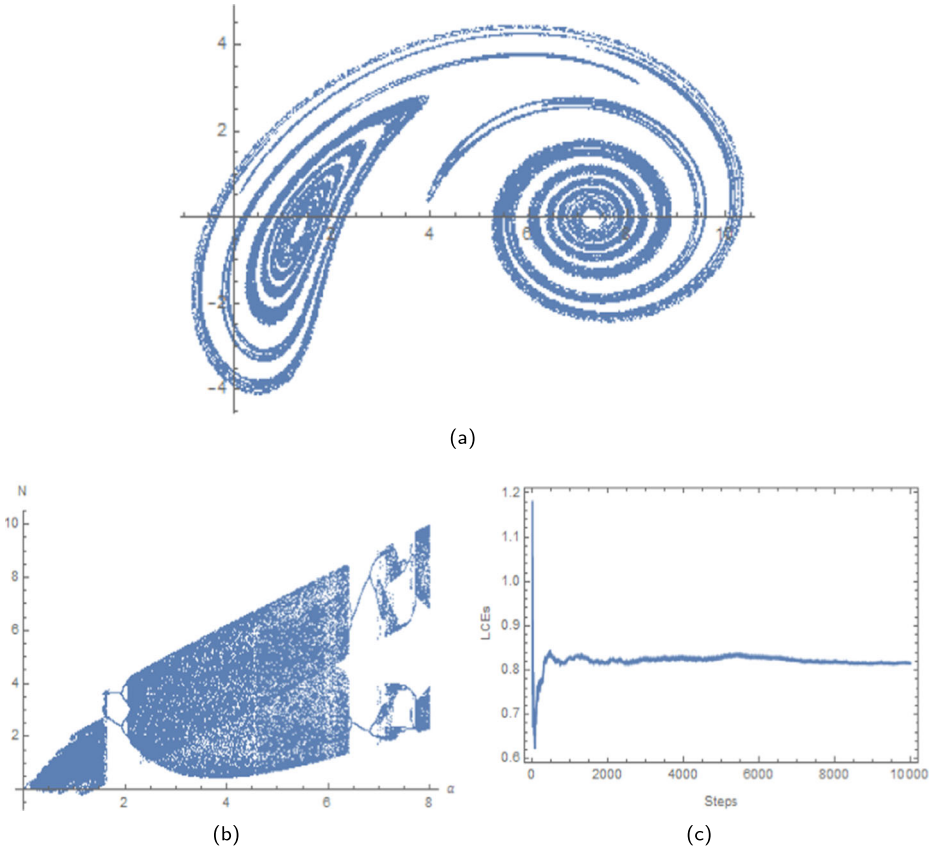


Fig. 1 Ikeda map (a) attractor (b) Bifurcation diagram (c) Lyapunov characteristic exponents

$\alpha = 5 + [b_{129}...b_{192}]_2/2^{64}$;
$\beta = e + [b_{193}...b_{256}]_2/(2^{64} \times 10)$;
$t = c - \frac{d}{1+x_k^2+y_k^2}$;
$c = 3 + [b_{257}...b_{320}]_2/2^{64}$;
$d = 43 + [b_{321}...b_{384}]_2/2^{64}$;
$[b_i...b_{i+63}]_2$ is the equivalent binary to integer conversion using 2 as base for each 64 binary bits, $e = 0.7$ and $e = 0.9$ for image scrambling and image encryption respectively.

3.2 Image scrambling

To generate a scrambled image from an colour input plain image the following steps are performed:

1 Input the plain RGB colour image and determine the image dimension ($M \times N$).
2 The Ikeda map is run for a loop of ($M \times N \times 3$) to generate a list of X and Y sequence using the initial parameters derived from hG_x.

3 The X and Y sequence are partitioned into three parts ($X_{Red}, X_{Green}, X_{Blue}$) and ($Y_{Red}, Y_{Green}, Y_{Blue}$) for each colour channel.
4 Each list is sorted and stored as ($X_{RSort}, X_{GSort}, X_{BSort}$) and ($Y_{RSort}, Y_{GSort}, Y_{BSort}$).
5 A permutation table ($P_{XRed}, P_{XGreen}, P_{XBlue}, P_{YRed}, P_{YGreen}, P_{YBlue}$) is generated by determining the position of each values of X_{Red} in X_{RSort}, X_{Green} in X_{GSort}, X_{Blue} in X_{BSort}, Y_{Red} in Y_{RSort}, Y_{Green} in Y_{GSort} and Y_{Blue} in Y_{BSort} respectively.
6 The input image is separated into the corresponding colour channels (I_R, I_G, I_B).
7 Each I_R, I_G and I_B are partitioned vertically into M parts and vertically scrambled using permutation table (P_{XRed}, P_{XGreen} and P_{XBlue} respectively to generate RGB vertically scrambled images.
8 The RGB vertically scrambled images are partitioned horizontally into N parts and horizontally scrambled using permutation table (P_{YRed}, P_{YGreen} and P_{YBlue} respectively to generate the horizontally scrambled images.
9 The output RGB images of Step [9] are combined to form the required scrambled image (S_{img}).

3.3 Image encryption

To generate the cipher image from the scrambled image (S_{img}) the following steps are performed:

1 The Ikeda map is run for a loop of $(M \times N \times 3)/12$ to generate a list of X and Y sequence using the initial parameters derived from hG_y.
2 Each values in X and Y are converted into 6 integers as follows:

$$\begin{aligned} S_\chi &= [X_i \times 10^{16}]_{256}[[2...7]] \\ S_\Psi &= [Y_i \times 10^{16}]_{256}[[2...7]] \end{aligned} \tag{2}$$

where, $[X_i \times 10^{16}]_{256}[[2..7]]$ and $[Y_i \times 10^{16}]_{256}[[2..7]]$ are the integer to base 256 conversion generating a list of values $\in (0 - 255)$ and the values from position 2 to 7 are taken.
3 The values in $S_X = (s\chi_1, s\chi_2, ..., s\chi_n)$ and $S_Y = (s\Psi_1, s\Psi_2, ..., s\Psi_n)$ are riffle as $S = (s\chi_1, s\Psi_1, s\chi_2, s\Psi_2, ..., s\chi_n, s\Psi_n)$ and used to generate a chaotic image C_{img}.
4 The chaotic image C_{img} is XORed with the scrambled image S_{img} to generate the encrypted image E_{img}.

3.4 Hiding encrypted image in audio data

To hide the encrypted image in an audio data the following steps are performed:

1 Import the audio file (.wav).
2 Extract the number of audio channel (ac), the sample rate (as) and the audio data (value ranges from -1 to 1) from the imported audio.
3 Combine all the channel (if more than one channel) and store as a single list. Apply Lifting Wavelet Transform (LWT) using Cohen-Daubechies-Feaveau (CDF) wavelet for 2 level (0,1,00,01) of refinement.
4 Generate three random integers (r_1, r_2, r_3) between 1 and the length of each list in the LWT 1, 00 and 01 refinement levels and share (r_1, r_2, r_3) secretly to the receiver.
5 The pixel values in encrypted image E_{img} is divided into 4 parts (P_1, P_2, P_3, P_4), where P_i are represented as triplet digits with necessary 0 padding at the left (for instance, if pixel value is 20, triplet digits=020). P_1 and P_2 are concatenated to form P_{12}.

6 The values in P_{12}, P_3 and P_4 replaces the fifth to seventh fractional part of the real digits starting at position (r_1, r_2, r_3) of the 1, 00 and 01 Level of LWT respectively with wrapping around if needed.
7 The Lifting Wavelet Data at level 0 along with the cipher data embedded DWD in Step [6] are combined together and Inverse Wavelet Transform is applied.
8 The data in Step [7] is partitioned based upon the number of audio channel (ac) and represented as audio (.wav) using the same sample rate (as).

3.5 Extracting encrypted image information

The following operations are performed to extract the encrypted image information from the stego-audio.

1 Import the stego-audio file.
2 Extract the audio data.
3 Apply Lifting Wavelet Transform (LWT) using Cohen-Daubechies-Feaveau wavelet for 2 level (0,1,00,01) of refinement.
4 Obtain the values of (r_1, r_2, r_3).
5 Extract the values of P_{12}, P_3 and P_4 from 1, 00, 01 level of LWT (step 2) from positions (r_1, r_2, r_3) respectively with wrapping around if necessary.
6 The encrypted image E_{img} information is obtained by combining P_{12}, P_3 and P_4.

Block diagram for enciphering and concealing in the proposed method is shown in Fig. 2a.

3.6 Image decryption

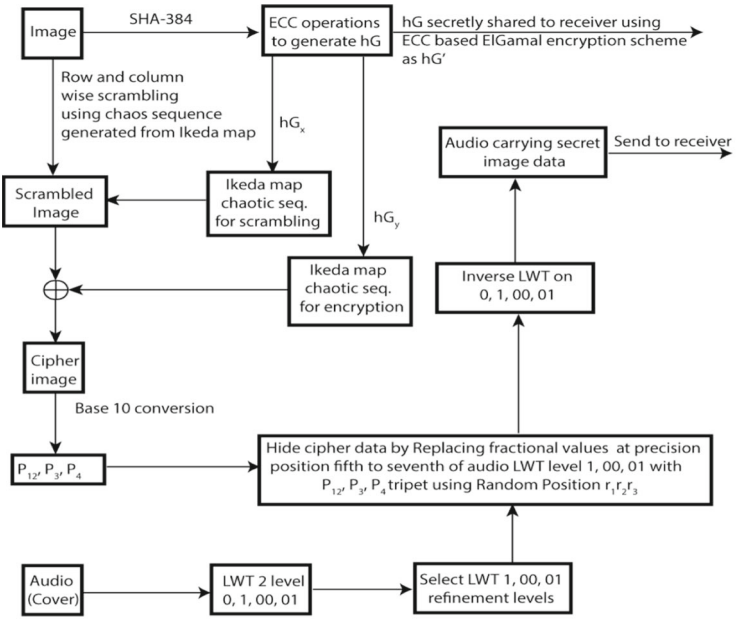
To generate the deciphered image from the encrypted image (E_{img}) the following steps are performed:

1 The Ikeda map is run for a loop of $(M \times N \times 3)/12$ to generate a list of X and Y sequence using the initial parameters derived from hG_y.
2 Each values in X and Y are used to generate S_X and S_Y as given in Step 2 of Section 3.3.
3 The chaotic image C_{img} is generated using the same process as given in Step 3 of Section 3.3.
4 The chaotic image C_{img} is XORed with the encrypted image E_{img} to generate the deciphered scrambled image S_{img}.

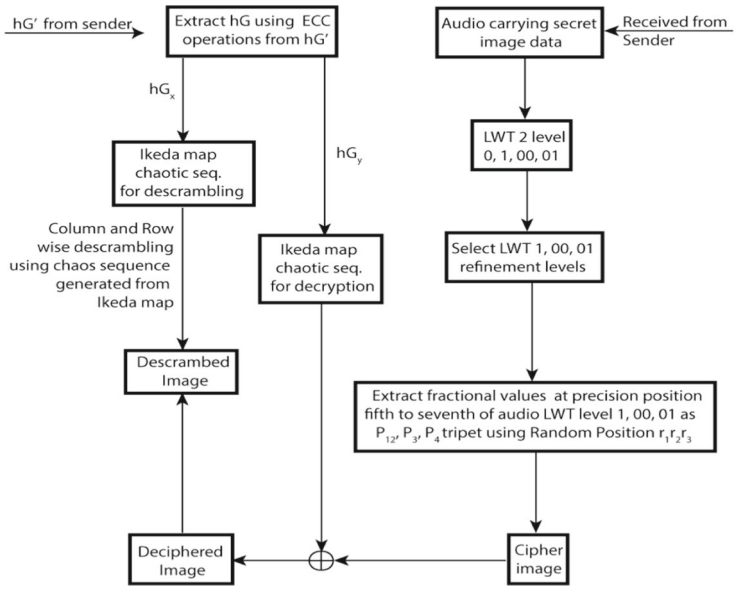
3.7 Image descrambling

To generate a descrambled image from the scrambled image S_{img} the following steps are performed:

1 Import the scrambled image S_{img}.
2 Generate X and Y sequence using initial parameters hG_x derived from the shared elliptic coordinate hG_x.
3 The X and Y sequence are partitioned into three parts $(X_{Red}, X_{Green}, X_{Blue})$ and $(Y_{Red}, Y_{Green}, Y_{Blue})$ for each colour channel.
4 Each list is sorted and stored as $(X_{RSort}, X_{GSort}, X_{BSort})$ and $(Y_{RSort}, Y_{GSort}, Y_{BSort})$.



(a)



(b)

Fig. 2 Proposed method block diagram for (a) enciphering and concealing the secret image in audio data. (b) deciphering and revealing the secret image from the audio carrying secret image data

5 Using the same process given in Step 5 of Image scrambling, a permutation table (P_{XRed}, P_{XGreen}, P_{XBlue}, P_{YRed}, P_{YGreen}, P_{YBlue}) is generated.
6 An inverse permutation table (IP_{XRed}, IP_{XGreen}, IP_{XBlue} is generated by determining the position of each values of 1 to M in (P_{XRed}, P_{XGreen}, P_{XBlue}. Similarly, an inverse permutation table (IP_{YRed}, IP_{YGreen}, IP_{YBlue} is generated by determining the position of each values of 1 to N in (P_{YRed}, P_{YGreen}, P_{YBlue}).
7 The scrambled image S_{img} is separated into the corresponding colour channels (SI_R, SI_G, SI_B).
8 Each SI_R, SI_G and SI_B are partitioned vertically into M parts and vertically descrambled using permutation table (IP_{XRed}, IP_{XGreen} and IP_{XBlue} respectively to generate RGB vertically descrambled images.
9 The RGB vertically descrambled images are partitioned horizontally into N parts and horizontally descrambled using permutation table (IP_{YRed}, IP_{YGreen} and IP_{YBlue} respectively to generate the horizontally descrambled images.
10 The output RGB images of Step [9] are combined to form the required descrambled image (I_{img}).

Block diagram for deciphering and revealing the secret image in the proposed method is shown in Fig. 2b.

4 Simulation and analysis of the proposed scheme

The proposed algorithm is simulated using Wolfram Mathematica 12.3 on Fujitsu Celsius workstation with configuration Intel(R) Xeon(R) W-2133 CPU @ 3.60 GHz 32 GB RAM. The sample images used are taken from the USC-SIPI Image Database [24]. The audios are taken from BBC Sound Effects [4]. The ECC technique uses the Brainpool [16] elliptic curve parameters for key exchange. Figure (3a) shows the plain image used as input. The input image is vertically and horizontally scrambled using the image scrambling technique given in Subsection 3.2 and shown in Fig. (3b) and (c) respectively. Figure (3d) shows the cipher image generated using the proposed method. The cover audio and the stego-audio is shown as an audio plot in Fig. (3e) and (f) respectively and the corresponding spectrogram is shown in Fig. (3g) and (h). Figure (3i-l) show the recovered cipher image, deciphered descrambled image, horizontally descrambled and vertically descrambled image, respectively. The absolute difference is calculated and depicted as an image to check the difference between the input image and deciphered descrambled image, as shown in Fig. (3m). Blacker the image, the lesser the difference. Figure (3n) depicts the absolute difference in audio magnitude between the cover audio and steg-audio. Lesser the difference, the magnitude of the amplitude tends to zero. The difference between the cover audio and stego-audio is minimal and lies between $+0.0001$ to -0.0001. The encrypted image data is concealed and distributed evenly across the audio data. The PSNR and SSIM value for the cover audio and stego-audio is tabulated in Table 1.

The PSNR and SSIM values of the stego-audio show that the stego-audio is very close to the cover audio, indicating high imperceptibility. The PSNR and SSIM value for the stego-audio, cipher image and the decrypted image is tabulated in Table 1. The PSNR and SSIM values of the decrypted image indicate that the original input image and the deciphered image are the same.

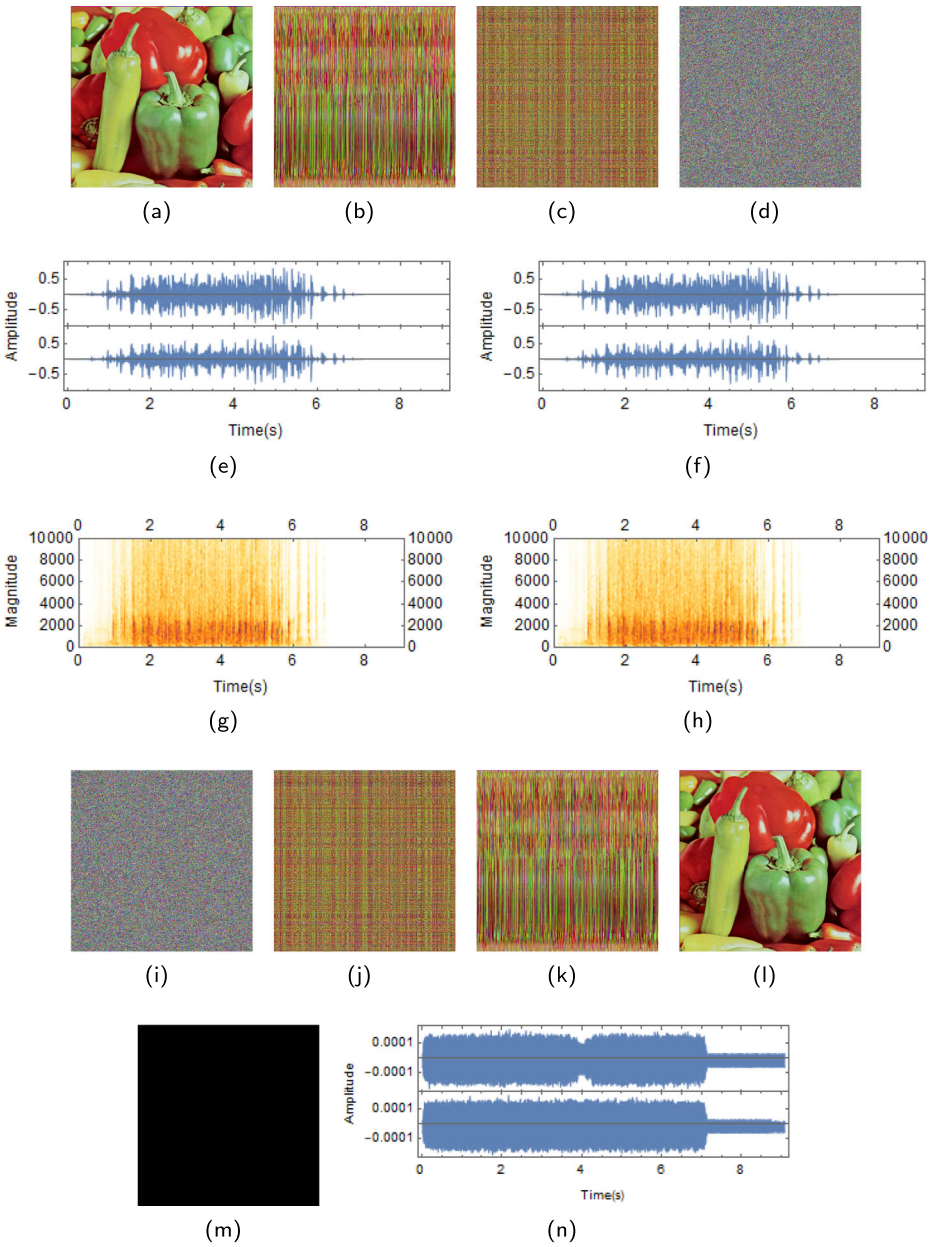


Fig. 3 (a) Sample image. (b) Vertically scrambled image of Fig. (3a). (c) Vertically scrambled image of Fig. (3b). (d) Encrypted image. (e) Cover audio. (f) Stego audio. (g) Spectrogram of cover audio. (h) Spectrogram of stego audio. (i) Recovered cipher image from stego audio. (j) Deciphered descrambled image. (k) Horizontally descrambled image. (l) Vertically descrambled image. (m) Image difference between Fig. (3a) and (l). (n) Audio difference between Fig. (3e) and (f)

Table 1 PSNR and SSIM of the stego-audio

	Stego-audio		Cipher		Decrypted	
Image	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Figure (3i)	86.424	0.999999	8.14766	0.0031375	∞	1

4.1 Embedding capacity

Embedding capacity is computed as:

$$\text{Embedding capacity} = \frac{\text{Size of secret data}}{\text{Size of cover data}} \times 100\% \tag{3}$$

The proposed method uses a 2200KB audio data to hide a colour image of size 768KB. The embedding capacity is 34.9%.

4.2 Noise attack

In a noise attack, random noise is induced in the stego-audio. The stego-audio is induced with noise whose magnitude ranges from $(-1, 1)$ for a certain percentage of the stego-audio duration to determine the robustness of the proposed algorithm against noise attack. Table 2 shows the PSNR, SSIM and BER values for the decrypted images under random noise attack (Fig. 4).

4.3 Random cropping attack

In a random cropping attack, certain audio signal parts are replaced by zeros. The stego-audio is applied with cropping attacks for specific durations to determine the robustness of the proposed algorithm against random cropping attacks. The random audio crop attack for (12.5%, 25%, 50%) on the stego audio are shown in Fig. (5a - c). The respective deciphered plain images are shown in Fig. 5d-f.

The PSNR, SSIM and BER values for the deciphered images under random cropping attack are tabulated in Table 2. The values show that the proposed method is robust against random cropping attacks and the generated deciphered images are visually perceivable.

4.4 Correlation analysis

The correlation coefficient shows how strongly two variables are related. Images are made up of pixel values arranged as a 2D matrix. The correlation coefficient analysis in cipher

Table 2 PSNR, SSIM and BER under noise attack and random cropping attack

	Decrypted under noise attack			Decrypted under random cropping		
% of attack	PSNR	SSIM	BER	PSNR	SSIM	BER
12.5%	17.8437	0.852625	0.123687	18.5274	0.873678	0.05928
25%	14.6807	0.705378	0.11411	15.0663	0.728767	0.11363
50%	11.7395	0.455331	0.24385	11.5033	0.427728	0.25760

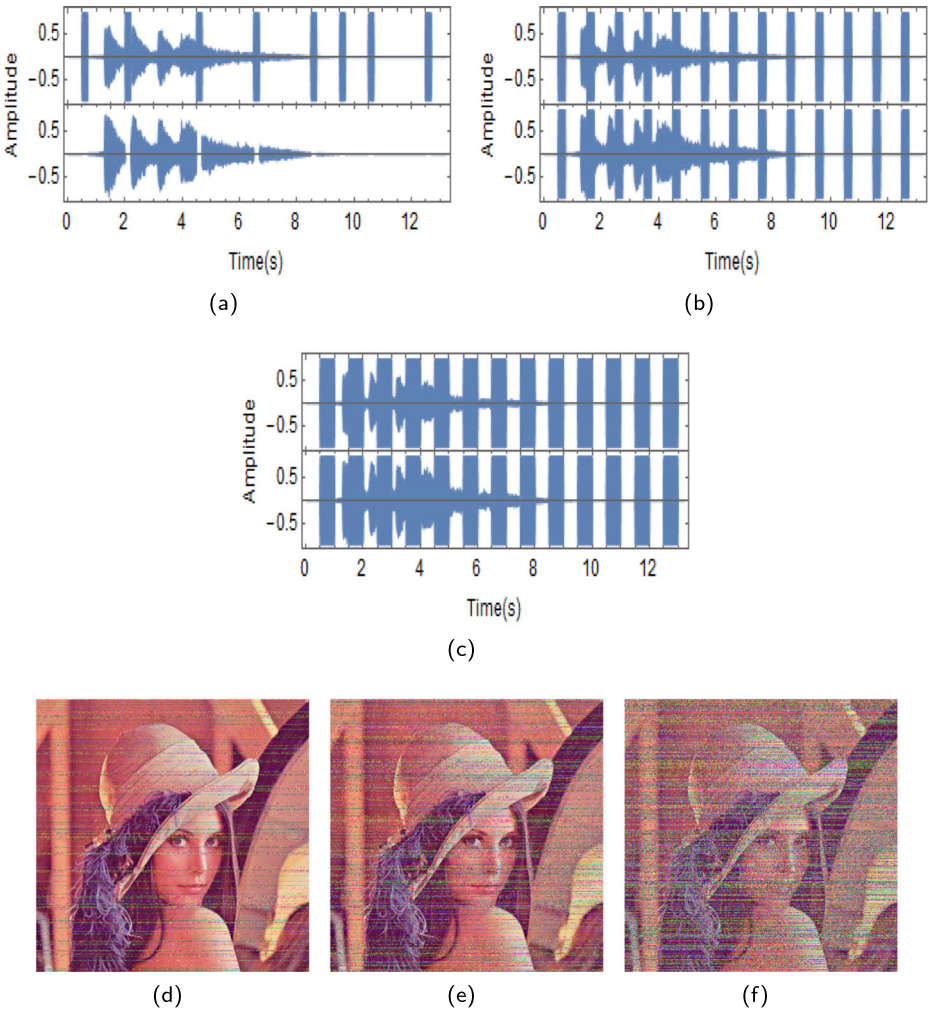


Fig. 4 (a-c) Steg-audio with 12.5, 25 and 50% noise attack respectively. (d-f) Deciphered images from steg-audio noise attack images respectively

image helps assess the encryption algorithm's statistical characteristic. The correlation coefficient is computed as:

$$\text{Correlation} = \frac{\sum_{i=1}^{n}(\chi_i - \overline{\chi})(\psi_i - \overline{\psi})}{\sqrt{\sum_{i=1}^{n}(\chi_i - \overline{\chi})^2}\sqrt{\sum_{i=1}^{n}(\psi_i - \overline{\psi})^2}} \tag{4}$$

where n is the number of pixels under consideration. Ten thousand points are randomly selected to calculate the correlation coefficient and horizontal, vertical and diagonal (HVD) directions. The correlation coefficient ranges from -1 to 1, where values toward -1 indicate anti-correlation, values towards 0 show no correlation and values toward 1 indicate high correlation. Standard images are usually highly correlated, indicated by values tending towards

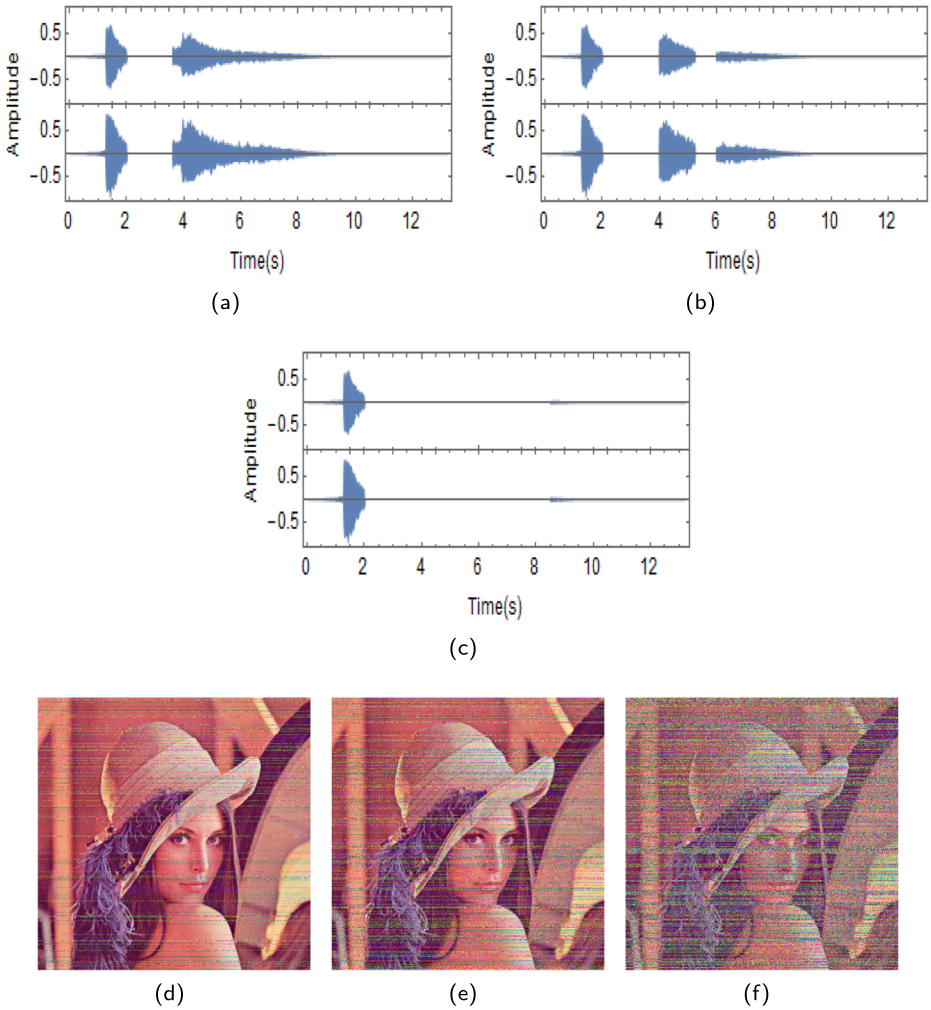


Fig. 5 (a-c) Random audio crop attack for (12.5%, 25%, 50%) on the stego audio respectively. (d-f) Deciphered images from stego audio under audio trim attack for (12.5%, 25%, 50%) respectively

1 and cipher images are lowlily correlated, indicated by values tending towards 0, as shown in Table 3.

Graphical plots of the correlation along HVD directions for plain and cipher images are shown in Fig. 6. The correlation graph is concentrated for plain images, indicating a higher correlation, while the correlation graph is dispersed for cipher images, indicating a lower correlation.

4.5 Attack analysis

The cipher-text-only attack is an attack where the adversary has access to only the cipher-text information. For the proposed model, if the adversary has access to the cipher-text, the

Table 3 Correlation coefficients for plain and cipher images

	Plain			Cipher		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	0.982023	0.982058	0.975287	0.000172	−0.005601	−0.004469
Peppers	0.979491	0.983450	0.971829	−0.002081	0.005132	0.000440
Splash	0.981477	0.982483	0.974056	0.008451	−0.002228	0.002109
Baboon	0.980496	0.982160	0.973199	−0.007918	−0.003943	0.003530

adversary cannot decipher the plain-text as he/she does not have the correct key. Determining the key would require solving ECDLP, a challenging problem, and trying a brute force attack is impractical. A known-plaintext attack is a type of attack where the adversary has access to cipher-text and the corresponding plain-text and tries to decipher a new cipher-text. A hash value is used for the proposed method to generate the secret keys using (SHA3-384). Every different image will have a different set of secret keys. So, the proposed method will be safe from cipher-text-only and known-plaintext attacks.

4.6 Histogram analysis

A histogram plot of an image depicts the frequency of each pixel value. A standard image usually has specific intensities conglomerated to provide meaningful information. So in

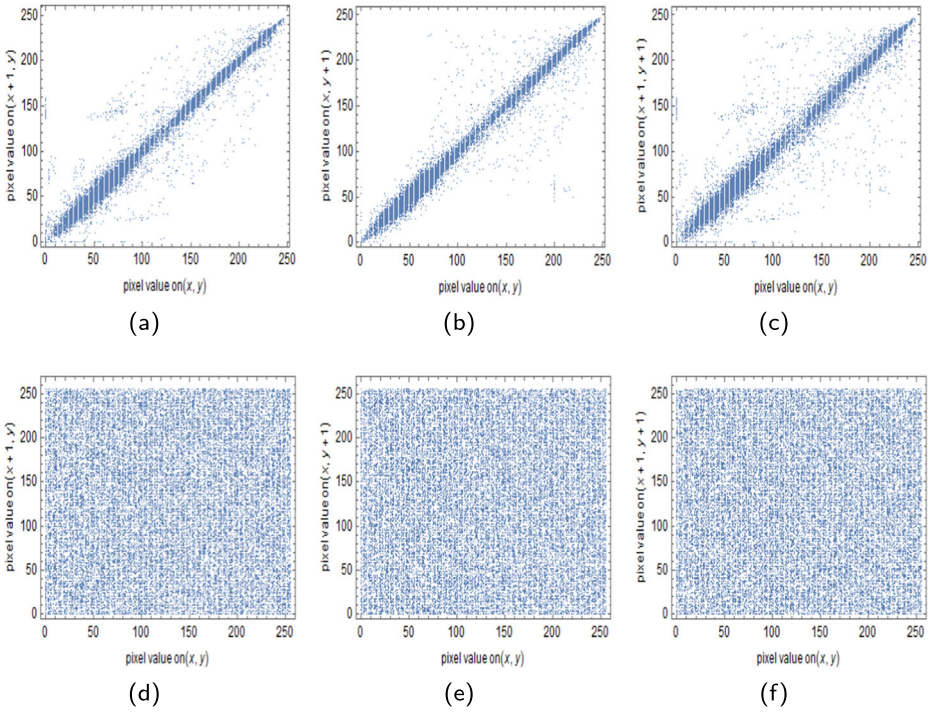


Fig. 6 HVD correlation graph for (a-c) Plain image. (d-f)Cipher image

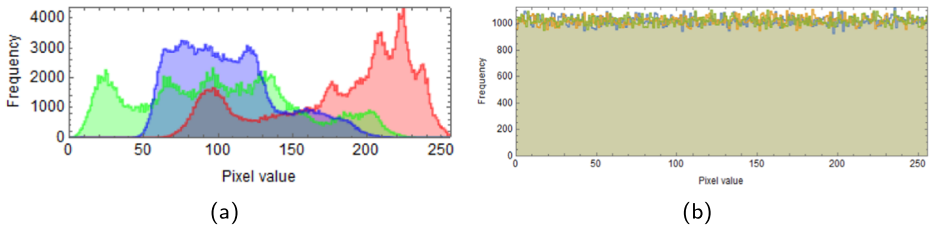


Fig. 7 Histogram plot of (a) Plain Lena image. (b) Cipher Lena image

plotting the histogram of a plain image, uneven distribution of pixels frequency is seen as shown in Fig. 7a. A good cipher image conceals any meaningful information with a uniform distribution of pixels frequency. So in plotting the histogram of a cipher image, uniform distribution of pixels frequency is seen as shown in Fig. 7b.

4.6.1 Variance

A variance of the histogram provides the mathematical confirmation for the histogram distribution. Lower the variance, the more uniform the histogram. Higher the variance, the more uneven the histogram. Variance is given by:

$$Variance = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{1}{2} (h_i - h_j)^2 \tag{5}$$

where, $N = 256$, h_i and h_j are number of grey pixels. Variance for the plain and cipher image are tabulated in Table 4.

4.6.2 Maximum deviation

The maximum deviation (MD) [5] is computed as:

$$MD = \frac{h_0 + h_{255}}{2} + \sum_{i=1}^{255} h_i \tag{6}$$

where, h_i is the absolute difference of the ith pixel count between the cipher and the plain image. MD measures the deviation of the cipher image from the plain image. Higher the difference better the encryption algorithm. The maximum deviation of the proposed method is tabulated in Table 4.

Table 4 Variance, maximum deviation and irregular deviation for plain and cipher images

	Plain	Cipher	MD	ID
Lena	2.72223×10^6	3535.49	355800	216088
Peppers	4.26665×10^6	3920.29	309242	254657
Splash	1.34622×10^7	3789.35	622609	416436
Baboon	2.49928×10^6	2903.29	356632	151984

4.6.3 Irregular deviation

The irregular deviation (ID) [5] is computed as:

$$ID = \sum_{i=0}^{255} (|H_i - A_h|) \tag{7}$$

where, H_i is the absolute difference of the ith pixel count between the cipher and the plain image; A_h is the mean of the histogram values. ID measures the closeness of statistical distribution between histogram deviation and the uniform distribution. A smaller ID indicated a better encryption algorithm. Irregular deviation for the proposed method is tabulated in Table 4.

4.7 Key space and key sensitivity analysis

The proposed algorithm uses 384 bits Brainpool parameters. The sender and receiver use a 384 bits private key to share the elliptic curve coordinate hG. Solving the private key from a public key in ECC requires solving the elliptic curve discrete logarithmic problem (ECDLP). Using Brute force requires computing CO computation, where CO is the cyclic order of the given 384 bits Brainpool parameter for a given Generator G. The best-known techniques, such Baby-step-Giant step method or the Pollard's rho method, requires $\sqrt{CO}$ steps to solve the ECDLP. $\sqrt{CO} = 4.65395 \times 10^{57}$ steps are large enough to baffle the attacker. The proposed algorithm is susceptible to keys. A single bit change drastically changes the data of the output. Figure (8a-b) shows the images of the deciphered image using the correct key and another utilizing a key that is just a bit different from the original key.

4.8 Differential attack

Cryptanalysis using differential attack tries to find non-random behavior in the cipher data generated from two minimally different plain inputs. Two input images are considered that differ just by a bit to test the robustness of the proposed encryption scheme against differential attacks. The cipher images' corresponding Number of Changing Pixel Rate (NPCR) and

Fig. 8 Decrypted using (a) Correct key $n_{receiver}$. (b) Incorrect key $n_{receiver} - 1$

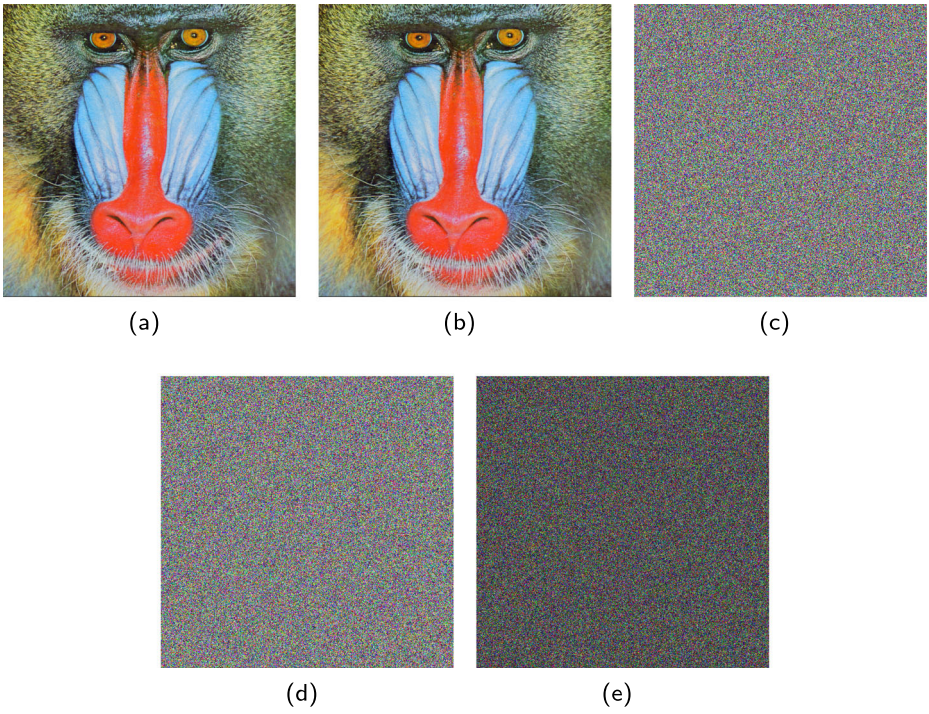


Fig. 9 (a) Original image. (b) One pixel value change from Fig. (9a) at location (255,255). (c) Encrypted image of Fig. (9a). (d) Encrypted image of Fig. (9b). (e) Image difference between Fig. (9c) and (d)

Unified Averaged Changed Intensity (UACI) values are analyzed. Wu et al. [27] proposed that a cipher image NPCR should be greater than the threshold criteria N^*_α, where α is the significance level. For UACI, the value should be in the interval of U^{*-}_α and U^{*+}_α. An image with dimension 512×512, should have N^*_α greater than 99.5717%, 99.5810%, 99.5893% and the interval of U^{*-}_α and U^{*+}_α of $33.3115\% - 33.6156\%$, $33.3445\% - 33.5826\%$ and $33.3730\% - 33.5541\%$ for significance levels 0.001, 0.01 and 0.05 respectively. Figure (9a) and (b) shows the two different images that just differ by a bit and the corresponding cipher image are given in Fig. 9c and d. Figure (9e) shows the image generating through absolute pixel difference between the two cipher images given in Fig. 9c and d.

The NPCR and UACI values are computed and tabulated in Table 5. The proposed encryption scheme passes the NPCR and UACI threshold criteria N^*_α, U^{*-}_α and U^{*+}_α.

4.9 Avalanche effect

The Avalanche effect is the desired property of a cryptographic algorithm where a slight change in the key or the plain image should drastically change the cipher image. The computed avalanche value for a single bit flipped in the input image and a single bit flipped in the input key is tabulated in Table 5. The avalanche values indicate that for a single bit flip in the input image or the key 50% of the cipher data got changed.

Table 5 Entropy, NPCR, UACI and avalanche effect

	Entropy		Cipher image		Avalanche effect	
	Plain image	Cipher image	NPCR (%)	UACI (%)	Bit change in image	Bit change in key
Lena	7.75020	7.99976	99.6284	33.4802	49.9743	50.0324
Peppers	7.66983	7.99977	99.6080	33.4867	50.0073	49.9840
Splash	7.24283	7.99977	99.5931	33.4528	49.9523	49.9903
Baboon	7.76244	7.99974	99.6040	33.4770	50.0086	50.0141

4.10 Entropy

In an image, entropy provides the randomness information on the distribution of pixels. Entropy is given by:

$$\text{Entropy} = -\sum_{i=1}^{2^n-1} P(p_i)\log_2 \frac{1}{P(p_i)} \tag{8}$$

where, n is the number of pixels and $P(p_i)$ denotes the probability of pixel p_i. In an image with pixels represented using 8 bits, the theoretical maximum entropy value is $log_2 2^8 = 8$. The higher the entropy values, the more uniformly the pixels are distributed. So, a cipher image tends to have entropy close to 8 compared to plain images. The Entropy value for the plain and cipher images is tabulated in Table 5.

4.11 Randomness test

Rukhin et al. [20] develop a standard statistical test suite (NIST test suite 800-22 revision 1a) that can be used to test the randomness of the cipher data. In each test a p-value is computed. A p-value greater than 0.01 suggests that the cipher data is random with the confidence of 99%. The fifteen tests are applied to each cipher image of Peppers, Lena, Splash and Baboon, and the p-values are tabulated in Table 6. The p-values indicates the encryption algorithm passes the randomness test with confidence of 99%.

4.12 Complexity of the proposed method

The proposed method consists of:

1. Computing the hash value (SHA3-384) of the input image: $O(n)$, where n is the number of bits in the input image.
2. Computation of elliptic curve point multiplication: $O(log_2 h)$, where h is the hash value of the input image.
3. Scrambling of the input image: $O(N^2)$, where N is the dimension of the image.
4. Chaotic sequence generation: $O(p)$, where p is the number of the pixels in the input image.
5. Computing LWT: $O(NlogN)$, where N is the audio data.
6. Concealing operation: $O(p)$, where p is the number of the pixels in the input image.

Table 6 Randomness test

	p-value			
Test	Peppers	Lena	Splash	Baboon
Monobit Test	0.030400	0.840126	0.769195	0.442572
Frequency Test	0.403648	0.327333	0.315931	0.346943
The Runs test	0.581766	0.817170	0.385508	0.373461
Longest-Run-of-Ones	0.640149	0.363535	0.166407	0.628545
Binary Matrix Rank Test	0.728893	0.964772	0.046054	0.754563
DFT Test	0.990075	0.078822	0.633310	0.147130
Non-overlapping test	0.234520	0.847266	0.510227	0.972892
Overlapping test	0.416913	0.501432	0.876996	0.084808
Universal Statistical Test	0.764317	0.341876	0.341433	0.341581
Linear Complexity Test	0.835431	0.090481	0.969487	0.565208
Serial test				
p-value 1	0.014753	0.828929	0.087447	0.175739
p-value 2	0.406904	0.551030	0.084788	0.370516
Approximate Entropy Test	0.184929	0.27646	0.397781	0.276460
Cumulative Sums Test	0.039204	0.724428	0.864933	0.171334
Random excursions test				
X=-2	0.117268	0.816181	0.487505	0.575564
X=-1	0.619707	0.997157	0.062991	0.247951
X=1	0.051927	0.792629	0.137640	0.041308
X=2	0.222671	0.553084	0.316819	0.200717
Random excursions variant test				
X=-2	0.419189	0.500721	0.077030	0.089489
X=-1	0.554243	0.628468	0.098711	0.041444
X=1	0.348446	0.456612	0.372315	0.041444
X=2	0.433719	0.302296	0.332716	0.116492

Overall, the complexity of the proposed method is $O(n) + O(log_2 h) + O(N^2) + 2O(p) + O(NlogN)$.

4.13 Comparison and discussion

Performance comparison of the proposed method with other techniques of data hiding is tabulated in Table 7. The proposed method had a good PSNR value for the embedded data, which is better than the existing techniques. The SSIM value of the embedded data for the proposed method is close to 1, where SSIM 1 indicates exact similarity with the original data. The embedding capacity of the proposed method is better than other existing methods. Discussion of the compared data hiding techniques are as follows:

1. In Ref. [30], the plain text is secure using the Vigenere cipher, and the payload is increased using the Huffman lossless compression technique. However, the Huffman dictionary table must be shared between the communicating parties for each secret message.

Table 7 Comparison with existing methods of data hiding

Method	Secret data	Cover data	Transformation	Security	PSNR/SSIM (Embedded)	Embedding capacity
Ref. [30] (2019)	Text (52.4 KB)	Image (256 KB)	KTA and EMD	VC	55.69/0.91	20.46
Ref. [10] (2018)	Text (1 KB)	Image (768 KB)	DCT	NIL	36.78/—	0.13
Ref. [3] (2022)	Image (6.6 KB)	Image (19 KB)	Differential Evolution	NIL	57.12/0.99	0.03
Ref. [23] (2018)	Image (192 KB)	Image (768 KB)	FRT and DWT	Scrambled using AT	57.37/—	25
Ref. [25] (2017)	Image (192 KB)	Image (768 KB)	IWT	Enc. using modified LM	54.87/0.99	25
Ref. [11] (2018)	Image (18.8 KB)	Audio (131.072 KB)	DWT and DCT	Scrambled using AT and CPP	72.87/0.99	14.3
Ref. [1] (2019)	Image (192 KB)	Image (768 KB)	Quantum walk	S-Box	44.41/0.93	25
Ref. [32] (2019)	Image (16 KB)	Image (768 KB)	IWT	NIL	50.75/0.99	2.08
Ref. [26] (2021)	Image (52 KB)	Image (768 KB)	Huffman coding or DWT	RSA	40.31/0.94	6.25
Ref. [2] (2022)	Text (48 KB)	Image (256 KB)	Elliptic curve point	ECC	48.13/0.97	18.75
Ref. [29] (2022)	Image (16.38 KB)	Image (192 KB)	SEDA	Scrambled using Chirikov map	61.31/0.98	8.53
Proposed	Image (768 KB)	Audio (2200 KB)	LWT	Scrambled and Enc. using IM	86.33/0.99	34.9

2. In Ref. [10], the confidential information of the nuclear reactor is hidden in middle band DCT coefficients by replacing the LSB. The technique possesses imperceptibility but has got low embedding capacity.
3. In Ref. [3], the confidential data is hidden in the insignificant perceptual region in the cover image determined by deploying the Differential Evolution optimization algorithm. The technique possesses imperceptibility and robustness but lacks embedding capacity.
4. In Ref. [23], the scrambled secret image is hidden in the DWT coefficient. The technique has a high embedding capacity; however, the scrambled secret image is obtained through Arnold's transformation, a permutation-only technique with finite cyclic order.
5. In Ref. [25], the secret image is enciphered using a modified Logistic map and embedded in the cover image using IWT. The modified Logistic map has got better key-space, thereby increasing the security. The method possess possesses imperceptibility and high embedding capacity.
6. In Ref. [11], the technique is blind steganography that uses both DCT and DWT to hide the secret image. Arnold's transformation is used to scramble the pixels to increase security. The technique possesses a strong imperceptibility and robustness but

has a moderate embedding capacity. The security can be improved because Arnold's transformation is a permutation-only technique with finite cyclic order.

7. In Ref. [1], a substitution box (S-Box) is developed using quantum walks. The secret data is expanded to the size of the cover image. The secret data is hidden in the cover image based on the S-box entries. The technique possesses a strong imperceptibility and embedding capacity but lacks robustness.
8. In Ref. [32], the secret image is embedded in the edge obtained from 3×3 non-overlapping blocks on the cover image. To avoid truncation errors, IWT is deployed. The problem of overflow or underflow in embedding for pixels with values close to 0 or 255 is also handled. The technique possesses a strong imperceptibility and shows robustness to some common attacks but has low embedding capacity.
9. In Ref. [26], the secret data is ciphered using RSA and compressed using Huffman coding, followed by embedding in the DWD LH, HL, and HH sub-bands. The method has got a moderate imperceptibility and embedding capacity. Choosing the primes of the RSA is a concern in this method. The smaller primes will generate a smaller value for the cipher, which will help better payload but is vulnerable to integer factorization attacks. Bigger primes will be secure, but the payload will decrease.
10. In Ref. [2], the plain text is secure through the elliptic curve encryption technique. However, the chosen elliptic parameter has a small cyclic order, and the technique uses a static table for each character.
11. In Ref. [29], the secret image is stored in the strong edges detected using SEDA, which helps in maintaining the robustness and imperceptibility, but it has low embedding capacity.

5 Conclusion

Images are one of the most transmitted digital data where specific images are confidential. Various image encryption methods are proposed by different authors that convert the plain image to an unintelligible image that looks noisy to maintain confidentiality. From an attacker's perceptive, these unintelligible noisy images indicate that something important is transmitted. Many such encryption schemes are cryptanalysed. Concealing the noise like encrypted data helps to baffle from cryptanalysis attack from the attacker as no clear indication is shown about the transmission of critical data. A method is proposed to safeguard the confidential image transmission by first converting the plain image into a cipher image based on scrambling and encryption operation using the Ikeda map. The cipher image is concealed in LWT audio data. Various statistical analyses are carried out to show that the technique blends the cipher image in the audio data without affecting the quality of the audio with high embedding capacity. The encryption algorithm passes the statistical and security analyses. The proposed technique shows robustness to noise attacks and random cropping attacks. Amongst the compared techniques given in Table 7, the proposed method has got the highest embedding capacity and better imperceptibility with PSNR and SSIM values of 86.33 and 0.99, respectively. The code of the proposed method can be obtained from https://github.com/Dolendro/Securing-encrypted-image-information-in-audio-data on demand. As a future work, concealing secret data in video using natural noise [12] for key generation can be researched.

Funding We confirm that there are no funding Sources.

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Competing interests We wish to confirm that there are no known conflicts of interest associated with this publication. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

References

1. Abd EL-Latif AA, Abd-El-Atty B, Venegas-Andraca SE (2019) A novel image steganography technique based on quantum substitution boxes. Opt Laser Technol 116:92–102
2. Bansal R, Badal N (2022) A novel approach for dual layer security of message using steganography and cryptography. Multimed Tools Appl 81:20669–20684
3. Basu S, Debnath A, Basu A et al (2022) An image data hiding technique using differential evolution. Multimed Tools Appl 1–18
4. Bbc sound effects. http://bbcsfx.acropolis.org.uk. Accessed 29 June 2020
5. Chattopadhyay C, Sarkar B, Mukherjee D (2015) Encoding by dna relations and randomization through chaotic sequences for image encryption. arXiv preprint arXiv:1505.01795
6. Ditta A, Azeem M, Naseem S et al (2022) A secure and size efficient algorithm to enhance data hiding capacity and security of cover text by using unicode. Journal of King Saud University - Computer and Information Sciences 34(5):2180–2191
7. Djebbar F, Ayad B, Meraim KA et al (2012) Comparative study of digital audio steganography techniques. EURASIP J Audio Speech Music Process 2012(1):1–16
8. El-Khamy SE, Korany N, El-sherif MH (2017a) Robust image hiding in audio based on integer wavelet transform and chaotic maps hopping. In: 2017 34Th national radio science conference (NRSC), IEEE, pp 205–212
9. El-Khamy SE, Korany NO, El-Sherif MH (2017b) A security enhanced robust audio steganography algorithm for image hiding using sample comparison in discrete wavelet transform domain and rsa encryption. Multimedia Tools Appl 76(22):24091–24106
10. El-Rahman SA (2018) A comparative analysis of image steganography based on dct algorithm and steganography tool to hide nuclear reactors confidential information. Comput Electr Eng 70:380–399
11. Gupta Banik B, Bandyopadhyay SK (2018) Blind key based attack resistant audio steganography using cocktail party effect. Secur Commun Netw. 2018:1781384:1–1781384:21
12. Hongjun Liu AK, Xu C (2020) Color image encryption with cipher feedback and coupling chaotic map. Int J Bifurcation Chaos 30(12):2050173(1)–2050173(14)
13. Hussain M, Wahab AWA, Idris YIB et al (2018) Image steganography in spatial domain: a survey. Signal Process Image Commun 65:46–66
14. Ikeda K, Daido H, Akimoto O (1980) Optical turbulence: Chaotic behavior of transmitted light from a ring cavity. Phys Rev Lett 45:709–712. https://doi.org/10.1103/PhysRevLett.45.709
15. Karakus S, Avci E (2020) A new image steganography method with optimum pixel similarity for data hiding in medical images. Med Hypotheses 139:109691
16. Lochter M, Merkle J (2010) Elliptic curve cryptography (ecc) brainpool standard curves and curve generation. https://doi.org/10.17487/RFC5639. https://www.rfc-editor.org/info/rfc5639
17. Paul G, Saha SK et al (2018) An efficient multi-bit steganography algorithm in spatial domain with two-layer security. Multimed Tools Appl 77(14):18,451–18,481
18. Qi B, Yang C, Tan L et al (2020) A novel haze image steganography method via cover-source switching. J Vis Commun Image Represent 70:102814
19. Ren Y, Cai S, Wang L (2021) Secure aac steganography scheme based on multi-view statistical distortion (sofmvd). J Inf Secur Appl 59:102863
20. Rukhin A, Soto J, Nechvatal J et al (2001) A statistical test suite for random and pseudorandom number generators for cryptographic applications. Booz-allen and hamilton inc mclean va, Tech. rep.
21. Sahasrabuddhe A, Laiphrakpam DS (2021) Multiple images encryption based on 3d scrambling and hyper-chaotic system. Inf Sci 550:252–267

22. Shafi K, Sankaranarayanan A, Prashanth G et al (2010) A novel audio steganography scheme using amplitude differencing. In: Trendz in Information Sciences & Computing (TISC2010), IEEE, pp 163–167
23. Thanki R, Borra S (2018) A color image steganography in hybrid FRT–DWT domain. J Inf Secur Appl 40:92–102. https://doi.org/10.1016/j.jisa.2018.03.004
24. The usc-sipi image database. http://sipi.usc.edu/database. Accessed 01 July 2020
25. Valandar MY, Ayubi P, Barani MJ (2017) A new transform domain steganography based on modified logistic chaotic map for color images. J Inf Secur Appl 34:142–151
26. Wahab OFA, Khalaf AA, Hussein AI et al (2021) Hiding data using efficient combination of rsa cryptography, and compression steganography techniques. IEEE Access 9:31805–31815
27. Wu Y, Noonan JP, Agaian S et al (2011) Npcr and uaci randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J Sel Areas Telecommun (JSAT) 1(2):31–38
28. Xiang S, Li Z (2017) Reversible audio data hiding algorithm using noncausal prediction of alterable orders. EURASIP J Audio Speech Music Process 2017(1):1–16
29. Younis AO (2022) A novel approach of steganography by using strong edge detection and chaos theory. Multimed Tools Appl 81(13):17875–17888
30. Younus ZS, Hussain MK (2019) Image steganography using exploiting modification direction for compressed encrypted data
31. Yu H, Wang R, Dong L et al (2020) A high-capacity reversible data hiding scheme using dual-channel audio. IEEE Access 8:162271–162278
32. Zhang H, Hu L (2019) A data hiding scheme based on multidirectional line encoding and integer wavelet transform. Signal Process Image Commun 78:331–344