Indonesian Journal of Electrical Engineering and Computer Science
Vol. 21, No. 3, March 2021, pp. 1692~1703
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v21.i3.pp1692-1703

Secure image hiding in speech signal by steganography-mining and encryption

Amal Hameed Khaleel, Iman Qays Abduljaleel
Computer Science Department, College of Computer Science and Information Technology, University of Basrah, Iraq

Article Info

Article history:

Received Oct 4, 2020
Revised Dec 6, 2020
Accepted Dec 20, 2020

Keywords:

Chaotic map
Cryptography
Data mining
Scrambling
Speech steganography

ABSTRACT

Information hiding techniques are constantly evolving due to the increased need for security and confidentiality. This paper proposes a working mechanism in three phases. The first phase includes scrambling the values of the gray image depending on a series of keys that are generated using a quantum chaotic map. The second phase generates hybrid keys by mixing a Zaslavsky and a 3D Hanon map that are used to encrypt the gray image values produced after the scramble. Finally, in the third phase, a new algorithm is suggested to hide the encrypted gray image at random locations within a speech file. This algorithm includes the LSB algorithm to determine the hidden bits and the zero-crossing K-means algorithm in selecting locations mining in a scattered manner so that hackers cannot easily retrieve the hidden data of any hacked person. Also used a fractional fourier transform to choose magnitude value as specific data to hide encoded image data. The measures MSE, PSNR, NSCR, and UACI are using to measure the work efficiency in the encryption algorithm, and in measuring the efficiency of the hidden algorithm, use the measures SNR, PSNR, and MSE. The results of the paper are encouraging and efficient compared to other algorithms that performed the same work. Hence our results show the larger the image dimensions used, the better the values.

Corresponding Author:

Amal Hameed Khaleel
Department of Computer Science
Basrah University, Basrah, Iraq
Email: amal_albahrany@yahoo.com

1. INTRODUCTION

Information hiding, a type of steganography, is one of the evolving techniques of embedding hidden information in digital media [1]. Digital steganography is the most effective protection technique used to preserve the privacy and confidentiality of data during transmission [2]. Steganography is an art that conceals data for any multimedia-based (cover medium) within the signals. It is used for invisible secret communication between parties by hiding the existence of secret data that changes within other media. Steganography's main goals include a high payload, enhanced robustness, and improved imperceptibility. Steganography requirements include a hidden file (text, image, video or audio), a carrier object (cover/host) media, a hide technique, and often a secret key and encryption technique to improve the security levels [1, 3].

Audio steganography is also one of the paper hotspots of security. Speech steganography is the way to hide the secret data in an audio medium which makes the secret data unnoticeable intruders [4]. Audio steganography is the most challenging technique in the world of steganography because the human auditory system is more extensive than the human visual system (HVS). The auditory system is the second largest source of information acquisition after the visual system. However, the diverse audio features of audio steganography,

such as loud noises tending to hide the quieter noises, make it a prominent steganography cover medium [5]. Combining a cryptography and steganography approach offers a higher degree of protection as a key is required to decrypt the secret message [2]. Also used The chaotic system improve robustness because it is a chaotic series of values used to generate a random series within a specified range. All audio (speech) encryption and speech steganography are used to secure sensitive information, but the security is somewhat different. Audio encryption hides the 'content' of the hidden secret signal, while the audio steganography hides the hidden 'behind' the host audio and converts the secret into something understandable [6].

The researchers have suggested many hiding methods and strategies, such as steganography. The method hiding of the image in the audio signal in [7]. The method is reliable, less complex and computation time and the discrete cosine transform and skin detection are used. A novel scheme of data hiding in video and audio files is suggested in [8]. The scheme used 4LSB and a phase encoding algorithm to preserve the quality of the video file even after secret data embedding. The application of the audio steganography with rivest cipher 4 (RC4) stream cipher encryption text is designed in [9]. The application is a windows shape that allows users to hide and recognize their message only in ".wav" audio. New technique is presented in [10] for data embedding uses a 10-digit number in the speech signal and embedded in frame samples of the LSB. The unique key is used to encrypt and decrypt signal for more secure transmission. The proposed method in [11] based on the function of an octal modulus modifying the cover samples. Three test implementations are performed to completely retrieve the audio cover file from the stego-one without using the position map by modifying the base of the unsigned integer (unit) function based on 8, 16, and 24 unit of the audio cover samples. The suggested approach in [12] to audio steganography used a combination of DCT coefficient computation and an AES encryption scheme to improve the security of the module. In [13], proposed a technique aimed to hide secret information in a host message (receiver). It used the LSB technique for embedding information in digital audio by replacing bits of samples in a circular manner from LSB to MSB or vice versa to hide a sequence of bits of secret text.

In this paper, speech file is chosen as carrier media and image data is hidden inside it using three algorithms: scrambling, encryption and hiding. This paper contains 5 sections. Section 2 discusses basic concepts; Section 3 elaborates on the methodology; Section 4 describes the results and simulation, and Section 5 ends with a conclusion.

2. BASIC CONCEPTS

In this paper, it used several methods and combined them in order to get good results. Below is an explanation of the basic concepts of these methods.

2.1. Quantum chaotic map

Combining quantum key with the chaotic signals could easily develop scrambling and cryptographic algorithms. This means that when it mixes two encoding methods (chaos cryptography and quantum cryptography), it produces a highly secure link [14]. It calculated chaotic map with the following (1) [15, 16]:

$$F(e_i, d_i) = \begin{cases} \frac{d_i}{e} & d_i = [0, e) \\ \frac{(1-d_i)}{(1-e)} & d_i = (e, 1] \end{cases} \quad (1)$$

where, (ei, di), respectively, the function variables and the initial state assigned to a specific map. It obtains values that have chaotic sequences that may be between (0-1) using the frequency F(e, d). It calculated quantum chaotic with the following Equations (2) [17]:

$$\text{Quantum Chaotic} = \begin{cases} x_{n+1} = d\,(x_n - |\,x_n\,|)^2 - dy_{n,} \\ y_{n+1} = -y_n e^{-2\beta} + e^{-\beta}\, d[(2 - x_n - x_n^*)y_n - x_n\, z_n^* - x_n^*\, z_n], \\ z_{n+1} = -z_n e^{-2\beta} + e^{-\beta}\, d[2(1 - x_n^*)z_n - 2x_n y_n - x_n]. \end{cases} \quad (2)$$

where, x∈[0,1], y ∈[0,0.1], z∈[0,0.2], x*=x, z*=z, β∈[6,∞), and d ∈[0,4]. δα represents a new quantum fluctuation about ⟨α⟩. Effects regarding quantum corrections produced by α=⟨α⟩+δα, x=⟨α⟩, y=⟨δαtδα⟩ and z =⟨δαδα⟩, x*, z* are complex conjugates of x and z respectively. β Is the particular dissipation parameter in the specific dynamic system and the intermediate values of β in addition to yn, zn≠0. If the quantum corrections yn, zn→0, the logistic map is one-dimensional.

2.2. Zaslavsky map

The Zaslavsky map is a discrete-time, nonlinear system. It exhibits complex behavior that is an integral part of the algorithms for encryption. It used it in this paper to generate pseudorandom bits of keys based on a speech encryption technique. Equations in a Zaslavsky map are as follows [18, 19]:

$$\text{Zaslavsky Map} = \begin{cases} x_{n+1} = \text{mod}\,(\,x_n + \text{v}(1 + \mu\, y_n) + \ \varepsilon\, \text{v}\, \mu \cos{(2\,\pi x_n)}, 1) \\ y_{n+1} = \ e^{-c}\,(y_n + \ \cos(2\pi x_n)) \end{cases} \quad (3)$$

where, v, c, ε are control parameters and e is exponentiation, and $\mu = \frac{1-e^{-c}}{c}$. The key set for the Zaslavsky map is $\{x_0, y_0, v, c, \varepsilon\}$.

2.3. Hénon map

The Hénon map is a complex 2D discrete-time system that exhibits chaotic behaviours. Michel Hénon introduced this as a comprehensible approach to the Poincare map resulting from the solution of the Lorenz complex Equations. The Hénon map is extremely sensitive to initial values, and changing parameters and initial values can create different chaotic sequences with a large translation, suggesting that it is suitable for creating cryptographic functions due to its ability to generate huge chaotic sequences, thus providing excellent pseudo randomness and unpredictability [20]. The Hénon map Equations [21] are defined as follows:

$$\text{Hénon map}(2D) = \begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \quad (4)$$

The 3D Hénon map refers to Equations [22] as follows:

$$\text{Hénon map}(3D) = \begin{cases} x_{n+1} = a - x_n^2 - bz_n \\ y_{n+1} = x_n \\ z_{n+1} = y_n \end{cases} \quad (5)$$

where, a, b are state variables (two bifurcation parameters a(>0)and b (>0)). x_n^2=The seed map.

2.4. Zero crossing rate (ZCR)

The zero crossing rate is a measure of the number of times the signal crosses the zero amplitude line by transition from a positive to negative or vice versa within a certain timeframe. It calculated ZCR with the following (6) [23, 24]:

$$ZCR = \frac{\sum_{n=1}^{N} |sg\ x(n) - sg\ x(n-1)|}{2N} \quad (6)$$

Where,
sg x(n)=the signal of the n-th sample x(n)
N=the total number of samples in the segment.
Sg x(n) can have three potential values:+1, 0,-1 depending on the sample being positive, zero or negative. The instantaneous accuracy is set at 20 ms because the human perceptual system is not usually accurate, also because speech signals stay stationary for (5-20) ms.

2.5. K-Means

Data mining is a common technique for data analysis that extracts useful knowledge from massive data. Clustering is an effective data-mining algorithm. K-Means is the most widely used method in clustering algorithms because of its easy and simple implementation, allowing fast learning for complex datasets. K-Means divides data into multiple clusters according to instance similarity, and has a strong dependency on the initial k cluster centers. Centroids in the initial cluster are typically chosen randomly [25, 26]. One calculates the distance between the sample xj ($1 \leq j \leq m$) and the center of each cluster μ_i (1<i<k) using the following (7) [27]:

$$d(x_j, \mu_i) = \sqrt{\sum_{t=1}^{n} |x_{jt} - \mu_{it}|^2} \quad (7)$$

The new mean vector is defined as follows:

$$new\ \mu_i = \frac{1}{c_i}\sum_{x \in c_i} x \quad (8)$$

It used K-Means algorithm to randomize the location to be hidden. After the locations in each block are divided into two classes, one of which has the value 1 and the other has the value 2, and the selection is made according to the category.

2.6. Least significant bit (LSB)

LSB is the simplest method to hide data in a speech file by replacing the bit of least significance of each bit in a message. It is good to increase the amount of data that can be covered, but it allows the amount of noise in the speech file to increase [28].

2.7. The fractional fourier transform (FrFT)

The Fractional Fourier transform provides generalization of conventional Fourier transform. FrFT is a generalized Fourier transform, which acts as an angle of rotation in the time-frequency plane and is also known as a rotational Fourier transform or angular Fourier transform. FrFT is a linear operator with angle (α) and signal (f(t)) according to the following (9)-(11) [28]:

$$F\alpha\ (x(t)) = \int_{-\infty}^{+\infty} f(t) k_\alpha\ (u,t) dt \quad (9)$$

$$k_\alpha(u,t) = \begin{cases} \sqrt{\frac{1-jcot\alpha}{2\pi}}\, e^{j((u^2+t^2)/2)cot\alpha - jutcosec\alpha} & if\ \alpha \neq k\pi \\ \delta(u-t) & if\ \alpha = 2k\pi \\ \delta(u+t) & if\ \alpha = (2k+1)\pi \end{cases} \quad (10)$$

The inverse FrFT is defined as [29]:

$$f'(t) = \int_{-\infty}^{+\infty} f_\alpha(u) k_{-\alpha}\ (u,t) du \quad (11)$$

Where,
δ(t): represents the Dirac function, α: represents the angle of rotation (α=απ/2)
It obtains a Fourier transform if α=π/2, while it gets the same signal if α=0, so the timefrequency representation of the signal is an intermediate value result of α:(0<α<π/2).

3. PROPOSED METHOD

The proposed algorithm (steganography-mining) consist of three phases as shown in Figure 1, while the pre-processing is not important because we did not need to distinguish or separate the speech signal from silence, but rather we dealt with it in the blocks form.

3.1. Scrambling phase

3.1.1. Quantum chaotic of index key algorithm

This algorithm generates new index keys for scrambling images. The steps are as follows.

a) Specify the two-dimensional size of the matrix for which you want to generate index key values, for example 64×64, 128×128, or 256×256 values.
b) Convert the two-dimensional matrix size to a one-dimensional vector size and save it in N; for example, if size is 64×64 values that means the vector size (N) is 4096 values.
c) Use the quantum chaotic map (2) with the specified initial values to generate a series of randomly generated key values with length N and save the result keys in one dimension vector of length N:
x1=0.45234444336, y1=0.003453324566, z1=0.001324523564
x*=0.00186, z*=0.00398, =4.489, d=3.99
d) Arrange the one-dimensional vectors in ascending order and save each value's original index before arranging it.
e) Save each value's result index in a one-dimensional vector for late use during scrambling.

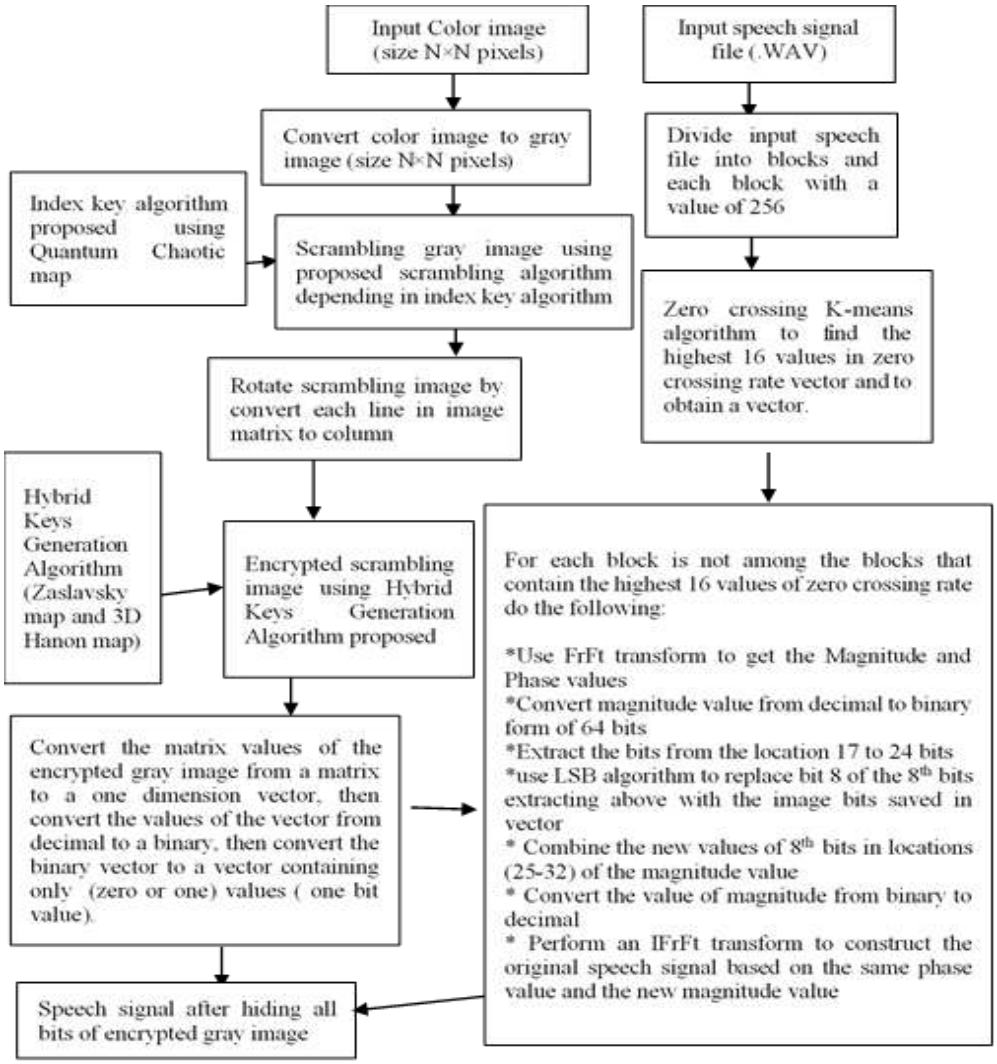


Figure 1. Flowchart of the proposed algorithm (steganography-mining)

3.1.2. Proposed scrambling algorithm

Input: color image
Output: gray image scrambled
The steps are as follows.

a) Convert the color image to a gray image size of 64×64 or 128×128 or 256×256 pixels.
b) Cut off the image into several segmentations by dividing the image dimensional by 4 (for example 256/4) and treat each segment independently in order to scramble the values.
c) Use the suggested algorithm (quantum chaotic of index key) to generate the index key vector with a size equal to the size of the divided matrix of the original image.
d) Arrange the data for each index of the divided image vectors, depending on the corresponding index key vector. In other words, arrange each value in the image vector, which contains, for example, 4096 values (if the image size is 265×256 pixels) according to the order of the keys in the index keys vector.
e) Collect segmentation matrices to reshape the original image after scrambling its values.
f) Repeat steps 2-5 by assuming that the size of the new image segmentation is due to dividing its dimensions by 2 (i.e. 256/2).
g) Repeat steps 2-5 again at the original image size (i.e. without cutting).
h) Note that the index key vector size that the quantum chaotic generated changes its size depending on the size of the new segment. So it becomes ((size×size) / 2) in step 6 and (size×size) in step 7.
i) Rotate the resulting image by converting each row into a column.
j) Save the resulting image for use in the next stage to encrypt the gray image.

3.2. Encrypting phase

3.2.1. The proposed hybrid zaslavsky-hénon map algorithm

This algorithm generates hybrid keys, which combine Zaslavsky and 3D Hanon maps for encrypting images. The steps are as follows.

a) Initialize the initial values for Zaslavsky map via experimental implementation using (3) as following: x_0=0.1, y_0=0.1, v=0.2, c=5, ε=9.
b) Initialize initial values for the parameters of a 3D Hénon map and based on the x, y values from the Zaslavsky map, generate a sequence of random values for x, y, and z for the 3D Hénon map using (5) as following: a=1.6, b=0.5
c) Convert the resulting fractional x values in to integer values based on the following relationship:

$$Xn=\text{mod}(\text{floor}(x\times 108), 256) \quad (12)$$

d) Save the integer string of random numbers resulting in a special vector for use in the encryption phase.

3.2.1. The proposed encrypting algorithm

Input: scrambled gray image
Outputs: gray encrypted image
The steps are as follows:

a) Use the suggested hybrid Zaslavsky-Hénon algorithm to generate the hybrid keys with lengths equal to the dimensions of the gray image (i.e., if the image is 256×256, the corresponding vector key size is 65,536 and so on).
b) Perform the XOR operation between the two values (one value from the input image matrix and another value from the key vector), and save the resulting value to the resulting encrypted image matrix.
c) Save the resulting value matrix in an image file for use in the next stage to hide the encrypted image in a speech file.

3.3. Hiding phase

3.3.1. Zero crossing k-means algorithm

One can use this algorithm to generate secret keys for hiding images in speech signals. It uses the zero crossing to mine locations and reduces the database for k-means clustering, which reduces the calculation time and makes the algorithm highly efficient. The mixing between products of the stego-key (secret) is very large. Since the length of the key decides the depth of security because of the large key size, then it is difficult to access the hidden data and vice versa.

Input: speech signal
Output: private secret key
The steps are as follow:

a) Read the input speech signal.
b) Divide the input signal into several blocks (i.e. each block has a size of 256 samples).
c) Calculate the coefficient of zero crossing rate for first block values according to (6)
d) Save the calculated zero crossing rate values in a vector (the length of the vector is the number of blocks for which it divided the entered speech file).
e) Choose the 16th largest coefficient in zero crossing rate vector calculated above and save the index of each block in the 16th largest zero crossing rate vector.
f) Convert each block sample (256 samples) in the 16th largest zero crossing rate vector to a two dimensional matrix of 16×16 values**.**
g) Divide the two-dimensional matrix values into two clusters (k cluster centers=2) using the k-means clustering algorithm, where the Euclidian distance is used.
h) Replace the first cluster with the value 1 and the second cluster with the value 2, and then combine the two clusters to obtain a vector with a length of 16 values**.**
i) Repeat 6-8 on all 16th largest zero crossing rate vectors to obtain a vector with 16 values for each one, then combine the 16 values vectors to create a new vector (A of 256 values), which is the private secret key that is used by the sender and recipient of the speech file**.**
j) Send the private secret key to both the sender (for use in hide algorithms) and recipient (for use in unhide algorithms).

3.3.2. Hide algorithm (steganography-mining)

The hide algorithm hides gray encrypted images in a speech signal.

Input: speech signal file, encrypted gray image

Output: Steg-speech file containing the hidden encrypted image
The steps are as follows:

a) Read the speech signal file. (Note: the speech signal file must be at least twice the size of the hidden image bits, to ensure that there is enough room to hide.)
b) Divide the signal into blocks (i.e. there are 256 samples of each block size).
c) Use the zero crossing k-means algorithm to obtain a vector (A of 256 values).
d) Convert the matrix values of the gray image from a matrix to a one-dimensional vector, then convert the values of the vector from decimal to binary, then convert the binary vector to a vector containing only zero or one value (one bit value).
e) For each speech block which does not contain the 16th largest zero crossing rate, do the following**:**
   – Use the FrFt transformation to obtain the magnitude and phase values.
   – Convert the magnitude value from decimal to binary with length equal to 64 bits.
   – Extract the bits at the locations (25-32) from the binary vector and save them in a new vector (B) of 8 bits (b1 b2 b3 b4 b5 b6 b7 b8).
   – Read a value from the gray image vector (G) and a value from the vector (A) and then perform the operations as follows:
      • If A=1, do XOR (a5, a7)=R1 as the first stage. Next do XOR (R1, G)=R2 as the second stage. Then use LSB algorithm to replace bit 8 of vector (B) with R2 value (i.e. the new values of vector (B) is b1 b2 b3 b4 b5 b6 b7 R2).
      • If A=2, do XOR (a6, a8)=R1 as the first stage. Next do XOR (R1, G)=R2 as the second stage. Then use the LSB algorithm to replace bit 8 of vector (B) with R2 (i.e. the new values of vector (B) are b1 b2 b3 b4 b5 b6 b7 R2).
   – Combine the new values of vector (B) in locations (25-32) of the magnitude value.
   – Convert the value of magnitude from binary to decimal.
   – Perform an inverse FrFt transform to construct the original speech signal based on the same phase value and the new magnitude value.
f) Repeat step 6 until the end of image encrypt vector binary values.
g) Combine the resulting blocks into one speech signal file and save it in an extension (".wav") file.

4. RESULTS AND DISCUSSION

4.1. Setting

The experiments in this paper used both the internet and dataset of 40 standard color images taken from the database "USC-SIPI Image Database: Miscellaneous" that includes various edgy and smooth color images in different dimensions. It used this after converting it to gray image to reduce the time because color images need to deal with three matrices that make up the color image. It has used speech files of the periods 3, 7, 11, 16, and 25 ms of the public dataset published by Lin et al. [30] containing more than 100 h of speech conversation in different languages and from different genders. Blocks are stored in ".wav" format. A variety of experiments were carried out on a PC with Pentium Intel (R) Corei7, CPU@2.60, 6.00 GB RAM, 64 bit Windows 10 to estimate the implementation time of the proposed new encryption method; and all of the above algorithms were coded using MATLAB R2018a software.

We assessed the effectiveness of the proposed solution to the formulated problem with the following measures to test the quality: SNR "Signal to Noise Ratio": used to quantify the distortion of a signal due to noise. PSNR "Peak Signal to Noise Ratio": This metric uses as a quality measurement between the original and stego-medium. MSE "Mean Square Error": It used this metric to measure the distortion in the cover after it had hidden the data within it. NPCR "Number of Pixels Change Rate": This is a measure to see the influence on the encrypted image by changing a single pixel in the original image (i.e. It determined the percentage of different pixel numbers between the two images). UACI "Unified Average Changing Intensity": This measures the standardized mean rate of the difference between the plain and encrypted images. The following Equations compute SNR, PSNR, MSE, NSCR and UACI [5, 14, 15, 19, 31-33]:

$$SNR = 10\ log_{10}\left(\frac{\sum_{i=1}^{n} o(i)^2}{\sum_{i=1}^{n}\left(o(i)-s(i)\right)^2}\right) \quad (13)$$

$$PSNR = 10log_{10}\left[\frac{\max(o(i),s(i))^2}{abs(o(i)-s(i))^2}\right] \quad (14)$$

$$MSE = 10log_{10}\sum_{i=1}^{m}\sum_{i=1}^{n}\frac{\left(o(i)-s(i)\right)^2}{M*N} \quad (15)$$

$$NPCR = \frac{1}{N \times M} \left(\sum_{i,j} D(i,j) \right) \times 100\% \quad (16)$$

$$UACI = \frac{1}{N \times M} \left(\sum_{i,j} \left| \frac{c_1(i,j) - c_2(i,j)}{255} \right| \right) \times 100\% \quad (17)$$

$$D(i,j) = \begin{cases} 0 & c_1(i,j) = c_2(i,j) \\ 1 & c_1(i,j) \neq c_2(i,j) \end{cases} \quad (18)$$

Where,
n,m =the numbers of rows and columns in speech signal or image
o =the block with index number in the original speech signal, s=the block with index number in the stego speech signal
C1 =an encrypted image without any change in the original image, C2=the encrypted image after changing one pixel in the original image.

4.2. Experiments and results

The experimental study demonstrates the efficiency of the suggested approaches because the results showed no difference in the listening tests between the original speech signal and the stego-speech signal after hiding an image in it. To measure and evaluate the quality of steganography approaches, it used many metrics to measure the overall performance of the proposed steganography techniques. The statistical metrics that it used are SNR, PSNR and MSE. The results show that even after embedding the secret message, the size of the speech file remains the same.

Table 1 demonstrates that the PSNR values are small, while there is a considerable difference in MSE between the original image and encrypted gray image. The NPCR and UACI closed to 100% and 33.3% respectively. Moreover, from the results can noted the effect of the image dimensions used, where the greater the dimension, the better the values. In Table 2, Table 3, and Table 4 we can see different speech signal files lengths were used to demonstrate its effect to hid on the images sizes (256×256, 128×128, and 64×64) respectively. As the longer the time period, the better the results. The SNR and PSNR values are large, but they decrease as the hidden file size increases, indicating that poor noise in higher layers does not affect the changed bits, while the MSE is very small.

Figure 2 and Figure 3 shows all the operations performed on the image with dimensional (256×256) and (64×64) respectively to convert the image into a gray image encoded through three scrambling operations for the purpose of scattering the values of the original image and separating the correlation between the values of the pixels in the image, and this process was essential because the encryption process was based on generating the keys and making fractions between those keys and the image values after the scrambling. While Figure 4 and Figure 5 shown the original speech signal before and after embedding the gray image (256×256) and (64×64) respectively with length (30ms). The algorithm is completely reversible due we used LSB to change only one bit at each location in the speech file. In principle, the hidden information retrieved is exactly same as the original covert information.

It is not possible to compare hiding the images with the sound, because the sizes of the images and the sound differ, as well as the research of hiding the images with the sound is little, so we made a comparison of the image coding stage, and these results indicate that our proposed method is better as we note in the Table 5.

Table 1. Gray image encoding results in different sizes

Gray Image	Dimension	PSNR	MSE	NPCR	UACI
Img1	256×256	9.8635	671.009	0.9963	0.32700
Img2	256×256	9.2372	775.088	0.9959	0.33097
Img3	256×256	8.9605	826.081	0.9956	0.32939
Img4	256×256	8.3892	942.237	0.9956	0.33114
Img5	256×256	9.1660	787.902	0.9961	0.32878
Img1	128×128	9.8252	676.950	0.9953	0.32705
Img2	128×128	9.1605	788.914	0.9949	0.32887
Img3	128×128	9.0089	816.933	0.9948	0.32913
Img4	128×128	8.3457	951.718	0.9962	0.33132
Img5	128×128	9.2021	781.385	0.9961	0.32867
Img1	64×64	9.8603	671.493	0.9956	0.32682
Img2	64×64	9.1708	787.043	0.9973	0.32884
Img3	64×64	8.9443	829.173	0.9963	0.32949
Img4	64×64	8.2884	964.340	0.9970	0.33170
Img5	64×64	9.0847	802.791	0.9970	0.32922

Table 2. Hide the different gray image size (256×256) in different speech signal lengths

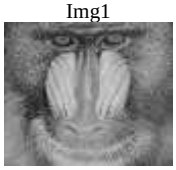
Cover	Length	Gray Image	Dimension	SNR	PSNR	MSE
Speech1	30ms	Img1 	256×256	109.6296	127.9767	5.1379E-14
Speech2	40ms		256×256	112.2630	130.6101	2.8018E-14
Speech3	50ms		256×256	113.7505	132.0976	1.9893E-14
Speech4	60ms		256×256	115.1321	133.4792	1.4472E-14
Speech1	30ms	Img2 	256×256	109.6296	127.9767	5.1379E-14
Speech2	40ms		256×256	112.2629	130.6100	2.8019E-14
Speech3	50ms		256×256	113.7510	132.0981	1.9890E-14
Speech4	60ms		256×256	115.1325	133.4796	1.4471E-14
Speech1	30ms	Img3 	256×256	109.6289	127.9760	5.1387E-14
Speech2	40ms		256×256	112.2629	130.6099	2.8019E-14
Speech3	50ms		256×256	113.7500	132.0971	1.9895E-14
Speech4	60ms		256×256	115.1316	133.4787	1.4474E-14

Table 3. Hide the different gray image size (128×128) in different speech signal lengths

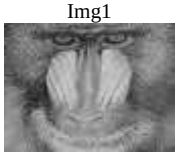
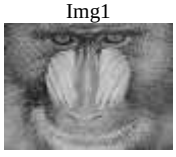
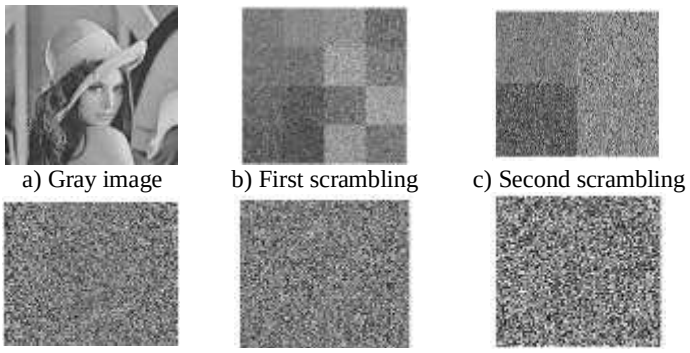
Cover	Length	Gray Image	Dimension	SNR	PSNR	MSE
Speech1	7ms	Img1 	128×128	103.6463	121.4892	1.5015E-13
Speech2	10ms		128×128	106.3241	124.6712	1.0999E-13
Speech3	20ms		128×128	107.1582	125.5053	9.0765E-14
Speech4	30ms		128×128	109.6879	128.0350	5.0694E-14
Speech1	7ms	Img2 	128×128	103.6462	121.4891	1.5015E-13
Speech2	10ms		128×128	106.3241	124.6712	1.0999E-13
Speech3	20ms		128×128	107.1581	125.5052	9.0767E-14
Speech4	30ms		128×128	109.6881	128.0352	5.0691E-14
Speech1	7ms	Img3 	128×128	103.6459	121.4889	1.5016E-13
Speech2	10ms		128×128	106.3240	124.6711	1.0999E-13
Speech3	20ms		128×128	107.1580	125.5051	9.0771E-14
Speech4	30ms		128×128	109.6879	128.0350	5.0693E-14

Table 4. Hide the different gray image size (64×64) in different speech signal lengths

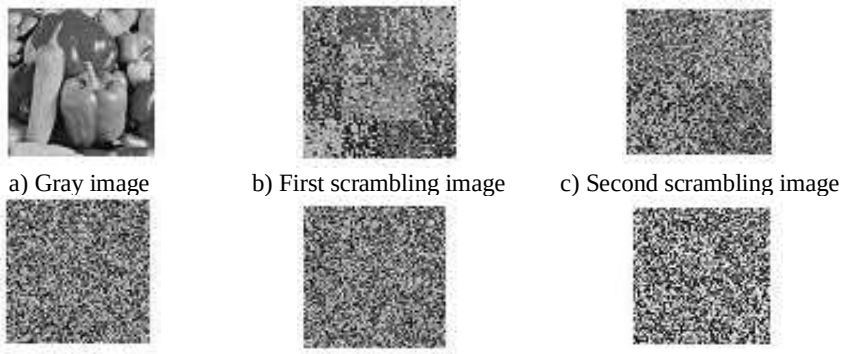
Cover	Length	Gray Image	Dimension	SNR	PSNR	MSE
Speech1	2ms	Img1 	64×64	109.5952	124.5720	3.9420E-14
Speech2	4ms		64×64	113.9923	131.7203	2.0899E-14
Speech3	7ms		64×64	103.6591	121.5020	1.4971E-13
Speech4	10ms		64×64	106.3411	124.6882	1.0955E-13
Speech1	2ms	Img2 	64×64	109.5948	124.5716	3.9423E-14
Speech2	4ms		64×64	113.9898	131.7177	2.0911E-14
Speech3	7ms		64×64	103.6589	121.5019	1.4971E-13
Speech4	10ms		64×64	106.3409	124.6880	1.0956E-13
Speech1	2ms	Img3 	64×64	109.5939	124.5707	3.9431E-14
Speech2	4ms		64×64	113.9959	131.7239	2.0881E-14
Speech3	7ms		64×64	103.6591	121.5020	1.4971E-13
Speech4	10ms		64×64	106.3411	124.6882	1.0956E-13



a) Gray image b) First scrambling c) Second scrambling

d) Third scrambling e) Rotated scrambling f) Encrypted scrambling

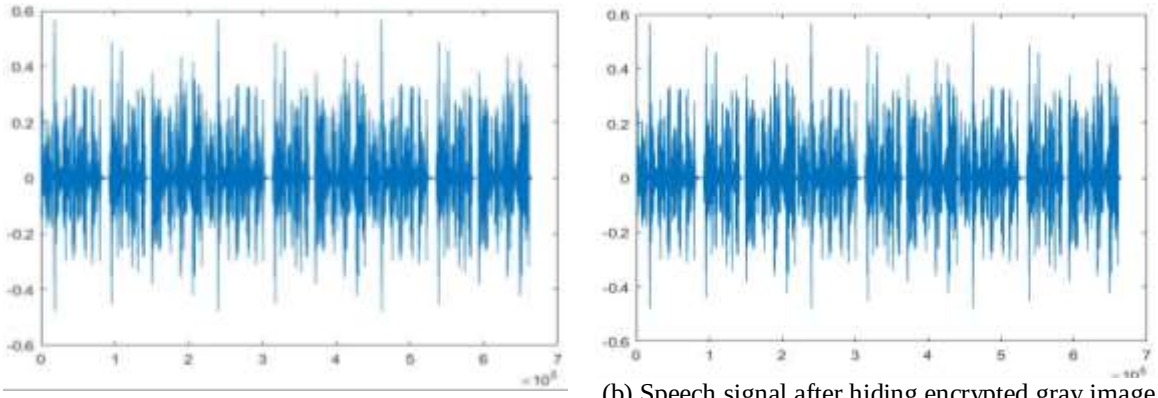
Figure 2. Shows all the operations (a-f) performed on the image (256×256 pixels)



a) Gray image b) First scrambling image c) Second scrambling image

d) Third scrambling image e) Rotated scrambling image f) Encrypted scrambling image

Figure 3. Shows all the operations (a-f) performed on the image (64×64 pixels)

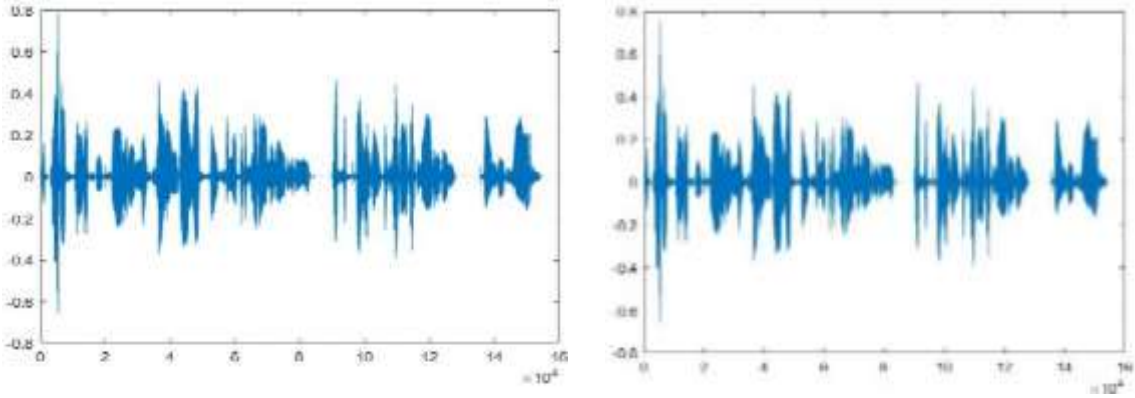


(a) Input speech signal (b) Speech signal after hiding encrypted gray image

Figure 4. Speech signal (30 ms) before and after hiding encrypted gray image (256×256 pixels)

Table 5. The comparison of NPCR & UACI measures in encryption image algorithms

Algorithm	NPCR	UACI
Proposed algorithm	0.9963	0.3270
[15]	0.9963	0.3051
[34]	0.9962	0.3359
[35]	0.9962	0.3354
[36]	0.9961	0.3328



(a) Input speech signal (b) Speech signal hiding gray encrypted image

Figure 5. Speech signal (4 ms) before and after after hiding encrypted gray image (64×64)

5. CONCLUSION

Data hiding is an important technique that embeds secret information into digital media and thereby ensures secure transfer. In our paper, it proposes an algorithm which hides the gray image in the speech signal file in different dimensions for both, where the proposed algorithm provides the advantages of embedding capacity and increases robustness against hackers. It used the proposed techniques in three phases: scrambling a gray image depending on quantum chaotic map, encrypting the gray image based on a combined Zaslavsky and 3D Hanon map, and finally, hiding the encrypted gray image using the LSB algorithm and the zero crossing k-means algorithm. Therefore, the embedding position is completely unknown to anyone who wants to hack the secret message and does not change the file size even after embedding. Hence, even if a person found the hidden message and attempted to decipher it, they could only get as far as the encrypted message without any way of being able to decrypt it and knowing the secret key. The suggested algorithm is new because of the use of sound characteristic (zero-crossing) to determine the locations within which it wants to be hidden.

Compared to similar researches, we obtained satisfactory and good results. We used MSE, PSNR, NPCR and UACI to measure the effectiveness of the encryption algorithm where MSE values were large, PSNR values were small, and the NPCR and UACI closed to 100% and 33.3% respectively. while The SNR and PSNR values were large and MSE was very small which measure the effectiveness of the hiding algorithm. Future research will concentrate on further enhancing the hiding methods using the same algorithm on color and medical images with additions commensurate with their difference from gray images and also dealing with different-dimensional images.

REFERENCES

[1] F. Q. A. Alyousuf, R. Din, and A. J. Qasim, "Analysis review on spatial and transform domain technique in digital steganography," *Bull. Electr. Eng. Informatics*, vol. 9, no. 2, pp. 573-581, 2020.
[2] S. P. Rajput, K. P. Adhiya, and G. K. Patnaik, "An Efficient Audio Steganography Technique to hide Text in Audio," 2017 Int. Conf. Comput. Commun. Control Autom., pp. 1-6, 2017.
[3] Q. M. Hussein, "New Metrics for Steganography Algorithm Quality," *International Journal of Advanced Science and Technology*, vol. 29, no. 02, pp. 2092-2098, 2020.
[4] D. Tan, Y. Lu, X. Yan, and X. Wang, "A simple review of audio steganography," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC*, pp. 1409-1413, 2019. ok
[5] M. Parthasarathi, T. Shreekala, T. Nadu, and T. Nadu, "Secured Data Hiding in Audio Files Using Audio Steganography Algorithm," *Int. J. Pure Appl. Math.*, vol. 114, no. 7, pp. 743-753, 2017.
[6] M. Liao, X. Dong, J. Chen, and D. Zeng, "An audio steganography based on Twi-DWT and audio-extremum features," *Chinese Control Conf. CCC.*, vol. 2019-July, pp. 8882-8888, 2019. doi: 10.23919/ChiCC.2019.8866035.
[7] R. Durgarao and T. Adithya, "Hiding Image in Audio Steganography Using Discrete Cosine Transform and Skin Detection," *IJRASET*, vol. 3, no. 1, pp. 216-221, 2015.
[8] V. B. Bhagat, P. N. Kulurkar, "Audio And Video Steganography: Using Lsb And Phase Encoding Algorithm," *IJPRET*, vol. 3, no. 9, pp. 1640-1648, 2015.
[9] C. T. Jian, C. C. Wen, N. H. Binti Ab Rahman, and I. R. B. A. Hamid, "Audio Steganography with Embedded Text," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 226, no. 1, pp. 1-10, 2017. doi:10.1088/1757-899X/226/1/012084.
[10] H. Chowdary N, K. Karan, K. P. Bharath and R. Kumar M, "Data Hiding in Speech Signal Using Steganography and Encryption," *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1219-1223, 2018. doi: 10.1109/RTEICT42901.2018.9012508.

[11] M. H. A. Al-Hooti, T. Ahmad, and S. Djanali, "Audio Data Hiding Using Octal Modulus Function Based Unsigned Integer Sample Values," *2018 Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2018*, pp. 48-53, 2018.
[12] S. Bandi and H. S. Manjunatha Reddy, "Combined audio steganography and AES encryption to hide the text and image into audio using DCT," *Int. J. Recent Technol. Eng.*, vol. 8, no. 3, pp. 1732-1738, 2019.
[13] S. Bahuguna, "Audio Steganography Technique Using Circular Bit Replacement," *International Journal of Computer Engineering & Technology (IJCET)*, vol. 10, no. 4, pp. 17-24, 2019.
[14] F. J. Farsana and K. Gopakumar, "Speech Encryption Algorithm Based on Nonorthogonal Quantum State with Hyperchaotic Keystreams," *Adv. Math. Phys.*, vol. 2020, pp. 1-12, 2020, doi: 10.1155/2020/8050934.
[15] H. M. Al-Mashhadi and I. Q. Abduljaleel, "Color image encryption using chaotic maps, triangular scrambling, with DNA sequences," *Int. Conf. Curr. Res. Comput. Sci. Inf. Technol. ICCIT 2017*, pp. 93-98, 2017.
[16] A. H. Khaleel and I. Q. Abduljaleel, "A novel technique for speech encryption based on k-means clustering and quantum chaotic map," *Bull. Electr. Eng. Informatics*, vol. 10, no. 1, pp. 160-170, 2021. DOI: 10.11591/eei.v10i1.2405.
[17] J. C. do Nascimento, R. L. C. Damasceno, G. L. de Oliveira, and R. V. Ramos, "Quantum-chaotic key distribution in optical networks: from secrecy to implementation with logistic map," *Quantum Inf. Process.*, vol. 17, 2018. https://doi.org/10.1007/s11128-018-2097-1.
[18] F. J. Farsana and K. Gopakumar, "A Novel Approach for Speech Encryption: Zaslavsky Map as Pseudo Random Number Generator," *Procedia Comput. Sci.*, vol. 93, pp. 816-823, 2016.
[19] D. Riadh and R. Shaker, "Implementation of Gray Image Encryption using Multi-Level of Permutation and Substitution," *Int. J. Appl. Inf. Syst.*, vol. 10, no. 1, pp. 25-30, 2015, doi: 10.5120/ijais2015451458.
[20] F. J. Farsana, V. R. Devi, and K. Gopakumar, "An audio encryption scheme based on Fast Walsh Hadamard Transform and mixed chaotic keystreams," *Appl. Comput. Informatics*, pp. 1-11, 2019. https://doi.org/10.1016/j.aci.2019.10.001.
[21] S. Zhou, F. Xu, P. Ping, Z. Xie, and X. Lyu, "Non-square colour image scrambling based on two-dimensional sine-logistic and hénon map," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 12, pp. 5963-5980, 2017. DOI: 10.3837/tiis.2017.12.015.
[22] A. Prof., "Proposed Hyperchaotic System for Image Encryption," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 37-40, 2016.
[23] D. Paul, "Automated Speech Recognition of Isolated Words Using Neural Networks," *Int. J. Eng. Sci. Technol.*, vol. 3, no. 6, pp. 4993-5000, 2011.
[24] K. Zvarevashe and O. Olugbara, "Speech Emotion Recognition," 2020.
[25] Q. Wang, X. Ouyang, and J. Zhan, "A classification algorithm based on data clustering and data reduction for intrusion detection system over big data," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 7, pp. 3714-3732, 2019. DOI: 10.3837/tiis.2019.07.021.
[26] J. J. Kim, M. Ryu, and S. H. Cha, "Approximate k values using Repulsive Force without Domain Knowledge in k-means," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 3, pp. 976-990, 2020. DOI: 10.3837/tiis.2020.03.004.
[27] Z. Wu, R. Li, and C. Li, "Adaptive speech information hiding method based on K-means," *IEEE Access*, vol. 8, pp. 23308-23316, 2020. doi: 10.1109/ACCESS.2020.2970194.
[28] K. Saroha and P. K. Singh, "A Variant of LSB Steganography for Hiding Images in Audio," *Int. J. Comput. Appl.*, vol. 11, no. 6, pp. 12-16, 2010.
[29] B. T. Krishna, "Fractional Fourier transform: A survey," *ACM Int. Conf. Proceeding Ser.*, 2012, pp. 751-757, 2012.
[30] Z. Lin, Y. Huang, and J. Wang, "RNN-SM: Fast Steganalysis of VoIP Streams Using Recurrent Neural Network," in *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1854-1868, July 2018. doi: 10.1109/TIFS.2018.2806741.
[31] R. Ismail Abdelfattah, H. Mohamed, and M. E. Nasr, "Secure Image Encryption Scheme Based on DNA and New Multi Chaotic Map," *J. Phys. Conf. Ser.*, vol. 1447, no. 1, 2020. doi:10.1088/1742-6596/1447/1/012053.
[32] R. Tanwar, K. Singh, M. Zamani, A. Verma, and P. Kumar, "An Optimized Approach for Secure Data Transmission Using Spread Spectrum Audio Steganography, Chaos Theory, and Social Impact Theory Optimizer," *J. Comput. Networks Commun.*, vol. 2019, pp. 1-10, 2019. https://doi.org/10.1155/2019/5124364.
[33] I. Q. Abduljaleel and A. H. Khaleel, "Hiding text in speech signal using K-means, LSB techniques and chaotic maps," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, pp. 5726-5735, 2020, doi: 10.11591/ijece.v10i6.pp5726-5735.
[34] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damaševičius, and T. Blažauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using Enhanced Logistic-Tent Map," *Entropy*, vol. 21, no. 7, pp. 656, 2019. doi: 10.3390/e21070656.
[35] C. Xu, J. Sun, and C. Wang, "A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems," *Multimed. Tools Appl.*, vol. 79, no. 9-10, pp. 5573-5593, 2020. DOI: https://doi.org/10.1007/s11042-019-08273-x.
[36] Y. Kang, L. Huang, Y. He, X. Xiong, S. Cai, and H. Zhang, "On a symmetric image encryption algorithm based on the peculiarity of plaintext DNA coding," *Symmetry (Basel).*, vol. 12, no. 9, pp. 3-18, 2020. https://doi.org/10.3390/sym12091393.