International Journal of Electrical and Computer Engineering (IJECE)
Vol. 10, No. 6, December 2020, pp. 5726~5735
ISSN: 2088-8708, DOI: 10.11591/ijece.v10i6.pp5726-5735

Hiding text in speech signal using K-means, LSB techniques and chaotic maps

Iman Qays Abduljaleel, Amal Hameed Khaleel
Department of Computer Science, Basrah University, Iraq

Article Info

Article history:

Received Mar 20, 2020
Revised May 8, 2020
Accepted May 22, 2020

Keywords:

Choatic maps
K-Means clusters
LSB technique
Steganography
Zaslavsky map

ABSTRACT

In this paper, a new technique that hides a secret text inside a speech signal without any apparent noise is presented. The technique for encoding the secret text is through first scrambling the text using Chaotic Map, then encoding the scraped text using the Zaslavsky map, and finally hiding the text by breaking the speech signal into blocks and using only half of each block with the LSB, K-means algorithms. The measures (SNR, PSNR, Correlation, SSIM, and MSE) are used on various speech files (".WAV"), and various secret texts. We observed that the suggested technique offers high security (SNR, PSNR, Correlation, and SSIM) of an encrypted text with low error (MSE). This indicates that the noise level in the speech signal is very low and the speech purity is high, so the suggested method is effective for embedding encrypted text into speech files.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

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

1. INTRODUCTION

In the modern digital communication age, information privacy is a huge concern. There are two essential areas for ensuring information security: encryption and hiding of information. As security needs increase, encryption alone is not sufficient, so hiding is also supplementary to encryption in security property [1]. Encryption is a system which transforms a message from a readable form into an unreadable form using a certain key, and the secret text can only be effectively isolated by a recipient who has the secret key [2, 3].

Hiding information is a method that hides sensitive information inside the media of an information carrier in a way that only senders and recipients who know about it can unlock. Steganography is one of these hidden data transformation techniques that is popularly employed [4]. Steganography is governed by four key elements: secret data, carrier file, type of carrier file, capacity carrier file [5]. In audio steganography the human auditory system's weakness is used to hide data in the audio, because the human ear cannot discriminate the small difference between the original file and the altered one [6]. Speech is a special case of audio signals which is distinguished in terms of spectral bandwidth, signal continuity, and intensity distribution [7]. Speech steganography is a particular challenge. In this paper, a speech file is selected as a carrier medium, and a text message is hidden within it; the secret text is encoded through the use of text scrambling using a Chaotic Map, and the scraped text is encoded using the Zaslavsky map, then the encoded text is hidden inside a speech signal using the LSB, K-means algorithms. The proposed algorithm has the advantages of increased capacity and protection. The proposed embedding technique did not change the original file size, so with the proposed method the data are difficult to extract.

The LSB encoding method means the Least Significant Bits of the audio sample are replaced with the hidden message bits, as the original audio has no effect [8]. The choice of pixels or the order of embedding capacity may be determined by a stego-key [9]. K-means clustering, which is a data mining

technique that divides or groups n items into groups of k, where k is chosen according to the number of groups needed. In the case of K-means, centroids represent the mean vectors [10]. The Zaslavsky Map is a nonlinear dynamic method of discrete-timentroduced by George M Zaslavsky. It exhibits deterministic dynamic behaviour, which is an integral part of the algorithms for contemporary data encryption [11]. A quantum logistic map is suggested by Goggin et al., [12]. Quantum chaos is based on the quantum system displaying chaotic dynamics within a given range. The very lowest-quantum corrections cause additive noise and quantum chaos, so the advantage of chaotic encryption is high-level security [13]. Several previous works were designed to perform data hiding including the Echo hiding method [14], Phase and amplitude coding methods [15].

In [16], the proposed method for LSB encoding in multiple layers is by using the bitwise operation, meaning two message bits are embedded at a time into audio cover to increase the robustness and capacity. In [17], the proposed technique enhances protection, where each hidden bit of information is placed in the selected audio media cover location. The embedding location is selected based on upper 3MSB bits of cover media. [18] proposes the enhanced technique of LSB with human detection as a key generator to extract a secret message from the video cover file and describe the location of the message. In [19], a new technique hides a secret text within a cover audio file using modulus function-based uint 8. In [20], the proposed IAMM scheme exploits the DWT properties to achieve efficient speech watermarking of the blind. Second level approximation and detail coefficients with appropriate intensity were chosen for the embedding of information bits, and synchronisation codes for the watermarking process, respectively. Bandi and Reddy [21] suggests encrypting the secret text message and concealing it in the cover audio, using a combination of DCT coefficient and AES encryption method to provide data protection. Chowdary [22] uses a different approach for data hiding in speech signals, with a unique key which contains a ten-digit number within a speech signal for better security. Ahmed [23] suggests an algorithm using pixel value gauge technology to mask the confidential message in the cover picture's most significant bits (MSBs).

In [24] the proposed encrypted data and address information is used to locate the next pixel, using the Fisher-Yates Shuffle algorithm. In [25], the proposed technique embeds English text into the wave audio using tone insertion method, which generates two frequency f1 and f2 and inserts them into an audio file in a suitable power level according to the specific stego-table. In [26] steganography of image is suggested using LSB and secret map techniques. The study is based on the principle of random insertion and the select of a pixel from a host image. Applying 3D chaotic charts, Chebyshev, and 3D logistic maps performs the technique.

The remainder of this paper is organised as follows: section 2 gives details of the general algorithms for the suggested method. Section 3 shows the experimental results and simulation. Conclusions and future work are presented in Section 4.

2. RESEARCH METHOD

The proposed algorithm of three stages:

- Scrambling text algorithm: it depends on the repetition of splitting a text consisting of bits into blocks, and using the quantum chaotic map on the resulting blocks. We repeat this process until the length of the block reaches 8 bits. The benefit of extending the circuit of the scrambling from small blocks to larger blocks is to scramble the text sites more than once
- Encryption algorithm: it uses the Zaslavsky Map, which includes converting the ciphertext from fractional values to integer values and then converting these values into bits. After that it applies the XOR operation between the keys generated by using the Zaslavsky algorithm and the ciphertext bits.
- Hiding algorithm: it hides the bits of encoded text in the speech signal. We divided the sound into blocks and we used only one part of each block, which was divided into two parts, then we used the K-means algorithm to produce an index key that represents the locations through which we hid the bits of text in the speech signal. The LSB algorithm was also used to choose the sites of speech that it hides. Thus, the proposed algorithm is more sophisticated and secure, and in this way, any hackers of the speech signal will not be able to recognise the hidden text. The proposed algorithm is shown in Figure 1.

2.1. Proposed chaotic scrambling bits algorithm

This algorithm increases resistance against attacks by supplying extra security with the help of a private key. The idea of our algorithm to generate the scrambling bits consists of the following steps:

- Read the text file.
- Convert the text file symbols from symbolic to ASCII.

- Convert the ASCII symbols to binary representation (zero and one) and store them in a vector with a length M, which represents the number of bits of the entered text; e.g., letter A=065 ASCII value=01000001 Binary value.
- Divide the result vector into blocks and each block with a length of n (n=8 initially).
- If Mod (vector of length m/n)=0 then go to the next step. Otherwise, keep the remainder of the vector text (Mod (vector of length m/n)=z) in the last block generation with length (z) without adding zeroes. This step of the algorithm distinguishes our research, since we depend on the original length of the text without additions. Thus, the processing time decreases, so we do not need additional time or a large memory.
- Generate sequences of keys using a chaotic map, where the number of sequences is equal to the number of blocks, and the length of each sequence is equal to (n). Also, if the length of the last block is less than (n), then we create a sequence of keys with the length of the last block.
- For each block with length (n) do the following:
  - Generate a sequence of keys using the chaotic quantum logistic map equation [12] as follows:

$$x_{n+1} = r\,(x_n - |\,x_n\,|)^2 - r y_n, \quad (1)$$

$$y_{n+1} = -y_n\, e^{-2\beta} + e^{-\beta}\, r[(2 - x_n - x_n^*)y_n - x_n\, z_n^* - x_n^*\, z_n], \quad (2)$$

$$z_{n+1} = -z_n e^{-2\beta} + e^{-\beta}\, r[2(1 - x_n^*)z_n - 2x_n y_n - x_n]. \quad (3)$$

    where X_1=0.39, Y_1=0.004, Z_1=0.002, X*=0.0018, Z*= 0.0039, β =4.46, r=3.89, x∈[0,1], x*=x, y ∈[0,0.1], z∈[0,0.2], z*=z, β∈[6,∞), and r ∈[0,4].
  - If the length of the sequence bit of the key is not satisfied (meaning not equal to n), go back to Step (a)
  - If the last block with length (z) is equal to length (n), use the sequence bit of key that results from Step (b), otherwise generate a sequence bit of key equal to length (z) using (1) in Step (7-a)
- Re-arrange all keys in each block in ascending order
- Re-order the bits of each block whose number is (n), according to the order of the values of the bit keys that we have arranged while keeping the original value index
- Combine the blocks in a single vector with a length of (m)
- Calculate n=n * 2
- Test the length of the vector=(m div new n); if greater than (8bits), go back to Step (5).
- Convert each 8bits of vector to ASCII symbols, after that convert it to the characters of the text that are used in the next stage (text encryption)

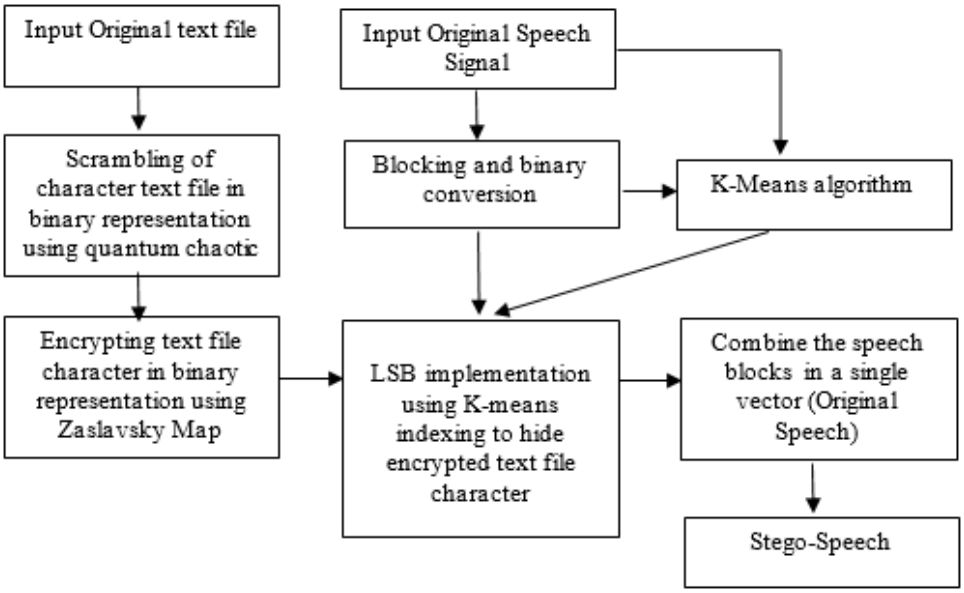


Figure 1. The General Structure of proposed model

2.2. Proposed zaslavsky text encryption algorithm

In this algorithm, for enhanced security the scrambling text is encrypted using random numbers generated using the Zaslavsky map before hiding it inside a speech signal. The idea of our algorithm to encrypt text consists of the following steps:

- Read the scrambling text file
- Convert the scrambling text file symbols from their symbolic form to ASCII symbols
- Convert the ASCII symbols to the binary representation (zero and one) and store them in a vector of length (M), which represents the number of bits of the entered scrambling text.
- Generate Zaslavsky keys with the length of the symbolic bits (m) using equation [11] as the following:

$$x_{n+1} = \text{mod}\ (\ x_n + \text{v}(1 + \mu\, y_n\ ) + \ \varepsilon\, \text{v}\, \mu \cos\ (2\ \pi x_n), 1) \quad (4)$$

$$y_{n+1} = \ e^{-r}\ (y_n + \ \cos(2\pi x_n)) \quad (5)$$

$$\text{and } \mu = \frac{1-e^{-r}}{r} \quad (6)$$

where v, r, ε are control parameters, and e is exponentiation. The key set for the Zaslavsky map is $\{x_0, y_0, v, r, \varepsilon\}$. Commonly used values for the parameters are: x_0=0.12, y_0=0.13, v=0.2, r=5, e=0.3, ε= 9.

- Convert Zaslavsky from fractional values to integer values
- Convert the values to a binary representation at 8bits for each numeric value
- Make an XOR operation between every 8bits of the key value and the text value
- Convert the text from a binary system to ASCII, then to characters, and save it in a text file for the purpose of hiding it

2.3. Proposed embedd K-means algorithm

In our work, to get increased security and accurate results we used the K-means clustering technique, which mainly consists of the following steps:

- Use the K-means algorithm on the matrix to divide it into 128 clusters,
- Use the index of each cluster in the index vector (128 values) to embed the binary text data in each first part of the speech block
- Store the encoded text vector, whose length is (m) in the first 128 bits of the first block of the original speech signal, according to the sequence that we obtained from K-means, to arrange the values from 1-128
- Begin from the second block of the speech signal, and only embed on the first part of each block according to the index of the K-means algorithm

2.4. Proposed advance LSB embedding algorithm

The average bit replacement must not exceed 50% of the bit length into the original speech signal, i.e. bit-length of N will affect only (n/2) bits of the host audio to produce Stegano Speech. The steps for embedding using Least Significant Bit (LSB) algorithm are as follows:

- Read the input speech signal in ".WAV" format and the scrambling text file.
- Divide the original speech signal into blocks (each block contains 256 values)
- Divide each block into two parts (each part has 128 values)
- Take the second part of the first blocks (1-128) of the speech signal, i.e. from each block, we take the values from 129-256, to create a two-dimensional matrix of 128 blocks, where each block has 128 values. Because ".WAV" documents has two parts, the header and the information, and the header is in the initial 44 bytes of the document, the first bits contain little data and the embedding is clear. In contrast, the last bits contain much data and embedding it is not known.
- Use the proposed embed K-means algorithm on the resulting matrix to embed text vector.
- Use a function (float2bin) to convert the values of speech signal files from their fractional form to their binary form using 64 bits, because the values in the speech signal file are characterised by their fractional form.
- Cut the eight bits from allocation 17 to 24 of the 64 bits for each binary number, and then apply to the LSB algorithm that hides the data in the last bit (b1), where it does not correlate with the original values and the form of the bits is as follows: b8 b7 b6 b5 b4 b3 b2 b1 where the bits (b8 b7 b6 b5) are MSB and bits (b4 b3 b2 b1) are LSB
- Test the MSB data by apply two relationships:
  - First relationship:
    If Xor (b_8, b_6)=Xor (b_7, b_5) then change the value of b_1 by doing the following
    Xor (Xor (b_8, b_6), $TextB_i$) the result insert in b_1
    Where $TextB_i$ is the bit for the text vector of length (m) to hide

This method makes it difficult to discover the original decoding values. This is one of the strengths of our algorithm.

- Second relationship:
  If Xor (b8, b6) <> Xor (b7, b_5) then change the value of b_1 by doing the following
  Xor (Xor (b_7, b_5), $TextB_i$) the result insert in b_1

Here there is the same idea, but with no equal Xor values, so it is difficult to retrieve because the sites will differ.

- Re-combine the eight bits cut out within the 64 bits after hiding the text data.
- Convert the 64 bits from their binary into a fractional number to return the values of the input speech signal
- Repeat steps (5-10) for all 128 values in each speech signal block, which is used on all bits of the encoded text vector that we want to hide
- Combine the speech signal blocks in a single vector that represents the resulting speech signal vector after the embedding process.
- Put the length of the text to be hidden into the last block of the speech signal, so that the recipient can know the length of the secret text. It is necessary for the recipient to know the last bit of text that is collected in order to retrieve the text from the speech signal.

3. RESULTS AND DISCUSSIONS

To evaluate the performance of the suggested technique, 21 (".WAV") English speech samples were used. Each audio file is a single-channel 16-bit PCM WAV with a sample rate of 22050 Hz whose duration ranges from four to ten seconds as a cover, and four text messages (.txt) of different lengths. The proposed technique was tested using a PC with the following properties: Pentium Intel (R) Corei7, CPU@2.60 GHz processor, 6.00 GB RAM, 64-bit Windows 10 operating system; and MATLAB R2018b software is the implementing and efficiency evaluation tool. We used (".WAV") audio file type as a cover file for the purpose of hiding data because it includes high data redundancy and its format is not subjected to any type of compression which allows higher data capacity to be hidden, so we used an LSB algorithm which depends on redundancy for hiding data.

The five measures: signal to noise ratio (SNR), Peak Signal to noise ratio (PSNR), correlation, structural similarity index metric (SSIM), and mean square error (MSE), are used. We used the SNR measure because it can show the quantity of noise in the signal for audio medium, PSNR because it shows the robustness of the proposed method, Correlation and SSIM used as quality metrices of strong interdependencies in the speech, and MSE to prove the algorithm reliability of the conveyed data, which must have a high retrieval rate and low error rate. The subsequent equations compute SNR, PSNR, MSE, where:

$$SNR = 10\ log_{10}\ (\frac{\sum_{i=1}^{n} o(i)^2}{\sum_{i=1}^{n}(o(i)-s(i))^2}) \quad (7)$$

$$PSNR = 10log_{10}\ \left[\frac{\max(o(i),s(i))^2}{abs(o(i)-s(i))^2}\right] \quad (8)$$

$$MSE = 10log_{10}\ \sum_{i=1}^{m}\sum_{i=1}^{n}\frac{(o(i)-s(i))^2}{M*N} \quad (9)$$

$$C_{os} = \frac{\sum_{i=1}^{n}(o_i-\mu_o).(s_i-\mu_s)}{[\sum_{i=1}^{n}(o_i-\mu_o)^2]^{1/2}\,[\sum_{i=1}^{n}(s_i-\mu_s)^2]^{1/2}} \quad (10)$$

$$SSIM(O,S) = \frac{(2\mu_o\mu_s+c_1)(2\sigma_{os}+c_2)}{(\mu_0^2+\mu_s^2+c_1)(\sigma_0^2+\sigma_s^2+c_2)} \times 100 \quad (11)$$

where:
n and m are the numbers of rows and columns in cover audio file input signals
o is the sample with index number in the original audio file
s is the sample with index number in the stego audio file
μ_o and μ_s are the meam values of o and s respectively
σ_o and σ_s are the standard deviation values of o and s respectively
$C_1=(k_1L)^2$, and $C_2=(k_2L)^2$ are two constants used to avoid null denominator
$k_1=0.01$ and $k_2=0.03$ by default
L is the dynamic range of the signal values (typically this is $2^{\#\text{ bits per signal}}$ -1).

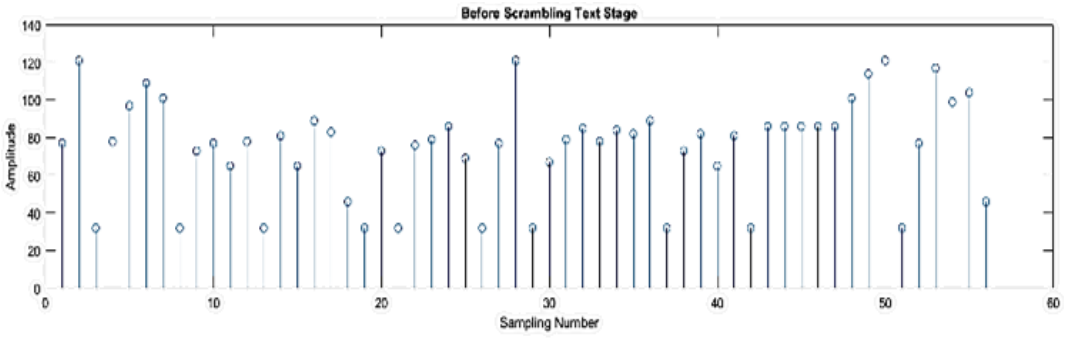
The results of the analysis showed that the longer the text message, the greater the MSE and the smaller the PSNR, and the noise rate is reduced. It can be noted from the outcome of the measures that the square error (MSE) of the proposed method is very small, which means no noise, and no difference between the cover speech and stegano speech. The suggested algorithm consists of three stages: scrambling text, encryption text and hide algorithm.

Example of the proposed algorithm:

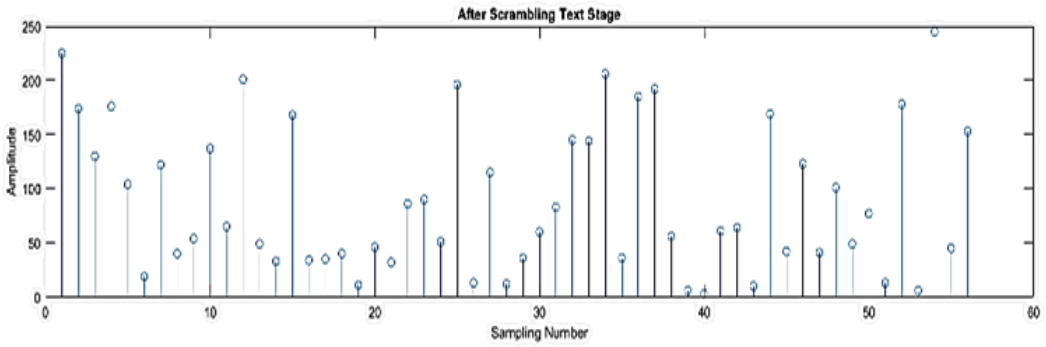
- Scrambling text stage: Figure 2 shown the text input and text after the scrambling algorithm
  Text input: My Name IMAN QAYS. I LOVE My COUNTRY IRAQ VVVVVery Much.
  Scrambling text stage result:
  'á®,°hz(6‰AÉ1!¨"#(. VZ3Ä s$<S‘ • Î$¹À8@©*{)e1M ²õ-™'
- Encryption text stage: We can notice that Figure 3 shows the output of the encryption text
  Input: the output of the scrambling stage
  Output encrypted text stage result: 'a®¯²¡Ñ¦¡€ }ÄêåÖð5xÑ^1HÐ9Óuã·©…R\-€ çÏH • –»a²F®?0›xÏ'
- Hide text stage result: Figure 4 shows the time waveform of the (speech1.wav) signal original before and after embedded (Txt3.txt) file, and the differences between them and it can be seen that the differences are very small using the method proposed.

The outcomes of the performance of the proposed algorithm which hides the text into a speech signal is shown as Tables 1 and 2. Table 1 shows the comparative analysis for Speech1, Speech2, and Speech3 using the existing algorithms in [17,1] and the proposed algorithms. Corresponding graphs are shown in Figures 5 and 6. Table 2 displays the measured result using SNR, PSNR, Correlation, SSIM, and MSE to evaluate the quality of speech before and after the texts were inserted. In Figure 7, the graphs are shown when embedding capacity increases, SNR, PSNR, Correlation, and SSIM decrease while MSE increases.

The different performance results are dependent on different lengths of the text message. Those findings indicate that the lower the MSE value and the higher of SNR, PSNR, Correlation, SSIM values, the better the stego-speech signal output. Whereas the correlation and SSIM values closed to +1, which means a high relationship between the original speech signal and the stego speech signal. Table 1 shows the SNR MES and PSNR values for different speech signal files and different text files after the steganography process.

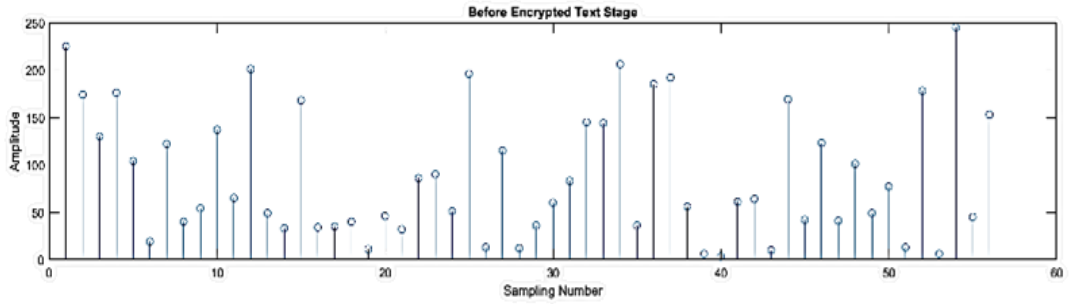


(a)

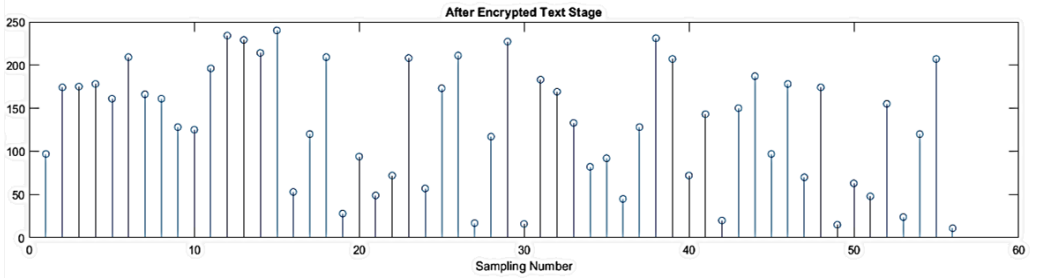


(b)

Figure 2. Scrambling text stage, (a) Before scrambling process, (b) after scrambling process

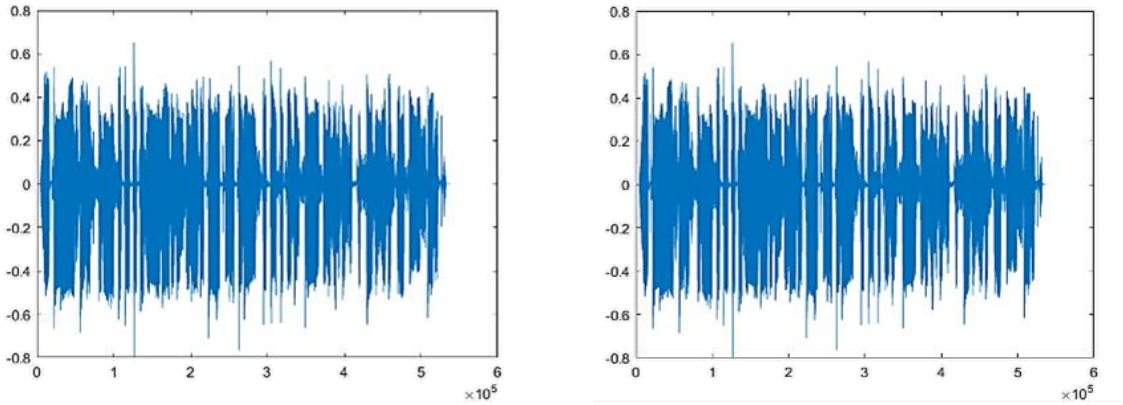


(a)



(b)

Figure 3. Encryption text stage, (a) Before encryption process, (b) after encryption process



(a) (b)

Figure 4. Hide text stage result, (a) Before embedding process, (b) after embedding process

Table 1. SNR MES and PSNR values for different speech signal files and different text files after the steganography process

Speech signal file name	Capacity (Bytes)	Input Text Data	Embedded (Bytes)	Existing Algorithm in [17]		Existing Algorithm in [1]		Proposed Algorithm		
				SNR	PSNR	SNR	PSNR	SNR	PSNR	MSE
Speech1.wav	443926	Txt1.txt	323	84.653	140.745	87.725	144.198	254.4630	142.2155	2.6654e-09
		Txt2.txt	836	83.303	139.776	85.41	142.214	196.4086	113.1883	2.1305e-06
		Txt3.txt	4931	78.782	134.959	79.074	135.584	176.7201	103.3441	2.0554e-05
		Txt4.txt	28959	74.78	131.267	75.054	131.527	162.1293	96.0486	1.1027e-04
Speech2.wav	571258	Txt1.txt	323	87972	144.771	89.914	147.483	257.0360	143.6621	2.4582e-09
		Txt2.txt	836	86.403	144.391	87.93	145.499	197.5288	113.9085	2.3226e-06
		Txt3.txt	4931	80.751	138.37	81.26	138.828	178.7935	104.5409	2.0079e-05
		Txt4.txt	28959	9139	134.569	77.231	134.799	163.6060	96.9471	1.1538e-04
Speech3.wav	900412	Txt1.txt	323	9053	150.934	93.865	153.41	261.4467	145.7098	2.4180e-09
		Txt2.txt	836	84.915	149.89	91.881	151.425	202.9425	116.4577	2.0355e-06
		Txt3.txt	4931	80.99	144.459	85.205	144.149	182.7941	106.3835	2.0706e-05
		Txt4.txt	28959	84.653	140.53	81.166	140.71	167.7983	98.8856	1.1638e-04

Table 2. stego-speech signal performances dependent on the length of text message

Speech signal file Duration	Message length	Performance Testing					
		SNR	PSNR	Correlation	SSIM	MSE	Success rate extraction %
10 ms	5 character	329.1504	187.8378	1	1	3.6415e-14	100%
	15 character	315.7006	181.1129	1	1	1.7131e-13	100%
	30 character	250.7466	148.6359	0.9999	0.9998	3.0302e-10	100%
	50 character	221.1720	133.8485	0.9999	0.989	9.1245e-09	100%
	75 character	213.8950	130.2101	0.9999	0.9888	2.1089e-08	100%
7 ms	5 character	301.4127	175.2949	1	1	4.6035e-13	100%
	15 character	292.2016	170.6893	0.9985	0.9977	1.3294e-12	100%
	30 character	226.0696	137.6233	0.9974	0.9899	2.6931e-09	100%
	50 character	200.6336	124.9054	0.9965	0.9888	5.0355e-08	100%
	75 character	189.1652	119.1711	0.9955	0.9888	1.8857e-07	100%
4 ms	5 character	263.8449	154.7290	1	1	3.1073e-11	100%
	15 character	251.0597	148.3364	0.9985	0.9977	1.3541e-10	100%
	30 character	224.7936	135.2033	0.9981	0.9899	2.7858e-09	100%
	50 character	210.3435	127.9783	0.998	0.9875	1.4705e-08	100%
	75 character	205.2477	125.4303	0.9975	0.9899	2.6439e-08	100%

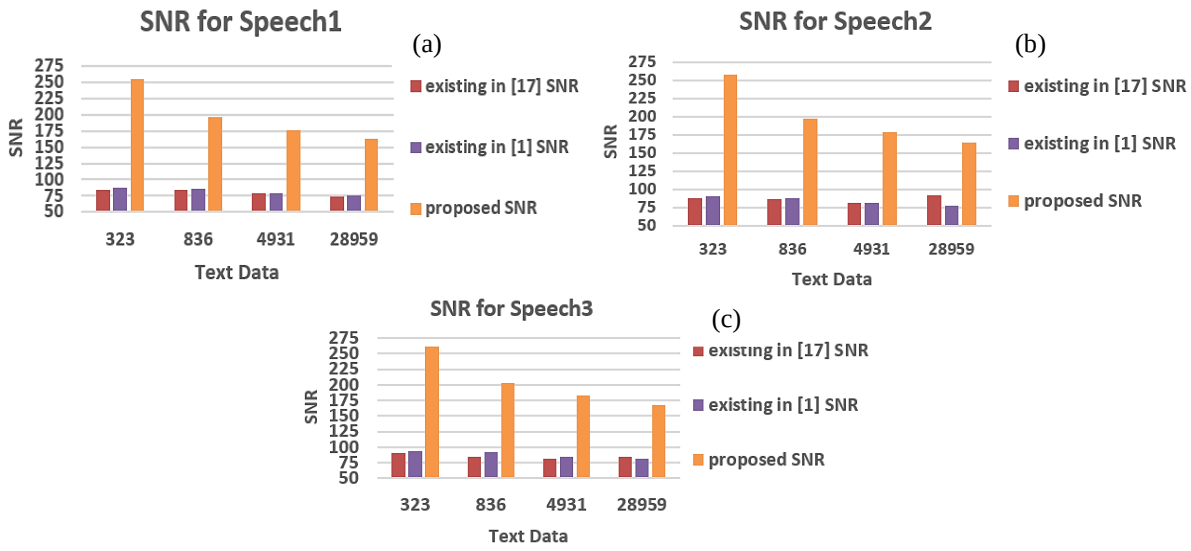


Figure 5. Corresponding graphs, (a) SNR analysis for Speech1, (b) SNR analysis for Speech2, (c) SNR analysis for Speech3

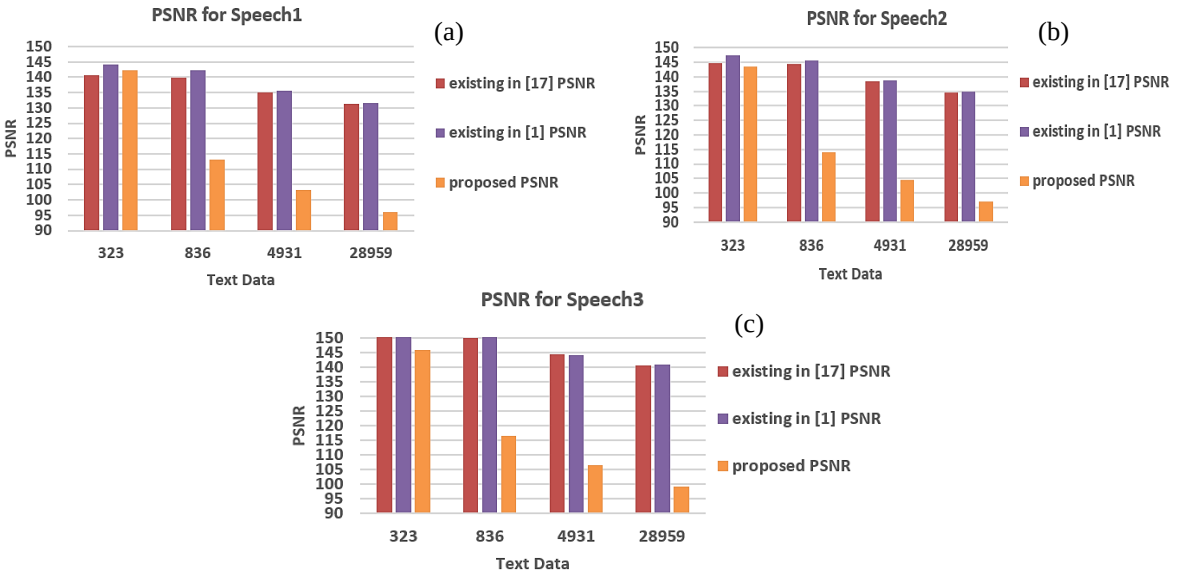


Figure 6. Corresponding graphs, (a) PSNR analysis for Speech1, (b) PSNR analysis for Speech2, (c) PSNR analysis for Speech3

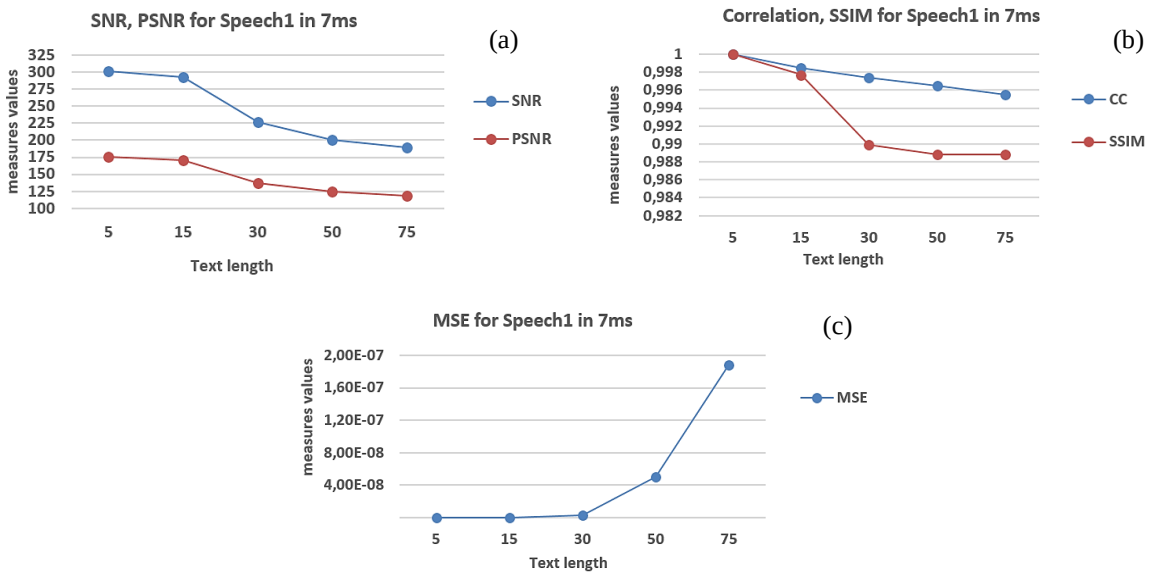


Figure 7. Graphs, (a) Analysis of the relation between the SNR, PSNR and the embedding text capacity, (b) analysis of the relation between the Correlation, SSIM and the embedding text capacity, (c) analysis of the relation between the MSE and the embedding text capacity

4. CONCLUSION

The main objective of our suggested method, which involves embedding textual information in wave audio when transferred over the internet, is to provide efficient and secure encryption by using a more complicated algorithm, hence hard-to-break. In this paper, a new algorithm of hiding text in the speech signal is proposed. The proposed algorithm consists of a scrambling text file using a quantum chaotic map, and an encrypting scramble text file using the Zaslavsky map; then, implementing the LBS algorithm using K-means indexing to hide the encrypted text file into a speech signal. This approach produces good results with a hidden text message that cannot be detected, or at least it cannot be recovered if it is detected. The results indicate that even after embedding the hidden message, the size of the speech file remains the same. This algorithm was applied to the same speech file to embed multiple text files with different sizes of text content and vice versa. Good results were obtained, without losing the text message or noticing any noise in the cover file. In the future, the method could be applied to domains other than speech, such as music and video.

REFERENCES

[1] S. P. Rajput, et al., "An efficient audio steganography technique to hide text in audio," *Int. Conf. Comput. Commun. Control Autom.*, pp. 1-6, 2017.
[2] H. A. Abdullah, and H. N. Abdullah, "FPGA implementation of color image encryption using a new chaotic map," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 129-137, 2019.
[3] I. U. W. Mulyono, et al., "Encryption of Text Message on Audio Steganography Using Combination Vigenere Cipher and LSB (Least Significant Bit)," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 1, pp. 63-74, 2018.
[4] A. B. W. Putra, et al., "Steganography for Data Hiding in Digital Audio Data using Combined Least Significant Bit and 4-Wrap Length Method," *Proc. 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIConCIT 2018*, pp. 336-339, 2018.
[5] Y. Bassil, "Audio Steganography Method for Building the Deep Web," *American Journal of Engineering Research (AJER)*, no. 5, pp. 45-51, 2019.
[6] M. Mustafa, et al., "A Novel Enhanced LSB Algorithm for High Secure Audio Steganography," *10th Comput. Sci. Electron. Eng. Conf. CEEC 2018 - Proc.*, pp. 125-130, 2019.
[7] Y. Xue, et al., "Robust speech steganography using differential SVD," *IEEE Access*, vol. 7, pp. 153724-153733, 2019.
[8] J. Hashim, et al., "LSB Modification based Audio Steganography using Advanced Encryption Standard (AES-256) Technique," *12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, MACS 2018 - Proceedings*, pp. 1-6, 2019.
[9] P. Bhattacherjee, et al., "A review of Steganography techniques suitable for ECG signal," *Conference: International Conference on Emerging Technologies for Sustainable Development (ICETSD '19)*, vol. 6, 2019.
[10] B. Pillai, et al., "Image steganography method using K-means clustering and encryption techniques," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 1206-1211, 2016.

[11] 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.
[12] A. Akhshani, et al., "Pseudo random number generator based on quantum chaotic map," *Commun. Nonlinear Sci. Numer. Simul.,* vol. 19, no. 1, pp. 101-111, 2014.
[13] H. Liu, et al., "A novel quantum image encryption algorithm based on crossover operation and mutation operation," *Multimed. Tools Appl.,* vol. 78, no. 14, pp. 20465-20483, 2019.
[14] wang yunlu, et al., "Steganalysis on positive and negative echo hiding based on skewness and kurtosis," *9th IEEE Conference on Industrial Electronics and Applications,* pp. 1235-1238, 2014.
[15] Kamesh and N. S. Priya, "A survey of cyber crimes Yanping," *Secur. Commun. Networks,* vol. 5, pp. 422-437, 2012.
[16] B. Datta, et al., "Robust multi layer audio steganography," *12th IEEE Int. Conf. Electron. Energy, Environ. Commun. Comput. Control (E3-C3),* pp. 1-6, 2016.
[17] S. Jadhav and A. M. Rawate, "A new audio steganography with enhanced security based on location selection scheme," *Int. J. Performability Eng.,* vol. 12, no. 5, pp. 451-458, 2016.
[18] P. G. P. S. T. Jaya, et al., "Enhanced LSB Steganography with people detection as stego key generator," *Proc. - Int. Conf. Signals Syst. ICSigSys 2017,* pp. 99-104, 2017.
[19] M. H. A. Al-Hooti, et al., "Audio Data Hiding Using Octal Modulus Function Based Unsigned Integer Sample Values," *Int. Conf. Comput. Eng. Netw. Intell. Multimedia, CENIM 2018 - Proceeding,* pp. 48-53, 2018.
[20] H. T. Hu and T. T. Lee, "Frame-synchronized blind speech watermarking via improved adaptive mean modulation and perceptual-based additive modulation in DWT domain," *Digit. Signal Process. A Rev. J.,* vol. 87, pp. 75-85, 2019.
[21] 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.
[22] H. Chowdary, "Data Hiding in Speech Signal Using Steganography and Encryption," *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology,* 2018.
[23] [S. Ahmed, "Data Hiding Using Green Channel as Pixel Value Indicator," *International Journal of Image Processing (IJIP),* no. 12, pp. 90-100, 2018.
[24] M. Cem kasapbaşi and W. Elmasry, "New LSB-based colour image steganography method to enhance the efficiency in payload capacity, security and integrity check," *Sadhana - Acad. Proc. Eng. Sci.,* vol. 43, no. 5, 2018.
[25] S. A. Yousif, et al., "Audio Steganography Using Tone Insertion Technique," *Int. J. Comput. Appl. Technol. Res.,* vol. 6, no. 6, pp. 254-258, 2017.
[26] A. Alabaichi, et al., "Image steganography using least significant bit and secret map techniques," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 1, pp. 935-946, 2020.