Soft Computing (2024) 28:5523–5540
https://doi.org/10.1007/s00500-023-09349-5

APPLICATION OF SOFT COMPUTING

A technique for securing digital audio files based on rotation and XOR operations

Anand B. Joshi[1] · **Abdul Gaffar**[2]

Accepted: 4 October 2023 / Published online: 31 October 2023

Abstract
Security of digital audio files is the need of the hour. In this context, researchers have proposed several techniques for the secure communication of audio files. But unfortunately, these are vulnerable to differential attack. So, we propose a WORD-oriented technique for securing digital audio files based on rotation and XOR operations. The key concepts of the designed encryption algorithm are the RX (Rotation-XOR) operations, i.e., the plain audio samples are first left-rotated by the sum-of-digits of the previous audio samples, and then XOR-ed with the previous audio samples. The designed encryption algorithm encodes a digital audio file into a random (noise-like) audio file. Several encryption and decryption evaluation metrics, such as Adjacent Sample Correlation Coefficient (ASCC), Crest Factor (CF), Number of sample Change Rate (NSCR), Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), etc., are applied on several digital audio files of varying sizes, to empirically assess the performance and efficiency of the proposed technique. The results of these metrics show that the cipher audio files have a very high key sensitivity, ideal ASCC, ideal CF, 100% NSCR score, zero MSE, and infinite PSNR. Moreover, the technique strongly resists the brute-force attack, differential attack, and other statistical attacks.

Keywords Cryptography · Digital audio security · WORD-oriented encryption · Rotation-XOR operations · Sum-of-digits function · Oscillogram

1 Introduction

Every day millions (perhaps billions) of messages in the form of texts, audio, images, and videos, are communicated on the Internet, which is an open (unsecured) network. So, there must be robust technique(s) to communicate secretly. In the context of secure communication, encryption is the best choice, which encodes a secret message into an unrecognizable form, except by the intended one. Broadly, there are two types of encryption schemes: symmetric-key encryption and asymmetric-key encryption. The symmetric-key encryption, also known as (a.k.a.) private-key encryption, uses the same secret key for encoding and decoding a message. The foremost application of private-key encryption is to provide confidentiality. On the other hand, asymmetric-key encryption, a.k.a. public-key encryption, uses different keys for encoding and decoding a message. In particular, the public-key is used for encoding, while the private (secret) key is used for decoding a message. The foremost applications of public-key encryption are authentication and non-repudiation, besides confidentiality.

Since symmetric-key encryption methods are much faster and more efficient for attaining confidentiality as compared to asymmetric-key encryption methods, therefore, we adopt symmetric-key encryption method in the proposed technique. Note that, the Rotation-XOR (RX) operations utilized in the proposed technique are primitive operations, which are efficiently and directly supported by most of the computer processors. These operations aid in the possible improvement of the speed of the designed technique.

The rest of the paper has been put in the following order: Sect. 2 provides related works; Sect. 3 gives preliminaries; Sect. 4 describes the encryption and decryption algorithms of the proposed technique; Sect. 5 describes the implementation and experimental results; Sect. 6 discusses security

✉ Anand B. Joshi
anandiitd.joshi@gmail.com

Abdul Gaffar
abdulgaffar.lu@gmail.com

[1] Department of Mathematics and Astronomy, University of Lucknow, Lucknow, UP 226 007, India

[2] Department of Mathematics and Statistics, Integral University, Lucknow, UP 226 026, India

analyses of the proposed technique; Sect. 7 gives comparison of the proposed technique with the recent state-of-the-art techniques; and Sect. 8 concludes the paper, followed by the references.

2 Related works

Suryadi et al. (2023) proposed a technique for securing digital audio data with the confusion and diffusion schemes based on the modification of the double-scroll function and SHA-256 (Secure Hash Function-256) function. In the first scheme, the confusion process is carried out by scrambling dual channels of plain audio using the keystream of the double-scroll function in the form of the proposed new nonlinear transformation function. The initial value of the double scroll function is obtained through the SHA-256 function. In the next scheme, the diffusion process is carried out by substituting the value of the dual channels based on the nonlinear transformation function, resulting in cipher audio. Although the technique is good, it does not provide the source(s) of the test audio files.

Demirtas (2023) in 2023 presented a lossless and secure audio encryption method based on the chaotic Chebyshev map. Firstly, the input audio samples are preprocessed to obtain the integer and decimal parts. The integer parts are rescaled to the interval [0, 255]. By iterating the Chebyshev map in the chaotic range using plain text dependent variables, the integer parts of the input audio sample are scrambled and then diffused. Finally, a post-processing operation is applied to the diffused audio samples. Although the method is good, it does not provide the source(s), duration, and size of the test audio files. Also, the method is vulnerable to differential attack.

Khalid et al. (2022) proposed a digital audio encryption scheme based on Mordell elliptic curve over a finite prime field. The scheme consists of a confusion-diffusion module. For the confusion module, the scheme initially generates 5×5 bijective S-boxes. The generated S-box is then used parallel in the substitution module, which provides optimum confusion in the cipher data. For the diffusion property, the scheme generates pseudo-random number sequences, to be used for block permutation, which achieves the property of diffusion. Although the scheme is good, it does not provide the source(s) and duration of the test audio files. Also, the scheme is vulnerable to differential attack. Abouelkheir and Sherbiny (2022) proposed a technique for the security of digital audio files based on a modified RSA (Rivest, Shamir, and Adleman) algorithm. The authors modified the RSA algorithm by using dynamic keys—for enhancing the security of the proposed technique, and five numbers (two primes and three random numbers)—for enhancing the speed of the proposed technique. Several metrics have been utilized to validate the aims of the designed scheme. Although the scheme performs well in terms of encryption, but in terms of decryption, it is not a good scheme. It performs lossy decryption, i.e., the decrypted audio files are not identical to the original audio files. Moreover, it does not provide the source(s) of the test audio files. Also, the technique is vulnerable to differential attack.

Shah et al. (2021b) proposed a technique for the secure communication of digital audio files based on finite fields. The authors generated a sequence of pseudo-random numbers via an elliptic curve, which is used to scramble the samples of the plain audio files. Further, the scrambled audio samples are substituted via the newly constructed S-boxes, to ensure the confusion-diffusion properties Shannon (1949) required for a secure encryption algorithm. Although the technique is good, it does not provide the source(s), duration, and size of the test audio files. Also, the technique is vulnerable to differential attack. Faragallah and El-Sayed (2021) proposed an encryption scheme for securing the audio files based on XOR (eXclusive OR) operation and Hartley Transform (HT). First of all, the plain audio file is reshaped into two-dimensional (2D) data blocks, and then it is XOR-ed with a grayscale image (treated as a secret key). The obtained XOR-ed blocks are then transposed via a chaotic map, followed by optical encryption using HT. Although the scheme is good, it does not provide the source(s), duration, and size of the test audio files. Also, the scheme is vulnerable to differential attack. Naskar et al. (2021) suggested an encryption scheme for audio files based on the distinct key blocks together with the Piece-Wise Linear Chaotic Map (PWLCM) and Elementary Cellular Automata (ECA). The scheme encrypts a plain audio file in three stages: cyclic shift, substitution, and scrambling. Cyclic shifting is utilized for reducing the correlation between the samples of each audio block. The shifted audio data blocks are substituted (modified) via the PWLCM, and finally, modified blocks are scrambled via ECA for better diffusion. Although the approach is good, it does not provide the source(s) and duration of the test audio files. Also, the approach is vulnerable to differential attack. Shah et al. (2021a) proposed a method for encrypting digital audio files based on a 3D chaotic map. This map is used for substituting as well as permuting the samples of the audio files. Although the method is good, it does not provide the source(s) and duration of the test audio files. Moreover, the method is vulnerable to differential attack. Stoyanov and Ivanova (2021) designed an algorithm for securing audio files using an Ikeda map (a chaotic map). The map is utilized to generate pseudo-random bytes, which are XOR-ed with the samples of the plain audio files, producing the encrypted audio files. Although the algorithm is good, it does not provide the source(s) of the test audio files. Furthermore, the algorithm is vulnerable to differential attack. Aziz et al. (2021) proposed an audio encryption algorithm based on PSN (Permutation-Substitution Network) (Shannon

1949). The permutation is performed via the application of Mordell elliptic curves, while substitution is performed via a symmetric group on eight symbols, i.e., S_8. The authors also utilized a chaotic map to further enhance the security of the audio files. Although the algorithm is good, it does not provide the source(s), duration, and size of the test audio files. Also, the algorithm is vulnerable to differential attack.

Abdelfatah (2020) proposed an algorithm for securing audio files in three phases utilizing three secret keys. The first phase is the self-adaptive scrambling of the plain audio files via the first secret key. The second phase is the dynamic DNA (Deoxyribonucleic Acid) encoding of the scrambled audio data via the second secret key. The last phase is the cipher feedback mode via the third secret key, which aids in achieving better confusion and diffusion properties. Although the algorithm is good, it does not provide the source(s) of the test audio files. Also, the algorithm is vulnerable to differential attack. Al-kateeb and Mohammed (2020) proposed an audio encryption algorithm based on Discrete Wavelet Transform (DWT) and hand geometry. Hand geometry is utilized for fetching biometric information, to be used in the encryption algorithm. Although the algorithm is good, it does not provide the source(s), and duration of the test audio files. Moreover, the algorithm is vulnerable to differential attack.

Wang and Su (2020) proposed an audio encryption approach using a PWLCM and DNA encoding, to attain the required confusion and diffusion properties. Although the approach is good, it does not provide the source(s) of the test audio files. Also, the approach is vulnerable to differential attack. Kordov (2019) designed a scheme for the security of audio files based on the PSN using a chaotic circle map and modified rotation equations. Although the scheme is good, it does not provide the source(s) of the test audio files. Also, the scheme is vulnerable to differential attack. Shah et al. (2020) suggested an audio encryption scheme based on PSN, wherein permutation is performed via the Henon map (chaotic map), while substitution is performed via the Mobius transformation. Although the scheme is good, it does not provide the source(s) and duration of the test audio files. Moreover, the scheme is vulnerable to differential attack.

Sasikaladevi et al. (2018) proposed an encryption scheme for encrypting audio files based on DWT and elliptic curves encryption. Although the algorithm is good, it does not provide the source(s) of the test audio files. Also, the algorithm is vulnerable to differential attack.

Sathiyamurthi and Ramakrishnan (2017) designed an encryption algorithm for encrypting audio files based on four chaotic maps: logistic map, tent map, quadratic map, and Bernoulli's map. Although the algorithm is good, it does not provide the source(s) and size of the test audio files. Also, the algorithm is vulnerable to differential attack.

Lima and Neto (2016) presented an approach for enciphering digital audio files based on cosine number transform over a finite field. Although the approach is good, it does not provide the source(s) of the test audio files. Also, the approach is vulnerable to differential attack.

Besides these techniques/approaches, several other methods Ghasemzadeh and Esmaeili (2017), Liu et al. (2016), Augustine et al. (2015), Naskar et al. (2019), Belmeguenai et al. (2017), Farsana and Gopakumar (2016), Faragallah (2018), Farsana et al. (2019) and Habib et al. (2017) have also been proposed in the literature.

It is noticeable that on studying existing techniques thoroughly, we conclude that some drawbacks need to be addressed, and can be listed as follows:

1. The existing techniques are vulnerable to differential attack.
2. The authors have not provided the source(s)/reference(s) of the test audio files.
3. Most of the authors have not provided duration(s) and size(s) of the test audio files.
4. Only a few of the authors have included the processing/execution time of their algorithms.

So, the proposed technique is designed to overcome these drawbacks. Moreover, to the knowledge of our best knowledge, this is the first paper on the security of audio files, which is unique/novel in the following ways:

1. The references of all the audio files have been provided.
2. All the necessary details, viz., number of channels, sample rate, total samples, duration, bits per sample, bit rate, and size, of the audio files have been given.
3. The source(s) of each definition/metric used in the paper have been provided.
4. The proposed technique is fully and strongly resistant to the differential attack.

3 Preliminaries

3.1 Digital audio

Digital audio, say, P is a l-by-c matrix, consisting of elements called samples, where l and c denote the number of samples and the number of channels in P, respectively. If $c = 1$, then P is said to be a single (or mono) channel audio file, and if $c = 2$, then P is said to be a dual (or stereo) channel audio file. Note that, the samples in P are floating-point values, i.e., real values. Figure 1 shows the oscillogram (a graph between amplitude and time) and spectrogram (a graph between frequency and time) of the audio file 'handel.wav', which is of size $73{,}113 \times 1$, i.e., a single-channel audio file containing 73,113 samples. For other details of the audio file 'handel.wav', namely, sample rate (in Hz—Hertz),

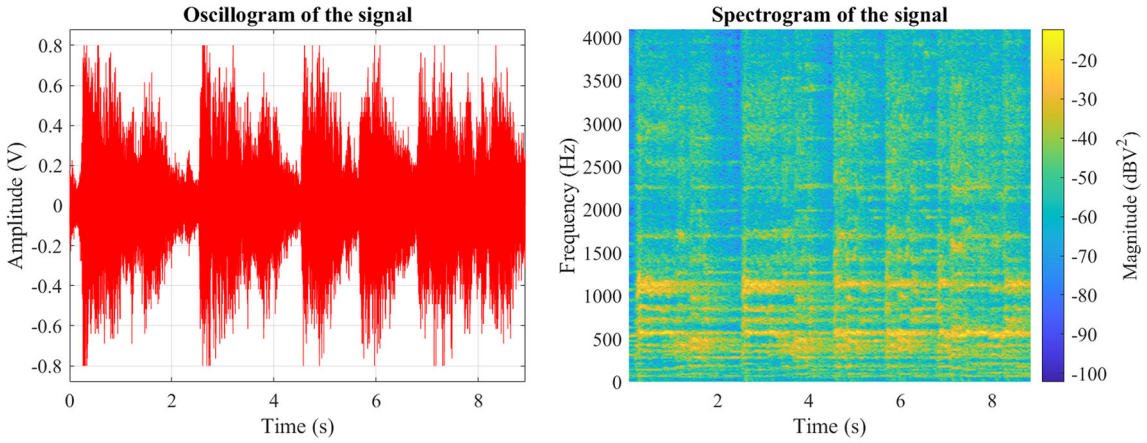


Fig. 1 Oscillogram and spectrogram of the audio file 'handel.wav'

Table 1 Description of the test audio files

File name (.wav)	Sample rate (in Hz)	Total samples (length)	Duration (in s)	Bits/samples	Bit rate (in kbps)	Size (in KB)
Handel	8192	73,113	8.9249	16	131.0720	142.7988
Gong	8192	42,028	5.1304	16	131.0720	82.0859
Zeros	8192	19,120	2.3340	16	131.0720	37.3438
Splat	8192	10,001	1.2208	16	131.0720	25.1562

duration (in s—seconds), bits per sample, bit rate (in kbps—1000 bits per second), and size (in KB—1024 Bytes); see Table 1.

3.2 Rotation operation

By rotation operation, we mean "circular shift" or "bit-wise" rotation. It is of two types:

1. Left rotation: It is denoted by '$\lll$'. By $x \lll y$, it is meant that x is left rotated by y bits. For example, if $x = 0001\ 0111$ and $y = 1$, then $x \lll y$ gives $0010\ 1110$. Figure 2a demonstrates the concept, wherein MSB is the Most Significant Bit and LSB is the Least Significant Bit.
2. Right rotation: It is denoted by '$\ggg$'. By $x \ggg y$, it is meant that x is right rotated by y bits. For example, if $x = 0001\ 0111$ and $y = 1$, then $x \ggg y$ gives $1000\ 1011$. Figure 2b demonstrates the concept.

3.3 XOR operation

It is one of the simplest operations in a computer's processor. It is a bit-wise operation that takes two strings of bits of equal length and performs XOR (denoted by $\oplus$) operation as: if two bits are same, the result is 0; and if not same, the result is 1. It's actually addition modulo 2.

For example, if $a = 1010\ 1011$ and $b = 0101\ 1100$, then $a \oplus b = 1111\ 0111$.

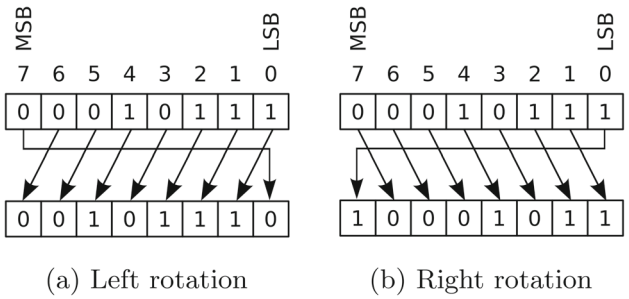


Fig. 2 **a** Left rotation of $x = 0001\ 0111$ by 1-bit and **b** right rotation of x by 1-bit

4 Description of the proposed encryption and decryption algorithms

4.1 Preprocessing on the audio file

Input. An audio file P of size $l \times 1$.

1. Convert the audio samples of P from floating point values (real values) to binary (matrix) via single-precision floating point (32-bit).[1]
2. Convert the binary (matrix) to non-negative integers (bytes) array, i.e., P is of size $1 \times l$. Note that, here samples of P are in bytes (0 to $2^8 - 1$).
3. Now, if l is a multiple of 4, then no padding is required, else pad $(4 - r)$ elements 'post' with zeros to P, where r is a remainder on dividing l by 4.
4. Convert the bytes of P into WORDS, where WORD is a collection of 4 bytes, and rename the audio file P as P_w.

Output. The audio file P_w of size $1 \times m$, where m denotes number of WORDS in P_w.

4.2 Reverse preprocessing on the audio file

Input. The audio file P_w of size $1 \times m$, where m being number of WORDS in P_w.

1. Convert the WORDS of the audio file P_w into bytes (0 to $2^8 - 1$), and now, the size of P_w is $1 \times 4m$. Rename P_w as P.
2. Remove 'last' zero (padded) bytes, if any, from P, and let the size of P becomes $1 \times l$ bytes.
3. Convert the bytes (non-negative integers—0 to $2^8 - 1$) into binary (matrix).
4. Convert the binary (matrix) into floating-point values via single-precision floating point (32-bit).
5. Take transpose of P, so that the size of P becomes $l \times 1$.

Output. The audio file P of size $l \times 1$.

4.3 Preprocessing on secret key

Input. Secret key $K = \{k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}, k_{17}, k_{18}, k_{19}, k_{20}, k_{21}, k_{22}, k_{23}, k_{24}, k_{25}, k_{26}, k_{27}, k_{28}, k_{29}, k_{30}, k_{31}, k_{32}\}$ of 32 bytes.

1. Split the secret key K into two equal parts, say, K_1 and K_2 as: $K_1 = \{k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}, k_{11}, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}\}$ and $K_2 = \{k_{17}, k_{18}, k_{19}, k_{20}, k_{21}, k_{22}, k_{23}, k_{24}, k_{25}, k_{26}, k_{27}, k_{28}, k_{29}, k_{30}, k_{31}, k_{32}\}$.
2. Convert the key-bytes of K_1 and K_2 into WORDS as: $K_{1w} = \{q_{1w}, q_{2w}, q_{3w}, q_{4w}\}$ and $K_{2w} = \{r_{1w}, r_{2w}, r_{3w}, r_{4w}\}$, where $q_{1w} = k_1k_2k_3k_4$, $q_{2w} = k_5k_6k_7k_8$, $q_{3w} = k_9k_{10}k_{11}k_{12}$, and $q_{4w} = k_{13}k_{14}k_{15}k_{16}\}$; $r_{1w} = k_{17}k_{18}k_{19}k_{20}$, $r_{2w} = k_{21}k_{22}k_{23}k_{24}$, $r_{3w} = k_{25}k_{26}k_{27}k_{28}$, and $r_{4w} = k_{29}k_{30}k_{31}k_{32}\}$.
3. ***Expansion of*** K_{1w}.
   - Expand K_{1w} to the size m as:
     (a) For $i = 1, 2, 3, 4$; $T_1[i] = K_{1w}[i]$, i.e., $T_1[1] = q_{1w}$, $T_1[2] = q_{2w}$, $T_1[3] = q_{3w}$, and $T_1[4] = q_{4w}$.
     (b) Calculate $T_1[5]$ as:

$$T_1[5] = \mathrm{mod}(\lceil \mathrm{mean}(T_1[i]) \rceil,\ 2^{32}), \quad i = 1, 2, 3, 4.$$

     where 'mean' denotes the average function and 'mod' denotes the modulus function.
     (c) Calculate $T_1[i]$, for $i = 6, 7, \ldots, m$, as:

$$T_1[i] = \mathrm{mod}(T_1[i-1] + T_1[i-2],\ 2^{32}), \quad i = 6, 7, \ldots, m.$$

4. ***Expansion of*** K_{2w}.
   - Expand K_{2w} to the size m as:
     (a) For $i = 1, 2, 3, 4$; $T_2[i] = K_{2w}[i]$, i.e., $T_2[1] = r_{1w}$, $T_2[2] = r_{2w}$, $T_2[3] = r_{3w}$, and $T_2[4] = r_{4w}$.
     (b) Calculate $T_2[5]$ as:

$$T_2[5] = \mathrm{mod}(\lceil \mathrm{mean}(T_2[i]) \rceil,\ 2^{32}), \quad i = 1, 2, 3, 4.$$

     where symbols have their usual meanings.
     (c) Calculate $T_2[i]$, for $i = 6, 7, \ldots, m$, as:

$$T_2[i] = \mathrm{mod}(T_2[i-1] + T_2[i-2],\ 2^{32}), \quad i = 6, 7, \ldots, m.$$

5. ***Generation of a third key.***
   - Generate a third key K_{3w} from K_{1w} and K_{2w} as:

$$K_{3w} = \mathrm{mod}(K_{1w} \cdot K_{2w}, 2^{32})$$

     where '$\cdot$' denotes component-wise multiplication.

Output. The expanded keys T_1 and T_2 of size m, and the generated key K_{3w} of size 4.

4.4 Encryption algorithm

Input. An audio file P of size $l \times 1$ and the secret key K of 32-byte.

1. Apply preprocessing on the audio file P (see Sect. 4.1), and let the obtained file be P_w of size $1 \times m$.
2. Apply preprocessing on secret key K (see Sect. 4.3) to obtain the expanded keys T_1 & T_2 of size m, and the generated key K_{3w} of size 4 (in WORDS).

[1] See Available at https://in.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html 2023, https://in.mathworks.com/help/matlab/ref/rms.html 2023.

3. ***Initial round substitution.*** XOR P_w with T_1, i.e.,

$$B[i] = P_w[i] \oplus T_1[i], \quad i = 1, 2, \ldots, m.$$

4. ***First round substitution.***

(a) Let $B = \{b_1, b_2, \ldots, b_m\}$, then do the following:

for $i = 1$ to m
$b_{i-1} = c_{i-1}$
$c_i = [b_i \lll \sigma(b_{i-1})] \oplus b_{i-1}$
end for

where $c_0 = b_m$; 'σ' in $\sigma(b_{i-1})$ denotes sum-of-digits function, and $\sigma(b_{i-1})$ denotes sum-of-digits of b_{i-1}; and '$\lll$' denotes left rotation operator.

(b) Let $C = \{c_1, c_2, \ldots, c_m\}$, then do the following:

$$C[i] = C[i] \oplus K_{3w}[i], \quad i = 1, 2, 3, \text{ and}$$

$$C[m] = C[m] \oplus K_{3w}[4].$$

5. ***Second round substitution.***

(a) Do the following:

for $j = 1$ to m
$c_{j-1} = d_{j-1}$
$d_j = [c_j \lll \sigma(c_{j-1})] \oplus c_{j-1}$
end for

where $d_0 = c_m$, and the rest symbols have their usual meanings.

(b) Let $D = \{d_1, d_2, \ldots, d_m\}$, then do the following:

$$E[j] = D[j] \oplus T_2[j], \quad j = 1, 2, \ldots, m.$$

6. Apply reverse preprocessing on the audio file E of size $1 \times m$ (see Sect. 4.2), and let the obtained audio file be F of size $l \times 1$.

Output. The encrypted audio file F of size $l \times 1$.

4.5 Decryption algorithm

Input. The encrypted audio file F of size $l \times 1$ and the secret key K (32-byte).

1. Apply the preprocessing on the audio file F (see Sect. 4.1) to obtain an audio file E of size $1 \times m$, m being number of WORDS in E.
2. ***Second round substitution.***

(a) XOR the audio file E with T_2, i.e.:

$$D[j] = E[j] \oplus T_2[j], \quad j = 1, 2, \ldots, m.$$

(b) Let $D = \{d_1, d_2, \ldots, d_m\}$, then do the following:

for $j = m$ to 1
$c_j = [d_j \oplus d_{j-1}] \ggg \sigma(d_{i-1})$
end for

where '$j = m$ to 1' means $j = m, m-1, \ldots, 2, 1$; $d_0 = d_m$; and '$\ggg$' denotes right rotation.

3. ***First round substitution.***

(a) Let $C = \{c_1, c_2, \ldots, c_m\}$, then do the following:

$$C[i] = C[i] \oplus K_{3w}[i], \quad i = 1, 2, 3, \text{ and}$$
$$C[m] = C[m] \oplus K_{3w}[4].$$

(b) Do the following:

for $i = m$ to 1
$b_i = [c_i \oplus c_{i-1}] \ggg \sigma(c_{i-1})$
end for

where $c_0 = C_m$, and the rest symbols have their usual meanings.

4. ***Initial round substitution.*** Let $B = \{b_1, b_2, \ldots, b_m\}$, then do the following:

$$P_w[i] = B[i] \oplus T_1[i], \quad i = 1, 2, \ldots, m.$$

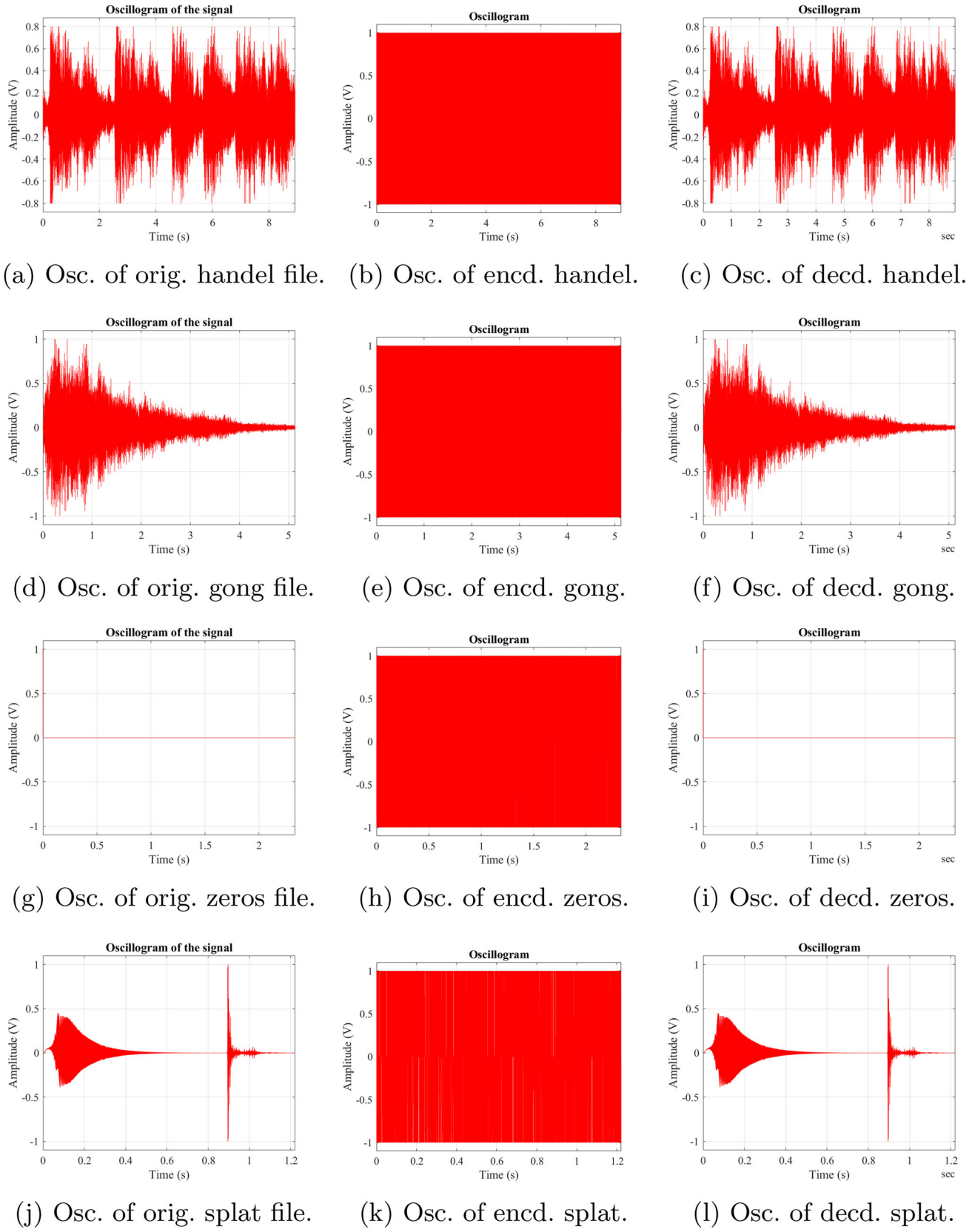
5. Apply the reverse preprocessing on the audio file P_w (see Sect. 4.2) of size $1 \times m$, to obtain the audio file P of size $l \times 1$.

Output. The decrypted (original) audio file P of size $l \times 1$.

5 Implementation and experimental results

The proposed technique is implemented on MATLAB ($R2021a$) software under the Windows 10 operating system. To evaluate the performance (encryption and decryption qualities) of the proposed technique, a number of mono-channel audio files of different sample lengths are taken from the MATLAB IPT (Image Processing Toolbox),[2] except the audio file 'zeros.wav', which is created in the MATLAB software. The details of these audio files are provided in Table 1. Also, the oscillograms (osc. for oscillogram—in short) of the

[2] Available in, C:\Program Files\Polyspace\R2021a\toolbox\images\imdata.



(a) Osc. of orig. handel file. (b) Osc. of encd. handel. (c) Osc. of decd. handel.

(d) Osc. of orig. gong file. (e) Osc. of encd. gong. (f) Osc. of decd. gong.

(g) Osc. of orig. zeros file. (h) Osc. of encd. zeros. (i) Osc. of decd. zeros.

(j) Osc. of orig. splat file. (k) Osc. of encd. splat. (l) Osc. of decd. splat.

Fig. 3 Experimental results: **a**, **d**, **g**, and **j** show the oscillograms of original audio files; **b**, **e**, **h**, and **k** show the oscillograms of the corresponding encrypted audio files; and **c**, **f**, **i**, and **l** show the oscillograms of the corresponding decrypted audio files

Table 2 Comparison of the key space

Our method	Demirtas (2023)	Sathiyamurthi and Ramakrishnan (2017)	Ghasemzadeh and Esmaeili (2017)	Augustine et al. (2015)	Liu et al. (2016)
2^{256}	2^{159}	2^{149}	2^{144}	2^{128}	2^{128}

original (orig. for original—in short), encrypted (encd. for encrypted—in short), and decrypted (decd. for decrypted—in short) audio files are shown in Fig. 3.

From Fig. 3, we observe that the oscillograms of the encrypted audio files are uniform, unlike those of the corresponding original audio files. Also, the oscillograms of the decrypted audio files are identical to those of the corresponding original files. Thus, our proposed technique performs robust encryption. Also, since the audio files are successfully decrypted without any data loss, so, the designed technique performs lossless decryption.

6 Security analyses

6.1 Key space analysis

The space of all potential combinations of a key constitutes a key space of any encryption/decryption algorithm. Key space should be very large so that attacks, such as brute-force (ECRYPT II yearly report on algorithms 2023), known/chosen plaintext (Stinson 2006), etc., could become unsuccessful. Our proposed technique is based on a secret key of 32 bytes (256 bits), which produces a key space of 2^{256}, and as of today, it is believed to be unbreakable. We also compare our key space with the key space of the existing methods. The results are provided in Table 2, whence we infer that our proposed technique has a very large key space as compared to the existing methods.

6.2 Key sensitivity analysis

This test is utilized to judge the confusion property (Shannon 1949) of any encryption/decryption algorithm. According to Shannon (1949), a secure cryptographic algorithm must have the confusion property to thwart statistical attacks. It is the property of confusion that hides the relationship between the encrypted data and the secret key. The key sensitivity test is utilized to judge this confusion property. The sensitivity of the secret key is assessed in two aspects:

1. Encryption: It is used to measure the dissimilarity between the two encrypted audio files E_1 and E_2 with respect to (w.r.t.) the same plain audio file P using two different encryption keys λ_1 and λ_2, where λ_1 and λ_2 are obtained from the original secret key K by altering merely LSB corresponding to the last and the first byte of K, respectively.
2. Decryption: It is used to measure the dissimilarity between the two decrypted audio files D_1 and D_2 w.r.t. the same encrypted audio file E, encrypted via secret key K, using the decryption keys λ_1 and λ_2, respectively. Note that, both encryption/decryption keys λ_1 and λ_2 differ from each other as well as from the secret key K merely by 1-bit.

The results of key sensitivity analysis w.r.t. the encryption (enc—in short) and decryption (dec—in short) aspects are shown in Figs. 4 and 5, respectively, whence we infer that the proposed technique has a very high bit-level sensitivity, and thus, ensures the property of confusion.

6.3 Encryption evaluation metrics

Since any single metric can not evaluate any encryption algorithm (or any encrypted audio file) fully, so we utilize several metrics, namely, spectrogram, adjacent sample correlation coefficient, signal-to-noise ratio, root mean square, crest factor, and a number of sample change rate.

6.3.1 Spectrogram analysis

The spectrogram (https://in.mathworks.com/help/signal/ref/spectrogram.html 2023) is a graph of an audio file between frequency and time. X-axis represents time in seconds, Y-axis represents frequency in Hertz, and the coordinate values represent energy values. The spectrograms of the original and the corresponding encrypted audio files are shown in Fig. 6. From Fig. 6, we observe that the spectrograms of the encrypted audio files have uniform darker color (yellow color), i.e., have stronger energy, unlike those of original audio files, which have (non-uniform) lighter color (mostly non-yellow), i.e., have weaker energy. Thus, the encrypted audio files are random-like audio files, which do not provide any relevant information regarding the original audio files.

6.3.2 Adjacent sample correlation coefficient (ASCC) analysis

ASCC test (Fisher and Yates 1958) is the frequently used measure to assess the concreteness of the novel techniques constructed for audio encryption, and in particular, to test the random distribution of samples in the encrypted audio file. Here, we have taken two thousand pairs of samples, which are chosen at random to estimate ASCC along vertical direction. Note that, ASCC along horizontal and diagonal directions can not be calculated since a single-channel audio file is merely a column vector, not a matrix.

Let I be an audio file of size $l \times 1$. Then, the correlation coefficient of adjacent samples of I is given by Eq. (1):

$$\rho_{XY} = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X) \cdot \mathrm{Var}(Y)}}, \tag{1}$$

where $\mathrm{Cov}(X, Y)$ denotes the covariance between the column vectors X and Y, while $\mathrm{Var}(X)$ denotes the variance of column vector X. The X and Y are computed as follows:

$$\left.\begin{array}{l} X = I(1 : l-1,) \\ Y = I(2 : l,) \end{array}\right\}.$$

Since the neighboring samples in the original audio file are strongly correlated so, the value of correlation coefficient ρ_{XY} tends to 1, and in the case of an encrypted audio file, the value of ρ_{XY} tends towards 0, cause the samples in the encrypted audio file are weakly correlated. The ASCC values of the plain and the cipher audio files along the vertical direction are shown in Table 3. For quick observation, the correlation graphs are also provided, which are shown in Fig. 7. From Fig. 7, we infer that the correlations graphs of the encrypted audio files are uniform, unlike those of the original audio files.

We also compare the obtained ASCC values with the most recent methods, and the comparison is provided in Table 3. Note that, the symbol '–' in Table 3 means "not available," i.e., the data is not available in the literature.

6.3.3 Signal-to-noise ratio (SNR) analysis

The SNR (https://in.mathworks.com/help/signal/ref/snr.html, 2023) is also a metric used to analyze an encrypted audio file. The more negative SNR implies better encryption quality. It is measured in decibel (dB) units. The SNR of an audio file, say, I can be calculated via Eq. (2):

$$\mathrm{SNR} = \frac{\mu}{\psi}, \tag{2}$$

where μ (mean) and ψ (standard deviation) are given by Eqs. (3) and (4), respectively:

$$\mu = \frac{\sum_{j=1}^{l} u_j}{l}, \tag{3}$$

$$\psi = \sqrt{\frac{\sum_{j=1}^{l} (u_j - \mu)^2}{l}}, \tag{4}$$

where 'u_j' denotes the samples of the audio (plain/cipher) file I and 'l' denotes the number of samples in the audio file. The SNR values of the plain and the cipher audio files are provided in Table 4.

6.3.4 RMS (root mean square) analysis

The RMS (Available at 2023) is used to calculate the average amplitude value of any (plain/cipher) audio file. For an original audio file, it should be closed to zero, while for an encrypted audio file, it should be closed to one. It can be calculated using Eq. (5):

$$\mathrm{RMS} = \sqrt{\frac{1}{l}\sum_{j=1}^{l} {u_j}^2}, \tag{5}$$

where symbols have their usual meanings.

The RMS values for the plain and the cipher audio files are provided in Table 4, whence we notice that the RMS values for the cipher audio files are close to the ideal value.

6.3.5 Crest factor (CF) analysis

The CF (https://in.mathworks.com/help/predmaint/ug/signal-features.html 2023), a.k.a. peak-to-average ratio, is another metric to analyze an audio file. It is measured in dB units. For an encrypted audio file, the crest factor should be closed to 3 dB. It can be calculated using Eq. (6):

$$\mathrm{CF} = \frac{u_p}{\mathrm{RMS}}, \tag{6}$$

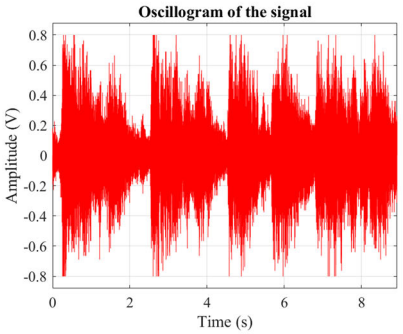
where u_p (peak value) is the maximum absolute value of an audio file, and RMS (average value) is the root mean square value, given by Eq. (5).

The CF values for the plain and the cipher audio files are provided in Table 4, whence we notice that the CF values for the cipher audio files are very close to the ideal value (3 dB).

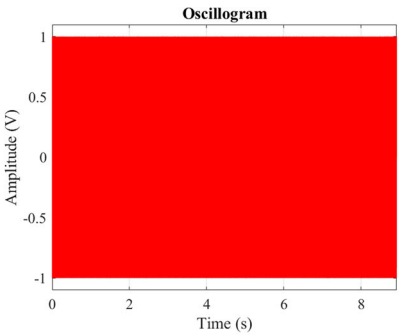
6.3.6 Number of sample change rate (NSCR) test

The NSCR (Wu et al. 2011) is used to test the resistance of differential attack (Biham and Shamir 1993), or to judge the

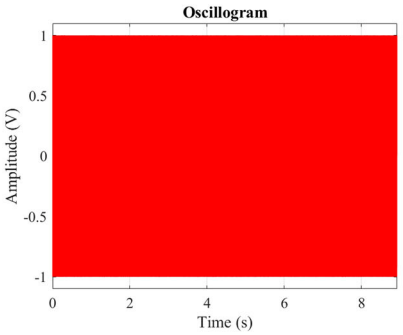
Fig. 4 Key sensitivity analysis w.r.t. encryption



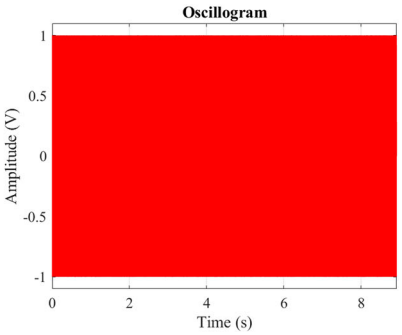
(a) Osc. of orig. handel file (P).



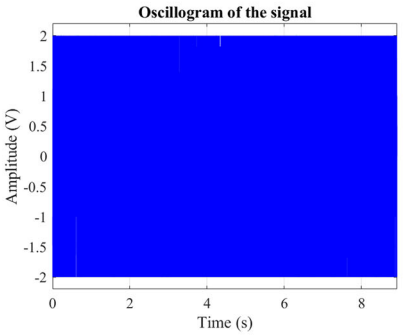
(b) Osc. of encrypted handel file E; $E = enc(P, K)$.



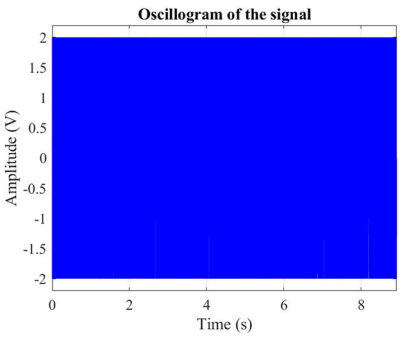
(c) Osc. of E_1; $E_1 = enc(P, \lambda_1)$.



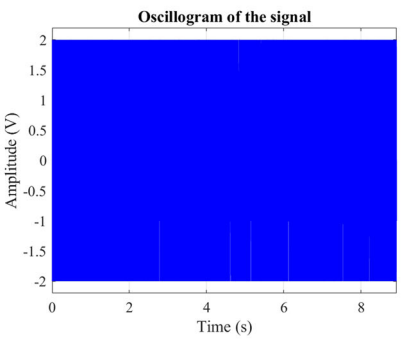
(d) Osc. of E_2; $E_2 = enc(P, \lambda_2)$.



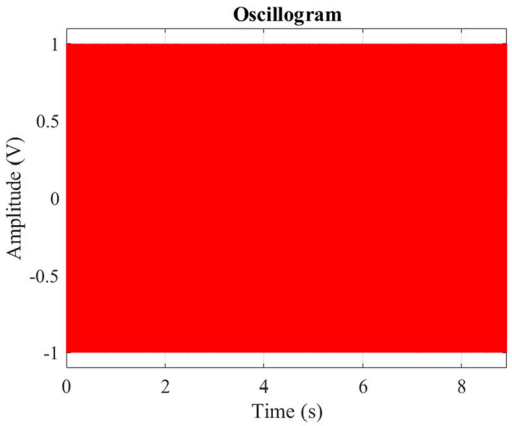
(e) Osc. of $|E_1 - E|$; $|E_1 - E|$ is the absolute difference between E_1 and E.



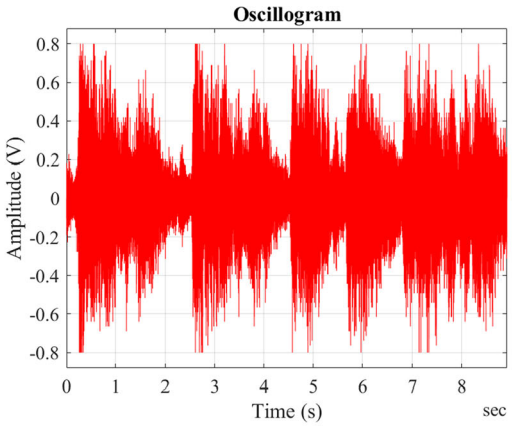
(f) Osc. of $|E_2 - E|$.



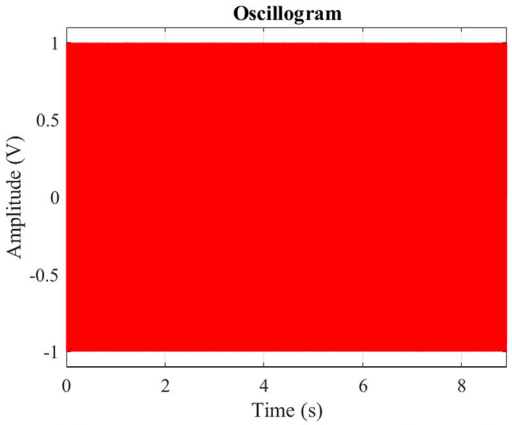
(g) Osc. of $|E_2 - E_1|$.



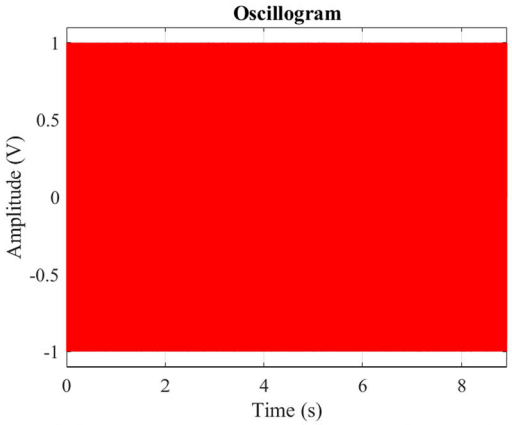
(a) Osc. of the encrypted handel file E; $E = enc(P, K)$.



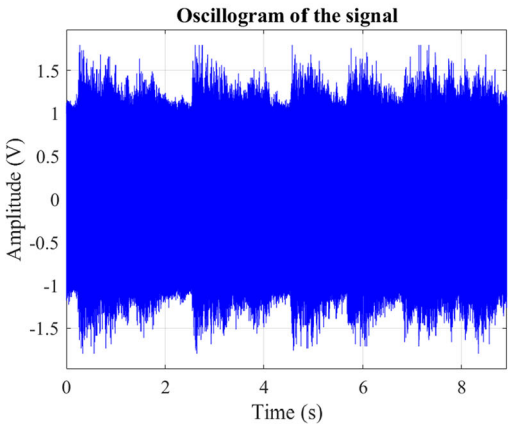
(b) Osc. of the decrypted handel (D) using correct secret key K.



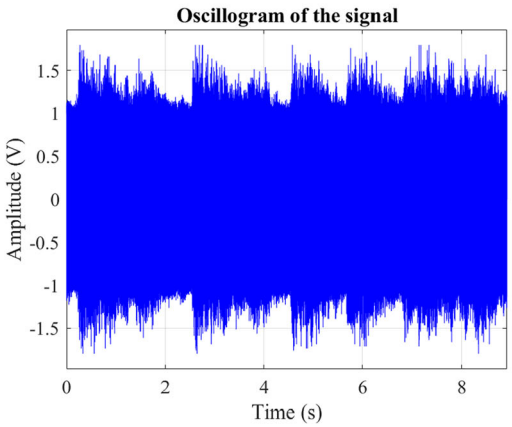
(c) Osc. of D_1; $D_1 = dec(E, \lambda_1)$.



(d) Osc. of D_2; $D_2 = dec(E, \lambda_2)$.



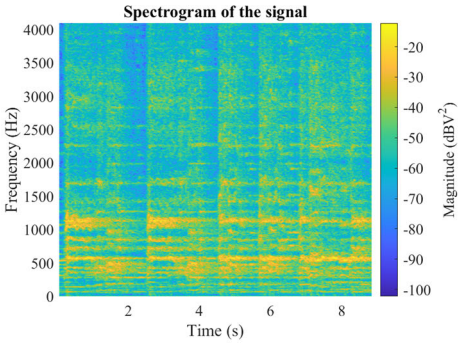
(e) Osc. of $|D_1 - D|$.



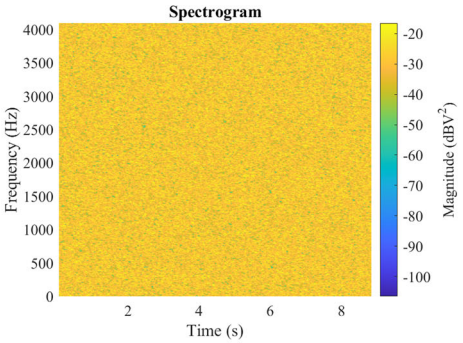
(f) Osc. of $|D_2 - D|$.

Fig. 5 Key sensitivity analysis w.r.t. decryption

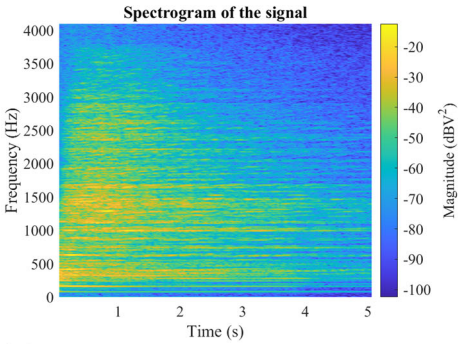
Fig. 6 Spectrograms of the original and the corresponding encrypted audio files



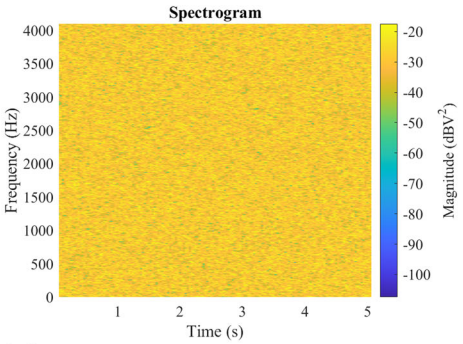
(a) Spectrogram of original handel file.



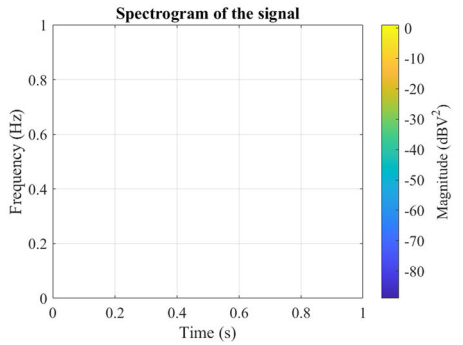
(b) Spectrogram of encrypted handel file.



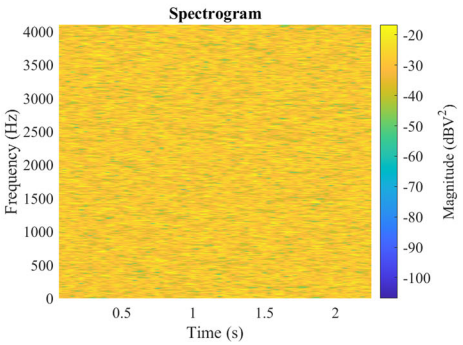
(c) Spectrogram of original gong file.



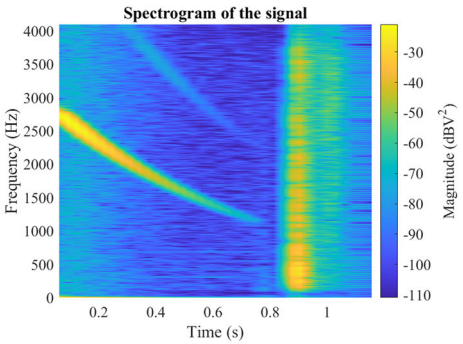
(d) Spectrogram of encrypted gong file.



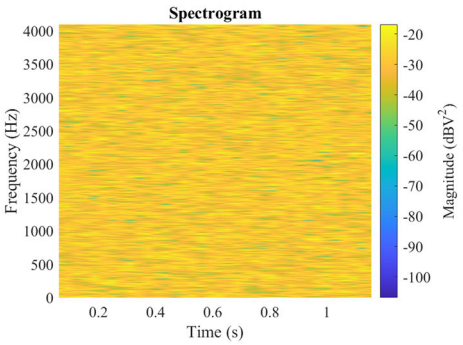
(e) Spectrogram of original zeros file.



(f) Spectrogram of encrypted zeros file.



(g) Spectrogram of original splat file.



(h) Spectrogram of encrypted splat file.

Table 3 ASCC values of the plain and the cipher audio files

Method	File name (.wav)	Duration (in s)	Size (in KB)	ASCC	
				Plain	Cipher
Proposed	Handel	8.9249	142.7988	0.7554	− 0.0065
	Gong	5.1304	82.0859	0.6413	0.0067
	Zeros	2.3340	37.3438	NaN	− 0.1041
	Splat	1.2208	25.1562	− 0.0958	0.0075
Reference Demirtas (2023)	Audio 2	12.2	–	0.9965	0.0089
Reference Shah et al. (2021b)	Music sound	–	–	0.9847	− 0.0081
Reference Shah et al. (2021a)	Female sound	–	31.25	0.9933	− 0.0019
Reference Abdelfatah (2020)	Audio-2	1.7600	296.875	–	− 0.0003

Shannon's diffusion property (Shannon 1949). The NSCR scores between the encrypted audio files E_1 and E_2 can be calculated via Eq. (7):

$$\text{NSCR} = \sum_{s=1}^{l} \frac{\beta(s,1)}{l} \times 100\%, \tag{7}$$

where $\beta(s,1)$ is given by Eq. (8):

$$\beta(s,1) = \begin{cases} 0, & \text{if } E_1(s,1) = E_2(s,1) \\ 1, & \text{if } E_1(s,1) \neq E_2(s,1) \end{cases}, \tag{8}$$

where $E_1(s,1)$ and $E_2(s,1)$ are the samples of the encrypted audio files before and after the alteration of only one sample of the original audio file.

We have calculated the NSCR scores by changing only one sample of the test audio files at different positions (from beginning—(1, 1)th sample as well as from the last—(l, 1)th sample), l being the total number of samples in an audio file. The obtained NSCR scores are shown in Table 5.

Note that, if the calculated/reported NSCR score is greater than the theoretical NSCR value, which is 99.5527 at 0.01 significance level and 99.5693% at 0.05 level (Wu et al. 2011), then the NSCR test is passed. The proposed technique passes the NSCR test for all the audio files, and thus, ensures the property of diffusion, and also, outperforms the methods listed in Table 5, which are vulnerable to the differential attack.

6.4 Decryption evaluation metrics

To evaluate the decryption algorithm, i.e., the decrypted audio files, we utilize two important metrics: mean square error and peak-signal-to-noise ratio.

6.4.1 Mean Squared Error (MSE) analysis

The MSE (https://en.wikipedia.org/wiki/Mean_squared_error 2023) is used to judge the decryption quality of any decrypted audio file. MSE value can be any non-negative integer. Lower the MSE, better is the decryption quality, in particular, value 0 denotes perfect decryption, i.e., the original and the decrypted audio files are identical—lossless decryption. The MSE can be calculated via Eq. (9):

$$\text{MSE} = \sum_{j=1}^{l} \frac{(P_j - D_j)^2}{l}, \tag{9}$$

where P_j and D_j denote the jth samples of the original and the decrypted audio files, respectively, while other symbols have their usual meanings.

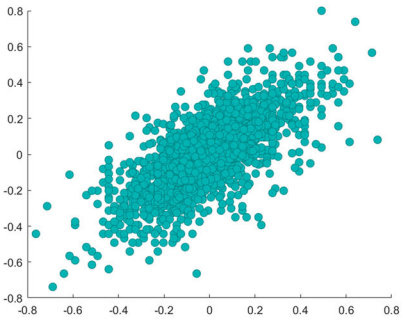
The values of MSE between the original and the decrypted audio files are provided in Table 6. From the table, we observe that the MSE values are 0 (zero), endorsing that the decrypted audio files are perfectly identical to the original audio files. Thus, the proposed approach performs lossless decryption.

Note that MSE is a straightforward and a better decryption evaluation metric, as compared to PSNR (see Sect. 6.4.2), since it does not require any other metric.

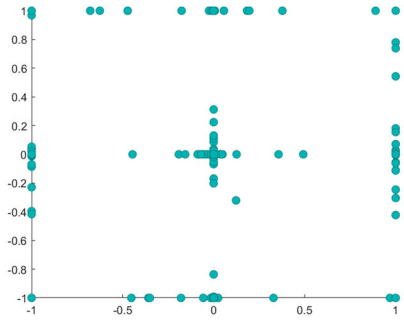
6.4.2 Peak-signal-to-noise ratio (PSNR) analysis

The PSNR (https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio 2023) metric is also used to measure the quality of the decrypted audio file. PSNR can be any positive real number and is measured in dB units. Higher the value of PSNR, better is the decryption quality. In particular, PSNR value equals to ∞ implies perfect decryption, i.e., lossless decryption. It can be calculated using Eq. (10):

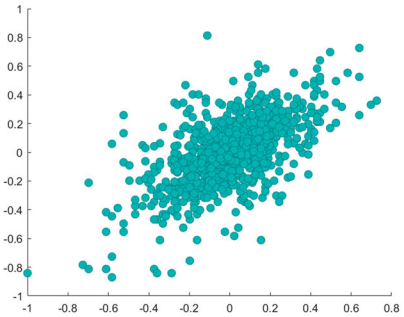
$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{h^2}{\text{MSE}}\right), \tag{10}$$



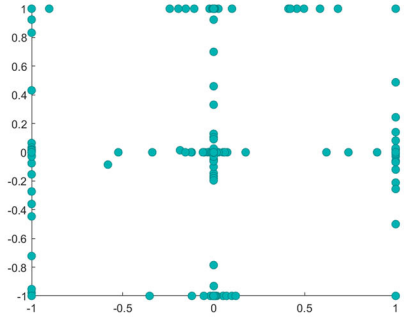
(a) ASCC graph of the original handel file.



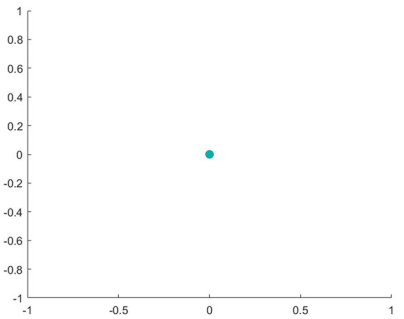
(b) ASCC graph of the encrypted handel file.



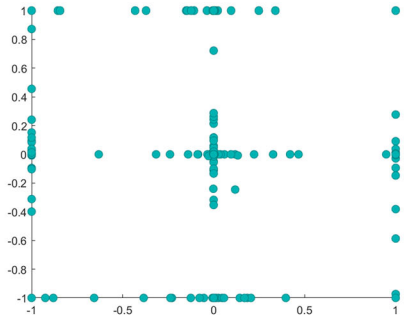
(c) ASCC graph of the original gong file.



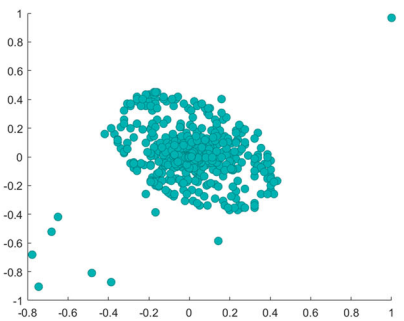
(d) ASCC graph of the encrypted gong file.



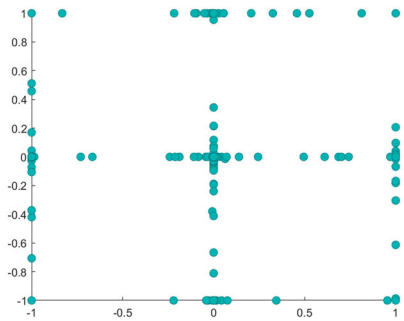
(e) ASCC graph of the original zeros file.



(f) ASCC graph of the encrypted zeros file.



(g) ASCC graph of the original splat file.



(h) ASCC graph of the encrypted splat file.

Fig. 7 ASCC graphs of the original and the encrypted audio files along vertical direction

Table 4 SNR, RMS, and CF values of the plain and the cipher audio files

Method	File name (.wav)	Duration (in s)	Size (in KB)	SNR (dB)		RMS		CF (dB)	
				Plain	Cipher	Plain	Cipher	Plain	Cipher
Proposed	Handel	8.9249	142.7988	− 16.1304	− 27.5990	0.1962	0.7089	12.2074	2.9892
	Gong	5.1304	82.0859	− 12.3680	− 25.5598	0.1538	0.7087	16.2633	2.9904
	Zeros	2.3340	37.3438	− 34.0016	− 23.9284	0.0072	0.7092	42.8149	2.9842
	Splat	1.2208	25.1562	− 20.4148	− 21.2276	0.0959	0.7119	20.3654	2.9527
Reference Abdallah and Meshoul (2023)	Test audio	180	–	–	− 2.69	–	–	–	–
Reference Shah et al. (2021b)	Music sound	–	–	–	–	–	0.6423	–	4.7610
Reference Naskar et al. (2021)	Audio-4	–	162	–	− 8.6030	–		–	–
Reference Aziz et al. (2021)	Audio 1	–	–	–	–	–	0.0034	–	49.3
Reference Abdelfatah (2020)	Audio-2	1.7600	296.875	–	− 28.1400	–	0.6027	–	4.3973
Reference Al-kateeb and Mohammed (2020)	A5	–	9.6436	–	− 18.2154	–	–	–	–
Reference Kordov (2019)	File 3	1.60	16.8457	–	− 8.7189	–	–	–	–
Reference Naskar et al. (2019)	–	–	–	–	–	–	0.6000	–	4.8
Reference Belmeguenai et al. (2017)	–	10.6300	–	–	–	–	–	–	
Reference Sathiyamurthi and Ramakrishnan (2017)	Audio-2	8.0	–	–	32.5781	–	–	–	–

where 'h' denotes largest possible value of an audio file and 'MSE' is defined by Eq. (9).

The values of PSNR between the original and the decrypted audio files are provided in Table 6. From the table, we observe that the PSNR values are equal to ∞, endorsing that the decrypted audio files are perfectly identical to the original audio files. In other words, the decryption algorithm performs lossless decryption.

6.5 Running time of the proposed technique

The encryption (or decryption) execution time of a cryptographic algorithm is an important factor to evaluate the performance of the algorithm. Lesser the running time, better is the performance. We compute the encryption time (in s) for the test audio files via MATLAB's function known as tic-toc'. Since we are using a symmetric key algorithm, so, the decryption time is the same as the encryption time. The encryption times are shown in Table 7.

7 Comparison with the existing techniques

The proposed technique is compared with the recent state-of-the-art techniques based on commonly available metrics, namely, key space, ASCC, SNR, RMS, CF, NSCR, and running time. The comparison of the proposed approach with the recent approaches based on key space is provided in Table 2; based on ASSC is provided in the Table 3; based on SNR, RMS, and CF metrics is provided in the Table 4; based on the metric NSCR is provided in the Table 5; and based on the running time is provided in the Table 7. From the Tables 2, 3, 4, and 5, we infer that our proposed technique performs well in terms of respective compared metrics.

8 Discussion and conclusion

In this paper, a technique for securing digital audio files, based on the WORD-oriented RX operations, has been proposed. The proposed technique encrypts a digital audio file into a random-like (noisy) audio file. To evaluate the encryption quality of the proposed technique, several metrics, viz., key sensitivity, ASCC, SNR, RMS, CF, and NSCR, have been employed. Analogously, to evaluate the decryption quality of the proposed method, the MSE and PSNR metrics have been employed.

The proposed technique has a very large key space of 2^{256} bits, which indicates resistance against brute-force attack, and a very high key sensitivity w.r.t. the encryption and decryption, which indicates the resistance against known-plain text, chosen-cipher text, etc., attacks. Also, the cipher audio files have attained the scores (nearest to ideal) for

Table 5 NSCR scores of the encrypted images

Method	File name (.wav)	Duration (in s)	Size (in KB)	Position altered	NSCR score (in %)
Proposed	Handel	8.9249	142.7988	(1, 1)	100
				(l, 1)	100
	Gong	5.1304	82.0859	(2, 1)	100
				($l-2$, 1)	100
	Zeros	2.3340	37.3438	(1, 1)	100
				(l, 1)	100
	Splat	1.2208	25.1562	(1, 1)	100
				(l, 1)	100
Demirtas (2023)	Audio 2	12.2	–	–	99.7501
Khalid et al. (2022)	Male sound	–	42.1245	–	99.9996
Shah et al. (2021b)	Bells sound	–	–	–	99.9884
Faragallah and El-Sayed (2021)	Alarm	–	–	–	99.7500
Naskar et al. (2021)	Audio-4	–	162	–	99.9958
Shah et al. (2021a)	Female sound	–	31.25	–	99.9958
Stoyanov and Ivanova (2021)	usb-headset-weird	2.3200	24.4141	–	99.9940
Aziz et al. (2021)	Audio 1	–	–	–	99.5316
Abdelfatah (2020)	Audio-2	1.7600	296.875	–	99.9700
Naskar et al. (2019)	–	–	–	–	99.9989
Shah et al. (2020)	Go ahead	–	139.6484	–	99.9973

Table 6 MSE and PSNR values between the decrypted and the original audio files

Method	File name (.wav)	Duration (in s)	Size (in KB)	MSE	PSNR
Proposed	Handel	8.9249	142.7988	0	∞
	Gong	5.1304	82.0859	0	∞
	Zeros	2.3340	37.3438	0	∞
	Splat	1.2208	25.1562	0	∞
Reference Abouelkheir and Sherbiny (2022)	Sen_4	1.1901	41.0156	3.3161×10^{-11}	–

Table 7 Running time of the proposed technique

Method	File name (.wav)	Duration (in s)	Size (in KB)	Encryption time (in s)
Proposed	Handel	8.9249	142.7988	379.4791
	Gong	5.1304	82.0859	158.2024
	Zeros	2.3340	37.3438	55.8380
	Splat	1.2208	25.1562	31.2809
Reference Suryadi et al. (2023)	Audio 1	5	105.006	55
Reference Demirtas (2023)	Audio 2	12.2	–	0.3832
Reference Khalid et al. (2022)	Male sound	–	42.1245	0.0073
Reference Abouelkheir and Sherbiny (2022)	Speaker 3	–	–	0.3395

ASCC as −0.0065; SNR as −27.5990; RMS as 0.7119; CF as 2.9904; and NSCR as 100%. Since the NSCR score is 100%, so, the proposed technique is strongly resistant to the differential attack. Moreover, since the deciphered audio files attained the ideal values of MSE and PSNR, which are 0 and ∞, so, the proposed approach performs lossless decryption. Furthermore, a thorough comparison with the recent state-of-the-art techniques, based on the commonly available metrics, has also been made. The results of the comparison show that the proposed approach outperforms the compared approaches in terms of key space, SNR, RMS, CF, NSCR, MSE, and PSNR. However, some compared approaches have better ASCC scores and execution times than our proposed approach.

Since the proposed approach takes more time for the large audio files, so, it is more suitable for the small-sized audio files.

The proposed approach can be improved further in respect of the execution time. Moreover, it can also be applied to dual-channel audio files, and other types of digital data, i.e., text, images, and videos.

Acknowledgements The authors are grateful to the referees and the editor for their valuable suggestions and remarks that definitely improve the paper. The author Dr. Abdul Gaffar would like to thank the Integral University, Lucknow, India, for providing the manuscript number IU/R&D/2023-MCN0002108, for the present research work.

Funding This work was partially supported by the UGC (University Grants Commission), India, under Grant no. [415024].

Data availability Enquiries about data availability should be directed to the authors.

Declaration

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this manuscript.

Ethical approval This manuscript does not contain any studies with human participants and/or animals.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Abdallah HA, Meshoul S (2023) A multi-layered audio signal encryption approach for secure voice communication. Electronics 12(1):2. https://doi.org/10.3390/electronics12010002

Abdelfatah RI (2020) Audio encryption scheme using self-adaptive bit scrambling and two multi chaotic-based dynamic DNA computations. IEEE Access 8:69894–69907. https://doi.org/10.1109/ACCESS.2020.2987197

Abouelkheir E, Sherbiny SE (2022) Enhancement of speech encryption/decryption process using RSA algorithm variants. Hum Cent Comput Inf Sci. https://doi.org/10.22967/HCIS.2022.12.006

Al-kateeb ZN, Mohammed SJ (2020) A novel approach for audio file encryption using hand geometry. Multimed Tools Appl 79:19615–19628. https://doi.org/10.1007/s11042-020-08869-8

Augustine N, George SN, Pattathil DP (2015) An audio encryption technique through compressive sensing and Arnold transform. Int J Trust Manag Comput Commun 3(1):74–92. https://doi.org/10.1504/IJTMCC.2015.072467

Aziz H, Gilani SMM, Hussain I, Janjua AK, Khurram S (2021) A noise-tolerant audio encryption framework designed by the application of S8 symmetric group and chaotic systems. Math Probl Eng 2021:5554707. https://doi.org/10.1155/2021/5554707

Belmeguenai A, Ahmida Z, Ouchtati S, Dejmii R (2017) A novel approach based on stream cipher for selective speech encryption. Int J Speech Technol 20:685–698. https://doi.org/10.1007/s10772-017-9439-8

Biham E, Shamir A (1993) Differential cryptanalysis of the data encryption standard (DES). Springer, Berlin

Demirtas M (2023) A lossless audio encryption method based on Chebyshev map. Orclever Proc Res Dev 2(1):28–38. https://doi.org/10.56038/oprd.v2i1.234

ECRYPT II yearly report on algorithms and keysizes, Smart N (ed) (BRIS), 2011–12. https://www.ecrypt.eu.org/ecrypt2/documents/D.SPA.20.pdf. Accessed 19 Aug 2023

Faragallah OS (2018) Secure audio cryptosystem using hashed image LSB watermarking and encryption. Wirel Pers Commun 98:2009–2023. https://doi.org/10.1007/s11277-017-4960-2

Faragallah OS, El-Sayed HS (2021) Secure opto-audio cryptosystem using XOR-ing mask and Hartley transform. IEEE Access 9:25437–25449. https://doi.org/10.1109/ACCESS.2021.3055738

Farsana F, Gopakumar K (2016) A novel approach for speech encryption: Zaslavsky map as pseudo random number generator. Procedia Comput Sci 93:816–823. https://doi.org/10.1016/j.procs.2016.07.302

Farsana FJ, Devi VR, Gopakumar K (2019) An audio encryption scheme based on fast walsh hadamard transform and mixed chaotic keystreams. Comput Inform Appl. https://doi.org/10.1016/j.aci.2019.10.001

Fisher RA, Yates F (1958) Statistical methods for research workers, 13th edn. Hafner, New York

Ghasemzadeh A, Esmaeili E (2017) A novel method in audio message encryption based on a mixture of chaos function. Int J Speech Technol 20(4):829–837. https://doi.org/10.1007/s10772-017-9452-y

Habib Z, Khan JS, Ahmad J, Khan MA, and Khan FA (2017) Secure speech communication algorithm via DCT and TD-ERCS chaotic map. 4th International conference on electrical and electronic engineering (ICEEE). IEEE, pp 246–250. https://doi.org/10.1109/ICEEE2.2017.7935827

https://en.wikipedia.org/wiki/Mean_squared_error. Accessed 19 Aug 2023

https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio. Accessed 19 Aug 2023

https://en.wikipedia.org/wiki/Single-precision_floating-point_format. Accessed 19 Aug 2023

https://in.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html. Accessed 19 Aug 2023

https://in.mathworks.com/help/matlab/ref/rms.html. Accessed 19 Aug 2023

https://in.mathworks.com/help/predmaint/ug/signal-features.html. Accessed 19 Aug 2023

https://in.mathworks.com/help/signal/ref/snr.html. Accessed 19 Aug 2023

https://in.mathworks.com/help/signal/ref/spectrogram.html. Accessed 19 Aug 2023

Khalid I, Shah T, Almarhabi KA, Shah D, Asif M, Ashraf MU (2022) The SPN network for digital audio data based on elliptic curve

over a finite field. IEEE Access 10:127939–127955. https://doi.org/10.1109/ACCESS.2022.3226322
Kordov K (2019) A novel audio encryption algorithm with permutation-substitution architecture. Electronics 8:530. https://doi.org/10.3390/electronics8050530
Lima JB, Neto EFS (2016) Audio encryption based on the cosine number transform. Multimedia Tools Appl 75(14):8403–8418. https://doi.org/10.1007/s11042-015-2755-6
Liu H, Kadir A, Li Y (2016) Audio encryption scheme by confusion and diffusion based on multi-scroll chaotic system and one-time keys. Optik 127(19):7431–7438. https://doi.org/10.1016/j.ijleo.2016.05.073
Naskar PK, Paul S, Nandy D, Chaudhuri A (2019) DNA encoding and channel shuffling for secured encryption of audio data. Multimedia Tools Appl 78(17):25019–25042. https://doi.org/10.1007/s11042-019-7696-z
Naskar PK, Bhattacharyya S, Chaudhuri A (2021) An audio encryption based on distinct key blocks along with PWLCM and ECA. Nonlinear Dyn 103:2019–2042. https://doi.org/10.1007/s11071-020-06164-7
Sasikaladevi N, Geetha K, Srinivas KNV (2018) A multi-tier security system (SAIL) for protecting audio signals from malicious exploits. Int J Speech Tech 21(2):319–332. https://doi.org/10.1007/s10772-018-9510-0
Sathiyamurthi P, Ramakrishnan S (2017) Speech encryption using chaotic shift keying for secured speech communication. J Audio Speech Music Proc. https://doi.org/10.1186/s13636-017-0118-0
Shah D, Shah T, Jamal SS (2020) Digital audio signals encryption by Mobius transformation and Henon map. Multimed Syst 26:235–245. https://doi.org/10.1007/s00530-019-00640-w
Shah D, Shah T, Ahamad I, Haider MI, Khalid I (2021a) A three-dimensional chaotic map and their applications to digital audio security. Multimed Tools Appl 80:22251–22273. https://doi.org/10.1007/s11042-021-10697-3
Shah D, Shah T, Hazzazi MM, Haider MI, Aljaedia HI (2021b) An efficient audio encryption scheme based on finite fields. IEEE Access 9:144385–144394. https://doi.org/10.1109/ACCESS.2021.3119515
Shannon CE (1949) Communication theory of secrecy systems. Bell Syst Tech J 28(4):656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x
Stinson DR (2006) Cryptography: theory and practice. Chapman and Hall CRC, London
Stoyanov B, Ivanova T (2021) Novel implementation of audio encryption using pseudorandom byte generator. Appl Sci 11(21):10190. https://doi.org/10.3390/app112110190
Suryadi MT, Satria Y, Boyke M (2023) Digital audio protection with confusion and diffusion scheme using double-scroll chaotic function. J Hunan Univ Nat Sci. https://doi.org/10.55463/issn.1674-2974.50.5.6
Wang X, Su Y (2020) An audio encryption algorithm based on DNA coding and chaotic system. IEEE Access 8:9260–9270. https://doi.org/10.1109/ACCESS.2019.2963329
Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. J Sel Areas Telecommun 1:31–38