Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067524

ARTICLE

A Secure Audio Encryption Method Using Tent-Controlled Permutation and Logistic Map-Based Key Generation

Ibtisam A. Taqi* **and Sarab M. Hameed**

Computer Science Department, University of Baghdad, Baghdad, 10001, Iraq

*Corresponding Author: Ibtisam A. Taqi. Email: ibtisam.taqi@sc.uobaghdad.edu.iq

Received: 06 May 2025; Accepted: 07 July 2025; Published: 29 August 2025

ABSTRACT: The exponential growth of audio data shared over the internet and communication channels has raised significant concerns about the security and privacy of transmitted information. Due to high processing requirements, traditional encryption algorithms demand considerable computational effort for real-time audio encryption. To address these challenges, this paper presents a permutation for secure audio encryption using a combination of Tent and 1D logistic maps. The audio data is first shuffled using Tent map for the random permutation. The high random secret key with a length equal to the size of the audio data is then generated using a 1D logistic map. Finally, the Exclusive OR (XOR) operation is applied between the generated key and the shuffled audio to yield the cipher audio. The experimental results prove that the proposed method surpassed the other techniques by encrypting two types of audio files, as mono and stereo audio files with large sizes up to 122 MB, different sample rates 22,050, 44,100, 48,000, and 96,000 for WAV and 44,100 sample rates for MP3 of size 11 MB. The results show high Mean Square Error (MSE), low Signal-to-Noise Ratio (SNR), spectral distortion, 100% Number of Sample Change Rate (NSCR), high Percent Residual Deviation (PRD), low Correlation Coefficient (CC), large key space 2^{616}, high sensitivity to a slight change in the secret key and that it can counter several attacks, namely brute force attack, statistical attack, differential attack, and noise attack.

KEYWORDS: Wave protection; 1D logistic map; Tent map; random permutation; key generation

1 Introduction

Audio encryption is crucial for ensuring the privacy, security, and integrity of voice communications and audio files. Ensuring secrecy in both personal and professional contacts is crucial, as is preventing unwanted access to private chats or recordings sent over social media platforms such as WhatsApp, Telegram, Instagram, and Viber. Encryption is one of the methods for protecting audio by converting information into a code or cipher to protect data from unauthorized access. End-to-end encryption for voice messages ensures that only the intended recipients can decrypt and listen to the messages [1,2].

Audio files have some properties such as bulk data capacity and high redundancy, which make conventional cryptography algorithms unsuitable, especially for fast applications. To address a high demand for real-time applications, researchers need to design a new method that requires less computational power while preserving an adequate level of security. Chaotic maps have applications in various scientific and engineering fields, especially in cryptography. Chaotic maps are also known as dynamical systems or nonlinear maps, which are mathematical models that exhibit chaotic behavior. Chaos refers to the sensitive dependence on initial conditions, where small changes in the starting conditions of a system can led to vastly

different outcomes over time. These properties make chaotic systems a potential choice for constructing cryptosystems [2–5].

This paper attempts to investigate how the chaotic maps' characteristics can be utilized in audio encryption. The contributions of this paper are:

1. To develop a highly effective encryption method for audio data, a 1D logistic chaotic map that exhibits extreme sensitivity to initial conditions and control parameters is used. This characteristic ensures that even the slightest variation in the input will lead to drastically different outputs, generating a strong encryption key that matches the size of the audio data being processed.
2. To propose a permutation method to disrupt the correlation between adjacent samples in the audio data. Audio samples often bear a strong temporal correlation, and this can lead to patterns that may be exploited if not properly addressed. A permutation method can effectively rearrange the audio samples, ensuring that the relationship between them is obscured.
3. To develop a secure encryption method capable of handling larger audio data volumes with different audio types and sampling rates that is resistant to several types of attacks.

The organization of this paper is as follows: Section 2 presents some related works on audio encryption. Section 3 explains the logistic chaotic map and the Tent map. Section 4 describes the proposed audio encryption method in detail. Section 5 presents the obtained results and discussion. Finally, the conclusions and future works are presented in Section 6.

2 Related Work

This section presents the significant developments in audio encryption, highlighting major contributions to identifying the gaps and potential developments.

A novel chaotic shift keying-based speech encryption was introduced that depends on switching locations in two stages [1]. The first stage divides the file into four parts. Each part's locations are changed based on the Logistic Map, Tent Map, Quadratic Map, and Bernoulli's Map, respectively. Chen's map is used in the second phase for another permutation to increase security. The method encrypts files ranging in size from 3 to 8 s, achieving average NSCR ranges from 99.9998 to 99.9999, CC ranges from 0.0119 to 0.0384, SNR ranges from 34.7112 to 32.5781, and (Peak Signal-To-Noise Ratio) PSNR ranges from 62.3189 to 59.2281.

Sheela et al. [5] presented an audio cryptosystem that mixed the chaotic maps, hybrid chaotic shift transform (HCST), and (Deoxyribonucleic Acid) DNA rules. HCST is performed using the standard map and 2D Modified Henon Map to create complex permutations of the audio data. It offers robust protection against various cryptographic attacks. It is particularly suitable for applications such as real-time encryption and narrow-band radio communication. Despite the complex method of integrating DNA with chaotic maps and the efficiency of the method, the results of CC ranges from 0.0043 to −0.0028, PRD ranges from 2.0496 $\times 10^7$ to 1.5012 $\times 10^8$ and SNR ranges from 189.6684 to 194.9421.

Kordov [6] suggested an innovative method to encrypt audio files. The encryption phase is built on traditional symmetric patterns using pseudo-random numbers, a chaotic gradient circle map, and a modified rotation equation. A new pseudo-random generator was proposed and applied to chaotic bit-level permutations and different substitutions over the audio file structure to encrypt them. The results showed that the wave files were encrypted only, and the maximum size was 2.33 MB with SNR equal to −16.0483, NSCR 99.9984%, and CC 0.00047.

Albahrani et al. [7] produced a new technique for encrypting two-channel audio files suitable for telecommunication areas. The original audio data must be encoded into a new data range for the suggested strategy to function. The chaotic state and chaotic parameters are used to carry out substitution and

permutation operations, and each value in the generated range is converted to a binary sequence. The permutation technique was based on the numerical sequence produced by a hyperchaotic system, while the substitution was accomplished using the XOR operation and Bernoulli substitution. Based on the characteristics of the square root of big prime numbers and a hyperchaotic system, a novel key generation algorithm is adopted to generate the keys. The method encrypts only wave files, and the largest size is 2.66 MB. The key space is equal to $10^{64} \approx 2^{240}$, NSCR ranges from 99.597 to 99.617, CC ranges from −0.0021 to 0.00008, SNR ranges from −21.7739 to −38.0568, PSNR is equal to 5.1042, and MSE is equal to 42 dB.

A secure and lossless technique for audio files using the Chebyshev chaotic map was presented [8]. First, the integer and decimal portions of the incoming audio samples are extracted by pre-processing. The input audio sample's integer components are first scrambled and subsequently diffused by employing plain text-dependent variables to iterate the Chebyshev map in the chaotic region. Finally, a post-processing technique was used to diffuse audio samples. The schema is suitable for voice transfer applications. The method encrypts only wave files, and the largest size is 1.30 MB. The key space is equal to 2^{159}, NSCR ranges from 99.8772 to 99.6076, and CC ranges from −0.00282 to 0.00288.

A strong digital audio encryption cryptosystem using an Elliptic Curve (EC) was introduced [9]. The approach first distorts the digital audio pixel position using a specific type of EC over a binary extension field. By lowering the inter-correlation between the original audio's pixels, it strengthens the system's defenses against statistical attacks. An EC over a binary extension field is used to create a different number of substitution boxes (S-boxes), which confuse the data. A special curve that depends on effective EC arithmetic operations in the diffusion module is used in the proposed schema. The experiment results show that the maximum encrypted file size is 31.25 MB, CC ranges from 0.0017 to −0.0024, and NSCR ranges from 99.9847 to 99.9990.

Farsana and Devi developed a keystream derived from the modified Lorenz-Hyperchaotic system for substitution in an audio encryption technique that shuffles audio samples using a discrete Henon map after augmentation [10]. Walsh-Hadamard first compresses the audio file to remove any lingering intelligibility in the transform domain. The produced file is then encrypted twice. In the first phase, the diffusion operation is performed using a modified discrete Henon map to decrease the correlation between adjacent samples. The second stage uses a modified Lorenz hyper chaotic system for replacement operations to fill in the silences in the voice exchange. The method encrypts only wave files with a sample rate of 8000. The SNR ranges from −110 to −133, NSCR ranges from 99.9999 to 99.9989, and CC ranges from 0.0013 to 0.0009.

Hu et al. suggested a new homomorphic audio signal encryption method for secure cloud communication and processing [11]. The actual audio stream is encrypted without being transformed into binary to reduce the computational complexity. Adaptive parameters were developed to manage a range of audio formats and properties. Users can choose the appropriate encryption level to achieve a balance between security and complexity. For audio operations such as loudness control and editing, the method allowed additive and multiplicative homomorphism. The method encodes wave files with a sample rate of 16,000 and MP3 files of 256 sample rates only and achieves NSCR ranges from 99.58 to 99.62 at the first level; after the fourth level, it achieves 100%, key space is equal to 2^{279}, MSE is equal to 14.74×10^{17}, CC is equal to −0.0020, and SNR is equal to −148.12.

Roy et al. developed a system for audio encryption based on DNA encoding and chaos theory that is appropriate for real-time applications [12]. The bulk data in audio files may be too large for traditional encryption algorithms made for text data, which could result in sluggish processing times and higher storage needs. A pseudo-random bit sequence is generated using the Recursive Chaotic Map (RCM). The proposed method encrypts only wave files with a maximum size equal to 7.92 MB and achieves NSCR ranges from 97.20% to 99.71%, CC ranges from 0.0013 to 0.0008, and SNR ranges from −23.0069 to −24.8017.

Maity and Dhara [13] proposed a 2D Cosine Logistic Map (2DCLM) by combining the logistic map with the cosine map. The suggested 2DCLM performs admirably in chaotic situations. Since the Secure Hash Algorithm (SHA3-512) is used to calculate the provided signal's hash value, the suggested approach is sensitive to audio signals. The hash value is used to jumble the provided audio signal. Empirical Mode Decomposition (EMD) breaks down the jumbled signal; to minimize the temporal complexity of the EMD process, the data is first divided into a 2D signal. The stream produced by 2DCLM and the residue given by EMD are XORed to generate the encrypted signal. The proposed method encrypts only wave files with a maximum size is equal to 2.52 MB and achieves 99.9982% average NSCR, average CC 0.00018, and key space of 2^{512}.

Joshi and Gaffar [14] suggested a WORD-oriented method based on rotation and XOR operations for protecting digital audio recordings. The main ideas behind the encryption design algorithm are the Rotation-XOR (RX) operations, which entail XORing the plain audio samples with the previous audio samples after they have been left-rotated by the sum of their digits. A digital audio file is converted into a random (noise-like) audio file using the encryption algorithm that was created. The proposed method encrypts only wave files with a maximum size equal to 142.79 KB, a sample rate of 8192, a total sample of 73.113, and a maximum duration of 8.9249 s, resulting in a long encryption time of 379.4791 s, and achieves 100% NSCR, a key space of 2^{256}, and SNR ranges from −16.1304 to −34.0016.

Previous studies concentrated on chaotic systems such as Lorenz and Logistic maps, hyperchaotic maps, (EMD), and elliptic curve encryption to improve security and flexibility. However, challenges in addressing audio quality, handling real-time processing, and ensuring resilience against evolving cryptographic attacks still persist. This paper combines two chaotic maps: the Tent map for controlling the permutation of audio samples and the logistic map for generating encryption keys. This dual-map design enhances the key space and improves resistance to attacks for different types of audio files, including .WAV and .MP3 formats, ensuring compatibility with both uncompressed and compressed audio data. It effectively encrypts both mono (1-channel) and stereo (2-channel) audio files, making it suitable for a wide range of applications. Additionally, the encryption scheme is designed to handle large audio files equal to 122 MB, ensuring scalability for high-quality and lengthy recordings. It also supports audio files with varying sample rates, including 22,050, 44,100, 48,000, and 96,000 Hz, thereby accommodating different audio qualities.

3 Chaotic Maps

Chaotic maps are mathematical models that show predictable rules but produce unpredictable behavior. They are widely used in cryptography, secure communications, and nonlinear dynamic systems. These maps are highly sensitive to initial conditions, meaning that even slight changes in the starting values can yield significantly different results.

3.1 1D Logistic Map

The 1D logistic map is a simple yet widely studied mathematical model that exhibits chaotic behavior. It is a discrete-time dynamical system that describes the population growth of a species in a simplified manner. Eq. (1) defines the recursive equation of the 1D logistic map [1]:

$$x_{n+1} = f(x_n, r) = r \times x_n \times (1 - x_n) \tag{1}$$

where $r \in (0, 4]$, $x_n \in (0, 1)$, $n = 0, 1, 2, \ldots$

The interesting dynamics of the one-dimensional logistic map become apparent when the parameter r is changed. System behavior can range from stable periodic orbits to chaotic patterns. Systems can exhibit complex and unpredictable behavior. Fig. 1 shows the Bifurcation and Lyapunov of the 1D logistic.

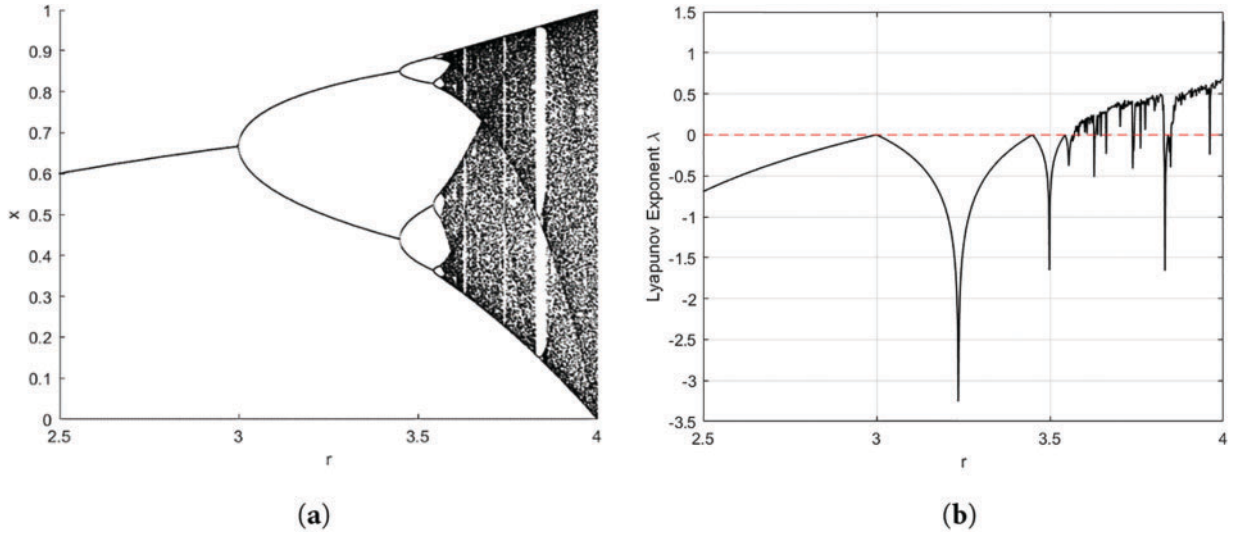

Figure 1: (**a**) Bifurcation of 1D logistic map, (**b**) 1D logistic map Lyapunov

3.2 Tent Chaotic Map

The real-valued function $f(\mu)$ is the Tent map with parameter μ. The unit interval $[0, 1]$ is mapped into itself by $f(\mu)$ for values of the parameter μ between 0 and 2, producing a discrete-time dynamical system on it or, conversely, a recurrence relation. Specifically, a series x_n is produced by iterating a point x_0 in $[0, 1]$ [1].

$$x_{n+1} = f(x_n, \mu) = \begin{cases} \mu \times (1 - x_n), & x_n >= 0.5 \\ \mu \times x_n, & Otherwise \end{cases} \tag{2}$$

where μ is a real positive number, $\mu \in [0, 2]$, $x_n \in [0, 1]$, $n = 0, 1, 2, \ldots$

4 The Proposed Audio Encryption Method

Fig. 2 shows the proposed encryption method safeguards audio data against unauthorized access by leveraging permutation and logistic map algorithms. Permutation ensures data shuffling, while logistic maps complicate the encryption process. The proposed audio encryption involves three processes. The first is permutation. The second is a key generation process that adopts the chaotic behavior of a new 1D logistic map to generate a key sequence. The third is the encryption of digital audio.

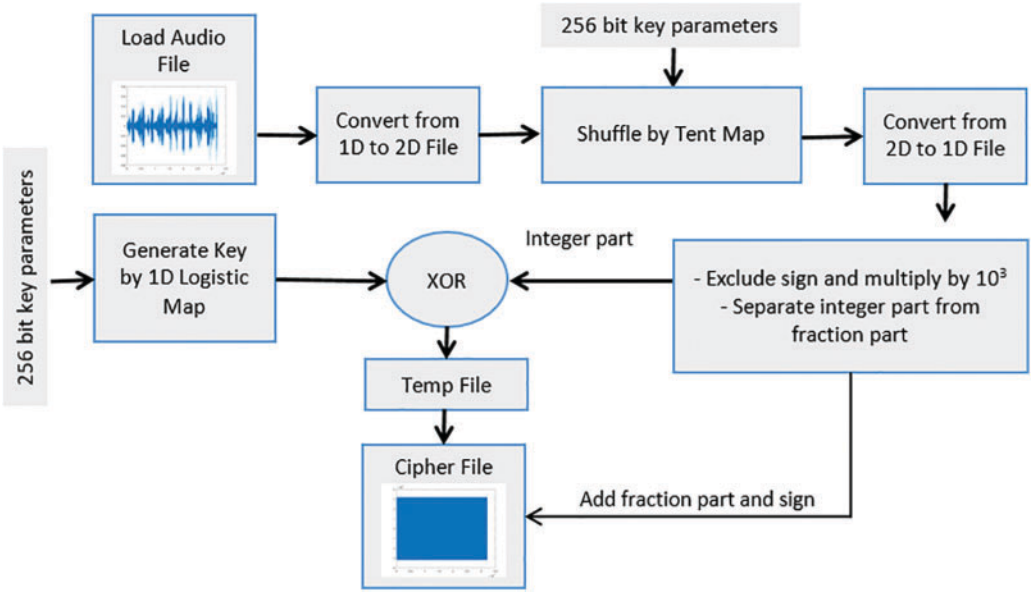

Figure 2: Proposed encryption method structure

4.1 Key Generation

A one-dimensional logistic map is used to generate the sequence of secret random keys as in Eq. (1). It is sensitive to the initial state and control parameters. A 256-bit secret key which (K_i) represents 64-bit hexadecimal, is used to produce the initial values. Eqs. (3)–(5) are used to initialize the parameters of the 1D logistic chaotic map.

$$x_0 = mod\left(\frac{k_1 k_2 \ldots k_{20}}{256}, 1\right) \tag{3}$$

$$x_1 = mod\left(\frac{k_{21} k_{22} \ldots k_{44}}{256}, 1\right) \tag{4}$$

$$x_2 = mod\left(\frac{k_{45} k_{46} \ldots k_{64}}{256}, 1\right) \tag{5}$$

In addition, r is obtained from the suggested Eq. (6):

$$r = 3.6 + y_0 \tag{6}$$

where (y_0) is obtained from Eq. (7):

$$y_0 = mod\left(\left(x_0 + x_1 + x_2\right), 1\right) \tag{7}$$

The sequence $Key_x = \{x_1, x_2, \ldots, x_{size}\}$, is generated using a 1D logistic map. After that, applying the new proposed Eq. (8) increases the randomness and the complexity of the 1D logistic map.

$$x_{n+1} = mod\left(\left\lfloor\left(|x_{n+1}| \times 10^{12}\right)\right\rfloor, size\right) \tag{8}$$

$$x_i = Transpose(x_{n+1}) \tag{9}$$

$\forall i, 1 \leq i \leq size$, Eq. (9) expresses the transpose operation to generate the secret key.

Fig. 3 shows a complementary view of the dynamics of the enhanced 1D logistic map. The enhancements to the logistic map may aim to improve stability, modify bifurcation points, and reduce the chaotic zone.

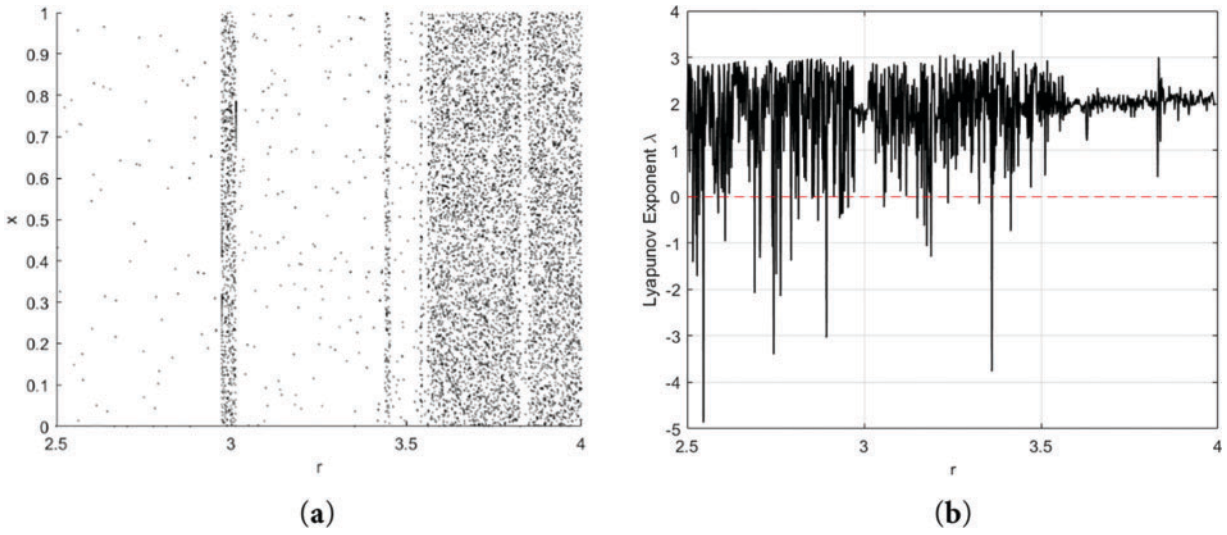

Figure 3: (**a**) Bifurcation of enhanced 1D logistic map, (**b**) The enhanced 1D logistic map Lyapunov

4.2 Permutation Mechanism

One approach to enhance the security of an audio file involves shuffling its data values using a secure permutation algorithm, introducing randomness and complexity. This operation rearranges the values in the file, effectively changing their positions according to a predetermined but random mapping. This process can be described mathematically as follows:

A one-dimensional Tent chaotic map is used to generate the new positions for shuffling the original file locations after being converted to 2D, as in Eq. (2). It is sensitive to the initial state and control parameters. A 256-bit secret key, which k_i represents 64-bit hexadecimal, and is used to produce the initial values. Eqs. (10)–(12) are used to initialize the parameters of the Tent map [1].

$$x_{00} = mod\left(\frac{k_1 k_2 \ldots k_{24}}{2^{46}}, 1\right) \tag{10}$$

$$x_{11} = mod\left(\frac{k_{25} k_{26} \ldots k_{44}}{2^{40}}, 1\right) \tag{11}$$

$$x_{22} = mod\left(\frac{k_{45} k_{46} \ldots k_{64}}{2^{40}}, 1\right) \tag{12}$$

In addition, (μ) is obtained from the suggested Eq. (13):

$$\mu = 1.9 + (y_{00}/100) \tag{13}$$

where (y_{00}) is obtained from Eq. (14), a new position is generated using Eq. (15), and then sorted as in Eq. (16) to find y_{new}:

$$y_{00} = mod\left((x_{00} + x_{11} + x_{22}), 1\right) \tag{14}$$

$$x_{new} = mod\left(\left\lfloor\left(|x_{n+1}| \times 10^{16}\right)\right\rfloor, numRows\right) \tag{15}$$

$$[y_{new}] = Sort\left(x_{new}\right) \tag{16}$$

After generating the new positions, convert the original file O_x into a 2D file and then shuffle positions as shown in Eq. (17):

$$Sh_x\left(i, j\right) = O_x(x_{new}(i), y_{new}(j)) \tag{17}$$

where $Sh_x\left(i, j\right)$ represents the value of the shuffled audio data at position $(i,\ j)$. $\forall i, j, 1 \leq i, j \leq numRows, numCols$. Fig. 4 depicts the proposed permutation mechanism.

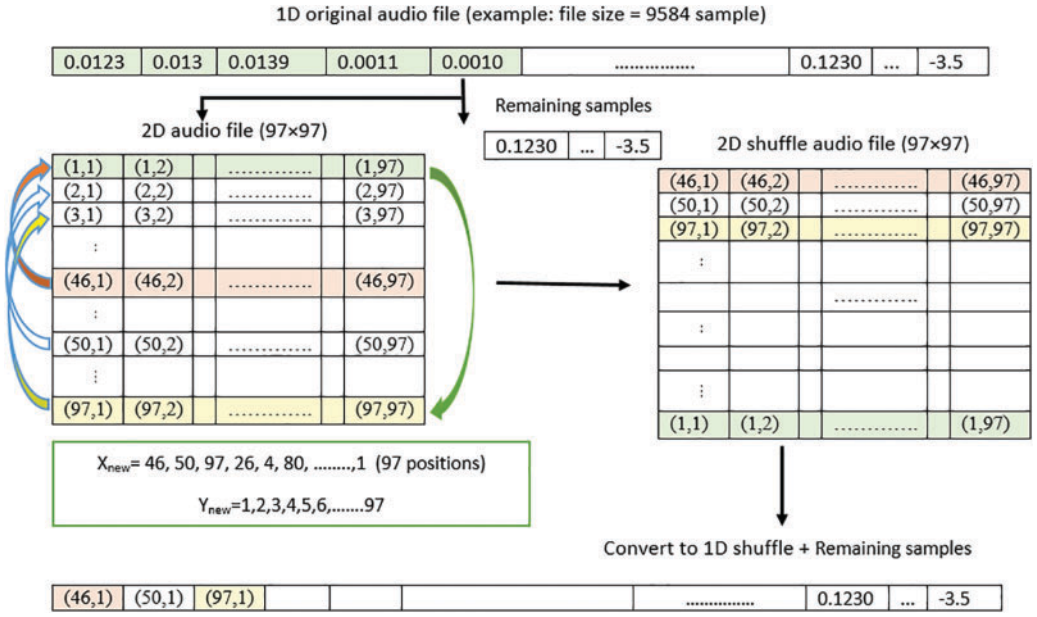

Figure 4: The suggested permutation mechanism

4.3 Audio Encryption Process

The proposed method must satisfy the two important cryptography properties: confusion and diffusion. The diffusion property is satisfied by shuffling the audio file using a random permutation function. The confusion property is satisfied by applying XOR operation between the key generated using Eq. (9) and the shuffled audio to generate the cipher audio using Eq. (22).

The encryption process by transforming the 1D audio file into a 2D matrix, and the Tent map is applied as per Eq. (17) to generate new positions for shuffling each value of the audio file. After the shuffling process, the audio file is converted back to its 1D form. Then, the shuffled wave values are analyzed to determine whether they are positive or negative, generating a binary array using Eq. (18), where values are assigned 1 if negative and 2 otherwise. To further enhance randomness, a new transformation equation is applied to the shuffled

wave by multiplying each value by 10^3, as defined in Eq. (19). This increases the distribution complexity, improving security. Next, the shuffled wave is converted into double form using Eq. (20), and then the integer and fractional parts are separated according to Eq. (21), For additional security, XOR operation is applied between a generated encryption key and the integer part of the shuffled wave producing the ciphered audio file. Finally, the fractional part and the sign information are included to generate the final encrypted audio file, ensuring a highly randomized and secure output. The steps of the proposed audio encryption technique are presented as follows:

Step 1: Load the original audio file and read the wave information $O_x(size)$;

Step 2: Convert a 1D original audio file into a 2D audio file $O(row, col)$ and save the remaining samples;

Step 3: Shuffle each value of the audio file using the new positions $Sh\,(x_{new}, y_{new})$ produced by Tent map. After that, convert the shuffled file to a 1D file $Sh(row \times col)$. Then add the remaining samples;

Step 4: Check each value of the shuffle audio if it is positive or negative, and generate an array to save the sign for each value using the following rule:

$$Check_{Sh} = f\,(x) = \begin{cases} 1, & Sh < 0 \\ 2, & otherwise \end{cases} \tag{18}$$

Step 5: Apply the new suggested equation to the shuffle file by multiplying by 10^3 to increase the randomness as follows:

$$Sh_i = \left(Abs(Sh_i) \times 10^3\right) \tag{19}$$

Step 6: Convert the shuffled audio file into double and extract the fractional component:

$$Sh_d = double\,(Int32\,(Sh_i)) \tag{20}$$

$$Sh_{fraction} = Sh_i - Sh_d \tag{21}$$

Step 7: Apply XOR between the key generated using Eq. (9) and the integer part of the shuffled file to generate the temporary cipher.

$$C_x = Key_x \bigoplus Sh_i \tag{22}$$

Step 8: Include the fraction part from Step 6 and the sign from Step 4 to generate the final Cipher audio file.

4.4 Audio Decryption Process

Fig. 5 shows that the decryption process is similar to the encryption process; however, the stages are handled in the reverse order. The decryption process begins with the ciphered audio file being loaded to reverse the encryption transformations. Then, the sign information that was previously extracted during encryption is excluded and stored separately. Additionally, the integer and fractional parts of the ciphered data are separated, mirroring the process used in Step 6 of encryption. This ensures that the data is properly structured for decryption. Moving to Step 3, a 1D logistic map is employed to generate the decryption key. The generated key is then XORed with the integer part of the ciphered file, effectively reversing the confusion step applied during encryption. Once the shuffled integer part is recovered, the previously separated fractional part is reintegrated into the data. To restore the original scale of the values, the result is then multiplied by 10^{-3} reversing the scaling applied in Step 5 of encryption.

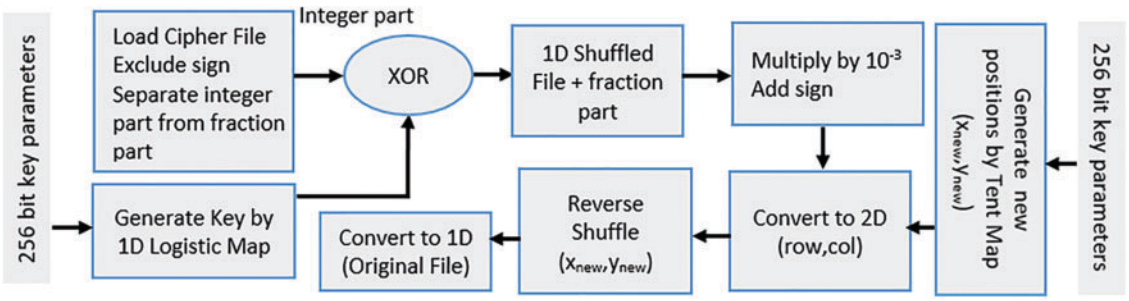

Figure 5: The proposed decryption method

Next, the sign information that was excluded in Step 2 is reapplied to the retrieved audio data, ensuring the correct polarity of the samples. Finally, the Tent map is utilized to regenerate the original positions used in the shuffling process. Using the same chaotic sequence applied in encryption, the shuffled file is restored to its correct sequence and then converted back into a 1D array, reversing the transformation that initially converted the 1D audio into a 2D matrix. At this stage, the original audio file is fully reconstructed and ready for playback or further processing. This decryption method ensures that the audio is restored with high accuracy while maintaining the integrity of the original data. The steps of the proposed audio decryption are clarified as follows:

Step 1: Load the cipher audio file;

Step 2: Exclude the sign and separate the integer part from the fraction part;

Step 3: Generate a key using a 1D logistic map;

Step 4: Apply XOR between the key generated and the integer part of the cipher to retrieve the shuffled file, then add the fraction part. Then, multiply the result by 10^{-3} and adding the sign excluded in Step 2;

Step 5: Generate the new positions (x_{new}, y_{new}) using the Tent map to reverse the shuffled file and then convert it to a 1D array to retrieve the original audio file.

5 Experimental Results

Different audio file samples in MP3 and WAV formats are used to evaluate the performance of the proposed method, including Windows wave files and samples from references [15–22], as reported in Table 1. The method was applied to an HP computer with a Core i7-10510U CPU 1.80–2.30 GHz, 16 GB RAM, and MATLAB R2020a.

Table 1: Audio file samples used to assess the performance of the proposed method

Audio file	Format	Source
Sample-1, Sample-2, Sample-5	WAV	https://getsamplefiles.com/sample-audio-files/wav/ (accessed on 06 July 2025) [15]
Sample1, Sample2, Sample4, Beethoven Symphony No. 6	WAV	https://toolsfairy.com/tools/audio-test/sample-wav-files (accessed on 06 July 2025) [16]

(Continued)

Table 1 (continued)

Audio file	Format	Source
Exotic Fiesta: Tropical Rhythm 132 bpm	WAV	https://samplefocus.com/samples/exotic-fiesta-tropical-rhythms (accessed on 06 July 2025) [17]
Synth Buzz Uprise 150 bpm	WAV	https://samplefocus.com/samples/synth-buzz-uprise (accessed on 06 July 2025) [18]
Rave Keys Synth Stabs 141 bpm	WAV	https://samplefocus.com/samples/rave-keys-synth-stabs (accessed on 06 July 2025) [19]
How Wet Guitar Passionate Loop 122 bpm	WAV	https://samplefocus.com/samples/how-wet-guitar-passionate-loop (accessed on 06 July 2025) [20]
Sample-15 s	MP3	https://samplelib.com/sample-mp3.html (accessed on 06 July 2025) [21]
Beethoven Symphony No. 6	MP3	https://www.mfiles.co.uk/mp3-downloads/beethoven-symphony6-1.mp3 (accessed on 06 July 2025) [22]

The performance of the suggested encryption technique is evaluated using PSNR and MSE metrics. The difference between the plain and encrypted audio is measured by MSE, as in Eq. (23) [7,9]:

$$MSE\left(I,C\right)=\frac{1}{\mathrm{N}}\sum_{i=0}^{\mathrm{N}-1}\left(I\left(i\right)-C\left(i\right)\right)^{2} \tag{23}$$

where $I\left(i\right)$ is the plain audio value, $C\left(i\right)$ is a cipher audio value, and N is the size of the audio file.

The mathematical demonstration of $PSNR$ is defined as in Eq. (24) [13,14]:

$$PSNR=20\log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{24}$$

NSCR is used to show the effect of changing one value in the original audio on the cipher audio as in Eq. (25) [4–6]:

$$NSCR\left(I,C\right)=\frac{\sum_{i=1}^{M}D(i)}{size}\times 100 \tag{25}$$

$$D\left(i\right)=\begin{cases}0, & I_i\neq C_i\\ 1, & otherwise\end{cases} \tag{26}$$

where $size$ is the audio size, I and C are the original audio and the encrypted one.

SNR is used to quantify the quality of the encoded signal as well as the remaining clarity of it. The encoded signal often has a lower SNR value, suggesting a higher level of noise. Eq. (27) is used to compute the SNR [1,3,8,11]:

$$SNR = 10\log_{10}\left(\frac{\sum_{i=1}^{N}(I(i))^2}{\sum_{i=1}^{N}(I(i)-C(i))^2}\right) \tag{27}$$

The correlation is counted in Eq. (28), which calculates the relationship of the close neighboring values [12]:

$$CC_{xy} = \frac{cov(x,y)}{\sqrt{D(x)\,D(y)}} \tag{28}$$

where,

$$E(x) = \frac{1}{S}\sum_{i=1}^{S} x_i \tag{29}$$

$$D(x) = \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x_i))^2 \tag{30}$$

$$cov(x,y) = \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x_i))(y_i - E(y_i)) \tag{31}$$

$D(x)$ is the variance, x and y are adjacent values, and 1000 is the number of adjacent values specified for the audio file.

PRD calculates how much the encrypted voice signal deviates from the original [5]. The PRD values for the original and encrypted voice signals for various audio signals were computed. The encrypted signal has been found to differ significantly from the original signal. The PRD is defined as in Eq. (32):

$$PRD = 100 \times \sqrt{\frac{\sum_{i=1}^{N}(I(i)-C(i))^2}{\sum_{i=1}^{N}(I(i))^2}} \tag{32}$$

5.1 The Proposed Method Results

Fig. 6 depicts encrypted and decrypted wave files. The figure shows that the proposed method provides good encryption because the cipher exhibits no distinct patterns and appears fully random.

5.2 Results of Mono Audio Files

Analyzing the encrypted mono audio files, as shown in Table 2, provides insights into their security and distortion levels after encryption. The dataset consists of 10 different alarm sound files, each with varying file sizes and total samples, but all maintaining the same sample rate of 22,050 Hz.

MSE values range from 2.2805E+09 (for Alarm02) to 1.1048E+10 (for Alarm05). A higher MSE indicates greater distortion between the original and the encrypted file, ensuring robustness against statistical attacks. The highest distortion (Alarm05, 1.1048E+10) suggests strong encryption, while the lowest distortion (Alarm02, 2.2805E+09) still provides effective security.

SNR values are consistently negative, ranging between −119.2771 dB (Alarm02) to −131.3678 dB (Alarm07). A highly negative SNR implies that the encrypted audio signal differs significantly from the original, making it indistinguishable and thus highly secure. The most distorted wave file (Alarm07, −131.3678 dB) demonstrates the highest level of encryption randomness.

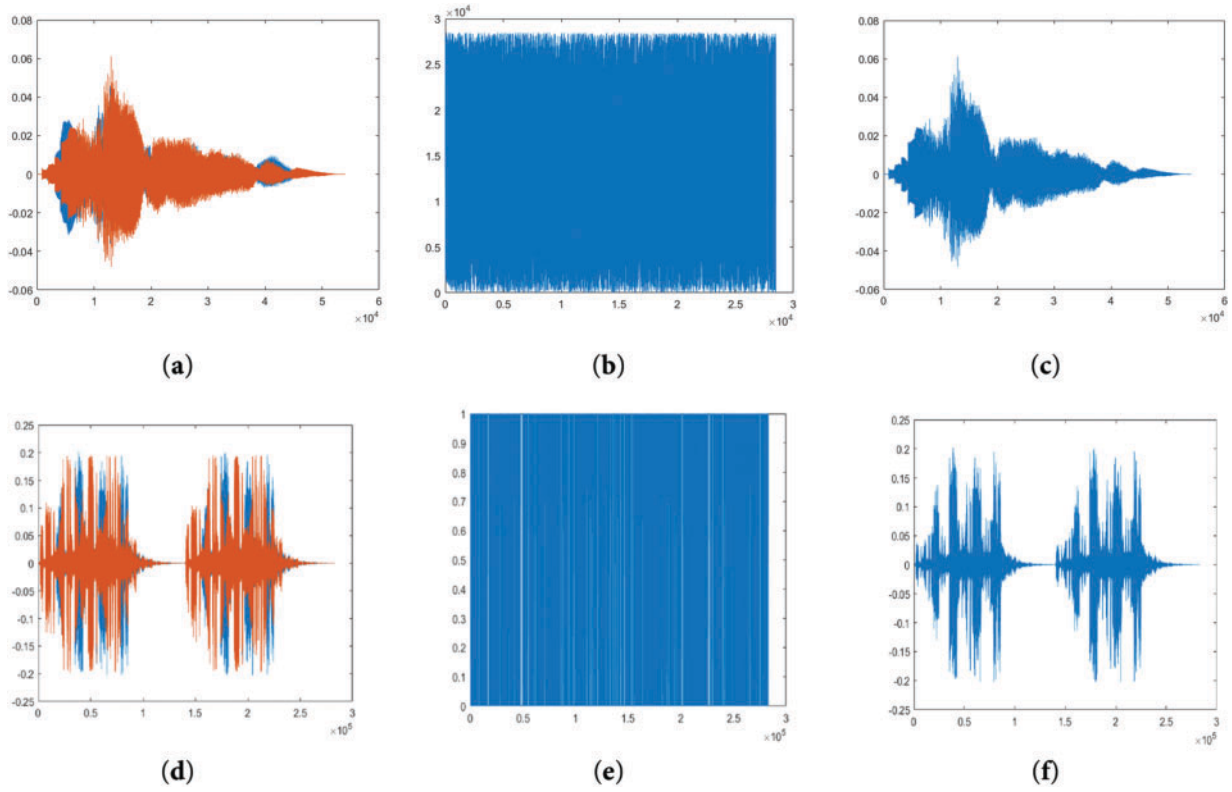

Figure 6: (**a**) chimes plain wave, (**b**) chimes encryption wave, and (**c**) chimes decryption wave, (**d**) ring05 plain wave, (**e**) ring05 encryption wave, and (**f**) ring05 decryption wave by the proposed method

Table 2: Results of the proposed method for 1-channel audio files

File name (1 Channel)	File size (KB)	Total samples	Sample rate	MSE	SNR	PSNR	NSCR
Alarm02	324	82,652	22,050	2.2805E+09	−119.2771	4.7647	100%
Alarm10	328	83,823	22,050	2.3366E+09	−120.0673	4.7814	100%
Alarm03	348	88,848	22,050	2.6378E+09	−123.0783	4.7605	100%
Alarm04	400	102,201	22,050	3.4888E+09	−127.3936	4.7624	100%
Alarm09	473	120,908	22,050	4.8752E+09	−122.9661	4.7692	100%
Alarm01	480	122,868	22,050	5.0430E+09	−121.9327	4.7619	100%
Alarm08	494	126,398	22,050	5.3374E+09	−128.6850	4.7615	100%
Alarm07	548	140,168	22,050	6.5341E+09	−131.3678	4.7811	100%
Alarm06	587	149,976	22,050	7.5080E+09	−125.8276	4.7652	100%
Alarm05	712	182,050	22,050	1.1048E+10	−129.5962	4.7711	100%

PSNR values remain around 4.76 dB, with minor variations. A low PSNR indicates that the encryption process has completely changed the signal characteristics, making reconstruction without the correct decryption process impossible.

NSCR value is 100% for all files, meaning every sample in the audio has been altered after encryption. This high NSCR confirms that the encryption is not localized but affects the entire signal, ensuring high security and randomness in the transformed data.

The encryption scheme effectively randomizes the audio data, achieving high distortion (high MSE, low PSNR), low correlation (negative SNR), and complete transformation (100% NSCR). These factors make the encrypted files highly secure and resistant to statistical and cryptographic attacks.

5.3 Results of Stereo Audio Files

Key insights into how various audio characteristics, such as file size, total samples, and sample rate, affect encryption performance can be gained by examining the encrypted stereo audio files displayed in Table 3. The dataset includes a variety of audio types, from short sound effects (e.g., "ding," "chord") to large music samples (e.g., "Symphony No. 6").

Table 3: Results of the proposed method for 2-channel audio files

File name (2 Channels)	File size	Total samples	Sample rate	MSE	SNR	PSNR	NSCR
Ding	68.4 KB	17,504	44,100	1.0287E+08	−129.268	4.7397	100%
Chord	111 KB	28,480	44,100	2.6953E+08	−116.162	4.7848	100%
Camera shutter	220 KB	37,460	96,000	4.6659E+08	−118.644	4.7820	100%
Chimes	211 KB	54,080	44,100	9.7354E+08	−132.272	4.7772	100%
Ring06	298 KB	76,401	22,050	1.9433E+09	−121.211	4.7766	100%
Ring02	307 KB	78,568	22,050	2.0623E+09	−124.154	4.7615	100%
Ring01	486 KB	124,556	22,050	5.1774E+09	−127.011	4.7662	100%
Ring04	682 KB	174,636	22,050	1.0169E+10	−129.557	4.7700	100%
Ring03	807 KB	206,668	22,050	1.4224E+10	−134.099	4.7752	100%
Ring05	1.08 MB	283,136	22,050	2.6716E+10	−135.434	4.7721	100%
Exotic Fiesta Tropical Rhythms	1.21 MB	317,940	44,100	3.3666E+10	−122.148	4.7750	100%
Synth Buzz Uprise	2.15 MB	564,480	44,100	1.4141E+08	−123.010	4.7658	100%
Rave Keys Synth Stabs	2.29 MB	600,510	44,100	1.2027E+11	−123.399	4.7689	100%
How Wet Guitar Passionate Loop	5.29 MB	1,388,291	44,100	6.4153E+11	−143.910	4.7775	100%
Sample-5	8.37 MB	2,194,286	48,000	1.6071E+12	−137.226	4.7655	100%
Sample-2	16.5 MB	4,350,378	48,000	6.3067E+12	−154.428	4.7725	100%
Sample-1	17.5 MB	4,350,378	48,000	7.0849E+12	−148.595	4.7729	100%
Sample1	20.5 MB	5,298,637	44,100	9.6661E+12	−142.438	4.7701	100%
Sample2	36.57 MB	9,585,510	44,100	3.0610E+13	−145.520	4.7736	100%
Sample4	41.13 MB	10,780,810	44,100	3.8805E+13	−150.504	4.7641	100%
BeethovenSymphony No. 6.wav	122.48 MB	32,108,544	44,100	3.4361E+14	−169.478	4.7717	100%
Sample-15 s.mp3	300 KB	869,019	44,100	2.3950E+11	−135.567	4.7681	100%
BeethovenSymphony No. 6.mp3	11.1 MB	32,108,543	44,100	3.4393E+14	−169.924	4.7681	100%

MSE values range from 1.0287E+08 (ding) to 3.4393E+14 (Symphony No. 6). Better encryption effects are indicated by larger files, which typically have higher MSE values. Small audio files like "ding" (1.0287E+08) and "chord" (2.6953E+08) still exhibit significant distortion, ensuring security even for shorter audio samples.

SNR values are consistently negative, ranging from −116.1623 dB (chord) to −169.924 dB (Symphony No. 6). Greater obfuscation of the original signal is indicated by a more negative SNR value, which makes

recovery practically impossible without the decryption key. The large files (sample-file-1, Symphony No. 6) exhibit the lowest SNR values, confirming high encryption strength.

PSNR values remain 4.73 dB, with small variation. Low PSNR prevents unauthorized reconstruction by indicating that the encrypted audio differs greatly from the original.

NSCR is 100% for all files, confirming that every sample in the audio signal has been altered after encryption. This ensures that encryption fully transforms the original audio form, preventing statistical attacks.

Audio files were encrypted across different sample rates (22,050, 44,100, 48,000, and 96,000 Hz). Higher sample rate files (e.g., 48,000 Hz) show larger MSE values because more data points are modified. The encryption scheme maintains uniform security properties across all sample rates.

Both small and big stereo audio files can be successfully transformed using the encryption approach, which guarantees complete transformation (100% NSCR), high randomness, and strong security (high MSE and low SNR). These results confirm that the encrypted audio files are highly resistant to statistical, cryptographic, and reconstruction attacks across various formats, file sizes, and sample rates.

5.4 Key Space and Key Sensitivity Analysis

The strength of the suggested approach to withstand the brute attack is demonstrated by the key space size. In the proposed method, the secret key consists of the logistic map parameters: x_0, x_1, x_2, r and y_0, and the Tent map parameters x_{00}, x_{11}, x_{22}, y_{00} and μ. Hence, the key space of the proposed method is $(10^{12} \approx 2^{40}) \times (10^{3} \approx 2^{10}) \times 2^{256} \times (10^{16} \approx 2^{54}) \times 2^{256} = 2^{616}$, which seems to be sufficient for countering the brute force attack.

A good encryption technique should react to even a small alteration in the secret key. By changing a single bit in the secret key and making a small adjustment to the chaotic map parameters, the suggested approaches' key sensitivity may be seen. Only one parameter is changed at a time in this experiment. However, some of the secret keys are altered. Suppose the secret key K and the chaotic parameters $y_0 = 0.39060$, $r = 3.9906250$, and $x_2 = 0.35156250$ are used to encrypt the original wave by the proposed method. Fig. 7b qualitatively depicts the decryption of the encrypted wave when y is changed to $y' = 0.39061$. Also, Type equation here is changed to $r' = 3.9906251$ and the decryption of the encrypted wave is shown in Fig. 7c. Finally, x_2 is changed to $x_2' = 0.35156251$ and the decryption process is shown in Fig. 7d. It is evident from the figure that when a small alteration is made to the secret key, the decryption procedure is always unable to reconstruct the original audio file. This shows how sensitive the suggested technique is to the secret key.

5.5 Histogram Analysis

Fig. 8 depicts the histogram of different audio files. From the figure, one can notice that the histogram is nearly uniform across the range. The height of the bars is consistent, indicating that the values in the encrypted data are evenly distributed. This histogram suggests that the logistic map, combined with the Tent map encryption process, has effectively transformed the original audio into a pseudo-random sequence, where all possible values are equally probable.

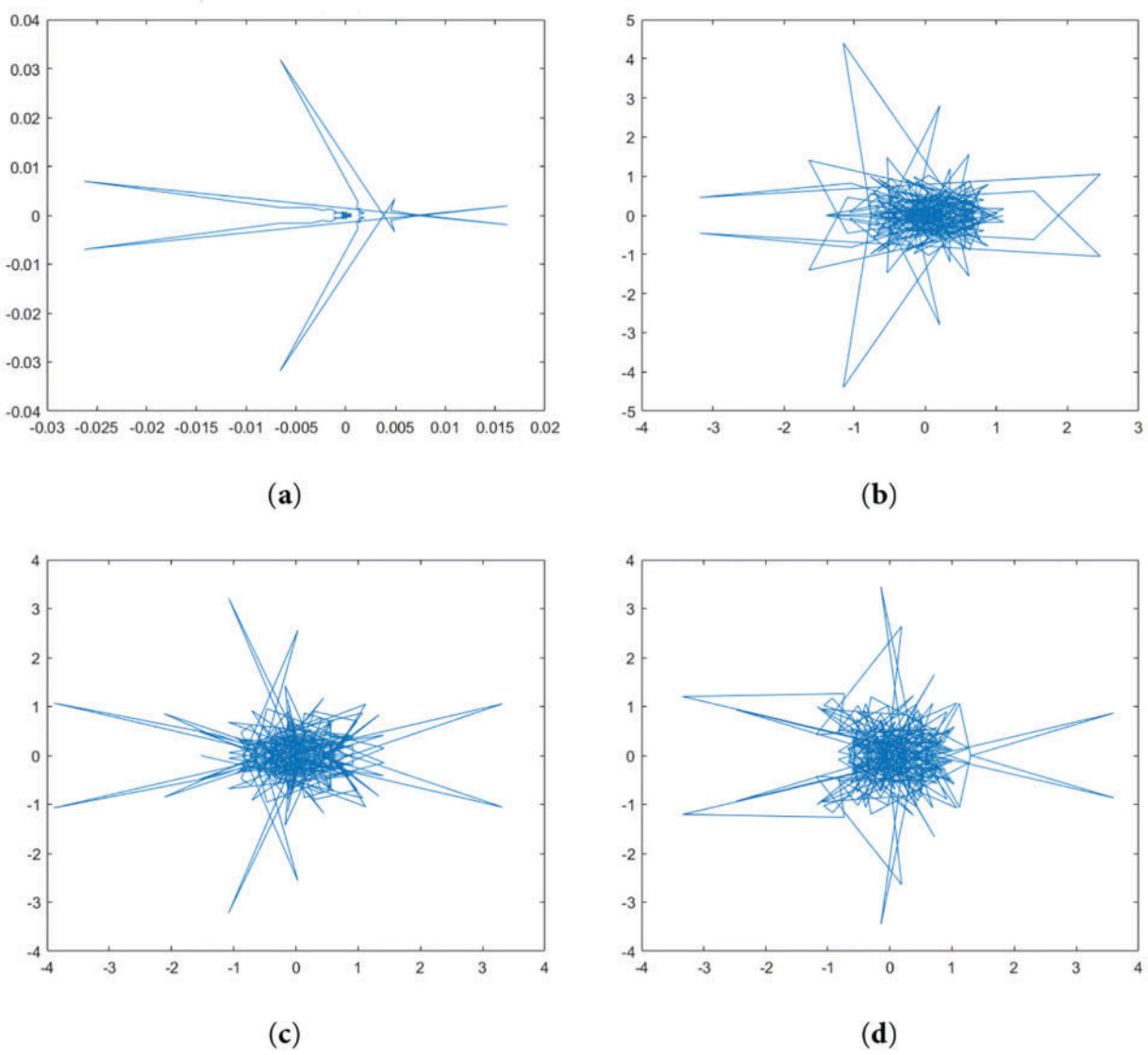

Figure 7: Key sensitivity results for the proposed method. (**a**) Fourier transform of plain wave. (**b**) The decrypted wave with y'. (**c**) The decrypted wave with r'. (**d**) The decrypted wave with $x_2{}'$

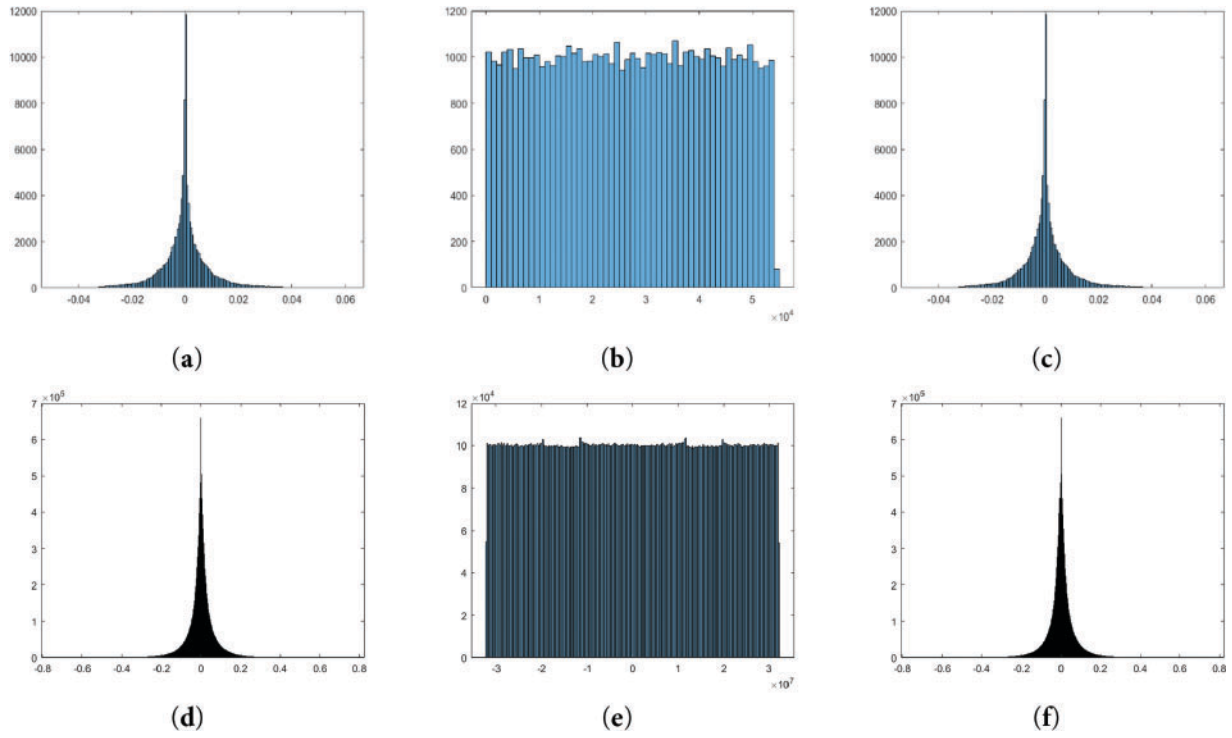

Figure 8: (**a**) Chimes plain histogram, (**b**) Chimes cipher histogram, (**c**) Chimes decrypt histogram, (**d**) Symphony plain histogram, (**e**) Symphony cipher histogram, and (**f**) Symphony decrypt histogram

5.6 PRD, Cross Correlation, and Time Analysis

Table 4 provides results for various audio files analyzed regarding File Size, RRD, CC, and Time. PRD is an important measure in audio transformation or encryption. It evaluates how much peak distortion is introduced between the original and the encrypted/transformed audio files.

Table 4: The PRD, CC, and encryption time for the audio files

Wave name	File size	PRD	CC	Time (s)
Ding	68.4 KB	2.9067E+08	−0.0069	0.2188
Chord	111 KB	6.4286E+07	−0.0027	0.1468
Camerashutter	220 KB	8.5554E+07	0.0057	0.1374
Chimes	211 KB	4.1077E+08	0.0016	0.1576
Ring06	298 KB	1.1497E+08	0.0018	0.1974
Ring02	307 KB	1.6132E+08	0.0002	0.1981
Ring01	486 KB	2.2415E+08	−0.0027	0.3297
Ring04	682 KB	3.0050E+08	0.00001	0.3564
Ring03	807 KB	5.0695E+08	−0.0014	0.5101
Ring05	1.08 MB	5.9118E+08	0.0012	0.6301
Exotic Fiesta Tropical Rhythms	1.21 MB	1.2806E+08	0.0009	0.6838
Synth Buzz Uprise	2.15 MB	1.4141E+08	0.0002	1.1373

(Continued)

Table 4 (continued)

Wave name	File size	PRD	CC	Time (s)
Rave Keys Synth Stabs	2.29 MB	1.4790E+08	−0.0004	1.2884
How Wet Guitar Passionate Loop	5.29 MB	1.5685E+09	−0.0001	2.9843
Sample-5	8.37 MB	7.2664E+08	−0.0006	5.4659
Sample-2	16.5 MB	5.2651E+09	−0.0004	12.7837
Sample-1	17.5 MB	2.6901E+09	0.0003	13.9662
Sample1	20.5 MB	1.3240E+09	−0.0004	15.778
Sample2	36.57 MB	1.8880E+09	−0.0003	36.2380
Sample4	41.13 MB	3.3512E+09	0.0002	42.5934
Beethoven Symphony No. 6.wav	122.48 MB	2.9779E+10	−0.0002	197.9206
Sample-15 s.mp3	300 KB	6.0027E+08	0.0015	1.5555
Beethoven Symphony No. 6.mp3	11.1 MB	1.7196e+16	−0.0001	225.06

PRD values range from 2.9067E+08 (e.g., “ding”) to 1.7196e+16 (e.g., “Symphony No. 6”). The very high positive PRD values indicate substantial transformation or distortion, reflecting that these files undergo a more significant change during the encryption/transformation process.

The CC (measures the similarity between the transformed/encrypted file and the original. A high CC means that the transformation or encryption has not significantly altered the audio signal, while a low CC indicates greater alteration. CC values reached 0.00001 (e.g., “ring04”), indicating minimal correlation and stronger encryption/transformation, which makes it more difficult to recognize the original file from the transformed one.

The processing time ranges from 0.1374 s (e.g., “camerashutter”) to 197.9206 s (e.g., “Symphony No. 6”). Smaller files, such as “ding” and “chord”, have relatively short processing times (e.g., 0.1468 s for “chord”). Larger files like “Symphony No. 6” with 197.9206 s take significantly longer to process due to their size and the complexity of encryption/transformation applied to the data.

The analysis shows that the encryption or transformation process is more impactful on larger files, with higher distortion (PRD) and lower similarity to the original file (CC). Processing time scales with file size; small files take less time to execute, while larger files require longer processing time, which indicates that the method is fast at encoding audio data. The method is efficient and scalable, with strong transformations for files of varying sizes.

5.7 NIST Test

To determine randomness, we applied the SP800-22 test criteria. The components of the SP800-22 test include 15 important tests. When the p-value is greater than 0.01, the test for randomness is successful; otherwise, the time series is not random [13]. The NIST test results for the proposed encryption method are clarified in Table 5 as follows:

Table 5: The NIST test of the proposed encryption method

Test	*p*-Value	Result
Frequency (Monobit) Test	0.911413	Pass
Frequency Test within a Block	0.804337	Pass
Runs Test	0.991468	Pass
Test for the Longest Run of Ones in a Block	0.77276	Pass
Binary Matrix Rank Test	0.350485	Pass
Fast Fourier Transform (FFT) test	0.299251	Pass
Non-overlapping Template Matching Test	0.976060	Pass
Overlapping Template Matching Test	0.949602	Pass
Maurer's "Universal Statistical" Test	0.297152	Pass
Linear complexity test	0.804337	Pass
Serial test	*p*-value 1: 0.378138 *p*-value 2: 0.637119	Pass
Approximate entropy test	0.739918	Pass
Cumulative Sums (Cusum) Test	*p*-value Forward: 0.602458 *p*-value Reverse: 0.804337	Pass
Random excursions test	0.149958	Pass
Random excursions variant test	0.280367	Pass

5.8 Noise Attack Analysis

To demonstrate the resistance of the proposed audio encryption method against noise attack, the proposed method is subjected to two types of noise, including white Gaussian noise and salt and pepper noise, with two densities: 0.25 and 0.50. Table 6 lists the Bit Error Rate (BER) between the original and retrieved noisy audio. The results show that smaller audio files consistently show higher BER values while larger audio files show lower BER values. This means that more complex audio files are more resilient to impulsive noise.

Table 6: BER under a white Gaussian noise, two densities of salt and pepper noise

File name	File size	Salt and pepper noise		Gaussian noise
		0.25	0.50	Awgn
Ding	68.4 KB	0.6737	0.6160	0.5029
Chord	111 KB	0.6700	0.6109	0.5025
Chimes	211 KB	0.6616	0.6074	0.4999
Camerashutter	220 KB	0.6471	0.5980	0.5012
How Wet Guitar Passionate Loop	5.29 MB	0.6017	0.5677	0.6357
Sample-5	8.37 MB	0.5403	0.5270	0.5000
Sample-2	16.5 MB	0.5797	0.5531	0.6062
Beethoven Symphony No. 6.wav	122 MB	0.5824	0.5550	0.6099
Sample-15 s.mp3	300 KB	0.5596	0.5397	0.5795
Beethoven Symphony No. 6.mp3	11.1 MB	0.5825	0.5549	0.6099

5.9 *Comparison with Other Works*

Table 7 compares the proposed audio encryption method with other existing methods, concentrating on key space, MSE, PSNR, SNR, NSCR, and CC to demonstrate the effectiveness, security, and efficiency of the proposed method. The proposed method stands out in terms of security and structural preservation, outperforming the other works. It encrypts a large file size of 122.48 MB for .WAV (32,108,544 total samples and 12:08 min duration) and 11.1 MB for .MP3, achieving low PSNR and negative SNR, which is less than that of other works. Furthermore, the key space of 2^{616}, and NSCR of 100% higher than other works, indicate that the proposed method is very effective in preserving the original structure of the signal while introducing a significant amount of distortion and noise, making it more resistant to attacks.

Table 7: Comparison with existing works

Ref.	File size	Key space	MSE	PSNR	SNR	NSCR	CC
[7]	2.66 MB	2^{240}	42 dB	5.1042	−21.7739	99.6170	−0.0023
[8]	1.30 MB	2^{159}	–	–	–	99.6076	0.0029
[9]	31.25 MB	–	3.26×10^4	10.8908	–	99.9990	0.0017
[10]	–	2^{548}	–	–	−133	99.9989	0.0009
[11]	–	2^{279}	After 4th level 14.74×10^{17}	−102.80	−148.12	99.6256 4th level 100%	−0.0020
[12]	7.92 MB	–	–	–	−24.0901	98.3800	0.0039
[13]	2.52 MB	2^{512}	–	–	–	99.9982	0.0001
[14]	142.7 KB	2^{256}	–	–	−16.1302	100%	−0.0065
Proposed	122.48 MB	2^{616}	3.4361×10^{14} 145 dB	4.7717	−169.4781	1st level 100%	−0.0006
method	11.1 MB MP3		3.4393×10^{14}	4.7681	−169.9240	100%	−0.0001

A negative CC value confirms the strong encryption, making the original signal highly unrecognizable after encryption. Compared to other methods, the proposed method strikes a balance of strong encryption with minimal distortion, preserving the structural content, making it an efficient and secure approach to audio file encryption or transformation.

6 Conclusions

This paper proposes an audio encryption scheme, leveraging a combination of the Tent chaotic map for permutation and a logistic chaotic map for key generation. The encryption process first involves shuffling the audio samples using the Tent map, then applying an XOR operation between the shuffled audio and a chaotic key stream generated by the logistic map. This dual-chaotic mechanism ensures high security and computational efficiency and is suitable for real-time audio applications. Extensive experiments were conducted using large audio files in both mono and stereo formats, including .WAV and .MP3 files up to 122 and 11 MB, respectively. The proposed method demonstrated excellent encryption performance, achieving high MSE, low SNR, 100% of NSCR, high PRD, and low correlation coefficients, all indicative of strong encryption strength and minimal residual information. The proposed method also showed exceptional sensitivity to small changes in the secret key and a large key space of 2^{616}, making brute-force attacks computationally infeasible. Security analysis further confirmed the robustness of the proposed approach against common cryptographic attacks, including brute-force, statistical, and differential attacks. Compared to existing audio encryption techniques, the proposed method offers superior performance in both encryption strength and computational efficiency. Generally, the proposed chaotic-based encryption scheme presents a viable solution for secure audio data transmission, especially in environments where low-latency and high-security

requirements are critical. Future work can be focused on optimizing the method for real-time streaming applications and exploring its integration with hardware-based audio processing systems.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: Ibtisam A. Taqi and Sarab M. Hameed; methodology: Ibtisam A. Taqi; software: Ibtisam A. Taqi; data collection: Ibtisam A. Taqi; analysis and interpretation of results: Ibtisam A. Taqi and Sarab M. Hameed; draft manuscript preparation: Ibtisam A. Taqi and Sarab M. Hameed. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All datasets and materials are publicly available.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Sathiyamurthi P, Ramakrishnan S. Speech encryption using chaotic shift keying for secured speech communication. EURASIP J Audio Speech Music Process. 2017;2017(1):1–11. doi:10.1186/s13636-017-0118-0.
2. Hameed Y, Ali N. An efficient audio encryption based on chaotic logistic map with 3D matrix. J Theor Appl Inf Technol. 2018;96(16):5142–52.
3. Hato E, Shihab D. Lorenz and Rossler chaotic system for speech signal encryption. Int J Comput Appl. 2015;128(11):25–33.
4. Wang X, Su Y. An audio encryption algorithm based on DNA coding and chaotic system. IEEE Access. 2019;8:9260–70. doi:10.1109/access.2019.2963329.
5. Sheela S, Suresh K, Tandur D. A novel audio cryptosystem using chaotic maps and DNA encoding. J Comput Netw Commun. 2017;2017(1):2721910. doi:10.1155/2017/2721910.
6. Kordov K. A novel audio encryption algorithm with permutation-substitution architecture. Electronics. 2019;8(5):530. doi:10.3390/electronics8050530.
7. Albahrani EA, Alshekly TK, Lafta SH. New secure and efficient substitution and permutation method for audio encryption algorithm. J Supercomput. 2023;79(15):16616–46. doi:10.1007/s11227-023-05249-5.
8. Demirtaş M. A lossless audio encryption method based on Chebyshev map. Orclever Proc Res Dev. 2023;2(1):28–38. doi:10.56038/oprd.v2i1.234.
9. Ur Rehman H. An efficient audio encryption scheme based on elliptic curve over finite fields. Mathematics. 2023;11(18):3824. doi:10.3390/math11183824.
10. Farsana F, Devi V, Gopakumar K. An audio encryption scheme based on Fast Walsh Hadamard Transform and mixed chaotic keystreams. Appl Comput Inform. 2023;19(3/4):239–64. doi:10.1016/j.aci.2019.10.001.
11. Hu Y, Zhang Q, Zhang Q, Ba Y An intelligent homomorphic audio signal encryption algorithm for secure interacting. Multimed Tools Appl. 2024;83(9):25675–93. doi:10.1007/s11042-023-16493-5.
12. Roy M, Chakraborty S, Mali K. Audio encryption framework based on chaotic map and DNA encoding. Appl Acoust. 2024;224:110152. doi:10.1016/j.apacoust.2024.110152.
13. Maity A, Dhara BC. An audio encryption scheme based on empirical mode decomposition and 2D cosine logistic map. IEEE Lat Am Trans. 2024;22(4):267–75. doi:10.1109/tla.2024.10472959.
14. Joshi AB, Gaffar A. A technique for securing digital audio files based on rotation and XOR operations. Soft Comput. 2024;28(6):5523–40. doi:10.1007/s00500-023-09349-5.
15. Sample WAV Audio File Download [Internet]. [cited 2025 Jul 6]. Available from: https://getsamplefiles.com/sample-audio-files/wav/.

16. Sample WAV Files [Internet]. [cited 2025 Jul 6]. Available from: https://toolsfairy.com/tools/audio-test/sample-wav-files.
17. Sample Focus. Exotic Fiesta Tropical Rhythms [Internet]. [cited 2025 Jul 6]. Available from: https://samplefocus.com/samples/exotic-fiesta-tropical-rhythms.
18. Sample Focus. Synth Buzz Uprise [Internet]. [cited 2025 Jul 6]. Available from: https://samplefocus.com/samples/synth-buzz-uprise.
19. Sample Focus. Rave Keys Synthstabs [Internet]. [cited 2025 Jul 6]. Available from: https://samplefocus.com/samples/rave-keys-synth-stabs.
20. Sample Focus. How Wet Guitar Passionate Loop [Internet]. [cited 2025 Jul 6]. Available from: https://samplefocus.com/samples/how-wet-guitar-passionate-loop.
21. Sample MP3 audio files. [cited 2025 Jul 6]. Available from: https://samplelib.com/sample-mp3.html.
22. Beethoven LV. Classical MP3 Files [Internet]. [cited 2025 Jul 6]. Available from: https://www.mfiles.co.uk/mp3-downloads/beethoven-symphony6-1.mp3.