applied sciences

MDPI

Article

Novel Implementation of Audio Encryption Using Pseudorandom Byte Generator

Borislav Stoyanov * and Tsvetelina Ivanova

Department of Computer Informatics, Faculty of Mathematics and Informatics, Konstantin Preslavsky University of Shumen, 9712 Shumen, Bulgaria; ts.r.ivanova@shu.bg

* Correspondence: borislav.stoyanov@shu.bg

Abstract: In this paper, we present an algorithm for encrypting audio files based on the Ikeda map, a mathematical function of chaos theory. Detailed experimental, security and theoretical analysis is provided on the proposed algorithm using histogram analysis, using different measurements including the signal-to-noise ratio, the peak signal-to-noise ratio, the number of samples change rate and the correlation coefficient. The provided results show a highly secure and strong algorithm against different types of attacks.

Keywords: audio encryption; chaotic maps; security analysis; Ikeda map

check for updates

Citation: Stoyanov, B.; Ivanova, T. Novel Implementation of Audio Encryption Using Pseudorandom Byte Generator. *Appl. Sci.* **2021**, *11*, 10190. https://doi.org/10.3390/app112110190

Academic Editor: Arcangelo Castiglione

Received: 15 September 2021
Accepted: 28 October 2021
Published: 30 October 2021

1. Introduction

The fast revolution in communication we are currently experiencing needs faster and more secure means of ensuring the security of audio messages. Nowadays, when everybody has smart-phone in their pocket and at any time has the opportunity to send a different message such as a text or audio message, it is very important to have better options to be more protected. Such fast-paced development is a big challenge when there are many different ways to attack audio data. Audio encryption has applications in online networks, interactive media and healthcare audio systems. Audio files are often larger than image files and common encryption methods such as RSA [1], AES (Rijndael) [2] and Blowfish [3] may not be directly applicable to audio files. The above algorithms require more time and resources when the file is bigger, which makes them inapplicable for audio encryption.

In this paper, our focus is on the encryption of wave audio files. Audio encryption refers to the coding of user data in an audio file. The waveform audio file enjoys the benefits of being a famous uncompressed audio file format, developed by IBM and Microsoft. It is a variant of the resource interchange file format (RIFF) and shares its container-based, tagged file structure. The wave file stores the audio recordings with different sampling rates and bitrates in a single container which consists of two sub-containers: the header part and actual sample data part. By using logical XOR, byte audio samples are encrypted with pseudorandom sequences.

Because of the resistance of the growing number of statistical attack methods, the use of chaotic maps in encryption schemes is becoming increasingly popular. The chaotic dynamical system includes deterministic systems that exhibit random-like, unpredictable behaviour as a result of their sensitivity depending on the starting values. In Reference [4], a novel audio signal encryption algorithm based on a combination of three chaotic maps was presented. In [5], four different voice signals with different running times were sampled and their values were permuted using the chaotic logistic function and the permuted numbers were further separated into four parts to present the novel audio encryption scheme. The use of chaotic functions and DNA coding to confuse and diffuse audio files was presented in [6]. In [7], a novel compressing sensing-based function to simultaneously compress and encrypt audio files was presented. Novel audio encryption with extensive

cryptographic analysis was proposed in [8]. In Reference [9], a novel audio encryption scheme based on the Henon–Tent chaotic pseudo-random number array was proposed.

In Reference [10], a noise-tolerant audio encryption technique designed by the application of the S_8 symmetric group and chaotic systems was proposed. Steps for audio signal encryption based on the chaotic Henon map were designed in [11]. In [12], a novel three-dimensional chaotic system was proposed for the audio encryption algorithm. Researchers in [13] proposed an algorithm for determining the maximum number of bits that can be used as the generator of a pseudo-random number output including chaos-based functions. In Reference [14], the authors proposed an efficient speech encryption method based on three-dimensional chaotic maps. In [15], the authors presented an audio encryption algorithm using confusion and diffusion based on multi-scroll chaotic function.

The motivation of our research was the ever-growing need for more sophisticated and robust encryption due to the evolution of PCs and network technologies. There is a widespread common belief that the chaos-based encryption set of rules allows to reduce the relation among audio content by increasing the cost of the entropy of the encrypted audio as well as lowering the correlation.

In our opinion, the main contributions of our paper can be summarised as follows:

- We applied the pseudo-random number generator based on the Ikeda map [16] to a novel audio encryption scheme;
- We examined the proposed algorithm and the results show that it has a very good sample change rate number, desirable signal-to-noise and peak signal-to-noise ratios as well as strong key sensitivity that is able to resist most common theoretical and statistical attacks;
- We allow ourselves to make the assumption that the proposed audio encryption is suitable for ensuring the security of different byte-oriented multimedia sources such as images and video files.

In Section 2, we present a well-known pseudo-random byte output generator based on the chaotic Ikeda map. In Section 3, we introduced the novel safe and efficient audio encryption using the chaotic Ikeda map and complete security analysis is given in Section 4. Finally, the paper is concluded in Section 5.

2. Ikeda Map Used as a Basis for a Pseudo-Random Generator

The main idea is to include the Ikeda map [17,18] in the audio file encryption algorithm, as its behaviour can generate close to random values [16].

2.1. Description of the Ikeda Map

The two-dimensional Ikeda map can be calculated using the following equations:

$$x_{n+1} = 1 + u(x_n \cos t_n - y_n \sin t_n) \quad (1)$$

$$y_{n+1} = u(x_n \sin t_n + y_n \cos t_n) \quad (2)$$

where u is parameter and:

$$t_n = 0.4 - \frac{6}{1 + x_n^2 + y_n^2}. \quad (3)$$

The Ikeda attractor exists when $u \geq 0.6$, as shown in Figure 1.

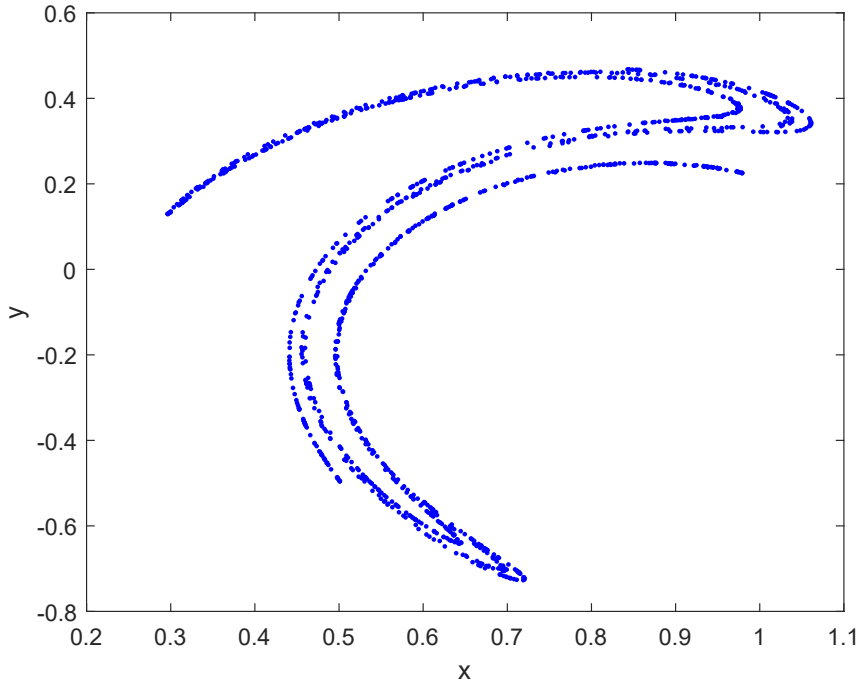

Figure 1. Plot of the Ikeda map.

Its bifurcation diagram (in the classical way with respect to parameter c_3 [19]) is shown in Figure 2. This diagram allows us to discover the existence of a period doubling cascade to chaos. The diagram of the Lyapunov exponent (LE) [20] is plotted in Figure 3. It can be seen that the highest value of the LE is 0.18947. One can choose initial values in the interval $[-0.5, 1.5]$.

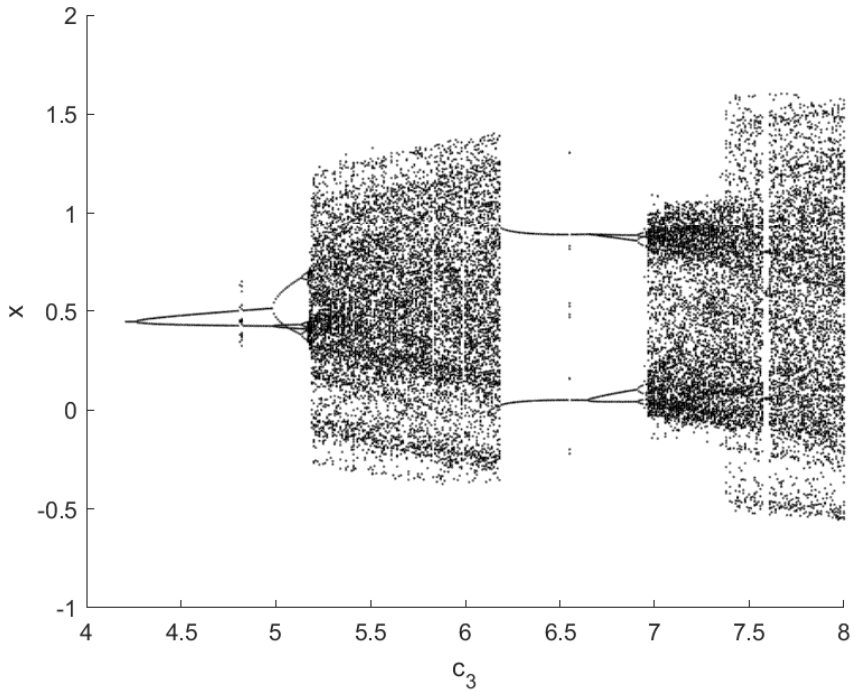

Figure 2. Bifurcation diagram of the Ikeda map.

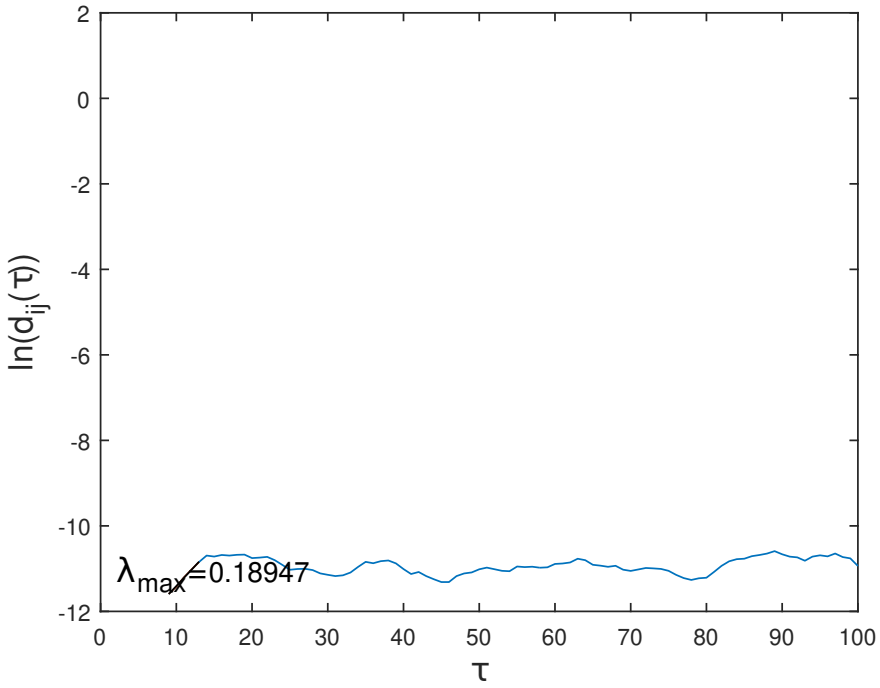

Figure 3. Lyapunov exponent of the Ikeda map.

The chaotic functions realised on a software program will always show dynamical degradation [21]. With respect to [22,23], we designed a pseudorandom byte generator based on two Ikeda functions with a different parameter u, combined with the logical XOR function. All program experiments described in this paper were carried out with double precision IEEE floating-point numbers [24]. It is possible to replace the XOR with feedback with a carry shift register (FCSR) which will randomly switch the Ikeda functions' output values.

2.2. Description of the Pseudo-Random Byte Algorithm

The pseudo-random byte generator includes five steps [16]:

1. The initial values and parameters forming the key set for the two Ikeda functions were obtained—$x_{1,0}$, $y_{1,0}$, $x_{2,0}$, $y_{2,0}$, u_1 and u_2;
2. More than 100 iterations of the Ikeda maps were made without retrieving any results;
3. The work of the algorithm continues with more iterations of the Ikeda maps to calculate and post-process the four real values—$x_{1,i}$, $y_{1,i}$, $x_{2,i}$, $y_{2,i}$. They are calculated as follows:

$$a = mod(abs(integer(x_{1,i} \times 10^{10}))), 256) \quad (4)$$

$$b = mod(abs(integer(y_{1,i} \times 10^{10}))), 256) \quad (5)$$

$$c = mod(abs(integer(x_{2,i} \times 10^{10}))), 256) \quad (6)$$

$$d = mod(abs(integer(y_{2,i} \times 10^{10}))), 256) \quad (7)$$

where $integer(z)$ returns the integer part of z—truncating the value at the decimal point—$abs(z)$ returns the absolute value of z, and $mod(z, w)$ returns the reminder after division;

4. Output byte s_i is generated when XOR operations are performed between a, b, c, and d: $s_i = a \oplus b \oplus c \oplus d$;
5. Perform Step 3 until the output stream is reached.

Theoretical and statistical tests were performed to determine the cryptographic security of the proposed generator and can be found in [16].

3. Novel Implementation of Audio Encryption Scheme

Audio encryption algorithm using the pseudo-random byte algorithm described above is presented in this section.

Proposed Encryption and Decryption Algorithm

These are the steps describing the audio encryption algorithm:

1. In file A′, the header bytes of a input audio file A are moved without cryptographic modifications;
2. Using logical XOR operation with the same amount of bytes as the bytes in the sample produced by the pseudo-random generator described above the bytes in the sample are encrypted;
3. Encrypted sample from Step 2 is processed into file A′;
4. Repeat Steps 2–3 until the end of input file A is reached;
5. The produced output file A′ is the final encrypted audio file.

For example, for the key set 1: $x_{1,0} = 0.61337047692752$, $y_{1,0} = 0.7315988807484$, $x_{2,0} = -0.90983014150675$, $y_{2,0} = 0.44066640271424$, $u_1 = 0.7941$, and $u_2 = 0.694$; after 200 iterations, we obtained $x_{1,200} = 0.814632$, $y_{1,200} = -0.854957$, $x_{2,200} = 0.571817$ and $y_{2,200} = 0.398801$. The output byte $s_0 = a \oplus b \oplus c \oplus d = 219 \oplus 138 \oplus 246 \oplus 191 = 24$. The first byte $a_0 = 114$ from the audio sample is encrypted by $ae_0 = a_0 \oplus s_0 = 114 + 24 = 106$.

For security reasons, we presented a few overall rounds of the audio encryption algorithm.

The decryption scheme is the same as the encryption algorithm because the proposed cryptographic algorithm is symmetric using the same steps and the same key for audio decryption.

4. Cryptographic Analysis

An important part of the encryption algorithms is their reliability. It is determined by cryptographic analysis. In this section, we present the results of empirical tests performed using proposed audio encryption algorithm described in the previous section.

Using the C++ programming language, the proposed audio encryption method was implemented. All experimental results discussed in the following subsections were taken by using one iteration of the novel technique.

Nine 2-bytes per sample audio files were encrypted for the security tests. The files were randomly selected from the huge collaborative database Freesound, accessed 1 September 2021 (http://freesound.org). The chosen files are currently stored in WAV format with an audio sample rate of 44.1 kHz.

4.1. Waveform Plotting

Waveform plotting represents the amplitude of the audio signal distributed in time [25]. Figure 4 shows the wave amplitude of the plain audio files (Figure 4a,c,e) and the wave amplitude of the same files after encryption (Figure 4b,d,f). The compared waves shown in Figure 4 are completely different, as can be seen, which shows the good properties of the proposed scheme for audio encryption. Waveform plotting was performed using MATLAB (R2021b, MathWorks, Natick, MA, USA).

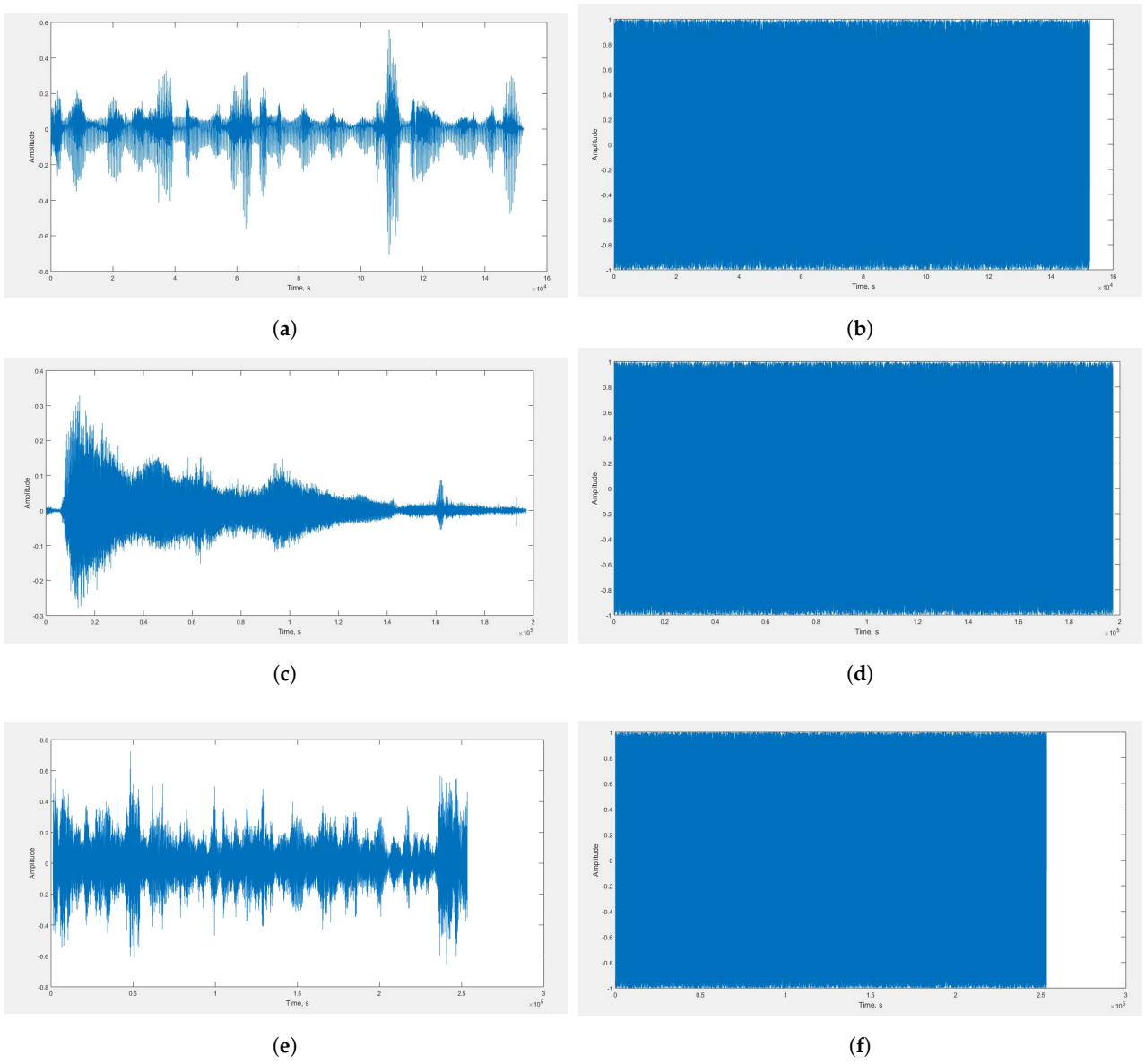

Figure 4. Waveform plotting of original and encrypted audio files: (**a**) cartoon-mumble-speak, (**c**) grito-wav and (**e**) a-strange-dream are the original input audio files; and (**b**,**d**,**f**) are the waveforms of the encrypted audio files.

4.2. Correlation Analysis

The correlation coefficient between two audio files is a mathematical relationship between groups of values [26]. To calculate the relationship between audio information between the primary and the encrypted file, it is necessary to compare the values of both files by working with the samples. The correlation coefficient represents the level of correlation and is always in the range of $[-1.0, +1.0]$. Proximity to 0.0 is considered as a lack of linear connection.

The correlation coefficient can be calculated as follows:

$$r_x y = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}. \tag{8}$$

where:

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2, \tag{9}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2, \tag{10}$$

$$cov(x,y) = \sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}), \tag{11}$$

Sample values of the plain and encrypted files are x_i and y_i, x and y are the mean values of samples, N is the total number of samples and finally $cov(x, y)$ is covariance between both files.

The correlation coefficient results are shown in Table 1.

Table 1. Correlation between plain and encrypted audio files.

Audio File	File Size	File Length	Corellation Coefficient
usb-headset-weird	200 kb	2.32 s	−0.0248023
cartoon-mumble-speak	298.1 kb	3.45 s	0.0008942
grito-wav	385.3 kb	4.47 s	0.0167187
a-strange-dream	495 kb	5.74 s	−0.0376280
radio	691 kb	8.02 s	0.0222683
insult	832 kb	9.66 s	0.0157122
sharpening-blade-long-blade	1228.8 kb	14.48 s	−0.0211372
airport-security-message	1740.8 kb	20.28 s	−0.0335765
shakuhachi-grave-5	2048 kb	23.41 s	−0.0068954
Ref. [5]	-	7 s	0.0233000
Ref. [8]	2.33 mb	13.85 s	0.0004710
Ref. [27]	-	-	0.0000900
AES [27]	-	-	0.0097100

The values in Table 1 indicate that there is no relation between the original file and the encrypted file. Compared with similar audio encryption techniques, the proposed one has correlation coefficients values close to the ideal value of 0.0.

4.3. Number of Sample Change Rate

The number of the sample change rate (NSCR) is a test that determines the quality of encryption algorithms. The test compares the corresponding original sample values and encrypted audio file and shows the difference in percentage. NSCR can be calculated as follows:

$$NSCR = \frac{\sum_{i=1}^{N} D_i}{N} x100\% \tag{12}$$

where:

$$D_i = \begin{cases} 1, x_i \neq y_i \\ 0, otherwise \end{cases} \tag{13}$$

In Equation (12), corresponding sample values of the plain and encrypted files are x_i and y_i and the total number of samples is N. The obtained test results are shown in Table 2. The numbers are similar to the results of other audio encryption schemes [5,8,27].

Table 2. Number of the sample change rate.

Audio File	NSCR
usb-headset-weird	99.994%
cartoon-mumble-speak	100.000%
grito-wav	99.998%
a-strange-dream	99.996%
radio	99.999%
insult	99.999%
sharpening-blade-long-blade	99.998%
airport-security-message	99.998%
shakuhachi-grave-5	99.998%
Ref. [5]	99.998%
Ref. [8]	99.998%
Ref. [27]	99.998%
AES [27]	99.603%

4.4. Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) test is excellent for measuring speech signal intelligibility. SNR measurement sets have precise criteria for measuring the performance of optimal signal processing [28] which can be calculated as follows:

$$SNR = 10log_{10}\frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N}[x_i - y_i]}dB, \tag{14}$$

where x_i and y_i are the corresponding sample values from the original audio file and the encrypted audio file, and N is the number of samples.

4.5. Peak Signal-to-Noise Ratio

Peak signal-to-noise ratio (PSNR) is another way to compute the power of the clean signal against the power of noise [29]. The PSNR is more commonly used in image encryption algorithms [30] and can be used to test the quality of the proposed encryption scheme in this article. PSNR is calculated as follows:

$$PSNR = 10log_{10}\frac{MAX^2}{MSE}dB \tag{15}$$

where the maximum possible value of audio stream is MAX. In this case, the maximum value can be 65,535. There is possibly a square error between the plain and encrypted files and this mean square error (MSE) can be computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{16}$$

The SNR and PSNR values for nine different audio files are represented in Table 3.

In this proposed method, we obtained a negative SNR value which shows that the encrypted files are very noisy and the clear signal is destroyed. As can be seen, PSNR values are small, which means that the encrypted audio files have a very high level of noise. Compared with the other audio encryption schemes [5,8,27], we can see that the proposed scheme has comparable and better SNR and PSNR values.

Table 3. Peak signal-to-noise ratio.

Audio File	SNR	PSNR
usb-headset-weird	−42.3697 dB	4.6583 dB
cartoon-mumble-speak	−44.6930 dB	4.7036 dB
grito-wav	−43.0053 dB	4.7551 dB
a-strange-dream	−44.8190 dB	4.6431 dB
radio	−40.7581 dB	4.6431 dB
insult	−43.2195 dB	4.7109 dB
sharpening-blade-long-blade	−28.5052 dB	4.7690 dB
airport-security-message	−44.7026 dB	4.6590 dB
shakuhachi-grave-5	−43.3945 dB	4.6990 dB
Ref. [5]	33.7464 dB	59.7989 dB
Ref. [8]	−16.0483 dB	1.4524 dB
Ref. [27]	−133.0000 dB	-
AES [27]	−1.4461 dB	-

4.6. Speed Performance

To measure the time required to encrypt, we used audio files of different sizes with hardware configuration—2.00 GHz, Intel(R) Core(TM) i3-6006CPU, Fujitsu, 4 GB RAM, Windows 10. In Table 4, we compare the speed of our method with [5,8,31]. The data show that the proposed audio encryption scheme has a satisfactory speed. The AES cipher takes less time than all other algorithms to encrypt audio files.

Table 4. Speed performance test.

Audio File	File Size	File Length	Encryption Time
usb-headset-weird	200 kb	2.32 s	0.865 s
cartoon-mumble-speak	298.1 kb	3.45 s	1.284 s
grito-wav	385.3 kb	4.47 s	1.676 s
a-strange-dream	495 kb	5.74 s	2.188 s
radio	691 kb	8.02 s	3.195 s
insult	832 kb	9.66 s	3.691 s
sharpening-blade-long-blade	1228.8 kb	14.48 s	5.651 s
airport-security-message	1740.8 kb	20.28 s	8.193 s
shakuhachi-grave-5	2048 kb	23.41 s	8.831 s
Ref. [5]	-	7 s	0.012 s
Ref. [8]	2.33 mb	13.85 s	5.767 s
AES [31]	800 kb	-	0.003 s

4.7. Encryption/Decryption Key Sensitivity

Another important characteristic of correlation analysis is the key sensitivity test. A good audio encryption scheme should be sensitive with respect to the secret key, that is, a slight modification of the secret key. We encrypted an audio file with key set 1 and then tried to decrypt it with the very slightly modified key set 2: x_{1d} = 0.62337047692752, y_{1d} = 0.7415988807484, x_{2d} = -0.91983014150675, and y_{2d} = 0.45066640271424.

Figure 5a,b present the original and encrypted file a-strange-dream; Figure 5c presents the decrypted file a-strange-dream with different key, Figure 5d is decrypted with the first key set. The proposed audio scheme demonstrates unsuccessful decryption even with a very similar key set. The right decryption needs the entire and correct knowledge of all of the secret key digit.

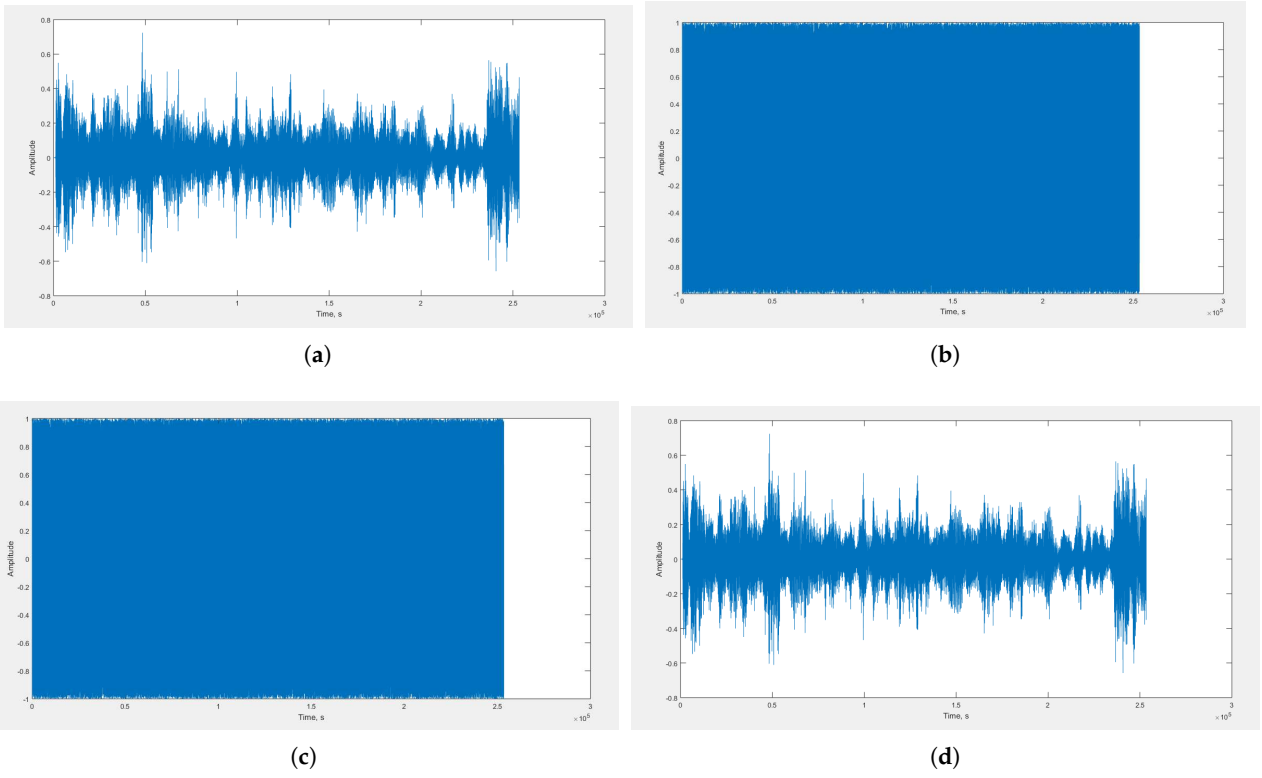

Figure 5. Waveform plotting of the original and encrypted audio file: (**a**) is the original audio file a-strange-dream; (**b**) is the encrypted audio file with the first key set; (**c**) is decrypted with a second key set; and (**d**) is the waveplot presented decrypted audio file with the first key set.

5. Conclusions

This paper presents a new design of the algorithm for audio files encryption. The high quality of the algorithm is proven by the tests which we conducted. The waveform plots of the tested audio files demonstrate the changes between the original files and the encrypted files. The number of the sample change rate test and the correlation coefficient verified that the sample values were completely different in the corresponding files. The peak signal-to-noise ratio values show that the encrypted files have a very high noise level—which means that encrypting the audio destroys the data in the original file. Key sensitivity is very high. An encrypted file can only be restored with the original key set and even with a small change in the keys, the original file cannot be restored. Based on the security analysis performed, we can say that the proposed algorithm has excellent audio encryption properties.

Author Contributions: Conceptualisation, B.S.; methodology, B.S.; software, B.S. and T.I.; validation, B.S. and T.I.; formal analysis, B.S. and T.I.; writing original draft preparation, B.S.; writing review and editing, B.S.; visualisation, B.S. and T.I.; supervision, B.S.; project administration, B.S.; funding acquisition, B.S. and T.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Scientific Program Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICTinSES), financed by the Ministry of Education and Science, Bulgaria.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rivest, R.L.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. Rijmen, V.; Daemen, J. Advanced encryption standard (AES). In *Federal Information Processing Standards Publications 197*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001; Available online: https://csrc.nist.gov/publications/detail/fips/197/final (accessed on 1 September 2021).
3. Schneier, B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Fast Software Encryption*; Anderson, R., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 191–204. [CrossRef]
4. Ghasemzadeh, A.; Esmaeili, E. A novel method in audio message encryption based on a mixture of chaos function. *Int. J. Speech Technol.* **2017**, *20*, 829–837. [CrossRef]
5. Sathiyamurthi, P.; Ramakrishnan, S. Speech encryption using chaotic shift keying for secured speech communication. *EURASIP J. Audio Speech Music Process.* **2017**, *2017*, 1–11. [CrossRef]
6. Wang, X.; Su, Y. An Audio Encryption Algorithm Based on DNA Coding and Chaotic System. *IEEE Access* **2020**, *8*, 9260–9270. [CrossRef]
7. Moreno-Alvarado, R.; Rivera-Jaramillo, E.; Nakano, M.; Perez-Meana, H. Simultaneous Audio Encryption and Compression Using Compressive Sensing Techniques. *Electronics* **2020**, *9*, 863. [CrossRef]
8. Kordov, K. A Novel Audio Encryption Algorithm with Permutation-Substitution Architecture. *Electronics* **2019**, *8*, 530. [CrossRef]
9. Adhikari, S.; Karforma, S. A novel audio encryption method using Henon–Tent chaotic pseudo random number sequence. *Int. J. Inf. Technol.* **2021**, *13*, 1463–1471. [CrossRef]
10. Aziz, H.; Gilani, S.M.M.; Hussain, I.; Janjua, A.K.; Khurram, S. A Noise-Tolerant Audio Encryption Framework Designed by the Application of S8 Symmetric Group and Chaotic Systems. *Math. Problems Eng.* **2021**, *2021*, 5554707. [CrossRef]
11. Roy, A.; Misra, A. Audio signal encryption using chaotic Hénon map and lifting wavelet transforms. *Eur. Phys. J. Plus* **2017**, *132*, 1–10. [CrossRef]
12. Shah, D.; Shah, T.; Ahamad, I.; Haider, M.I.; Khalid, I. A three-dimensional chaotic map and their applications to digital audio security. *Multimedia Tools Appl.* **2021**, *80*, 22251–22273. [CrossRef]
13. Tutueva, A.V.; Karimov, T.I.; Moysis, L.; Nepomuceno, E.G.; Volos, C.; Butusov, D.N. Improving chaos-based pseudo-random generators in finite-precision arithmetic. *Nonlinear Dyn.* **2021**, *104*, 727–737. [CrossRef]
14. Hato, E.; Shihab, D. Lorenz and rossler chaotic system for speech signal encryption. *Int. J. Comput. Appl.* **2015**, *128*, 25–33. [CrossRef]
15. Liu, H.; Kadir, A.; Li, Y. Audio encryption scheme by confusion and diffusion based on multi-scroll chaotic system and one-time keys. *Optik* **2016**, *127*, 7431–7438. [CrossRef]
16. Stoyanov, B.; Ivanova, T. CHAOSA: Chaotic map based random number generator on Arduino platform. *AIP Conf. Proc.* **2019**, *2172*, 090001. [CrossRef]
17. Ikeda, K. Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. *Opt. Commun.* **1979**, *30*, 257–261. [CrossRef]
18. Ikeda, K.; Daido, H.; Akimoto, O. Optical Turbulence: Chaotic Behavior of Transmitted Light from a Ring Cavity. *Phys. Rev. Lett.* **1980**, *45*, 709–712. [CrossRef]
19. Buscarino, A.; Fortuna, L.; Frasca, M. *Essentials of Nonlinear Circuit Dynamics with MATLAB® and Laboratory Experiments*; CRC Press: Taylor and Francis Group: Boca Raton, FL, USA, 2017.
20. Mohammadi, S. *LYAPROSEN: MATLAB Function to Calculate Lyapunov Exponent*; Boston College, Department of Economics: Boston, MA, USA, 2009; Available online: https://EconPapers.repec.org/RePEc:boc:bocode:t741502 (accessed on 1 September 2021).
21. Li, S.; Chen, G.; Mou, X. On the dynamical degradation of digital piecewise linear chaotic maps. *Int. J. Bifurcation Chaos* **2005**, *15*, 3119–3151. [CrossRef]
22. Patidar, V.; Sud, K.K.; Pareek, N.K. A pseudo random bit generator based on chaotic logistic map and its statistical testing. *Informatica* **2009**, *33*, 441–452.
23. Tutueva, A.; Pesterev, D.; Karimov, A.; Butusov, D.; Ostrovskii, V. Adaptive Chirikov Map for Pseudo-random Number Generation in Chaos-based Stream Encryption. In Proceedings of the 2019 25th Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 5–8 November 2019; pp. 333–338. [CrossRef]
24. IEEE Standard for Floating-Point Arithmetic. In *IEEE Std 754-2019 (Revision of IEEE 754-2008)*; IEEE: Piscatway, NJ, USA; New York, NY, USA, 2019; pp. 1–84. [CrossRef]
25. Jozwiak, M.; Monnet, X.; Teboul, J.L. Pressure waveform analysis. *Anesth. Analg.* **2018**, *126*, 1930–1933. [CrossRef] [PubMed]
26. Taylor, R. Interpretation of the correlation coefficient: a basic review. *J. Diagnost. Med. Sonogr.* **1990**, *6*, 35–39. [CrossRef]
27. Farsana, F.; Devi, V.; Gopakumar, K. An audio encryption scheme based on Fast Walsh Hadamard Transform and mixed chaotic keystreams. *Appl. Comput. Inform.* **2019**. [CrossRef]
28. Johnson, D.H. Signal-to-noise ratio. *Scholarpedia* **2006**, *1*, 2088. [CrossRef]
29. Korhonen, J.; You, J. Peak signal-to-noise ratio revisited: Is simple beautiful? In Proceedings of the 2012 Fourth International Workshop on Quality of Multimedia Experience, Melbourne, Australia, 5–7 July 2012; pp. 37–38. [CrossRef]

30. Poobathy, D.; Chezian, R.M. Edge detection operators: Peak signal to noise ratio based comparison. *Int. J. Image Graph. Signal Process.* **2014**, *10*, 55–61. [CrossRef]
31. Ahamad, M.M.; Abdullah, M.I. Comparison of encryption algorithms for multimedia. *Rajshahi Univ. J. Sci. Eng.* **2016**, *44*, 131–139. [CrossRef]