2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4) | 978-1-6654-0096-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/WorldS451998.2021.9514037

Audio Encryption With Computational Chaotic System Error

Thiago A. Santos[†], Erivelton G. Nepomuceno[†*], *IEEE Senior Member*,
Thomas M. Chen[‡] and Denis N. Butusov[§]
[†]Department of Electrical Engineering, Federal University of São João del-Rei, Brazil.
[‡]Department of Electrical and Electronic Engineering, City, University of London, UK
[§]Youth Research Institute, St Petersburg Electrotechnical University "LETI", Russia

***Abstract*—This paper proposes a novel method of chaos-based encryption for audio. Our method outperformed the minimal requirement of speed for real time audio transfers while maintaining its high security features. The paper exploits finite errors derived from the computation of chaotic systems. The cipher is built on the lower bound error, which is computed by means of two interval extensions of a chaotic system. It was found that the method was effective, and required little computational power in order to be completed, proving to be faster and still reliable compared to other works.**

***Index Terms*—Audio encryption, chaotic systems, stream ciphers, lower bound error.**

I. Introduction

AUDIO encryption has become an important research issue to guarantee the privacy of users [1], and is at the core of almost all mobile devices and multimedia technologies. Among the several techniques applied to audio encryption, chaotic systems have received great attention. Transmitting speech or audio files usually requires an encryption process, which is essential for protecting files against unauthorized reading or modifying the signals [2].

Among the numerous techniques applied to encrypt audio, chaotic systems have recently attracted considerable attention [1], [3]–[5]. According to [1], some advantages of chaotic systems relies on the fact these systems present initial value sensitivity, no periodicity, pseudo-randomness, and ergodicity of chaotic sequences. These systems are also more effectively applied to practical applications. Nevertheless, in these recent works less attention has been given to some disadvantages of chaotic systems, particularly related to degradation of its chaotic behavior due to finite precision of digital devices [6].

This paper exploits finite error derived from the computation of chaotic systems to encrypt audio signals . Using two interval extensions, the cipher is built on the lower bound error [7]–[9]. The result is efficient, enabling even low computational capacity, such as embedded devices, to use the encryption of audio streams or recordings. Successful tests have been made on audio files samples recorded at 44100 Hz.

This paper was divided into parts. In the introduction, we made a brief contextualization of the problem and latest work and in the key preparation section, we explained the whole process in order to obtain the key from a chaotic system model. The audio encryption section details the source files characteristics, as well as how the obtained key was applied to encode the information. In the results section, we present our findings about our new method, while comparing some indicators with other works, followed by a conclusion that summarizes our research.

Erivelton G. Nepomuceno was supported by Brazilian Research Agencies: CNPq/INERGE (Grant No. 465704/2014-0), CNPq (Grant No. 425509/2018-4 and Grant No. 311321/2020-8) and FAPEMIG (Grant No. APQ-00870-17)
*Corresponding author: nepomuceno@ieee.org

II. Problem formulation

The paper has investigated the encryption of audio signals. Three samples were used to demonstrate the encryption method downloaded from a free online audio library[1]. The audios are encoded in 16-bit signed integers in *flac* format, with a sampling frequency of 44100 Hz. Such a sampling frequency is widely used and presents a challenge, since there is a need for processing speed when dealing with encryption of *real-time* audio transmission. Thus, selective encryption schemes emerge as a method to increase the process speed [10]–[12]. Our method instead, focuses on simplicity in order to accelerate the encrypting of all parts of the data, while facilitating implementation and maintaining cryptanalysis and attacks infeasible.

The audio signal is encrypted with a key in the same data format, which allows for each bit of the audio to be operated by a respective bit of the key by means of binary exclusive-or operation, denoted by the symbol "$\oplus$". The 16-bit groups of the data are each operated, using XOR, with one or more groups of the same size from the key. The number of each 16-bit number is operated bit-by-bit with one or multiple 16-bit numbers of the key. The number of operations is dictated by the number of rounds the user defined.

Since there is a distribution bias on the key values evidenced by Figure 2, we used multiple rounds of encryption in order to shuffle the key. We do this in a way that no significant additional iterations of the chaotic systems are required. Each key value will be used multiple times, by applying rounds of encryption. Using multiple rounds drastically decreased correlation between the original and encrypted file, while adding insignificant delay to the process. We found heuristically a minimum of $R = 20$ rounds of encryption, but it should

[1]https://freesound.org/

be noted that the user can change this value to best fit each application, as long as the key remains statistically random. Also, by reusing values from the key, this process allows for live data transmission to be entirely encrypted. In fact, every type of data transmission could make use of this method, since digital data is always represented by binary digits.

A. Decryption process

The decryption process follows exactly the same algorithm as the encryption phase. The only change that the receiver has to make is to use the encrypted audio as the input to the algorithm. In possession of some information such as which chaotic system was used, and what were the initial conditions and parameters, the receiver is able to repeat the process. Since the exclusive-or operation is reversible, the output on the receiver side will be the original human-readable file.

III. Key preparation

Previous works have used stream ciphers to encrypt an image. RC4 is a well-known scheme that uses an insecure pseudo-random algorithm to encrypt data [13], [14]. Other famous and more recent stream ciphers are Salsa20 and ChaCha20 [15], [16], which are an attempt to make stream ciphers more secure. In order to solve the RC4 key generation problem and propose a simpler method for stream ciphers, we based on previous works [7], [8] to develop a new scheme for obtaining a strong encryption key suitable for audio encryption.

In this section, we describe in detail the process of obtaining a strong stream cipher key. Chaos-based encryption relies on simulating a mathematical description of a chaotic system. Lorenz describes chaotic systems as having complex dynamics and deterministic behavior [17]. By simulating a chaotic system, a pseudo-random sequence is obtained and could be reproduced taking advantage of the deterministic property, given the same initial conditions as inputs to the system.

Encrypting data is an effort to keep third parties unable to read the information concealed, by altering it in a manner that only the sender and the recipient knows how to undo. The deterministic property of chaotic systems presents a great tool to generate encryption keys to alter the data, as in order to obtain the same pseudo-random sequence, one must only know the mathematical description of the system used and the set of initial conditions.

In this work, the well-known chaotic logistic map is used. This map was used as a demographic model by May [18], and is written as follows:

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

where x_n is a sequence of results obtained from the chaotic model, with $x_n \in (0; 1)$, and $r = 3.6$ to guarantee chaotic behavior. x_1 assumes values of a list M:

$$M = [m_1\ m_2\ m_3\ m_4\ m_5], \tag{2}$$

where $m_i \in [0; 1]$ are the perturbation values used to generate one fifth of the key. Choosing a list of values as the perturbation conditions drastically increase the key space, as evidenced by Nepomuceno et al. [19].

A. On the finite precision of computers

Although chaotic systems can be used for encryption, some remarks have been made about the accuracy on which that computer simulates it [20], [21]. Finite precision is an inherent characteristic of any computing device. In fact, computers often are unable to store an exact value of a number, e.g., binary repeating decimals. Thus, various methods of approximation are standardized by IEEE 754 standard for floating point arithmetic [22].

As a means to model the uncertainty of results, R. E. Moore [23] defines interval extensions, as follows.

Definition 1. *An interval extension of f is an interval valued function F of an interval variable X, with the property*

$$F(x_1, \cdots, x_n) = f\left(x_1, \cdots, x_n\right), \tag{3}$$

where an interval means a closed set of real numbers such that $X = [\underline{X}, \overline{X}] = \{x : \underline{X} \leq x \leq \overline{X}\}$ *with* $x \in R$.

Thus, by simulating an interval extension of a chaotic map, it is possible to obtain two pseudo-orbits that represent an interval in which the exact solution is contained. An unique chaotic sequence is computed by interval extension from the logistic map. The pseudo-orbits were obtained by synthesizing two mathematically equivalent equations 4 and 5. Such equations, although equivalent, yield different results as the order of operations matter when dealing with floating-point arithmetic. [9].

$$x_{a,n+1} = r x_{a,n} \times (1 - x_{a,n}) \tag{4}$$

$$x_{b,n+1} = r x_{b,n} - r x_{b,n} x_{b,n} \tag{5}$$

where $x_{a,n}$ denotes the sequence obtained from the first equivalent equation, and $x_{b,n}$ represents the results obtained from the second equivalent model.

In our work, interval extensions are used to estimate the Largest Lyapunov Exponent (LLE), in order to indicate that the system is chaotic. The LLE quantifies the system's sensibility to initial conditions, and can be evaluated by observing the size of each iteration interval, as described in [24], using the following formula:

$$\delta_n = \frac{|\hat{x}_{a,n} - \hat{x}_{b,n}|}{2}, \tag{6}$$

where $\hat{x}_{a,n}$ and $\hat{x}_{b,n}$ are pseudo-orbits and δ_n is the lower bound error of a map $f(x)$. The LLE also means the loss of information, which allows the use of multiple perturbation and an increase of key-space for the proposed encryption scheme.

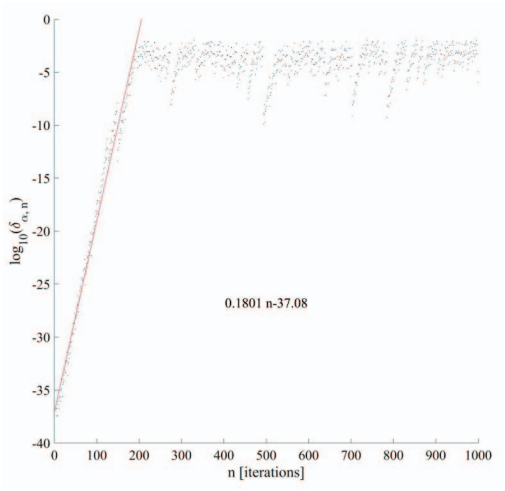


Fig. 1: Plot of the logarithm of the lower bound error for the first 1000 iterations n. The interval grows until the simulation reaches around 200 iterations, when the error reaches the magnitude of the calculated variable. The red line approximates the growth rate of the lower bound error. This rate represents the largest Lyapunov exponent [24].

B. Key requisites

In order to obtain a successful secure data transference, the sender and the receiver computers must obtain the same key. Some mandatory characteristics of the key for this method to work are:

- In order to apply our method, the key has to be at least the same size of bits of the data, since we are using symmetric key encryption;
- As we are using symmetric key encryption, both the sender and receiver must obtain the same key in order to achieve a successful data transfer;

IV. Results

In this section, we present the tests applied to verify the viability, security and effectiveness of our method. We compare the results with other works presented in the literature.

A. Key obtaining and scaling

After simulating an interval extension of the logistic map, we calculated the lower bound error between both pseudo-orbits using Eq. 6. Figure 1 shows a logarithmic plot of this vector for the first 1000 iterations. After that, the sequence obtained from the lower bound error vector to compose the key, instead of the values from one pseudo-orbit. This choice presents several security improvements, as these values have dissociated themselves from the original initial conditions of the map, and also from the map itself. Some dynamic systems, such as Lorenz's system [17], have well known behavior. By observing only the lower bound error of such a system, one would not be able to identify which system generated the data. Also, calculating the LBE makes it possible to easily estimate the LLE, confirming that the simulated system is chaotic.

In order to encrypt information, the key should have a range of values that match the original data domain. Our prerecorded audio is represented by binary 16-bit signed integers, which means that it assumes decimal values from 0 to 255. In order to distribute and normalize these values over the domain of the data, we used the following formula:

$$K_n = \mathrm{fix}\left(\frac{255 \times \delta_{\alpha,n}}{\max(\delta_{\alpha,n})}\right), \tag{7}$$

where $\mathrm{fix}()$ rounds the elements to the nearest integers towards zero. This generates integer values between 0 to 255, which are represented by 16-bit signed integers.

In order to analyze the value distribution of the key vector, Figure 2 shows the frequency of values in the set. It can be seen that a bias is present towards smaller values on the set. This poses a problem, as an encryption key ideally should be equally distributed to avoid cryptanalysis and has been successfully mitigated by multiple application of the cipher into the signal on 20 rounds of XOR, as shown in Figure 3.

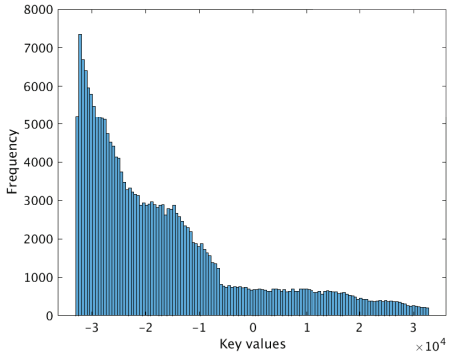


Fig. 2: Probability distribution of the key values, by the frequency each value appears in the set. It can be seen that there is a bias towards smaller numbers, which demonstrates that both pseudo-orbits are often close to each other and is a problem from the cryptanalysis point of view. We eliminate this bias in a later phase.

B. Waveform plotting

The waveform of the audio is a way to visualize the plot of audio data into values versus time. In Figure 4, a plot of an excerpt of the audio is shown. One should note that in the encrypted audio, the data is not lost, but concealed by our method in a manner that third parties are not able to read.

C. Speed requisites

Since there is a great concern about the capacity of the computers to handle real-time audio encryption, we propose a simple rule to determine whether our method can be applied. In our example, the sampling rate of the audio was 44100

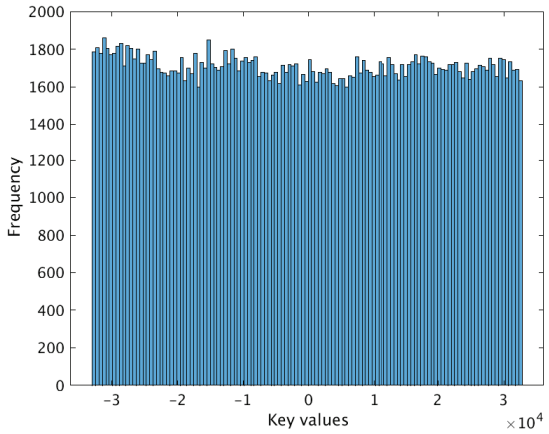


Fig. 3: Probability distribution of the key after 20 rounds of XOR. Now it is possible to note that the values are distributed more evenly, eliminating the security concerns caused by the bias towards small numbers of the original key.

Hz. So, in order to apply this method, we propose using the following formula:

$$T_{\delta_{a,n}} + T_{\delta_{b,n}} + T_{\delta_{\alpha,n}} + T_{\delta_{\alpha,n} \to Key} + T_{Key \oplus Data} \times R < \frac{1}{f_s}, \quad (8)$$

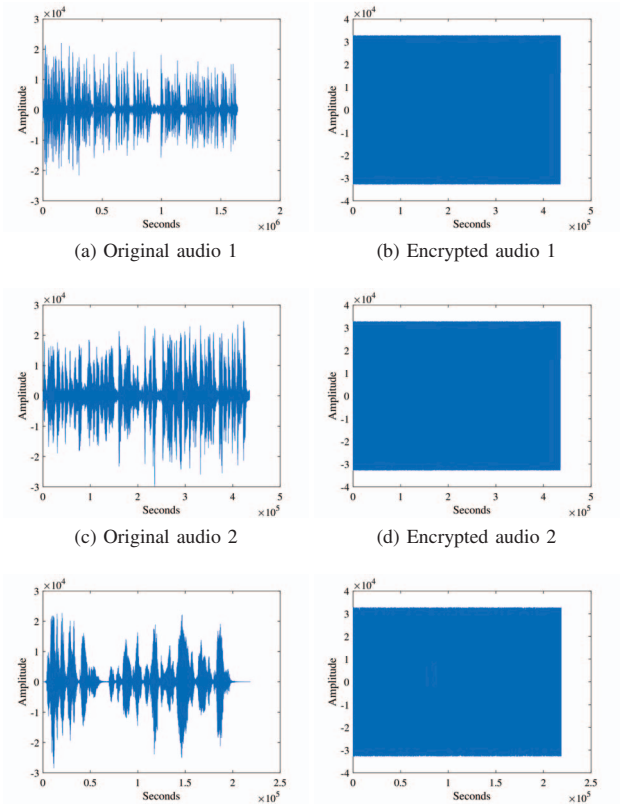
where $T_{\delta_{a,n}}$ and $T_{\delta_{b,n}}$ is the time to simulate one iteration of the pseudo-orbit, $T_{\delta_{\alpha,n}}$ is the time needed to calculate the lower bound error between one iteration of the pseudo-orbits, $T_{\delta_{\alpha,n} \to Key}$ is the time needed to scale the lower bound error to the same data type and range of the audio, $T_{Key \oplus Data} * R$ is the time needed to perform the exclusive-or operation between one position of the key and one position of the data, multiplied by R rounds of encryption, and f_s is the sampling frequency of the audio.

Equation 8 summarizes the entire process. It states that the sum of the time needed by each operation should not exceed the time taken for new data to be collected, which is dictated by the sampling frequency. In fact, our results show that not only does our method meet these requirements, but takes far less time than the minimum needed to complete the entire operation. The time needed to encrypt a byte of the data is $8.4577e-07$s, much smaller than the interval for the next sample of the data to be encrypted $2.2676e-05$s. This allows for the computer to store more iterations of the chaotic system while waiting for more data input.

D. Histogram and spectrogram analysis

The histograms are a tool that allows to visualize the probability distribution of values on data. When applied to audio files, it allows the measurement of the quantity of each value present in the audio file.

Figure 5 shows spectrograms and histograms of each audio file, in its encrypted and decrypted versions. It can be seen that, in the original files, the values are concentrated at middle values, while in the encrypted file, our method was able to distribute that expressive bias towards central values throughout the whole range, avoiding any third party to access the content of the data.



(a) Original audio 1 (b) Encrypted audio 1

(c) Original audio 2 (d) Encrypted audio 2

(e) Original audio 3 (f) Encrypted audio 3

Fig. 4: An excerpt of a part of the test audio used to test our encryption method is shown in (a). In (b), it can be seen that our encryption method yielded completely unreadable and noisy data. After decryption, the waveform assumed the original form in (a).

E. Correlation analysis

The correlation coefficient is a measure of relationship between two data sets. This coefficient can assume values in range [-1,1], and values closer to 0 are considered weak or no correlation. Such an indicator can provide a measure of how much the encrypted data is similar to the original data.

The correlation coefficient can be calculated as follows:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}, \quad (9)$$

in which $cov(x,y)$ is the covariance between both files, σ_X is the standard deviation of the original file and σ_Y is the standard deviation of the encrypted file.

Other studies also have calculated the correlation coefficient in order to evaluate their method. Although the sample audio was not the same, Table I shows that our value is close to

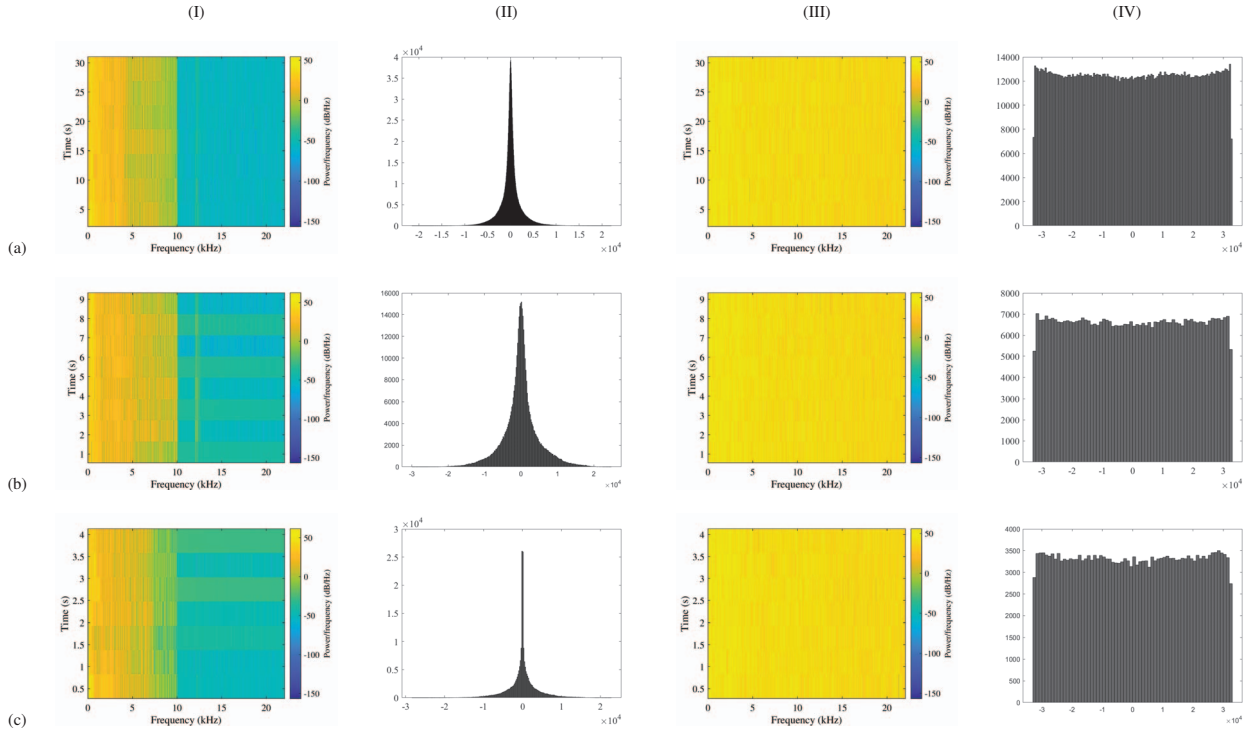


Fig. 5: Histogram and spectrograms of the three audio files. The amount of data in both files remains the same, but the values are spread over the interval, making the content of the encrypted data unrecognizable.

what is present in literature, and also very close to 0. Another encryption was made with 32 rounds, in order to illustrate that the correlation coefficient is affected by the number of rounds we use in the encryption process.

In order to further investigate the key randomness, we evaluated the autocorrelation, shown by Figure 6. The results indicate that one is unable to infer the next value of the key stream based on past values.

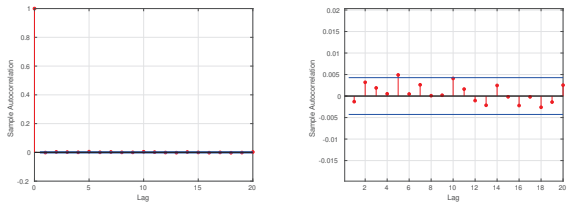


Fig. 6: Autocorrelation for the key obtained after 20 rounds of XOR. It can be seen that, except for lag 0, which is always 1 by definition, the autocorrelation has its values within the 99% confidence limit. On the right, a zoom of the first plot.

V. Conclusion

In this paper, we presented a novel method to encrypt audio. The method is feasible for recorded media and live streams without much computational cost. We were able to provide the user with some control of how scrambled the encrypted data must be, by changing the number of encryption rounds. One should note that changing the number of rounds has little influence in the computational time, because of the efficiency of our method. Although stream ciphers are generally more difficult to implement correctly, we were able to describe the guidelines which should lead to successful encryption. Future works will aim at validating some keystream requisites to ensure security. By applying well known indicators, we were able to ensure our technique was effective for the purpose

TABLE I: Comparison of correlation coefficient with other works. We made an additional attempt to use 32 rounds of encryption, in order to demonstrate that the choice of R can alter this indicator. As Kordov [4] and Kalpana et al. [25] made tests with various files, we included their best and worst results, in order to represent the overall performance achieved.

Work	Correlation
This work (R=20)	-0.0091, -0.0122, -0.01425
This work (R=32)	-0.0014, -0.0005, -0.0109
Kordov [4]	-0.00038781
Kordov [4]	0.0263
Naskar et al. [12]	-0.000208
Naskar et al. [12]	-0.00230
Kalpana et al. [25]	0.00037789

in question. By exploiting the computational error we have mitigated problems with chaos degradation. Further works will aim at developing a communication software using the proposed encryption scheme presented in this work.

VI. References

[1] X. Wang and Y. Su, "An Audio Encryption Algorithm Based on DNA Coding and Chaotic System," *IEEE Access*, vol. 8, pp. 9260–9270, 2020.

[2] O. M. Abu Zaid, M. A. Tawfeek, and S. Alanazi, "Applying and Comparison of Chaotic-Based Permutation Algorithms for Audio Encryption," *Computers, Materials and Continua*, vol. 67, no. 3, pp. 3161–3176, 2021.

[3] H. K. Zghair, S. A. Mehdi, and S. B. Sadkhan, "Speech Scrambler Based on Discrete Cosine Transform and Novel Seven-Dimension Hyper Chaotic System," in *Journal of Physics: Conference Series*, vol. 1804, no. 1, 2021, p. 012048.

[4] K. Kordov, "A novel audio encryption algorithm with permutation-substitution architecture," *Electronics*, vol. 8, no. 5, 2019.

[5] Z. N. Al-kateeb and S. J. Mohammed, "A novel approach for audio file encryption using hand geometry," *Multimedia Tools and Applications*, vol. 79, no. 27-28, pp. 19 615–19 628, 2020.

[6] S. Li, G. Chen, and X. Mou, "On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps," *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005.

[7] E. G. Nepomuceno, L. G. Nardo, J. Arias-Garcia, D. N. Butusov, and A. Tutueva, "Image encryption based on the pseudo-orbits from 1D chaotic map," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 6, p. 061101, 2019.

[8] L. G. Nardo, E. G. Nepomuceno, J. Arias-Garcia, and D. N. Butusov, "Image encryption using finite-precision error," *Chaos, Solitons & Fractals*, vol. 123, pp. 69–78, 2019.

[9] E. Nepomuceno, S. Martins, G. Amaral, and R. Riveret, "On the lower bound error for discrete maps using associative property," *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 462–473, 2017.

[10] Chung-Ping Wu and C. . J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, 2005.

[11] H. Wang, M. Hempel, D. Peng, W. Wang, H. Sharif, and H. Chen, "Index-based selective audio encryption for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 12, no. 3, pp. 215–223, 2010.

[12] P. K. Naskar, S. Paul, D. Nandy, and A. Chaudhuri, "DNA Encoding and Channel Shuffling for Secured Encryption of Audio Data," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 25 019–25 042, 2019.

[13] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of rc4," in *Selected Areas in Cryptography*, S. Vaudenay and A. M. Youssef, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 1–24.

[14] S. Paul and B. Preneel, "A new weakness in the rc4 keystream generator and an approach to improve the security of the cipher," in *Fast Software Encryption*, B. Roy and W. Meier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 245–259.

[15] D. Bernstein, "The salsa20 family of stream ciphers," *New stream cipher designs*, 2008.

[16] ——, "Chacha, a variant of salsa20," *Workshop record of SASC*, 2008.

[17] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[18] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[19] E. G. Nepomuceno, L. G. Nardo, J. Arias-Garcia, D. N. Butusov, and A. Tutueva, "Image encryption based on the pseudo-orbits from 1d chaotic map," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 6, p. 061101, 2019.

[20] S. M. Hammel, J. A. Yorke, and C. Grebogi, "Do numerical orbits of chaotic dynamical processes represent true orbits?" *Journal of Complexity*, vol. 3, no. 2, pp. 136 – 145, 1987.

[21] E. G. Nepomuceno, S. A. Martins, B. C. Silva, G. F. Amaral, and M. Perc, "Detecting unreliable computer simulations of recursive functions with interval extensions," *Applied Mathematics and Computation*, vol. 329, pp. 408 – 419, 2018.

[22] IEEE, "IEEE standard for floating-point arithmetic," *IEEE Std 754-2008*, pp. 1–70, 2008.

[23] R. E. Moore, *Methods and Applications of Interval Analysis*. Society for Industrial and Applied Mathematics, 1979.

[24] E. M. A. M. Mendes and E. G. Nepomuceno, "A very simple method to calculate the (positive) largest lyapunov exponent using interval extensions," *International Journal of Bifurcation and Chaos*, vol. 26, no. 13, p. 1650226, 2016.

[25] M. Kalpana, K. Ratnavelu, P. Balasubramaniam, and W. A. M. Othman, "Double-Key Secure for N-1-N Sound Record Data (SRD) by the Drive-Response of BAM NNs," *Neural Processing Letters*, vol. 50, no. 3, pp. 2925–2944, 2019.