Mathematical and Software Engineering, Vol 3, No 2 (2017), 183–189
Varεpsilon Ltd, varepsilon.com

Using Circle Map for Audio Encryption Algorithm

Krasimir Kordov and Lachezar Bonchev

Department of Computer Informatics, Faculty of Mathematics and Informatics, Konstantin Preslavski University of Shumen, 9712 Shumen, Bulgaria

Abstract

In this paper we propose audio encryption algorithm based on standard circle map. The proposed scheme contains bit level sample permutation using pseudorandom generator. Provided cryptanalysis includes number of tests demonstrating the security of proposed encryption algorithm.

Subject Codes: 68P25, 94A60

Keywords: encryption, audio encryption, wav encryption, encryption algorithm

1 Introduction

Digital data encryption is a process for securing stored and transmitted data. According to the specific type of data, encryption algorithm are designed to work in specific way.

In recent years multimedia files are often preferred as data carries in social networks, communications and Internet as general. Such files are audio files, video files and digital images. Therefore our focus is on multimedia files and more concrete on audio files. Securing audio files for safe storing and transmission requires designing of new encryption algorithms.

Most common approach for modeling encryption algorithms is bit level symmetric encryption using preudorandom generators(PRG) [8, 9, 11]. Pseudorandom bit generators are based on complex discrete mathematics or chaos based [5, 13, 14] but simple for programming realization. Their goal is to produce unlimited sequence of random bits therefor they can be used as building units for cryptographic [12] and steganographic [15, 16] algorithms.

Audio encryption algorithms require strong level of security therefor they are subject of detailed cryptanalysis. Various experiments can be performed for determining resistance of proposed algorithm to attacks, defining their level of security [3, 7].

2 Pseudorandom Bit Generator Based on Two Circle Maps and XOR Function

2.1 Circle Map Description

The Circle map [2] maps points on the circle back onto a circle. The Circle map is a nonlinear iterated map calculated by

$$\theta_{n+1} = (\theta_n + \Omega - \frac{K}{2\pi}\sin(2\pi\theta_n)) \mod 1, \tag{1}$$

where Ω is a constant that is the fixed angular progression of the sinusoidal oscillator, and K is the coupling strength.

2.2 Bit generation scheme

In our previous work we proposed a pseudorandom bit generator based on two circle maps and XOR function [4]. The proposed scheme is based on the following two Circle map equations:

$$\begin{aligned} \theta_{1,m+1} &= (\theta_{1,m} + \Omega_1 - \frac{K_1}{2\pi}\sin(2\pi\theta_{1,m})) \mod 1 \\ \theta_{2,n+1} &= (\theta_{2,n} + \Omega_2 - \frac{K_2}{2\pi}\sin(2\pi\theta_{2,n})) \mod 1, \end{aligned} \tag{2}$$

where $\theta_{1,m}$, Ω_1, K_1, $\theta_{2,n}$, Ω_2 and K_2 are the initial conditions. The new generator consists of the following steps:

Step 1: The initial values $\theta_{1,m}$, Ω_1, K_1, $\theta_{2,n}$, Ω_2 and K_2 of the two Circle maps from Eq. (2) are determined.

Step 2: The two chaotic maps from Eq. (2) are iterated for L_1 and L_2 times, respectively.

Step 3: The iteration of the Eq. (2) continues, and as a result, two real fractions $\theta_{1,i}$ and $\theta_{2,j}$, are generated and post-processed as follows:

$$\begin{aligned} s_{1,i} &= mod(integer(\theta_{1,i} \times 10^9)), 2) \\ s_{2,j} &= mod(integer(\theta_{2,j} \times 10^9)), 2), \end{aligned}$$

where $integer(x)$ returns the integer part of x, truncating the value at the decimal point, and $mod(x, y)$ returns the reminder after division.

Step 4: Perform XOR operation between $s_{1,i}$ and $s_{2,j}$ to get a single output bit.

Step 5: Return to Step 3 until the bit stream limit is reached.

Used initial values are: $\theta_{1,m} = 0.5$, $\theta_{2,n} = -0.25$, $\Omega_1 = \Omega_2 = 0.7128281828459045$, $K_1 = K_2 = 0.5$, $L_1 = L_2 = 200$.

Statistical tests were made for determining the necessary cryptographic security of the proposed generator in [4] and similar scheme can be found in [6, 10].

Key space includes all the initial values indicating the key space is more than 2^{179} [4] providing sufficient security against brute-force key search attacks [1].

3 Audio Encryption Scheme

In this section we present audio encryption algorithm using the pseudorandom generator described in 2.2. Produced random bits are extracted for sample permutation of audio files. Sample permutation is realized by using simple XOR function.

3.1 Encryption and Decryption Algorithm

Audio encryption algorithm consists of the following steps:

Step 1: Header bits from plain audio file A are transfered into file A' without cryptographic modifications.

Step 2: The bits in the sample are encrypted using XOR operation with the same amount of bits produsedby the pseudorandom generator from 2.2.

Step 3: Encrypted sample from Step 2 is transfered into file A'

Step 4: Repeat Steps 2-3 until end of plain file A is reached.

Step 5: The produced output file A' is the final encrypted audio file.

The proposed audio encryption method is implemented in programing language C++. Keeping the header bits in both plain and encrypted audio files allows us to perform further cryptanalysis tests by comparing the samples values of both files.

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

4 Cryptographic Analysis

The cryptanalysis determines the reliability of encryption algorithms. In this section we provide the results of empirical tests performed using proposed audio encryption algorithm in 3.1.

4.1 Waveform Plotting

Waveform plotting represents the amplitude of audio signal distributed in time. Figure 1 shows wave amplitude of plain audio file 1(a) and wave amplitude of the same file, after encryption 1(b).

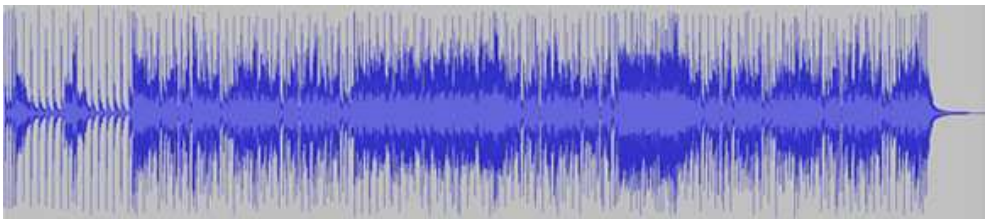
Simple visual analysis can determine that the two waves shown in 1 are completely different, indicating the good properties of the proposed audio encryption algorithm.

4.2 Correlation Analysis

Measuring correlation between samples values of plain and encrypted files is standard method for evaluating cryptographic algorithms. Calculating correlation coefficient determines the level of correlation between two files.

Correlation coefficient can be calculated as follows:

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{3}$$



(a) Plain audio file

(b) Encrypted audio file

Figure 1: Waveform of Plain audio file 1(a) and corresponding Encrypted audio file 1(b)

where

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

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

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

N is the number of samples processed from audio file (plain or encrypted), x_i and y_i are values of corresponding samples of both files, $\overline{x}$ and $\overline{y}$ are mean values of samples for each file, and finally $cov(x, y)$ is covariance between both files.

Table 1 shows some of the obtained result values from our tests. Experiments were made with different file size.

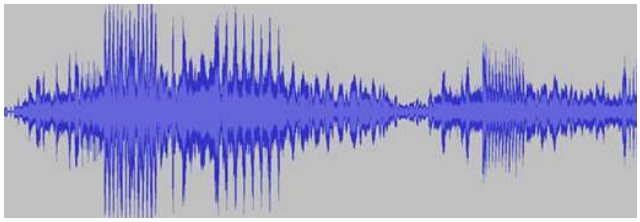
Plain Audio File	**Encrypted Audio File**	**File Size**	**Bytes in Sample**	**Correlation Coefficient**
Sample1.wav	encSample1.wav	100kb	2	-0.020451
Sample2.wav	encSample2.wav	150kb	2	0.049001
Sample3.wav	encSample3.wav	500kb	2	-0.013219
Sample4.wav	encSample4.wav	1MB	2	-0.018381
Sample5.wav	encSample5.wav	5MB	2	0.006555

Table 1: Correlation Between Plain and Encrypted Audio Files

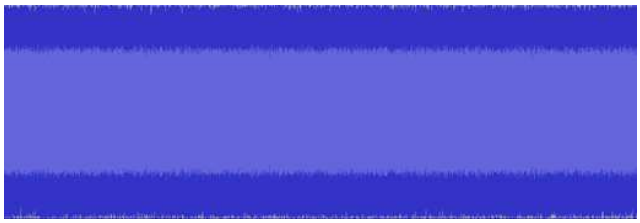
Results in Table 1 indicates that all values are close to zero which means there is no dependence between the two files. This demonstrates the good cryptographic properties of the proposed audio encryption algorithm.

4.3 Key Sensitivity Analysis

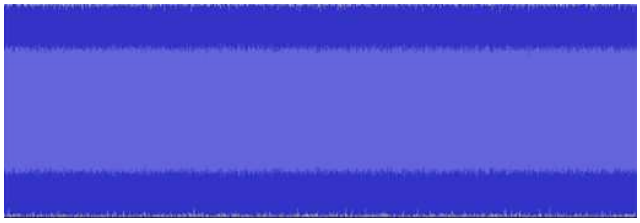
The secure encryption algorithms require strong resistance to secret key attacks. The following key sensitivity analysis provides encryption and decryption experiment using very similar secret keys. K_1 is the secret key used for encryption and it contains all the initial values of the parameters and variables for PRG in 2.2. Very similar secret key K_2 is used for decryption of the encrypted file. K_2 is obtained by slightly changing a single variable from K_1 and keeping all the other variables unchanged. $\theta_{2,n}$ for K_1 is -0.25 and for K_2 is -0.26.



(a) Plain audio file



(b) Encrypted audio file with K_1



(c) Decrypted audio file with K_2

Figure 2: Waveform of Plain audio file 2(a), encrypted file using K_1 2(b) and Decrypted file using K_2 2(c)

Figure 2 demonstrates unsuccessful attempt for decryption using very similar secret key. This indicates highly key sensitivity of the proposed audio encryption algorithm, proving resistibility against secret key attacks.

4.4 Speed Test Performance

The encryption time of cryptographic algorithms is important aspect for their evaluation. We have tested the encryption time for variety of files with different size to determine the algorithm encryption time. Testing computer configuration was 2.40 GHz Intel ® Core^{TM} i7-3630QM Dell Inspiron laptop. Part of the results are shown in the following Table 2

The results from Table 2 determine encryption speed $\approx$ 0.45 mb/sec which indicates fast time performance of the proposed audio encryption algorithm.

File Name	File Size	Bytes in Sample	Encryption Time
Sample1.wav	100kb	2	0.202s
Sample2.wav	150kb	2	0.320s
Sample3.wav	500kb	2	1.246s
Sample4.wav	1MB	2	2.230s
Sample5.wav	5MB	2	12.728s

Table 2: Speed Test Performance

5 Conclusion

New model of audio encryption is presented in this paper. The audio encryption algorithm is modeled by using preudorandom bit generator based on two circle maps. The proposed method permutates data in samples with bit level operation XOR using actual data in sample and binary sequence produced by pseudorandom generator.

Security analysis is provided by waveform plotting of plain and encrypted files, correlation coefficient is calculated for plain and encryption files, key sensitivity is tested, and speed test performance is measured. The obtained results demonstrate good cryptographic properties of the proposed audio encryption algorithm. Considering the results, we can conclude that the proposed algorithm is resistant to cryptographic attacks and can be used for audio encryption.

Acknowledgement

This work is partially supported by the Scientific research fund of Konstantin Preslavski University of Shumen under the grant No. RD-08-124/06.02.2017.

References

[1] Alvarez, G., Li, S., *Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems*, International Journal of Bifurcation and Chaos, 16 (2006), 2129–2151.

[2] Essl, G., *Circle Maps as a Simple Oscillators for Complex Behavior: I. Basics*, in ICMC, 2006, pp. 356359.

[3] Hato, E., Shihab, D., *Lorenz and Rossler Chaotic System for Speech Signal Encryption*, International Journal of Computer Applications, Vol 128, 2015, no. 11, 09758887.

[4] Kordov, K., *Modified pseudo-random bit generation scheme based on two circle maps and XOR function*, Applied Mathematical Sciences, Vol. 9, 2015, no. 3, 129-135, `http://dx.doi.org/10.12988/ams.2015.411887`.

[5] Kordov, K., *Signature Attractor Based Pseudorandom Generation Algorithm*, Advanced Studies in Theoretical Physics Vol. 9, 2015, no. 6, pp. 287-293, `http://dx.doi.org/10.12988/astp.2015.517`

[6] Malchev, D., Ibryam, I., *Construction of Pseudorandom Binary Sequences Using Chaotic Maps*, Applied Mathematical Sciences, Vol. 9, 2015, no. 78, 3847-3853, `http://dx.doi.org/10.12988/ams.2015.52149`.

[7] Sheu, L. J.,*A speech encryption using fractional chaotic systems*, Nonlinear Dynamics, Vol. 65 2011, no.1, 103-108.

[8] Stoyanov, B.P., *Chaotic cryptographic scheme and its randomness evaluation*, in 4th AMiTaNS'12, AIP Conference Proceedings, 1487, 397–404, 2012, http://dx.doi.org/10.1063/1.4758983.

[9] Stoyanov, B.P., *Pseudo-random bit generator based on Chebyshev map*, in 5th AMiTaNS'13, AIP Conference Proceedings, 1561 (2013), 369–372, http://dx.doi.org/10.1063/1.4827248.

[10] Stoyanov, B.P., *Using Circle Map in Pseudorandom Bit Generation*, in 6th AMiTaNS14, AIP CP 1629 (2014), 460 - 463, http://dx.doi.org/10.1063/1.4902309.

[11] Stoyanov, B., *Pseudo-random Bit Generation Algorithm Based on Chebyshev Polynomial and Tinkerbell Map*, Applied Mathematical Sciences, Vol. 8, 2014, no. 125, 6205–6210, http://dx.doi.org/10.12988/ams.2014.48676.

[12] Stoyanov, B., Kordov, K. *Image encryption using Chebyshev map and rotation equation*, Entropy vol. 17 (2015), no.4, 2117-2139.

[13] Stoyanov, B., Kordov, K., *Novel Zaslavsky Map Based Pseudorandom Bit Generation Scheme*, Applied Mathematical Sciences, Vol. 8, 2014, no. 178, 8883-8887, http://dx.doi.org/10.12988/ams.2014.410879.

[14] Stoyanov, B., Kordov, K., *Pseudorandom Bit Generator with Parallel Implementation*, In Large-Scale Scientific Computing 2014, Lecture Notes in Computer Science, vol. 8353, pp. 557-564, ISSN: 0302-9743

[15] Zhelezov S., *Modified Algorithm for Steganalysis, Mathematical and Software Engineering*, Vol. 1, No. 2 (2015), 31-36. ISSN 2367-7449

[16] Zhelezov, S., Paraskevov, H., *Possibilities for steganographic parallel processing with a cluster system*, Contemporary Engineering Sciences, Volume 8, Issue 20, 2015, ISSN:1313-6569