NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2024; 13(2), 540-549

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Novel true random bit generation and its audio encryption application with Lorenz chaotic circuit-based entropy source

Lorenz kaotik devre tabanlı entropi kaynağı ile özgün gerçek rasgele sayı üretimi ve ses şifreleme uygulaması

Esra İnce[1], Barış Karakaya[2,*], Mustafa Türk[3]

[1,2,3] *Fırat Unıversity, Faculty of Engineering, Department of Electrical-Electronics, 23200, Elazığ, Türkiye*

Abstract

This paper introduces a methodology for generating secure cryptographic key bits from analog values obtained from the Lorenz chaotic circuit. The analog values are transferred from the chaotic circuit to the computer via an Analog Discovery-2 device and then post-processed by using a fixed-point number representation format and Von-Neumann corrector for the sampled values on the MATLAB program. By employing this method, the analog values are digitized and randomized as the main idea is to obtain secure and efficient statistically random bits. Furthermore, the generated random bits are utilized for secure audio encryption by demonstrating a practical application of the proposed methodology. In addition to the classical randomness test criteria, NIST 800.22 statistical test suite, the throughput bit stream is also subjected to Chi-square and FIPS 140-1 tests to further evaluate its effectiveness. The results of these comprehensive tests confirm the successful performance of the proposed system in generating statistically random bits and its secure audio encryption system. The utilization of the Lorenz chaotic circuit as an entropy source in generating true random bits for secure audio transmission applications showcases the potential of the proposed system.

Keywords: Audio encryption; Chaos; Post-processor; Random number generator; Statistical tests

Öz

Bu makale, Lorenz kaotik devresinden elde edilen analog değerlerle güvenli kriptografik anahtar bitleri üretme yöntemini tanıtmaktadır. Analog değerler, kaotik devreden Analog Discovery-2 cihazı aracılığıyla bilgisayara aktarılır ve ardından MATLAB programında örnekleme değerleri için sabit noktalı sayı formatı ve Von-Neumann düzeltici kullanılarak işlenir. Bu yöntemi kullanarak, ana fikir güvenli ve verimli istatistiksel olarak rasgele bitler elde etmektir, bu nedenle analog değerler sayısallaştırılır ve rasgele hale getirilir. Ayrıca, önerilen yöntemin pratik bir uygulamasını göstererek üretilen rasgele bitler güvenli ses şifreleme için kullanılmaktadır. Klasik rasgelelik test kriteri olan NIST 800.22 istatistiksel test paketine ek olarak, üretilen bit dizisi ayrıca Chi-square ve FIPS 140-1 testlerine tabi tutularak etkinliği daha fazla değerlendirilmektedir. Bu kapsamlı testlerin sonuçları, önerilen sistemin istatistiksel olarak rasgele bitler üretme ve güvenli ses şifreleme sisteminde başarılı bir performans sergilediğini doğrulamaktadır. Lorenz kaotik devresinin bir rasgelelik kaynağı olarak kullanılması, önerilen sistemin potansiyelini göstermektedir.

Anahtar kelimeler: Ses şifreleme; Kaos; Son-işlemci; Rasgele sayı üreteci; İstatistiksel testler

1 Introduction

In recent years, chaos-based cryptography has gained increasing attention due to its inherent complexity and unpredictability while generating true random numbers [1]. Random number generation is a fundamental requirement in many cryptographic applications and secure communication systems. The quality and unpredictability of the generated random numbers play a crucial role in ensuring the security and reliability of this kind of system [2].

In the literature, chaos-based true random number generator (TRNG) designs are very popular at both discrete and continuous time chaotic systems are preferred as entropy sources of TRNG. In [3], a novel chaos-ring-based TRNG design proposed with high operating frequency and high throughput has been performed in favor of the FPGA platform. As an entropy source, the chaotic oscillator is preferred and the 32-bit IQ-Math fixed point number standard is examined in FPGA programming language. Gong et. al. in 2022 [4], proposed a new 4D chaotic system with hidden attractors and self-excited attractors and they used this chaotic system as an entropy source of RNG to encrypt images securely. In [5], the author has proposed a novel RNG in which the entropy source is a fractional order chaotic Chua system [6] that is one of the most and first popular chaotic systems in the literature. In this study, a fractional order version of the Chua chaotic system is used and the advantages of the fractional order computing technique are put forward.

Among various chaotic systems, the Lorenz system has emerged as a popular choice to be used as an entropy source of random number generator (RNG) designs. The Lorenz system is a three-dimensional autonomous system that

* Sorumlu yazar / Corresponding author, e-posta / e-mail: bkarakaya@firat.edu.tr (B. Karakaya)
Geliş / Recieved: 06.12.2023 Kabul / Accepted: 27.01.2024 Yayımlanma / Published: 15.04.2024
doi: 10.28948/ngumuh.1401243

exhibits chaotic behavior at specific parameter configurations of the differential equations characterized by its sensitivity to initial conditions and the presence of a strange attractor [7]. This chaotic behavior makes the Lorenz system an attractive candidate for random number generation applications. Several studies are using the Lorenz system itself [8], its discrete-time implementation on FPGA [9], and an improved version of the system [10]. Especially, in all studies, complex and high flexibility properties of the Lorenz system are attractive for it in cryptographic applications such as optimization for initial value space [11], secure random bit generation [12], and image encryption [13].

In this context, we propose a novel method for generating true random bits (TRB) from the output of a continuous-time integer-order Lorenz chaotic circuit [14]. The analog values of the state variables of the system are sampled and stored by using the Analog Discovery-2 device and transferred to the computer for post-processing in MATLAB. We employ the fixed-point number representation format and Von-Neumann corrector for the sampled values to obtain the cryptographic key bits from the digitized signals, which provides a good balance between accuracy and computational complexity.

The obtained true random bits (TRB) are then utilized for secure audio communication by encrypting and decrypting any public audio data. The proposed system provides a high level of security, as the key generation process is based on the inherent randomness and complexity of the chaotic system [15]. The use of real analog values of state variables from the chaotic circuit provides an additional layer of security to the generated random bits as they are inherently unpredictable and difficult to reproduce without the post-processor algorithm. Furthermore, the usage of the fixed-point number representation format and Von-Neumann corrector together is a novelty for digitizing any analog values. Overall, the proposed TRB generation system and the post-processor algorithm are utilized for the potential of secure audio transmission application.

Herewith this introduction; numerical analysis, chaotic behavior, and the electronic circuit implementation of the Lorenz system are detailed in Section 2. The structure of the proposed post-processor algorithm, TRB design with Lorenz chaotic circuitry, and audio encryption application is given in Section 3 where entropy source, fixed-point number representation format-based digitizer algorithm, Von-Neumann corrector, and XOR process for encryption are detailed. The statistically randomness of the proposed TRB design is proven with NIST, Chi-Square, and FIPS 140-1 tests, and histogram analysis of the proposed audio encryption application is examined in Section 4. In the end, future works and conclusions are discussed.

2 Material and methods

2.1 Entropy source: Integer order chaotic Lorenz system and its electronic circuit

The Lorenz chaotic system refers to a mathematical model that exhibits chaotic behavior over continuous time. It is based on the Lorenz system, which was originally introduced by Edward Lorenz in 1963 as a simplified model of atmospheric convection [7]. The Lorenz system consists of a set of three nonlinear ordinary differential equations that describe the evolution of three variables: x, y, and z. These variables represent the state of the system and describe its position in the three-dimensional phase space;

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{1}$$

where $\sigma, \rho,$ and β are called control parameters [16]. The chaotic behavior (*the strange attractor*) is observed when (σ, ρ, β)=(10, 28, 8/3) and the initial conditions (x_0, y_0, z_0) = (0.02, 0.02, 0.02) as shown in Figure 1. The behavior of the Lorenz chaotic system is characterized by sensitivity to initial conditions, meaning that small changes in the initial state can lead to significantly different trajectories over time. This property is one of the defining features of chaotic systems.

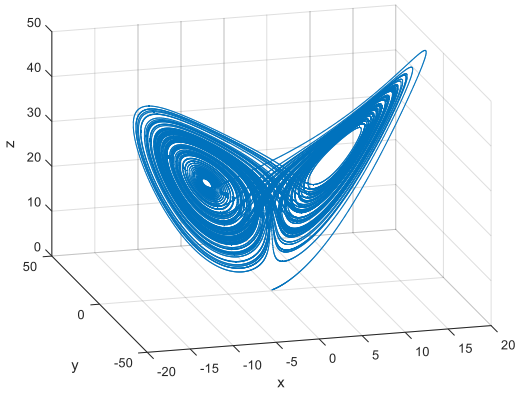

Figure 1. 3D phase portrait of Lorenz system; the strange attractor

Chaos dynamical analysis is a branch of study within the field of dynamical systems that explores the fascinating behavior of chaotic systems. Chaos refers to a complex and seemingly random behavior that arises from deterministic systems characterized by sensitivity to initial conditions. By using Lyapunov exponents and bifurcation diagrams for the dynamic analysis of the system, the study can gain insight into the nonlinear properties of the system and understand its chaotic behavior. For example, positive Lyapunov exponents suggest that the system is exhibiting chaotic behavior, and bifurcation diagrams can reveal how the system's behavior changes as a parameter is varied.

Since the Lorenz system has three state variables, and the system has three Lyapunov exponents. These exponential values were obtained as $\lambda_1 = 0.783, \lambda_2 = 0.014,$ and $\lambda_3 = -14.47$ at steady state as shown in Figure 2. a. In the system for which three or more Lyapunov exponential values are calculated, if the condition $\lambda_1 > 0, \lambda_2 \cong 0, \lambda_i < 0$ is met for $i = 3 \ldots n$, it is said to exhibit chaotic behavior. Furthermore, how the system changes despite the parameter change is observed with the bifurcation diagram. The behavior of the system despite the change of the β parameter for the z state variable is shown in Figure 2. b.

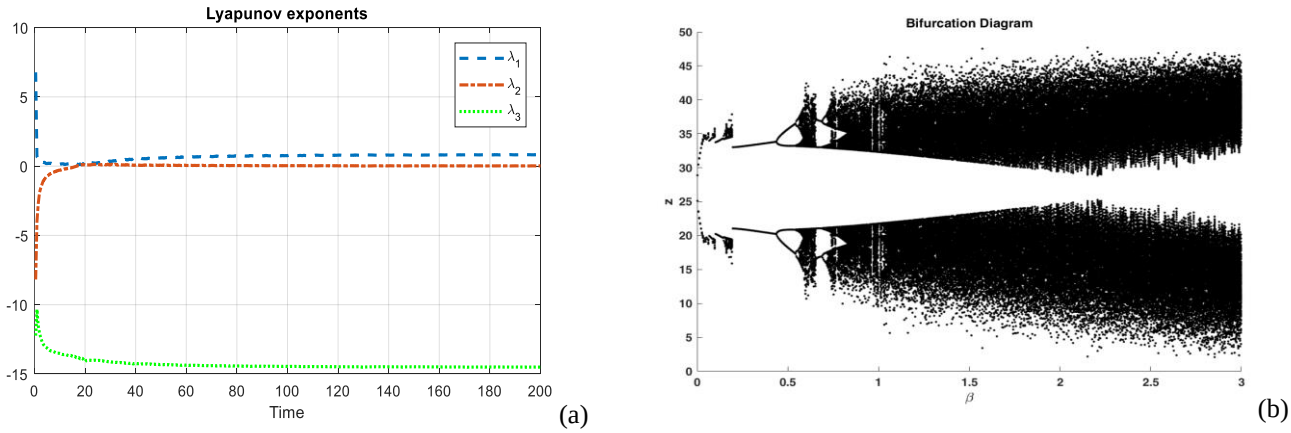

Figure 2. a) Variation of Lyapunov exponents, b) bifurcation diagram for the interval β ∈ [0,3] of the Lorenz system

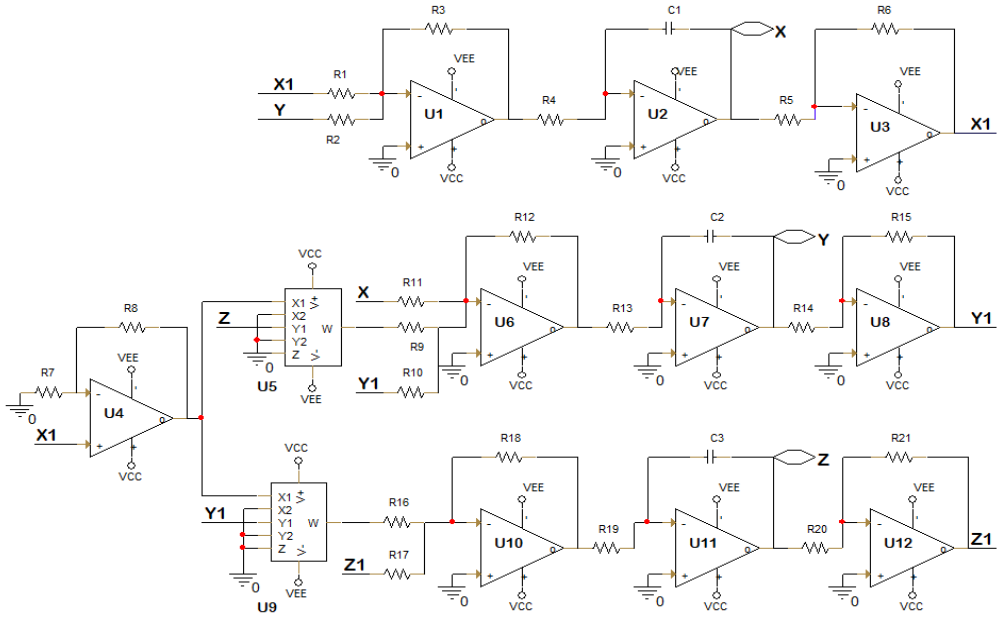

Figure 3. Electronic circuit realization of the Lorenz chaotic system in Orcad-Pspice environment. Components: $R_1 = R_2 = R_8 = 10$ kΩ, $R_3 = R_5 = R_6 = R_{10} = R_{12} = R_{14} = R_{15} = R_{18} = R_{20} = R_{21} = 100$ kΩ, $R_4 = R_7 = R_{13} = R_{19} = 1$ kΩ, $R_9 = R_{11} = R_{16} = 3.3$ MΩ, $R_{17} = 39$ kΩ, $C_1 = C_2 = C_3 = 220$ nF, $V_{CC} = -V_{EE} = 15$ V; U_5 and U_9 are AD633 analog multipliers where the other integrated circuits are TL082 general purpose operational amplifiers

Thus, when the three-dimensional change of the system, bifurcation diagram, and Lyapunov exponents are examined, it is seen that the Lorenz system exhibits chaotic behavior by the definition of strange attractive.

In the proposed TRB design, the Lorenz system is used as an entropy source to obtain a chaos-based random bit stream. This study aims to utilize the Lorenz chaotic system for generating true random bits. Hence, the electronic circuitry for the system is designed using the Orcad-Pspice environment, and the corresponding configuration is illustrated in Figure 3. In this circuit design, regarding to the power supply limitation of the circuit, the Lorenz system differential equations are scaled as $\frac{1}{20}$, $\frac{1}{20}$, $\frac{1}{50}$ for the state variables *x*, *y* and *z* respectively. Active circuit elements such as TL082 operational amplifiers, AD633 analog multiplier circuits, and passive circuit components (resistor and capacitor) are used to implement the equations of the chaotic systems [17, 18].

The initial condition of the state variables for each circuitry is applied as the initial voltage value of integration capacitors on the circuit. The simulation outcomes of electronic circuitries are exported from the Orcad-Pspice environment to the computer to plot and analyze the behavior of the state variables. The electronic circuit implementation of a Lorenz chaotic system is achieved within the Orcad-Pspice environment. After obtaining the desired results by analyzing the attractor observation, the electronic circuit board is designed finally on board. To gain insights into the dynamics and behavior of the Lorenz chaotic system, the outputs of each dimension are visually displayed utilizing high-speed oscilloscopes. The experimental setup allowed

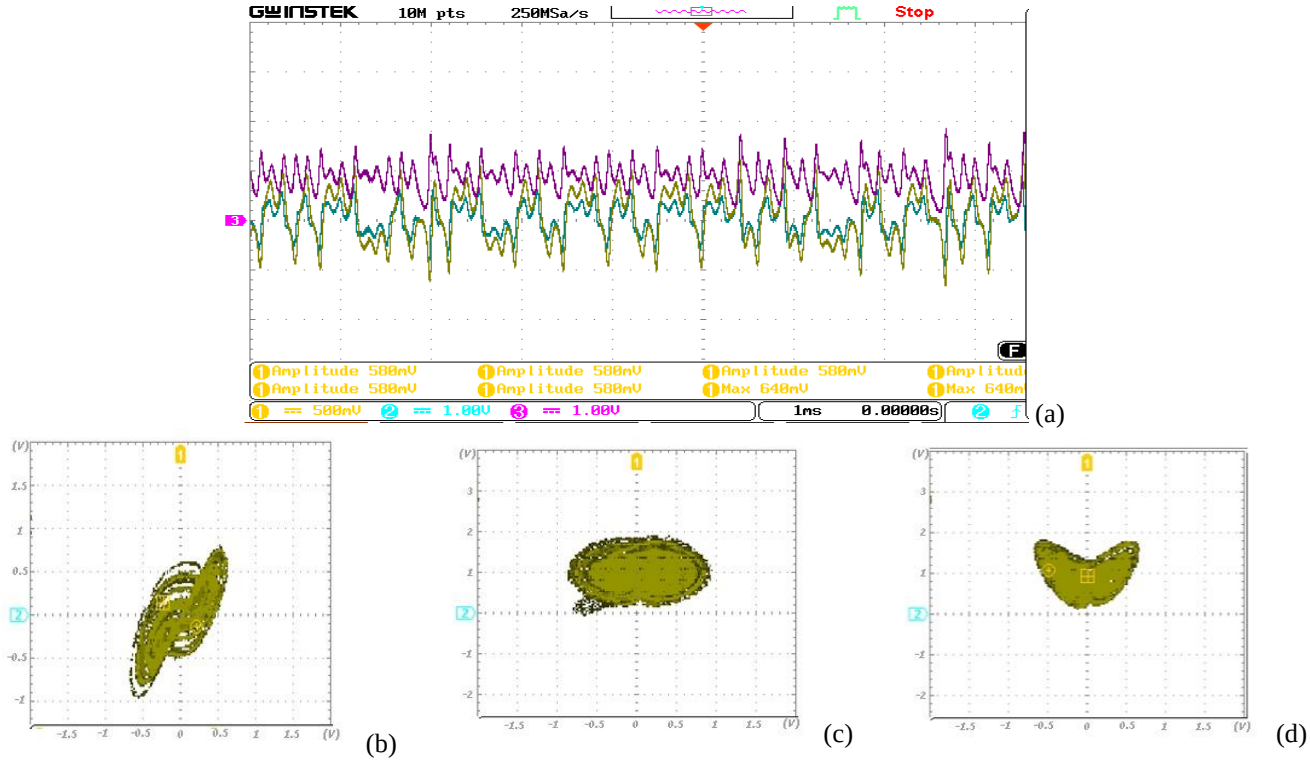

Figure 4. The oscilloscope outputs display a) the time domain representation of state variables and b) x-y, c) x-z, d) y-z phase portraits of the Lorenz chaotic circuit

for the direct observation and verification of the anticipated chaotic behavior, thereby affirming the precision of the circuit design. The circuit design incorporates the necessary components and connections to accurately emulate the behavior of the chaotic system. Furthermore, Figure 4 displays the time domain representation of the state variables and the phase portraits of Lorenz chaotic system measured from the electronic circuit, providing a visual representation of the dynamic behavior in phase space. These phase portraits depict the trajectories and attractors associated with the Lorenz chaotic system, enabling a deeper understanding of its complex dynamics. Together, these figures illustrate the successful implementation of the electronic circuit and provide valuable insights into the behavior of the Lorenz chaotic system.

2.2 Proposed TRB design, statistical test results, and audio encryption system

The proposed TRB design and its procedure for audio encryption and decryption are illustrated in Figure 5. As an entropy source of TRB design, Lorenz chaotic circuit outputs, specifically the x and y outputs, are subjected to quantization in MATLAB due to their analog values ranging from negative to positive. The scaled x and y state variables of the system observed from the electronic circuit vary *-1* to *+1*.

The fixed-point binary conversion algorithm is chosen as the quantization method as digitizer algorithm on MATLAB where each analog value consists of 16 bits. Out of these 16 bits, 1 bit is assigned for the sign, 1 bit for the integer part, and the remaining 14 bits for the fractional part of the analog value. A total of 2 million values are obtained from the Lorenz circuit board at a frequency of 25 kHz and a period of 40 microseconds.

Each state variable of the Lorenz system (x and y) is inputted to a fixed-point binary conversion algorithm. For each analog value input, a 10-bit output is obtained with the sample number as 4. Thus, a total of 5 million serial bits are obtained for both the x and y outputs. These bits are further processed using the Von-Neumann corrector. As a result, 1074403 serial bits are obtained from the x output and 1052726 serial bits from the y output through the final processing algorithm. By passing these obtained digital bits through an XOR gate, a secure key of 1052726 bits is derived. To ensure the confidentiality and security of the audio information, and make it resistant to unauthorized access or deciphering, the TRB has to pass all statistical tests such as NIST, FIPS 140-1, and Chi-Square tests.

2.2.1 NIST statistical test suite

The effectiveness of the proposed true random bit generator design, TRB has to pass some popular statistical tests. Therefore, the generated bit stream is applied to NIST 800.22 statistical randomness tests [19, 20]. Table 1 presents the results of the NIST tests conducted on the original audio data generated TRB and encrypted audio. According to the test results, the original audio data passed only two of the tests, indicating a poor behavior of randomness. On the other hand, both the generated TRB and the encrypted audio successfully passed all the statistical tests, demonstrating a high degree of randomness and robustness against cryptographic attacks. The statistical validation of the

randomness of the bit stream provides strong evidence for its quality and suitability in cryptographic applications.

2.2.2 FIPS 140-1 tests

FIPS 140-1 is a standard that establishes criteria for evaluating the security of cryptographic modules. The test requirements defined in FIPS 140-1 aim to ensure that the modules demonstrate sufficient levels of randomness and unpredictability. There are four different tests in the module.

In the Poker Test, a 20.000-bit block is tested by dividing it into 5.000 consecutive 4-bit chunks. The occurrences of each of the 16 possible chunks (with 15 degrees of freedom) are counted, and they are represented by *f(i)*. To evaluate the test, *X* is calculated using the following equation:

$$X = \left(\frac{16}{500}\right) * \sum (f(i)^2) - 5000 \qquad (2)$$

In this equation, *f(i)* represents the occurrences of each value for the 4-bit chunks. The summation (∑) is performed over all the 16 possible chunks. Once *X* is calculated, it is compared to the specified ranges to determine the outcome of the test. For the FIPS 140-2 test to be considered passed, *X* must fall within the range of 2.16 to 46.17. On the other hand, FIPS 140-1 has slightly different requirements, requiring X to be within the range of 1.03 to 57.4. Both versions use the same equation to evaluate the test, but they have distinct passing criteria.

The Monobit Test involves counting the number of 1's in a 20.000-bit block. Let's denote the number of 1's as *Y*. For FIPS 140-1, the test is passed if *Y* is between 9,654 and 10.346. The goal of this test is to ensure that the number of 1's in the block is within the specified ranges, indicating a satisfactory distribution of ones and zeros [21].

The Runs Test examines the presence of consecutive sequences, known as runs, consisting of either 1's or 0's within a 20.000-bit block. This test count runs with lengths ranging from 1 to 6. The passing intervals for this test are specified in Table 2 for the FIPS 140-1 version of the test. The purpose of this test is to evaluate the distribution and occurrence of runs within the block, ensuring they fall within the designated passing intervals as defined in the FIPS 140-1 standard [22].

The Long Run Test examines the presence of extended sequences, known as long runs, within a 20.000-bit block. In the context of FIPS 140-1, a long run is defined as a consecutive series of identical values exceeding 34 bits. If the number of consecutive identical values exceeds this threshold, the test is considered failed under FIPS 140-1.

These tests, along with other requirements, ensure that cryptographic modules conform to the security standards defined by FIPS 140-1. In Table 2, we can find the conditions and results of all the tests associated with this test module. The proposed design has successfully passed all the tests present in the FIPS 140-1 module.

2.2.3 Chi-Square test

The chi-square test is a statistical test used to determine the significance of the relationship between two categorical variables. It compares observed and expected frequencies and calculates a chi-square statistic. This statistic measures the deviation from randomness. By comparing the chi-square statistic to a table of critical values, a p-value is obtained. If the p-value is below a predetermined significance level (e.g., $\alpha = 0.05$), the null hypothesis is rejected, indicating a significant relationship.

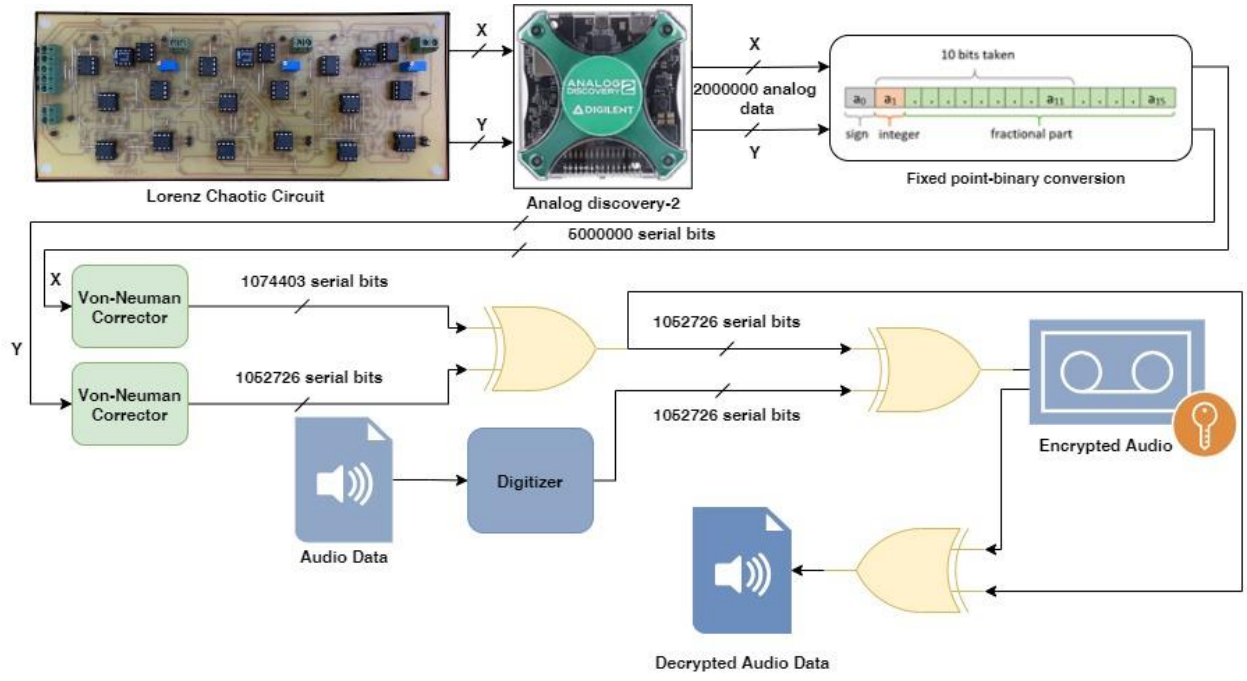

Figure 5. Experimental setup for audio encryption

2.2.4 *Chi-Square test*

If the p-value is above the significance level, the null hypothesis is accepted, suggesting no relationship [23].

In this study, the chi-square test is applied to the entire bit sequence with a length of 1052726. The test aimed to determine if the sequence exhibited randomness. The obtained p-value was 0.42530455279488244, indicating that there is no significant deviation from randomness. Additionally, the chi-square statistic is calculated as 0.6356111318508977. Based on these findings, we cannot reject the null hypothesis, indicating that the bit sequence can be considered random.

2.2.5 *Encryption process*

After the confirmed randomness, a simple audio encryption process is performed, utilizing the generated true random bit stream to encrypt the audio data securely. The original audio source (the mp3 version of Beethoven's 9th symphony) contains 5998269 analog values ranging from 0 to 255 after the digitizing process on MATLAB. After digitizing the audio source, a portion of the resulting digital audio data, specifically 1052726 bits, is selected as the length of the generated TRB and used for encryption. The encryption process consists of logical XOR operation of digitized audio data and the generated true random bits which is statistically proven by the means of randomness. The whole encryption algorithm and MATLAB codes are given in Table 3. In Figure 6, MATLAB outputs are provided for the digitized original audio data, the generated TRB, the encrypted digital data, and the decrypted audio data, respectively. To enhance the visual representation, 1000 bits from each source are selected and displayed in the figures. These samples allow for a more meaningful interpretation of the transformations and operations performed on the audio data during the encryption and decryption processes. After the decryption stage, it is better to determine the bit error rate (BER) by using the original digital audio data and decrypted digital audio data in order to obtain how much data has been changed after decryption process. The expected BER is 0 if the encryption application is executed successfully. The BER is determined by following equation;

$$BER = \frac{1}{M}\sum_{k=1}^{M}\frac{HD(r_k, \dot{r_k})}{n} * \%100 \quad (3)$$

where Mstands for digital data length, HD for hamming distance, r_k for *k-th* bit of the original digital data and $\dot{r_k}$ for *k-th* bit of the decrypted digital data. The encryption-decryption system can successfully recover the original audio data with zero error as the BER is determined *0*.

Table 1. NIST test results for proposed TRB design

NIST TESTS (1052726 BIT)	P- value Original Audio	Result	P- value TRB	Result	P- value Encrypted Audio	Result
Frequency (monobit) Test	-	Failed	0.4253	**Passed**	0.7154	**Passed**
Frequency Test within a Block	-	Failed	0.901	**Passed**	0.9415	**Passed**
Runs Test	-	Failed	0.5923	**Passed**	0.1148	**Passed**
Test for the Longest Run of Ones in a Block	-	Failed	0.0614	**Passed**	0.725	**Passed**
Binary Matrix Rank	0.4805	**Passed**	0.2222	**Passed**	0.9715	**Passed**
Discrete Fourier Transform Test	-	Failed	0.9672	**Passed**	0.2678	**Passed**
Non-overlapping Template Matching Test	-	Failed	0.0691	**Passed**	0.4942	**Passed**
Overlapping Template Matching Test	-	Failed	0.5482	**Passed**	0.4548	**Passed**
Maurer's Universal Statistical Test	-	Failed	0.0488	**Passed**	0.4351	**Passed**
Linear Complexity Test	-	Failed	0.2903	**Passed**	0.12	**Passed**
Serial Test 1/2	-	Failed	0.55 / 0.74	**Passed**	0.93 / 0.886	**Passed**
Lempel Ziv Test	75350	**Passed**	77710	**Passed**	77815	**Passed**
Approximate Entropy Test	-	Failed	0.3268	**Passed**	0.8088	**Passed**
Cumulative Sums Test	-	Failed	0.4377	**Passed**	0.668	**Passed**
Random Excursions Test	-	Failed	0.3271	**Passed**	0.6227	**Passed**
Random Excursions Variants Test	-	Failed	0.7311	**Passed**	0.6288	**Passed**

Table 2. FIPS 140-1 test result

FIPS 140-1 TESTS	Criteria for Success	Values	Result
Monobit Test	9654 < x < 10346	10163	**Successful**
Poker Test	1.03 < x < 57.4	16.454400000000533	**Successful**
Long Run Test	dvalue < 34	3	**Successful**
Runs Test-0	For 0 [2267-2733,1079-1421,502-748,223-402,90-223,90-223]	[2395, 1223, 677, 320, 151, 167]	**Successful**
Runs Test-1	For 1 [2267-2733,1079-1421,502-748,223-402,90-223,90-223]	[2496, 1194, 596, 315, 182, 151]	**Successful**

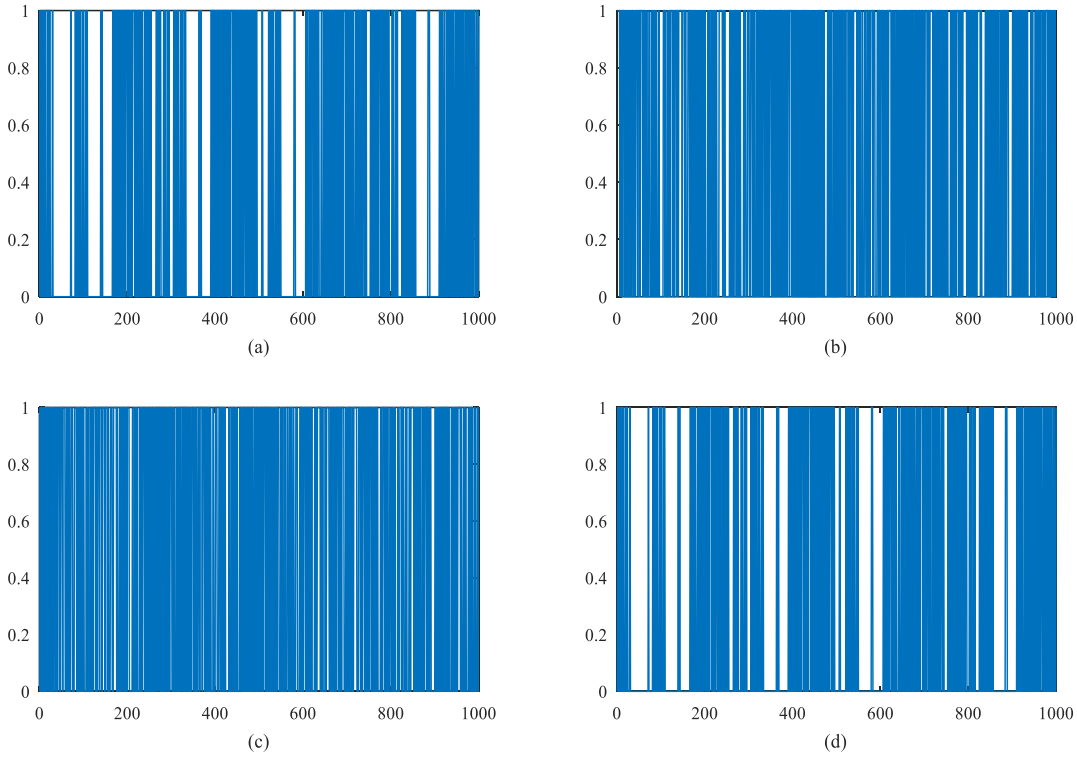

Figure 6. a) Original audio digital data (1000 bit) b) True random digital data (1000 bit) c) Encrypted digital data (1000 bit) d) Decrypted digital data (1000 bit)

3 Results and discussion

The most popular analysis method that demonstrates the effectiveness of the design is histogram analysis. The distribution of the audio data can be displayed using histogram analysis. Flat audio data has an uneven and centralized distribution of values. The histogram of audio data that has been encrypted using the encryption algorithm is anticipated to have a smooth distribution of values. The performance of encryption and attack vulnerability increase with the smoothness of the histogram of the encrypted audio data. In Figure 7, histogram analyses are presented for the original audio data, the generated TRB, the encrypted audio data, and the decrypted audio data, respectively. Upon examining these graphs, the unique distributions observed for the generated TRB and encrypted audio data indicate the success of the proposed audio encryption algorithm. Furthermore, the similarity between the histograms of the original audio data and the decrypted audio data signifies a successful decryption process. The consistency in histogram shapes validates the accuracy of the decryption operation.

In Table 4, a comparison of our proposed system and the TRNGs in the literature is given. When examining different TRNGs (True Random Number Generators) with various entropy sources, it is observed that the designs are implemented using simulation, FPGA, and analog circuits. In these studies, the randomness of the generated true random bits is mostly validated through the NIST test suite. In this study, the design is implemented using an analog circuit, and the randomness of the generated bits is demonstrated through several different tests. In addition to testing the generated bits, histogram analysis is performed on the encrypted and decrypted audio.

Table 3. Proposed encryption algorithm procedure

***Step* 1:** Chaotic Lorenz *y* state variable and its digitizing process codes on MATLAB, the same code is applied for the *x* state variable as well which is transferred from the electronic circuit to the PC via Analog Discovery-2.

```
q = quantizer('fixed',[16 14]);
j=1; dim=length(y);

          for k=1:4:dim
s_y= num2hex( q, y(k) );
y_bin(j,:)= hex2bin(q, s_y);
j=j+1;
          end

[a b]=size(y_bin);

          for i=1:1:a
yy=logical(y_bin(i,:)'-'0');
y_final(((i-1)*(b-6)+1):i*(b-6),1)=yy(2:11);
          end

y_final=double(y_final);
```

***Step* 3:** Digitizing the plain audio data

```
fid=fopen('beethoven9thsymphony.mp3','r');
mp3_content=fread(fid, [1 inf], '*uint8');
fclose(fid);
binaryform=dec2bin(mp3_content);
binaryform2=binaryform((1:132000),:);
[a b]=size(binaryform2);

          for i=1:1:a
reg=logical(binaryform2(i,:)'-'0');
mp3_bin(((i-1)*(b)+1):i*(b),1) = reg;
          end

mp3_bin_son=double(mp3_bin(1:length(TRB)));
```

***Step* 2:** Pseudo code implementation of Von-Neumann corrector algorithm for Lorenz *y* state variable on MATLAB, the same code is applied for the *x* state variable as well.

```
dim=length(y_final);
k=1;
          for j=1:2:dim
sel(j)= y_final (j);
sel(j+1)= y_final (j+1);

          if sel(j) < sel(j+1)
key_y(k)=0;
k=k+1;
          elseif sel(j) > sel(j+1)
key_y(k)=1;
k=k+1;
          else
k=k;
          end
          end
key_y=key_y';
```

***Step* 4:** True random bit generation process by using XOR and the output of ***Step* 2** for both *x* and *y* state variables. Encryption and decryption of digitized audio data with TRB

```
dim=length(key_y);
          for i=1:dim
TRB(i)= xor(key_x(i),key_y(i));
          end
TRB=double(TRB');

         for i=1:dim
encrypted(i)= xor(TRB(i), mp3_bin_son (i));
         end

encrypted=double(encrypted');

         for i=1:dim
        decrypted(i)= xor(TRB (i),encrypted(i));
         end
decrypted=double(decrypted');
```

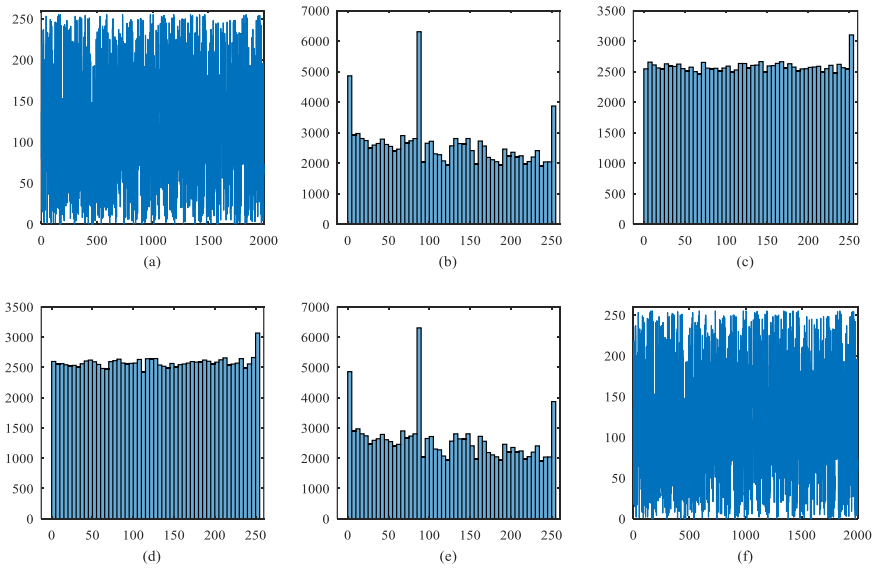

Figure 7. a, b) Original audio and its histogram c) TRB histogram d) Encrypted audio histogram e, f) Decrypted audio and its histogram

Table 4. Literature comparison for TRNG designs

Reference	Generator Type	Entropy Source	Realization Type	Post-processor	Tests
[24]	TRNG	SAR Residue	Analog circuit	-	NIST
[25]	TRNG	Chaotic Map	FPGA	-	NIST
[26]	TRNG	EM Waves	FPGA	-	Online health test
[27]	TRNG	PLL period jitter	FPGA	Yes	NIST, Kolmogorov-Smirnov test
[28]	TRNG	three-terminal magnetic tunnel junction	Simulation	-	-
[29]	TRNG	3T-MTJ	Simulation		NIST
[30]	TRNG	Chaotic Map	Analog Circuit	Yes	NIST
[31]	TRNG	Bistable Josephson Junction	Simulation	-	NIST
[32]	TRNG	Audio Signal	Simulation	-	NIST, TestU01
Proposed design	TRNG	Chaotic Map	Analog Circuit	Yes	NIST, FIPS 140-1, Chi-Square tests

4 Conclusions

In this study, audio encryption is successfully performed using the Lorenz chaotic circuit. The analog values obtained from the circuit board output contributed to generating TRB. The reliability of the generated key bits is verified through NIST 800.22 statistical randomness tests, Chi-Square, and FIPS 140-1 tests. The results obtained from this implementation demonstrate the versatility of the proposed design, which can be applied not only to ensure secure communication but also to address information security requirements in various domains. Future research aims to fully implement secure key generation in a hardware-based approach and conduct more independent studies that are less reliant on software environments. These advancements will offer new possibilities in information security and contribute to the development of more reliable communication systems.

Acknowledgement

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) Project Number: 121E210.

Conflict of Interest

There is no conflict of interest.

Similarity Rate (Turnitin): 14%

References

[1] V. Lynnyk, N. Sakamoto, S. Čelikovský, "Pseudo-random number generator based on the generalized Lorenz chaotic system," IFAC-PapersOnLine, 48 (18), pp. 257-261, 2015. https://doi.org/10.1016/j.ifacol.2015.11.046

[2] M. D. Gupta, R. K. Chauhan, "Hardware efficient pseudo-random number generator using Chen chaotic system on FPGA," Journal of Circuits, Systems and Computers, 31 (3), 2250043, 2022. https://doi.org/10.1142/S0218126622500438

[3] İ. Koyuncu, et al., "Design, FPGA implementation and statistical analysis of chaos-ring based dual entropy core true random number generator," Analog Integrated Circuits and Signal Processing, 102, pp. 445-456, 2020. https://doi.org/10.1007/s10470-019-01568-x

[4] L.-H. Gong, et al., "New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG," Physica A: Statistical Mechanics and its Applications, 591, 126793, 2022. https://doi.org/10.1016/j. physa. 2021.126793

[5] F. Ozkaynak, "A novel random number generator based on fractional order chaotic Chua system," Elektronika ir Elektrotechnika, 26 (1), pp. 52-57, 2020. https://doi.org/10.5755/j01.eie.26.1.25310

[6] L. O. Chua, "Chua's circuit: An overview ten years later," Journal of Circuits, Systems, and Computers, 4 (2), pp. 117-159, 1994. https://doi.org/10.1142/S0218126694000090

[7] E. N. Lorenz, "Deterministic nonperiodic flow," Journal of Atmospheric Sciences, 20 (2), pp. 130-141, 1963. https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

[8] M. S. Azzaz, et al., "Design and FPGA implementation of TRNG based on a new multi-wing attractor in Lorenz chaotic system," The European Physical Journal Special Topics, 230 (18), pp. 3469-3480, 2021. https://doi.org/10.1140/epjs/s11734-021-00234-6

[9] A. A. Rezk, et al., "Reconfigurable chaotic pseudo random number generator based on FPGA," AEU-International Journal of Electronics and Communications, 98, pp. 174-180, 2019. https://doi.org /10.1016/j.aeue.2018.10.024

[10] C. Zou, et al., "Image encryption based on improved Lorenz system," IEEE Access, 8, pp. 75728-75740, 2020. https://doi.org/10.1109/ACCESS.2020.2988880

[11] G. Yildirim, E. Tanyildizi, "An innovative approach based on optimization for the determination of initial conditions of continuous-time chaotic system as a random number generator," Chaos, Solitons & Fractals,

172, 113548, 2023. https://doi.org/10.1016/j.chaos.2023.113548
[12] B. Aricioğlu, S. Kaçar, "Circuit Implementation and PRNG Applications of Time Delayed Lorenz System," Chaos Theory and Applications, 4 (1), pp. 4-9, 2022. https://doi.org/10.51537/chaos.976593
[13] B. Karakaya, "Chaotic System-based Pseudo Random Bit Generator and Post-processor Design for Image Encryption," in Proc. 2022 13th National Conference with International Participation (ELECTRONICA), IEEE, 2022.
[14] C. García-Grimaldo, et al., "FPGA Implementation of a Chaotic Map with No Fixed Point," Electronics, 12 (2), 444, 2023. https://doi.org/10.3390/electronics12020444
[15] S. M. Basha, P. Mathivanan, A. B. Ganesh, "Bit level color image encryption using Logistic-Sine-Tent-Chebyshev (LSTC) map," Optik, 259, 168956, 2022. https://doi.org/10.1016/j.ijleo.2022.168956
[16] Y. Yu, et al., "Dynamic analysis of a fractional-order Lorenz chaotic system," Chaos, Solitons & Fractals, 42 (2), pp. 1181-1189, 2009. https://doi.org/10.1016/j.chaos.2009.03.016
[17] M. Preishuber, et al., "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," IEEE Transactions on Information Forensics and Security, 13 (9), pp. 2137-2150, 2018. https://doi.org/10.1109/TIFS.2018.2812080
[18] Q. Lai, L. Wang, "Chaos, bifurcation, coexisting attractors and circuit design of a three-dimensional continuous autonomous system," Optik, 127 (13), pp. 5400-5406, 2016. https://doi.org/10.1016/j.ijleo.2016.03.014
[19] B. Karakaya, A. Gülten, M. Frasca, "A true random bit generator based on a memristive chaotic circuit: Analysis, design and FPGA implementation," Chaos, Solitons & Fractals, 119, pp. 143-149, 2019. https://doi.org/10.1016/j.chaos.2018.12.021
[20] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, S. Vo, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.
[21] D. Hurley-Smith, C. Patsakis, J. Hernandez-Castro, "On the unbearable lightness of FIPS 140–2 randomness tests," IEEE Transactions on Information Forensics and Security, 17, pp. 3946-3958, 2020. https://doi.org/10.1109/TIFS.2020.2988505
[22] S. Çiçek, "Design and Implementation of an FPGA based Chaotic Communication System with a New Chaotic System," Ph.D. Dissertation, Sakarya University, Turkey, 2016.
[23] A. Vardasbi, M. Salmasizadeh, J. Mohajeri, "Multiple-chi-square tests and their application on distinguishing attacks," in Proc. 2011 8th International ISC Conference on Information Security and Cryptology, IEEE, 2011. https://doi.org/10.1109/ISCISC.2011.6062336
[24] A. Jayaraj, et al., "0.6–1.2 V, 0.22 pJ/bit true random number generator based on SAR ADC," IEEE Transactions on Circuits and Systems II: Express Briefs, 67 (10), pp. 1765-1769, 2019. https://doi.org/10.1109/TCSII.2019.2949775
[25] C. Wannaboon, P. Ketthong, "A Simple Random-Bit Generator Implemented on FPGA Based on Signum Chaotic Map," in 2022 International Conference on Digital Government Technology and Innovation (DGTi-CON), IEEE, 2022.
[26] S. Osuka et al., "A Study on Output Bit Tampering of True Random Number Generators Using Time-Varying EM Waves," in 2021 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), IEEE, 2021.
[27] G. D. P. Stanchieri et al., "An FPGA-Based Architecture of True Random Number Generator for Network Security Applications," in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2018.
[28] S. Mukaida, N. Onizawa, and T. Hanyu, "Design of a low-power MTJ-based true random number generator using a multi-voltage/current converter," in 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL), IEEE, 2018.
[29] A. Tamakoshi, et al., "Design of an energy-efficient true random number generator based on triple read-write data-stream multiplexing of MTJ devices," in 2020 18th IEEE International New Circuits and Systems Conference (NEWCAS), IEEE, 2020.
[30] C. Wannaboon, P. Ketthong, and W. San-Um, "On-Chip True-Random Bit Generator Through a Robust Tent-Based Chaotic Map," in 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), IEEE, 2019.
[31] E. Elmitwalli and S. Köse, "Bistable Josephson Junction-Based True Random Number Generator Without Inductors," in IEEE Transactions on Circuits and Systems II: Express Briefs, 70 (4), pp. 1615-1619, 2022. https://doi.org/10.1109/TCSII.2022.3226166
[32] T. Etem and T. Kaya, "Self-generated encryption model of acoustics," Applied Acoustics, 170, p. 107481, 2020. https://doi.org/10.1016/j.apacoust.2020.107481