IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 2, April, 2024, pp. 791-799
Journal homepage: http://iieta.org/journals/ts

Optimization of Acoustic Entropy Source for Random Sequence Generation Using an Improved Grey Wolf Algorithm

Erdinç Avaroğlu*, Semih Kahveci, Ramazan Akkurt

Computer Engineering Department, Faculty of Engineering, Mersin University, Mersin 33343, Turkey

Corresponding Author Email: eavaroglu@mersin.edu.tr

https://doi.org/10.18280/ts.410220

Received: 24 April 2023
Revised: 3 October 2023
Accepted: 8 March 2024
Available online: 30 April 2024

Keywords:
acoustic entropy source, swarm intelligence, random number generator

ABSTRACT

The functionality of cryptographic systems necessitates unpredictable, high-quality random numbers. High-quality random numbers must possess unpredictability, non-reproducibility, and strong statistical properties. To achieve these qualities, True Random Number Generators (TRNG) are employed. The randomness quality of TRNG-derived sources depends on the entropy source used. Physical noise sources, ring oscillators, metastable, acoustic sources, and chaotic attractors are commonly used as entropy sources. In recent years, the use of acoustic signals as entropy sources has attracted attention. However, the noise in the signals affects the bit sequence to be generated. In addition, the threshold and sampling interval applied to the frequency values obtained from the signals also determine the quality of the bit sequence to be produced. Choosing the most appropriate values for the entropy source, randomness and unpredictability of these values are important for the bit sequence to have good statistical properties and to be used in strong cryptographic applications. In this paper, a swarm intelligence-based approach is proposed to determine the optimal threshold and sampling interval by exploiting the power of randomness and unpredictability such as random initialization of swarm intelligence algorithms and obtaining different optimal solutions in each run. In the proposed approach, a bit sequence is generated by applying the values determined by the Improved Grey Wolf Optimization algorithm on the data taken from the MUSDB18 dataset as an entropy source. The generated bit sequences have been shown to be usable as initial value, seed value or additional input for cryptographic key and random number generators by obtaining a p-value greater than 0.01 from the National Institute of Standards and Technology (NIST) test, statistical complexity test (SCM) and autocorrelation test with results close to 0 by obtaining 0.013, 0.074 respectively. Furthermore, to show that the obtained bit sequences can be used as cryptographic keys in the encryption system, we perform encryption on two different images and present histogram and differential attack (UACI, NPCR) test results.

1. INTRODUCTION

Random numbers are used in many areas around the world. Reliable random number generation (RNG) is essential for cryptography, randomization, simulation, and many other applications. Therefore, RNG have been developed [1]. RNG are designed to generate random numbers according to a specified distribution. Pseudo-RNG generate random numbers using a deterministic algorithm and generally provide a sufficient level of randomness. However, for some applications, true random number generators (TRNGs) are more convenient to use [1].

TRNG use devices that generate random numbers from physical processes or noisy signals. When designing RNG, entropy, randomness level, reliability and performance are important factors. These generators have broad use, particularly in cryptography and other security applications [1]. Sound-based RNGs, also referred to as acoustic RNGs, are an example of using physical processes to generate random numbers, offering a supplementary method for generating randomness. Acoustic RNGs rely on the unpredictability of sound to generate random numbers. The randomness of sound arises from its complex signal that constantly changes over time and is influenced by multiple factors, including the environment, sound source, and receiver [2].

Previous research has provided important information on generated bit sequences by using acoustic sources. The most important thing in cryptography is to have an unpredictable key. That is, it is very important that the keys generated are substantially random and difficult to predict. In this work, a statistically independent and homogeneously distributed random bit sequence is generated by sampling the audio signals from noise sources via the microphone, And FIPS 140-1 is applied for the test results by using XOR as the post-processing algorithm [2]. TRNG have a key role in cryptography that require unpredictable and non-deterministic random number sequences. It's so important to utilize the powerful entropy sources while generating random numbers. The authors used a computer microphone to provide a powerful source of entropy and demonstrated that the

proposed TRNG by passing the statistical tests such as NIST SP 800-22, DIEHARD and ENT is capable of generating a great percentage of true random numbers [3].

In a study, white noise of video and audio sources used as entropy source and NIST SP800-22 test suite was used to test the randomness of the proposed system [4]. In literature [5], FM radio signals used as the entropy source and the results demonstrated an increase in the entropy rate. In another work, true random bits were generated using the hardware of a computer sound card, where a random environmental noise signal was input to the audio input using a microphone. Autocorrelation test results are presented by using the proposed MiBiS&XOR as a post-process [6]. Moreover, the random numbers generated by using an audio source can also be used in voice communication. Approximately 80% of true random numbers generated based on audio passed 15 statistical tests, demonstrating that the proposed ARNG can be used to protect personal privacy in a PDA or smartphone [7]. The author proposed a scheme, which is generating 256 bits keys based on human voice or speech data and demonstrated that the proposed scheme passed 13 NIST tests [8]. In literature [9], a random bit sequence generated by using acoustic data as the entropy source. Audio encryption was performed with a sequence generated. NIST and Test U1 test suite were used for results.

Sound-based RNGs have several characteristics that make them attractive for certain applications. First, they are relatively inexpensive and easy to implement, requiring only a microphone and a few electronic components. Second, they are inherently unpredictable because the noise they capture is unpredictable and uncontrollable. Third, they are independent of external factors such as temperature or electromagnetic interference that can affect other types of RNGs. Despite advantages, acoustic-based RNGs have several limitations. First, they are vulnerable to acoustic attacks, where an attacker can manipulate the acoustic environment to distort the random numbers generated by the RNG. Second, they are sensitive to environmental factors such as noise levels and acoustic reflections from the environment, which can affect the quality of the random numbers generated. Third, they have relatively low entropy rates, which means they may not be suitable for applications that require high-quality random numbers [2, 9]. Particularly low entropy rates are seen in single sound source recordings. In contrast, high entropy is seen in recordings containing different sound sources. Despite the use of high entropy sources obtained from different sound sources, it is seen that bit sequences generated according to manually set threshold and sampling interval do not always show valid statistical features [10].

Two parameters play an essential role in producing a bit sequence from the entropy source. The first is the threshold value which is the reference parameter for bit sequence generation and is used to convert the values received from the entropy source into 0 and 1 bits. Secondly, the sampling interval is utilized to generate a bit sequence by drawing the values from the entropy source in a different order according to the sample interval, rather than sequentially, in order to maximize the randomness and unpredictability of the bit sequence. These parameters are normally set manually, such as the mean or standard deviation of the values for the Threshold value and numbers such as 10, 100, and 1000 for the sample interval. Even if the created bit sequence has a high entropy source, it may not always display good statistical properties [10]. In addition, since the predictability of the parameters determined in this way is high, the unpredictability of the bit sequence is low and is not suitable for cryptographic applications. In this paper, we use an optimization technique for selecting the threshold and sampling interval to tackle the above-mentioned problem. Reaching the optimal solution in mathematical models or data in a acceptable time is defined as an optimization problem and there are various optimization methods in the literature for solving this problem. The fact that optimization algorithms randomly generate the values in the phases they pass through while achieving the optimal solution leads to different results at each phase and each time when tackling the same problem with the same method [11]. Therefore, the results obtained with optimization algorithms have high randomness and unpredictability. Taking advantage of this aspect of optimization algorithms, we consider the selection of two parameters that play an important role in bit sequence formation as an optimization problem. We adapt the I-GWO [12] optimization algorithm, which is an improved version of the GWO [13] algorithm, one of the most powerful algorithms in recent times, to this problem. The GWO algorithm was inspired by the social life of grey wolves in nature, such as exploration, hunting, and acting as a group, and the hierarchical structure within the group, and was mathematically modeled by Mirjalili et al. [13]. It was mathematically modeled and created by Mirjalili et al. [13]. I-GWO is a meta-heuristic developed by Nadimi-Shahraki et al. [12], due to some problems of the GWO algorithm such as decreasing population diversity and early convergence while reaching the optimal solution. It is a meta-heuristic method developed by Nadimi-Shahraki et al. [12], I-GWO and has recently been recognized as a powerful algorithm adapted to many problems such as engineering problems, signal processing, and feature extraction. In this study, audio data from the MUSDB18 dataset is used as the acoustic entropy source [14].

The scheme of the proposed method is given in Figure 1. NIST, SCM and autocorrelation test results are shown to prove that the generated pure bit sequences are statistically sufficient. Acoustic-based RNGs have various uses in cryptography, security, and gaming. One of the most common applications is in the generation of cryptographic keys. Cryptographic keys are used to safeguard communications and transactions and must be unpredictable and truly random. Acoustic-based RNGs can provide a source of randomness that is difficult to predict, making them useful for generating cryptographic keys. In order to prove that the generated bit sequences can be utilized as a cryptographic key in encryption systems, image encryption/decryption has been conducted in F-AES [15]. Then, encryption and decryption operations have been done on two separate photos. It has been proved that the proposed system resulted in success by providing the histogram analysis results of the operations executed. It has been found that there is no difference between the original image and the decoded image when the value derived from the mean square error is zero. In addition, the number of pixel change rate (NPCR) and the unified averaged changed intensity (UACI) values are reported.

The primary contributions of the proposed approach are as follows:

I. More accurate and faster sampling intervals and threshold values are obtained in bit sequence generation.
II. The randomness and unpredictability of the bit sequence increase with the optimization process.

III. The availability of the entropy source is determined faster.

The study is structured as follows: the literature review is presented in the introduction. In Chapter 2, Grey Wolf and the improved Grey Wolf algorithm are first explained. Then, the proposed system design and bitstream generation are detailed. Section 3 presents the statistical analysis of the bitstream produced by the proposed system, as well as the practical implementation of the bitstream and the results obtained. Finally, Chapter 4 discusses the findings of the study and provides conclusions.

2. PROPOSED METHOD

In this section, we will initially present the GWO and its enhanced variation - the I-GWO metaheuristic algorithm, which takes inspiration from the life of grey wolves in nature to optimize bit sequence generation. Next, we will explicate the conventional methodology of generating bit strings and its limitations before delving into a thorough explanation of the recently developed approach for optimizing bitstream generation. The schematic of this suggested approach is depicted in Figure 1.

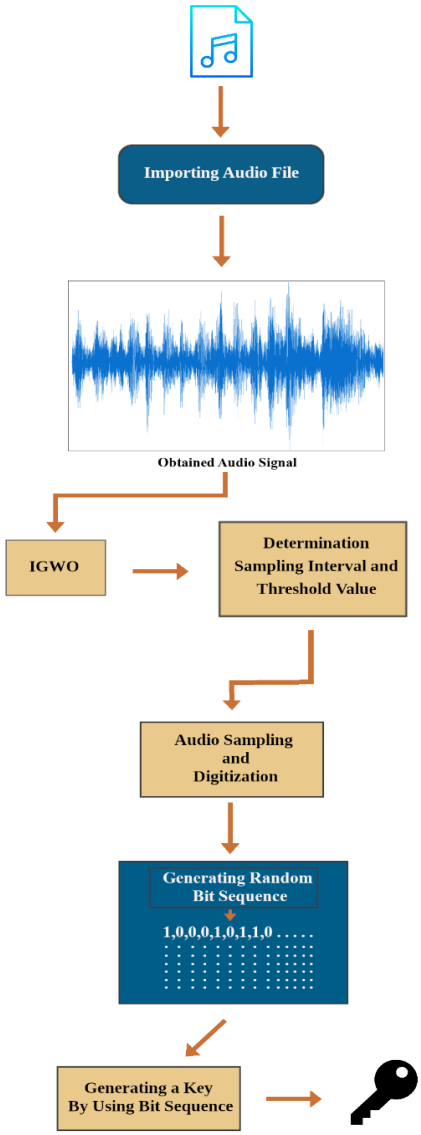

Figure 1. Proposed system scheme

2.1 GWO algorithm

The social life and hierarchical order of grey wolves in the wild are modelled mathematically and a grey wolf metaheuristic algorithm is introduced. In the algorithm, grey wolves are divided into four different groups: alpha (α), which leads the group and is hierarchically dominant, beta (β), which supports alpha and is hierarchically second, omega (Ω), which is hierarchically the weakest, and delta (δ), which is superior to omega wolves. Mathematically, the position of alpha wolves is recognised as the best, beta as the second best and delta as the third best. The remaining solutions represent the omega wolves. In the algorithm, processes such as prey encirclement and hunting are managed under the leadership of alpha, beta and delta wolves [13].

2.1.1 Encircling prey

First, Grey wolves encircle their prey. Eq. (1) and Eq. (2) provide a mathematical model for the encircling behaviour of wolves (2).

$$D = \left|C * X_p(t) - X(t)\right| \tag{1}$$

$$X(t+1) = X_p(t) - (A * D) \tag{2}$$

Eq. (1) computes the distance D between the prey and the wolf, while Eq. (2) updates the position of the wolf. In the equations, the t value stands for the current iteration, the A and C values for the coefficient vectors, X_p for the prey's position vector, and X for the wolf's position. Eq. (3) and Eq. (4) are used to calculate the values of A and C, respectively [13]:

$$A = 2a * r_1 - a \tag{3}$$

$$C = 2r_2 \tag{4}$$

The variable a is linearly reduced from 2 to 0 at each valid iteration. The r_1 and r_2 are vectors that take random values in the range of 0-1.

2.1.2 Hunting

After the wolves find and surround the prey, the hunting process begins with the alpha wolves taking the lead. Therefore, the positions of the beta and delta wolves are considered the top three solutions, with the position of the alpha wolf being the best. Omega Wolves represents other candidate solutions. This behavior of wolves is mathematically given below by Eqs. (5)-(7) [13].

$$D_k = |C_d * X_k - X| \tag{5}$$

$$X_k = X_k - A_d * D_k \tag{6}$$

$$X(t+1) = \frac{(X_1 + X_2 + X_3)}{3} \tag{7}$$

Eqs. (5)-(7) above can be utilized to estimate the distance and final position of the current solution between the alpha, beta, and delta worms, respectively. In these equations, k = (α, β, δ) and d = (1,2,3).

2.1.3 Attacking prey

The Grey wolves finish the hunt by attacking their prey once it stops moving. The value of 'a' is decreased in order to

mathematically represent how a grey wolf approaches its prey. On a mathematical level, a also reduces the fluctuation range of A. In other words, over iterations, a decreases from 2 to 0, with A being a random value between -a and a. Wolves attack their prey when |A|<1.

2.1.4 Prey search

In accordance with the roles of the Alpha, Beta, and Delta wolves, Grey wolves hunt for prey. While hunting, wolves separate, then reunited attack. To make the search agent give up the prey, use A with random values greater than or lower than -1. This modelling, which also allows the GWO algorithm to search globally, defines the exploration phase.

To provide context for the search process, a randomized population of possible solutions is initially created, with Grey wolves selving and representatives. Through iterations, alpha, beta, and delta wolves assess the potential location of the prey. The distance between each solution and the prey is then updated. In order to emphasize exploration and exploitation, the parameter a is decreased from 2 to 0. Candidates move away from prey when |A|>1 and toward prey when |A|<1, respectively [13].

2.2 I-GWO

In GWO, α, β, and δ wolves guide ω wolves towards promising solutions in the search space. This behaviour of GWO may lead to trapping in local optimal solutions. From another point of view, it can lead to a decrease in population diversity and to getting stuck at the local optimum. To overcome such problems, an improved Grey wolf optimization algorithm is proposed in literature [12]. In I-GWO, a new movement approach is proposed. Individual hunting is another intriguing social behavior of grey wolves in addition to group hunting [16], which is the main motivation for I-GWO, and additional steps of I-GWO are described below [12].

Canonical GWO search strategy: In GWO, the top three best wolves α, β and δ are considered as the best fitness values found in the population. After that, the linearly reduced coefficient α and the values A and C are determined by Eqs. (3) and (4). Then, the encirclement of the Prey is identified by taking into the positions of X_α, X_β and X_δ by Eqs. (5) and (6). The initial candidate for the new location is finally of the wolf Xi(t), called Xi-GWO (t+1), is determined by Eq. (7) [12].

Search technique using dimension learning-based hunting (DLH): DLH is a new search strategy that exchanges information between the current wolf and neighboring wolves during the exploration phase. This strategy solves the problems of GWO such as early convergence and reduced population diversity. DLH works as follows: First, a radius Ri(t) is calculated between the current location Xi(t) and the candidate location Xi(t+1) using Euclidean distance and a circle is formed with Xi(t) as the center. Then, within this circle, the neighborhoods Ni(t) of Xi(t) are formed and the multi-neighbor learning step is performed with Eq. (8) [12].

$$X_{i-DLH,d}(t+1) = X_{i,d} + rand(X_{n,d}(t) - X_{r,d}(t)) \quad (8)$$

where, $X_{n,d}(t)$ is the dth size of the randomly selected neighbor wolf from $N_i(t)$, $X_{r,d}$ is the dth size of the randomly selected wolf from the population.

The d'th dimension of $X_{i\text{-}DLH,d}(t+1)$ is calculated using the d'th dimension of a wolf randomly selected from the population and a neighbor selected from $N_i(t)$. Finally, the fitness scores of $X_{i\text{-}DLH,d}(t+1)$, and $X_i(t+1)$ are calculated and the location with the best score is determined as the new current location.

Phase of selection and updating: Here, the fitness values of the two candidates are first compared, and the best candidate is chosen. Then, to update the new position of $X_i(t+1)$, if the chosen candidate's fitness value is less than $X_i(t)$, $X_i(t)$ is updated by the selected candidate. Otherwise, $X_i(t)$ in the population doesn't change [12].

2.3 Traditional bit sequence generation

To use a bit sequence in cryptographic applications, the sequence have certain specific statistical properties such as randomness and unpredictability. There is a specific process for obtaining a bit sequence with these properties. The process begins with the selection of an entropy source. Next, the data obtained from this entropy source is utilized to generate a bit sequence using a sampling interval value and a threshold value. The generated bit sequence is subjected to statistical tests to assess its reliability. If the sequence fails these tests, a new sequence is generated by adjusting the sampling interval value and threshold value in the next step. This manual process continues until a reliable bit sequence is generated or the entropy source is changed.

2.4 Proposed bit sequence generation method

In this paper, we approach the generation of bitstreams as an optimization problem to overcome the drawbacks of manually generated bitstreams. We choose acoustic sound waves as the entropy source, using audio files from the MUSDB18 dataset [14]. To design the optimization algorithm based on the process described in the previous section, it goes through the following steps:

I. The development of a transformation function and constraints to generate a bit sequence from the data obtained from the entropy source.

II. The design of a fitness function that evaluates whether the generated bit sequence satisfies certain statistical properties, such as randomness and unpredictability.

We detail these two important steps in the following sections and present the flowchart of the proposed approach in Figure 2.

2.4.1 Transformation function

By applying Eq. (9), defined below, a new bit sequence is generated based on the selected sampling interval value and a threshold value.

$$\sum_{i=1}^{n} \begin{cases} 0 & k_{(i+d)} < T \\ 1 & k_{(i+d)} \geq T \end{cases} \quad (9)$$

The variable n in the equation represents the total number of data obtained from the entropy source, k represents the data in the i'th index, d represents the sampling interval and T represents the threshold value. For these values from I-GWO, a lower and upper limit needs to be defined.

2.4.2 Fitness function

As mentioned in the previous sections, the generated bit sequence must have randomness and certain statistical properties. Therefore, a new fitness function was defined to

ensure that the bit sequence generated with the selected values meets these criteria. NIST tests were used to construct the fitness function. The definition of the function is given below in Eq. (10).

$$F_{\text{fitness}}(\text{x}) = F_{freq}(\text{x}) \begin{cases} 0 & p < \text{t\_v} \\ p & p > \text{t\_v} \end{cases} + F_{block}(\text{x}) \begin{cases} 0 & p < \text{t\_v} \\ p & p > \text{t\_v} \end{cases} + F_{run}(x) \begin{cases} 0 & p < \text{t\_v} \\ p & \text{p} > \text{t\_v} \end{cases} + F_{Lrun}(x) \begin{cases} 0 & p < \text{t\_v} \\ p & \text{p} > \text{t\_v} \end{cases} + F_{DFT}(x) \begin{cases} 0 & p < \text{t\_v} \\ p & \text{p} > \text{t\_v} \end{cases} \quad (10)$$

According to the equation, X represents the generated bit sequence. $F_{frequnecy}$ represents the frequency test, F_{block} represents the block frequency test, F_{run} represents the flow test, F_{Lrun} represents the flow test of the longest ones in the block and F_{DFT} represents the Discrete Fourier test . For all the defined tests to produce successful results, the p-value obtained from the tests must be greater than 0.01 Therefore t_v represents the threshold value and is equal to 0.01. In the defined application function, if the p value obtained from the frequency test is greater than 0.01, the other tests are applied. The results of the p-values obtained are summed and the conformity value is calculated. According to Figure 2, the flowchart of the proposed method can be observed.

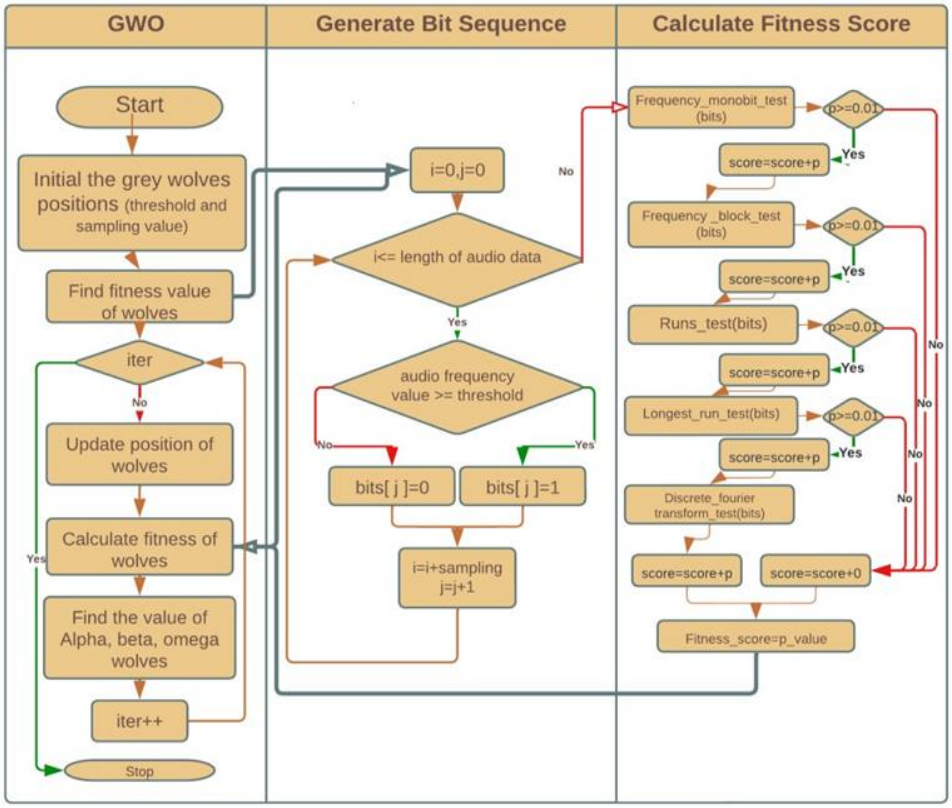

Figure 2. Proposed system flowchart

2.4.3 Experimental setup of the proposed method

In the optimization algorithm used, the lower limit for the sampling interval value is 1, the upper limit is n/5000, the lower limit for the threshold value is min(n) and the upper limit is max(n). The population size was also set to 50. Experimental studies of the proposed system were carried out on the MUSDB18 dataset. MUSDB18 is a dataset consisting of 150 full-length music tracks (~10 hours in duration) from different genres with decomposed drums, bass, vocals, and other tracks [14]. To test the stability of the system, we randomly selected audio files from the data set. Since a single trial for each audio file would be insufficient to test the stability of the system, 20 different trials were performed. Since the convergence rate usually decreases after 100 iterations, each trial was performed in 100 iterations. In addition, to prove that the bit sequences obtained with the proposed system can be used in cryptographic applications, they are set as keys for the F-AES [15] image encryption algorithm. All of these processes and tests were performed on an Intel core i5 (1135G7), 16 GB DDR4 system.

3. APPLICATION OF THE PROPOSED METHOD IN PRACTICE AND RESULTS OF THE ANALYSIS

This section describes the experimental studies performed on the designed system. First, the NIST800.22 test suite [17] was applied to the bit sequences obtained in different ways throughout the study to verify their statistical properties. The NIST test results of the bit sequences obtained with manually determined sampling interval and threshold values and those obtained by applying post-processing algorithms are presented in Table 1. The NIST results of the bit sequences obtained using the proposed method are shown in Table 2. Then, to demonstrate the security of the bit sequence obtained by the proposed method in cryptographic applications, it was used as a key value for the F_AES algorithm. Two different images, randomly selected from the General-100 [18] dataset, were used for image encryption with F-AES. The security of the key is proven by presenting the results of histogram analysis and differential attack analysis at the end of this section.

Table 1. NIST test results of bit sequence generated according to manually determined threshold and sampling interval

Test	Original Data P-Value	Result	Post-Processing Algorithms Applied to Raw Acoustic Source P-Value					
			XOR	Result	Von Neumann	Result	H Function	Result
Freq	0.4465	S	-	F	-	F	0.1472	S
Freq BL	-	F	-	F	-	F	0.0101	S
Run	-	F	-	F	0.9761	S	0.1433	S
Long Run	-	F	-	F	-	F	0.1527	S
BMR	0.0183	F	0.0321	F	0.9642	S	0.8057	S
DFT	-	F	0.0147	F	-	F	0.6348	S
Non-Over T.M.	-	F	-	F	-	F	0.1742	S
Over T.M.	-	F	-	F	-	F	-	F
MU	-	F	-	F	-	F	-	F
LCT	0.2697	F	0.0484	F	0.5748	F	-	F
Serial test		F	-	F	-	F	-	F
App Ent	-	F	-	F	-	F	-	F
Cum Sum	-	F	-	F	-	F	0.2667	S

Notes: In the table, S = Success means that the test was passed and F = Fail means that the test was failed. In addition, values where the calculated p-value is much smaller than the reference value (0.01) and is shown as NA by the NIST test environment are indicated by a hyphen (-). Abbreviations of NIST Tests are given in Table 3.

Table 2. NIST test results of bit sequence generated according to I-GWO determined threshold and sampling interval

Test	Optimization Based Bit Sequence P-Value	Result
Freq	1	S
Freq BL	0.6149	S
Run	0.9404	S
Long Run	0.8069	S
BMR	0.5326	S
DFT	0.6069	S
Non-Over T.M.	0.2710	S
Over T.M.	0.5525	S
MU	0.326	S
LCT	0.7794	S
Serial test	0.8317	S
	0.9755	S
App Ent	0.1790	S
Cum Sum	0.9662	S

Notes: In the table, S = Success means that the test was passed and F = Fail means that the test was failed. Abbreviations of NIST Tests are given in Table 3.

Table 3. Abbreviations of NIST tests

Test	Abbreviation
Frequency (Monobit) test	Freq
Frequency test within a block	Freq BL
Runs test	Run
Test for the longest Run of ones in a block	Long Run
Binary matrix rank test	BMR
Discrete Fourier transform Testi	DFT
Non-overlapping template matching test	Non-Over T.M.
Overlapping template matching test	Over T.M.
Maurer's Universal statistical test	MU
Linear complexity test	LCT
Approximate entropy test	App Ent
Cumulative Sums test	Cum Sum

3.1 Statistical test results

The statistical test results of the bitstream obtained with the proposed approach were validated using the NIST800.22 test suite. The NIST statistical tests primarily evaluate the probability of small non-random segments in the bitstream. The key parameters involved in these tests are α and P. The significance level is called the α value, and a value of 0.01 indicates that the bitstream is random with a 99% confidence level. The measure of randomness is called the P-value. If the P-value is 1, the numbers are completely random. Conversely, if P=0, the numbers are not random at all. To ensure cryptographic reliability, an appropriate significance level (α) must be chosen. A test is deemed successful if the P-value is equal to or greater than α, otherwise the test fails and the numbers are not random. The standard significance level typically ranges between 0.001 and 0.01. For this paper, a significance level of 0.01 is chosen. In order to consider a test successful, the obtained P-value must exceed 0.01. Based on the results presented in Table 1 and following the predetermined threshold, it is clear that the P-value for the pure bitstream obtained from the manually recorded pure audio source was significantly lower than 0.01 in most of the conducted tests. Some algorithms were used to modify the existing order in the raw bit sequence and improve its randomness. However, the post-processing algorithms did not yield favorable results as the p-value of the bit sequence obtained was considerably lower than the reference value in most tests. Upon analysis of the NIST test results presented in Table 2, it is clear that the proposed method has successfully passed all tests by achieving values exceeding the reference p-value. Furthermore, the bit sequence generated by the proposed method exhibits a fully uniform distribution of 0s and 1s, which is evident by the frequency test result of 1. This approach yields bit sequences that are more random and unpredictable than those generated manually or through post-processing algorithms.

Table 4. Autocorrelation test results

Autocorrelation	D Value	X5 Value	Result
Optimization based bit sequence	8	0.074	Passed
	15	0.059	Passed
Original data	8	-49.05	Unpassed
	15	8.79	Unpassed

Table 4 presents the outcomes of the autocorrelation test. The correlation indicates the linear relationship among multiple variables, accepting values between +1 and -1. If it is zero or approaching zero, it implies there is no linear relationship between the variables. As illustrated in Table 4, optimization-based bit sequence produces favorable results for D values 8 and 15 [19].

The statistical complexity measure is displayed in Table 5. The statistical complexity measure for aperiodic sequences must be zero or nearly zero. The successful outcomes for the optimization-based bit sequence [19] are displayed in Table 5.

Table 5. Statistical complexity measure result

	Original Data	Optimization Based Bit Sequence
Statistical complexity measure	0.103	0.013

3.2 Leveraging the generated random bit sequence as a cryptographic key

The key value for the AES-based F-AES encryption algorithm in literature [15] was generated using the bit sequence obtained by our proposed method. We then utilized this generated key to perform encryption and decryption operations on randomly selected images from the General 100 [18] dataset. Histogram analysis was conducted to confirm the successful encryption of the images, and the ensuing results are shown in Figures 3 and 4. To demonstrate the strength of the key value employed, differential attack analysis was performed.

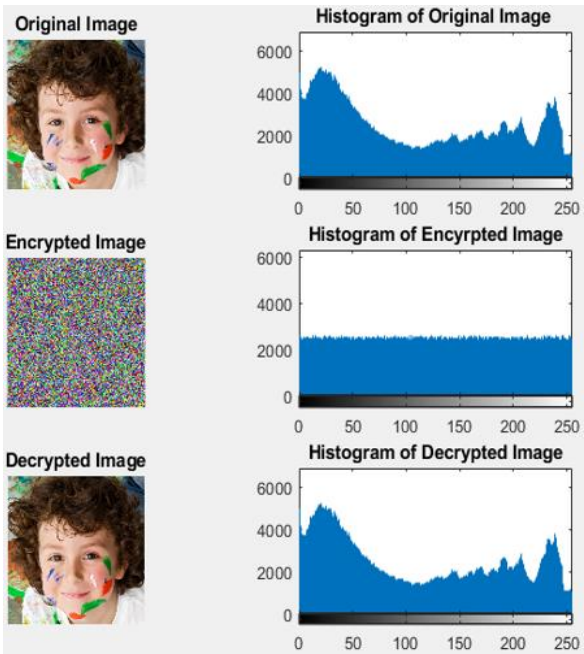

Figure 3. Histogram analysis graph of the boy image

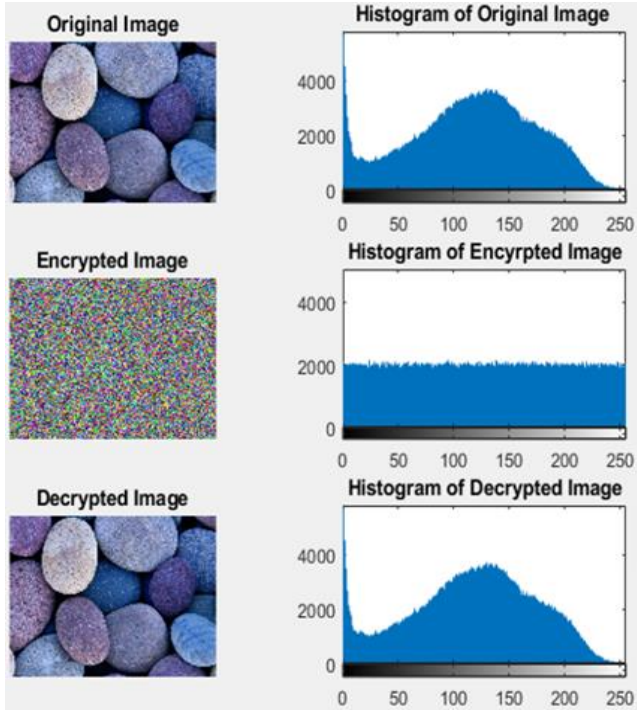

Figure 4. Histogram analysis graph of the stone image

The histogram contains valuable statistical data on the image and also finds relevance in other applications like image compression and segmentation. Images that are encrypted should have a consistently balanced histogram as stated in literatures [8, 9]. Figures 3 and 4, we display sample images in their original, encrypted, and decrypted formats. Histogram analysis graphics for grayscale values are presented on the right hand side. A uniform distribution is evident for all values in the coded histogram analysis figures. However, the histogram of the original and the encrypted images' differ significantly. Uniformity makes it difficult to draw statistical inferences and develop statistical attacks targeting the proposed encryption technique.

Differential attacks attempt to introduce a certain difference in the original text pair using a fixed-key encryption algorithm, and investigate the effect of the original text information by analyzing the effect of the corresponding encrypted output difference. The effect of a small change in the original text or key on the ciphertext is defined as an avalanche effect. The avalanche effect is increased in cryptographic systems to strengthen their defence against differential attacks. The avalanche effect is a desirable property in cryptographic algorithms because it allows small changes in the input to result in large changes in the output [20]. There are two quantities that are commonly used to evaluate the strength of image encryption algorithms against differential attacks. These are the number of pixel change rate (NPCR) and the unified average change intensity (UACI). These measures are calculated using an original image and an image that has been modified by a few pixels (typically one pixel). NPCR focuses on the number of changed pixels in the encrypted versions of these two images, while UACI focuses on the average changed intensity. Optimal values for NPCR and UACI are considered in the literature to be around 100% and 33% respectively.

The results of our experiments are presented in Table 6, displaying the NPCR and UACI scores. The encrypted images show uniformly distributed histograms, rendering it problematic to derive statistical inferences from them via

histogram analysis. However, upon examining the statistics presented in Table 6, it is apparent that the NPCR and UACI values for both images align with those acknowledged in the literature. Based on the experimental results, it is evident that the bit sequence generated is appropriate for utilization in cryptographic applications.

Table 6. UACI and NPCR test results

Sample Pictures	NPCR	(UACI)
Boy	0,99716	0.330
Stone	0,99704	0.333

Comparisons of the RNG are given in Table 7.

Table 7. RNG comparisons

Ref.	Randomness Source	Post Processing	Tests
[21]	LFSR, discrete chaotic map	-	NIST, DIEHARD, TestU01
[22]	Memristor	Trivium	NIST, scale index
[23]	Electromagnetic noise	XOR	NIST, scale index, autocorrelation
[24]	MARC-bb	XOR	TestU01
[25]	Analog circuit	XOR	NIST
Proposed method	environmental sounds	-	NIST, statistical complexity, autocorrelation

4. CONCLUSION

To ensure that the random number generators produce statistically valid numbers that cannot be guessed or regenerated, it is imperative to use resilient entropy sources. When extracting a bit sequence from an entropy source, there are two key factors to consider: sampling interval and threshold value. Accurately determining these parameters is crucial for the efficient use of the entropy source. These conventional selection parameters can hinder the statistical properties of bit sequences derived from robust sources of entropy, consequently impairing the usability of the entropy sources.

This study aims to identify the parameters that have a significant influence on enhancing the usability of the entropy source in a faster and more precise manner. The study further aims to increase the randomness and unpredictability of these parameters beyond a certain order and generate bit sequences with improved statistical properties. Swarm intelligence algorithms achieve varying results when solving the same problem due to their use of random initial values and subsequent random values. There findings suggest that meta-heuristic algorithm solutions possess a strong degree of unpredictability and randomness. In the proposed approach, discovering the best threshold and sampling interval values through leveraging swarm intelligence algorithms' power is viewed as an optimization challenge. The Grey Wolf Optimization (GWO) algorithm is based on the hunting and exploration abilities of grey wolves, which makes it a suitable approach for tackling problems. Entropy sources were chosen from various audio files within the MUSDB18. Bit sequences acquired through the proposed, conventional, and post-processing methods underwent NIST tests, with subsequent calculations of their statistical features including randomness. As a result of the calculations, it is clearly seen that the p values obtained from the NIST tests of the proposed method are greater than 0.01, and the SCM and autocorrelation test results are close to 0 by obtaining 0.013, 0.074, showing better random distribution and better statistical properties compared to other bit sequences. Additionally, to demonstrate the applicability of the produced sequence in cryptography, it served as a cryptographic key in the F-AES encryption algorithm, and two images were subsequently encrypted. According to the results of histogram analysis and differential attack analysis, it is appropriate to use the generated bit sequence as the cryptographic key.

In future studies, it is believed that swarm intelligence algorithms could functions as post-processing algorithms in TRNG systems.

REFERENCES

[1] Koç, Ç.K. (2009). About Cryptographic Engineering. Springer US.

[2] Morrison, R. (2001). Design of a true random number generator using audio input. Journal of Cryptology, 1(1): 1-4.

[3] Teh, J.S., Teng, W., Samsudin, A. (2016). A true random number generator based on hyperchaos and digital sound. In 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, pp. 264-269. https://doi.org/10.1109/ICCOINS.2016.7783225

[4] Chen, I.T. (2013). Random numbers generated from audio and video sources. Mathematical Problems in Engineering, 2013: 285373. https://doi.org/10.1155/2013/285373

[5] Lee, K., Lee, M. (2019). True random number generator (TRNG) utilizing FM radio signals for mobile and embedded devices in multi-access edge computing. Sensors, 19(19): 4130. https://doi.org/10.3390/s19194130

[6] Nikolic, S., Veinovic, M. (2016). Advancement of true random number generators based on sound cards through utilization of a new post-processing method. Wireless Personal Communications, 91: 603-622. https://doi.org/10.1007/s11277-016-3480-9

[7] Chen, I.T., Tsai, J.M., Tzeng, J. (2011). Audio random number generator and its application. In 2011 International Conference on Machine Learning and Cybernetics, Guilin, China, pp. 1678-1683. https://doi.org/10.1109/ICMLC.2011.6017002

[8] Bano, A. (2013). Random key generator using human voice. In IMPACT-2013, Aligarh, India, pp. 41-45. https://doi.org/10.1109/MSPCT.2013.6782084

[9] Etem, T., Kaya, T. (2020). Self-generated encryption model of acoustics. Applied Acoustics, 170: 107481. https://doi.org/10.1016/j.apacoust.2020.107481

[10] Avaroglu, E., Tuncer, T., Özer, A.B., Türk, M. (2014). A new method for hybrid pseudo random number generator. Informacije MIDEM, 44(4): 303-311.

[11] Abualigah, L., Elaziz, M.A., Khasawneh, A.M., et al. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: A comprehensive survey, applications, comparative analysis, and results. Neural Computing and

Applications, 34: 4081-4110. https://doi.org/10.1007/s00521-021-06747-4
[12] Nadimi-Shahraki, M.H., Taghian, S., Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications, 166: 113917. https://doi.org/10.1016/j.eswa.2020.113917
[13] Mirjalili, S., Mirjalili, S.M., Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software. 69: 46-61.
[14] Rafii, Z., Liutkus, A., Stöter, F.R., Mimilakis, S.I., Bittner, R. (2017). MUSDB18-a corpus for music separation. HAL Open Science. https://doi.org/10.5281/zenodo.1117372
[15] Dişkaya, O., Avaroğlu, E., Menken, H., Emsal, A. (2022). A new encryption algorithm based on Fibonacci polynomials and matrices. Traitement du Signal, 39(5): 1453-1462. https://doi.org/10.18280/ts.390501
[16] MacNulty, D.R., Mech, L.D., Smith, D.W. (2007). A proposed ethogram of large-carnivore predatory behavior, exemplified by the wolf. Journal of Mammalogy, 88(3): 595-605. https://doi.org/10.1644/06-MAMM-A-119R1.1
[17] Rukhin, A., Soto, J., Nechvatal, J., et al. (2001). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (Vol. 22). Gaithersburg, MD, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology.
[18] Dong, C., Loy, C.C., Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, pp. 391-407. https://doi.org/10.1007/978-3-319-46475-6_25
[19] Menezes, A J., Van Oorschot, P.C., Vanstone, S.A. (2018). Handbook of Applied Cryptography. CRC Press.
[20] Hussain, I., Shah, T. (2013). Literature survey on nonlinear components and chaotic nonlinear components of block ciphers. Nonlinear Dynamics, 74: 869-904. https://doi.org/10.1007/s11071-013-1011-8
[21] Alhadawi, H.S., Zolkipli, M.F., Ismail, S.M., Lambić, D. (2019). Designing a pseudorandom bit generator based on LFSRs and a discrete chaotic map. Cryptologia, 43(3): 190-211. https://doi.org/10.1080/01611194.2018.1548390
[22] Kaya, T. (2020). Memristor and Trivium-based true random number generator. Physica A: Statistical Mechanics and Its Applications, 542: 124071. https://doi.org/10.1016/j.physa.2019.124071
[23] Etem, T., Kaya, T. (2020). A novel true random bit generator design for image encryption. Physica A: Statistical Mechanics and Its Applications, 540: 122750. https://doi.org/10.1016/j.physa.2019.122750
[24] Li, J., Zheng, J., Whitlock, P. (2018). Efficient deterministic and non-deterministic pseudorandom number generation. Mathematics and Computers in Simulation, 143: 114-124. https://doi.org/10.1016/j.matcom.2016.07.011
[25] Guler, U., Pusane, A.E., Dundar, G. (2017). Design of efficient CMOS ring oscillator-based random number generator. International Journal of Electronics, 104(9): 1465-1482. https://doi.org/10.1080/00207217.2017.1312704