BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ
JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGIES
http://dergipark.org.tr/tr/pub/bibted

Araştırma Makalesi

Real Random Number Generation by Chemical Reactions Based on Quantum Wave Equation

Muharrem Tuncay Gençoğlu*[1], Tuncay Genç[2]

[1] *Fırat Üniversitesi, Teknik Bilimler MYO, Elâzığ, Türkiye*
[2]*Emniyet Müdürlüğü, Elâzığ, Türkiye*

Keywords:
Chemical reaction,
Random number generator,
True random number generator

ABSTRACT

Random Number Generators are software or hardware components that allow the production of unpredictable number sequences without any pattern or relationship between them. Various studies have been conducted with different techniques regarding RNG. In these studies, the difficulties of random number generation and the high cost negatively affect the efficiency of the developed generators. Many different methods have been used in real random number generation, and even quantum random number generators have been developed to make predictability difficult. Quantum Random Number Generators; are a tepe of generator based on the laws of Quantum physics instead of classical physics. In photonic-based RNG, random numbers are generated after various software and hardware operations by utilizing the uncertainty of photons. This study, it is aimed to develop a true random number generator using chemical reactions that have not been studied before. Data was produced by using sensors and other hardware elements together, the values produced were taken as seed values and assigned as input to the algorithm used in generating random numbers, and true random numbers were produced and these numbers were tested in detail with known test methods.

Kuantum Dalga Denklemi Tabanlı Kimyasal Reaksiyonlarla Gerçek Rastgele Sayı Üretme

Anahtar Kelimeler:
Kimyasal reaksiyonlar,
Rasgele sayı üretimi,
Gerçek rasgele sayı üretimi

ÖZ

Rastgele Sayı Üreteçleri, aralarında herhangi bir örüntü veya ilişki olmayacak şekilde tahmin edilemeyecek sayı dizileri üretilmesini sağlayan yazılımsal veya donanımsal bileşenlerdir. RSÜ ile ilgili farklı tekniklerle çeşitli çalışmalar yapılmıştır. Bu çalışmalarda rastgele sayı üretiminin zorlukları ve maliyetin yüksek olması geliştirilen üreteçlerin verimliliğini olumsuz etkilemektedir. Gerçek rastgele sayı üretiminde çok farklı yöntemler kullanılmış hatta tahmin edilebilirliği zorlaştırmak için kuantum rastgele sayı üreteci dahi geliştirilmiştir. Kuantum Rastgele Sayı Üreteçleri; klasik fizik yerine Kuantum fiziği yasalarının temel alındığı bir üreteç çeşididir. Fotonik tabanlı KRSÜ'de fotonların belirsizliğinden faydalanılarak çeşitli yazılımsal ve donanımsal işlemlerden sonra rastgele sayılar üretilir. Üretilen bu sayılar, tahmin edilemeyecek seviyede güçlü rastgele sayılardır. Ancak bu yöntemin hem insan sağlığı hem de maliyet açısından olumsuzlukları mevcuttur. Bu çalışmada, özellikle radyoaktif rastgele sayı üreteçlerine alternatif olacak ve maliyeti düşürmek adına daha önce çalışılmamış olan kimyasal reaksiyonlar kullanılarak gerçek rastgele sayı üreteci geliştirilmesi amaçlanmıştır. Donanımsal kaynaklar ve kimyasal reaksiyonlar birlikte kullanılarak gerçek rastgele sayılar üretilmiştir. Sensörler ve diğer donanım elemanlarının ortak kullanımıyla veri üretilmiş, üretilen değerler tohum değeri olarak alınıp, rastgele sayı üretmede kullanılan algoritmaya girdi olarak atanarak gerçek rastgele sayılar üretilmiş ve bu sayılar bilinen test yöntemleriyle detaylı olarak test edilmiştir.

* **Muharrem Tuncay Gençoğlu**

*(mt.gencoglu@firat.edu.tr) ORCID ID 0000 – 0002 – 8784 – 9634
(tncygnc@gmail.com) ORCID ID 0000 – 0002 – 8325 – 3243

e-ISSN: 2717-8579

Geliş Tarihi: 05/10/2024; Kabul Tarihi: 16/12/2024

1. INTRODUCTION

The random number (RN) is the number we obtain by mathematically and evenly distributing the elements in a series whose members are known so that new choices cannot be predicted from previous choices (Chaitin,2001).

The history of random numbers goes back a long way. Dice, coins, and other devices have been used to generate random numbers in random elections and games of chance. In particular, dice were used to make important decisions such as inheritance sharing and presidential elections. In addition to dice, card games, coins, spinning wheels, etc. objects were also used as early random number generators.

Random numbers began to be used in later years, especially in cryptology. It has been used to generate keys in encryption.

Currently, images, patterns and 3D objects are created using random data through certain programs and computers. Random numbers are used in secure communication applications where only the receiver and transmitter know the content or in data-hiding applications where only the user needs to know the content (Daemen,2013). The fields where random numbers are used include sampling, entertainment, modeling, simulation and testing, decision-making, cryptography, computer games, computer programming and electronic design(Robinson,1998;Schoukens,1988; Schindler,2002).

Random numbers are very important for ensuring the confidentiality and reliability of the encryption process (Avaroğlu,2017; Tuncer and Genç,2019). The use of random numbers in cryptographic applications increases the encryption strength.

The situations that can be used as sources of randomness are listed as follows:

- Time spent during electrostatic release in radioactive decay
- Thermal noise caused by a resistor or diode element
- Parameter instability between independently operating oscillators
- The charging time of the semiconductor capacitor for a certain period
- Air turbulence on a hard disk
- An arbitrary amount of software-based audio from a microphone or image from a camera [8].

Although there are many different RNG structures for generating random numbers, it is generally possible to divide them into three classes. These are called pseudo random number generators (PRNG), true random number generators (RRNG) and hybrid random number generators (HRNG). The classification of random number generators is shown in Figure 1 (Koç,2009):

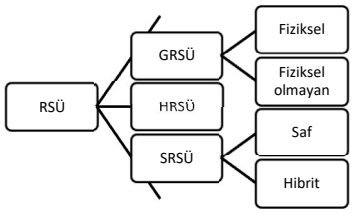

Figure 1. Classification of Random Number Generators

In general, random properties must be met in random number generation. RRNG is based on a physical state known to be random. Sources that generally produce noise or noise sources found in nature can be used as examples. In other words, even if the RRNG is run twice under exactly the same conditions, it produces two unrelated sequences of random numbers. PRNGs are number generators that are based on predictable equations, contain random data, and calculate random data generation in their processor in a limited situation.

The differences between RRNG and PRNG are shown in Table 1 (Von Neumann,1951):

Table 1. Differences between RRNG and PRNG

RNG	Sufficiency	Determinism	Periodicity
PRNG	Perfect	Deterministic	Periodic
GRNG	Weak	Nondeterministic	Nonperiodic

Figure 2 shows the general structure of real random number generation.

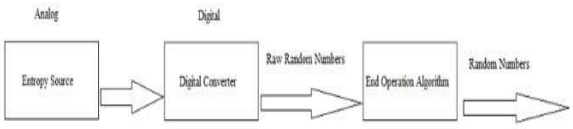

Figure 2. General structure of random number generation

Random numbers form the most important part of many systems and applications where security is at the forefront. In critical areas such as cryptographic applications, games of chance, and password generators, the security of the application is based on random numbers (Wold,2011; Yakut et al.,2019). Yakut proposed a random number generator that can be easily generated using any digital data source (Yakut,2021).

In such security applications, RRNG is generally preferred. The security of encrypted systems is based on the fact that confidential data or keys are known by authorized individuals and cannot be guessed by other individuals.

Random values are needed here to make it difficult for others to guess the secret information.

Malicious users can create security vulnerabilities by exploiting the weaknesses of random number generation methods. Therefore, the

use of random numbers in important areas such as security makes it important that these numbers are close to real random numbers and have the properties of real random numbers (Chaitin,2001; Sanguinetti,2014). Apart from security, random numbers also play an important role in simulating real events. The use of different sources as entropy sources in RRNG is available in the literature (Wold,2011). However, the generation of random numbers in uncontrolled environments outside the system poses a problem for the security of the system (Avaroğlu,2014).

RRNG generally consists of three blocks. These:

- Entropy (noise) source,
- Sampler (digitizer),
- End operation algorithms.

The concept of entropy constitutes the second law of thermodynamics theory. Entropy is defined as a measure of the qualitative disorder and randomness of a system (Kapur and Kesavan,2014). The sampler ensures the necessary sampling of the noise signal, and this structure can be expressed as the production mechanism for physical noise sources (Wold,2011). Thus, it is possible to obtain a digitized signal from an analog signal.

There are different approaches to sampling, and the sampler, along with the entropy source, has an important role in determining the quality of the numbers produced. Postprocessing is often used to increase the randomness present in the signal. This operation is applied to long-bit sequences and propagates in an autocorrelated manner. Here, the coefficient of two adjacent bits in the connected bit stream is greater than the coefficient of connection between the distant bits. Therefore, the relationships between bits that are close to each other are stronger than the relationships between bits that are far away from each other. In post processing algorithms, better results are obtained from statistical tests by rearranging the autocorrelation instead of a simple compression process. The postprocessed signal has a more uniform distribution and random appearance compared to its pure form.

Random numbers generated by post processing are more resistant to side-channel analysis attacks and less affected by environmental factors. Therefore, post processing algorithms make the generator more secure.

There are different post processing algorithms, such as XOR verification, Von Neumann verification, extractor function, cryptographic hash algorithms and resilient function (Dichtl,,2007). RRNG cryptological applications and chaos-based PRNG-RRNG applications have been developed on FPGAs (Yıldırım,2012; Özkaynak,2013; Özkaynak,2014; Özkaynak,2015). Cirauqui et al. performed correlation analysis and compared the effects that potentially hidden correlations in random or pseudo-random flows can have in some physical MC simulations (Cirauqui et al.,2024) Daojing et al. have created a Software Random Number Generator Applicable to the Internet of Things (Daojing et al. ,2024). Luis et al. used quantum mechanics for pseudo-random number generation based on simulated quantum processes as a source of entropy Luis et al.,2025).

1.1. RRNG Designs in Literature

In their study, Voris et al. suggested that the accelerometer and temperature on the Wireless Identity and Sensing Platform (WISP) are better sources of entropy than other sensors (Voris et al.,2011). However, using only these two sensors in motionless environments where the temperature does not change is insufficient for random number generation. In Mitra's study, a true random number generator suitable for generating seed values was proposed. RRNG was implemented with a dual-fed operational amplifier (Bedekar and Shee,2015). Hennebert et al. found that the best candidates as possible sources of entropy are accelerometers, magnetometers, vibration sensors, and internal clock sensors (Hennebert et al.,2013).

In their study, Bedekar and Shee presented a practical method for assessing the GRSS by using microelectromechanical system sensors (MEMS) (accelerometer, gyroscope and compass) (Bedekar and Shee,2015). In another study by Vivier et al., a pseudorandom number generator was designed based on an n-cube without a Hamilton cycle. Since this method, which has passed classical tests, is carried out only with integers, the security was evaluated as weak as a result of the NIST test (Vivier et al.,2017). In another study, Akgül et al. designed only an interface and did not introduce a generator (Akgül et al. 2019). In 2020, a study on pseudorandom number generation was conducted and tested by Rezk et al. (Rezk et al.,2020).

In one recent study, Avaroğlu and Tuncer designed a new true random number generator based on an S-box (Avaroğlu and Tuncer,2020). The disadvantage of this work is that there is a correlation with the generalized bit sequence coming from the entropy source.

Cryptography is a fundamental component of network security and therefore cybersecurity [28]. The most important problem in public key cryptography is finding a unique and nonrepeatable key. There are two methods for generating the key. The first is a rigorous and powerful mathematical algorithmic approach. The second is to imitate nature.

In 2020 and beyond, studies focused on quantum random number generators (Smith et al.,2020; Lin et al.,2020; Kavulich et al. 2021). In a study on random number generation with quantum technology, unpredictable random numbers were produced by using photons obtained from photo frames taken from phone cameras (Dutang and Wuertz,2009). Since generating random numbers

using quantum technology is costly and these techniques are not widely used today, there are thoughts that they will be widely used in the future with the development of quantum technology (Gençoğlu,2021).

It is important to ensure certain features when generating numbers in a random number generator. It should be as random as possible, randomness should be ensured over long periods, and the generated random numbers should be reproducible, calculable and reusable when necessary. Can a true random number generator be created that provides all these features and is efficient, low-cost, and easy to use? question became the source of motivation for this study.

1.2. Quantum Wave Equation

The wave equation is a partial differential equation that has a very important place in physics. Wave equations, which have a very wide usage area, have started to be used in cryptography in recent years (Gençoğlu and Agarwal,2021). When the wavefunction is used in the Schrödinger equation, it is also called a quantum wavefunction. This equation provides information about the future behavior of a dynamic system and predicts the distribution of outcomes by analytically and precisely predicting the likelihood of events. The combination of a physical system consisting of a particle and a wavefunction is one of the assumptions of quantum mechanics. The wavefunction can be complex (Gençoğlu,2013).

2. MATERIALS AND METHODS

In this study, a new hybrid approach is proposed for the use of quantum wave equationbased algorithms in cryptography by using seed values obtained through chemical reactions, combining mathematical calculations and natural phenomena. This work aims to generate a quantum wave equation-based, low-cost random number using chemical reactions for nonreproducible, unpredictable and efficient real random number generation (RRNG) that exhibits good statistical properties. For this purpose, the following hypotheses have been proposed:

1. The use of chemical reactions to generate real random numbers has a positive impact on the efficiency and cost of the generated random numbers.
2. The use of chemical reactions as a noise source in generating real random numbers is important because of their good statistical properties.
3. Chemical reactions that can be used as alternatives to radioactive random number generators positively affect the development of lowcost random number generators.

The greatest disadvantages of existing RRNGs and PRNGs are that they are costly and predictable. The disadvantages of QRNGs, which are the most reliable in terms of unpredictability, are cost, negativities caused by radioactivity and difficulty of use. In the model we propose, since the numbers obtained from different seed values in the random number generation process will be combined with a function f;

- Even if a part of the generated number sequence is obtained, it is impossible to obtain the other part.
- The number sequence does not contain periodic results.
- The produced sequences will not have any hidden correlations within themselves.

The first step in the theoretical approach and method followed in this study is seed data generation. For this, a seed was planted from a corn cob and data was obtained from the chemical reactions occurring in the plant and the environment during the germination and growing process. Then, these data were used as input in a mathematical algorithm based on quantum wave equations to produce true random numbers.

The first step in the theoretical approach and method followed while carrying out this study is seed data generation. For this purpose, a seed was planted from a corn cob, and data were obtained from the chemical reactions occurring in the plant and the environment during the germination and growing process. Then, these data were used as input in a quantum wave equation-based mathematical algorithm to generate true random numbers.

Afterwards, the obtained random numbers were tested.

Stage 1

Chemical reactions have been used as noise sources. The light source, soil, water, precision scale were used to calculate the weight gain, and humidity and thermometer were used to measure the ambient humidity and temperature. The weight values of the corn plants on the precision scale were recorded at regular intervals, as were the humidity and temperature changes. The data obtained in this direction are shown in Table 2.

Table 2. Weight, Humidity and Temperature Values According to Measurement Order

Measurement Order	Weight	Humidity	Temperature(°C)
1	277.33	51.0	26.8
2	271.81	45.0	28.1
3	272.24	52.0	23.6
4	271.41	55.0	24.1
5	270.68	54.0	24.3
6	268.67	44.0	24.1
7	268.35	48.0	24.3
8	263.73	54.0	23.0
9	262.33	47.0	23.0
10	261.8	42.0	22.6
11	261.28	53.0	23.0
12	260.0	52.0	22.6
13	258.50	53.0	22.9
14	258.31	48.0	22.3
15	258.07	55.0	23.1
16	257.75	44.0	27.2
17	257.61	45.0	27.1
18	257.37	45.0	27.3
19	257.27	46.0	27.4
20	257.1	48.0	27.5
21	255.12	48.0	27.3
22	299.15	47.0	28.8
23	298.91	46.0	28.8
24	298.48	48.0	28.9
25	296.28	45.0	28.9
26	295.88	45.0	29.0
27	295.41	44.0	29.7
28	288.49	45.0	29.5
29	287.42	46.0	29.6
30	286.92	43.0	29.3
31	286.74	43.0	29.4
32	286.45	42.0	29.6

33	286.02	40.0	29.8
34	283.39	41.0	28.3
35	282.82	41.0	29.3
36	281.79	45.0	28.3
37	281.63	47.0	28.4
38	281.42	52.0	23.7
39	281.34	51.0	23.8
40	281.14	50.0	23.5
41	280.98	56.0	24.1
42	280.85	53.0	24.2
43	279.71	51.0	24.4
44	278.92	51.0	22.9
45	278.77	51.0	23.7
46	277.39	51.0	24.2
47	276.92	57.0	24.5
48	276.82	56.0	24.5
49	276.67	55.0	24.7
50	276.39	52.0	24.0
51	276.32	56.0	24.3
52	276.23	63.0	24.4
53	276.06	59.0	24.2
54	276.0	62.0	24.4
55	275.84	59.0	24.3
56	275.77	60.0	24.3
57	275.38	57.0	24.0
58	274.7	57.0	23.6
59	287.25	53.0	23.1
60	287.18	52.0	23.1

Stage 2

A mathematical formulation was developed by taking into account existing applications in the literature. The quantum wave equation, is a quadric differential equation known as the Schrödinger equation;

$$-\frac{\hbar^2}{2m}\frac{\partial^2\Psi(x)}{\partial x^2} + V(x)\Psi(x) = E\Psi(x) \qquad (1)$$

The general solution of this equation is a linear combination as follows:

$$\Psi(x) = ACos(kx) + BSin(kx) \ (2)$$

Here, a new wavefunction is obtained from the solution of equation (2), where t: time, k: wave vector ($2\pi/\lambda$), λ: wavelength, x: position and w: frequency [35];

$$\psi(x,t) = \frac{1}{\sqrt{2}}[\cos(wt - kx) + sin(wt - kx)] \quad (3)$$

Each of the values obtained from moisture, heat and mass changes was taken as the seed value in equation (3), and different random numbers were generated. The algorithms used for RRNG, which are based on the quantum wave equation, were written using the Python program. Python codes are shown in Figure 3.

The generation of random numbers using the available data is shown in Figure 4.

The general view of the proposed architecture for the random number generator algorithm obtained using the Python program is given in Figure 5 The architecture proposed in Figure 5 is a mathematical function-based random number generator architecture recommended by the data analyzed using the determined parameters. Blocks other than dashed arrows represent analysis stages of statistical randomness processes, which is the basic condition that must be met for generated random numbers.

Stage 3

A run test was used to check the randomness of the results. The run randomness test is a statistical test used to check randomness in data. This nonparametric test uses datasets to determine whether the data presented are random or tend to follow a pattern (Bujang and Sapri, 2018).

The first step in run testing is to count the number of runs in the data array. A run is defined as a series of consecutive positive or negative values.

$$Z = \frac{R - \bar{R}}{S_R}$$

Here,

R= Observed number of runs

R=Expected number of runs

$$R = \frac{n_1 n_2}{n_1 + n_2}$$

S_R= Standard deviation of the number of runs

$$S_R^2 = \frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{(n_1+n_2+)^2\,(n_1+n_2-1)}$$

n_1, n_2= Number of positive and negative values in the series

Comparing the calculated Z-statistic with the Z critical value for a certain confidence level (Z critical =1.96 for 95% confidence level), if |Z|>Z critical, the numbers are not random Bujang and Sapri, 2018). The test Python codes are given in Figure 6.

3. FINDINGS AND DISCUSSION

In this study, in the real random number generator design prepared using corn plants, the plant's weight change and the humidity and temperature values of the environment when the plant's weight was measured were taken as seed values and these values were used to generate random numbers in precise functions using the Python language. The Run test was used to measure the reliability of the numbers.

Run test, which is one of the methods used to test the homogeneity of the data, is a test in which the data to be examined is assumed to come from the same mass and are independent of each other or the opposite assumption can be checked. According to the result of this test, if the data are from the same mass and independent of each other, these series are called simple random numbers. Therefore, it was evaluated that the most reliable analysis could be made with the run test according to the data we have.

In the run test, to say that the numbers are random, the Z value must be less than 1.96. In our study, the Z value was found to be 0.5242377083205431 as a result of the Run test.

```
import math
import xlrd
import xlwt

#excelin olduğu adres
loc = ("C:\\Users\\Tuncay\\Desktop\\proje\\veri.xlsx")
wb = xlrd.open_workbook(loc)
sheet = wb.sheet_by_index(0)
EXCEL_FILES_FOLDER = 'C:\\Users\\Tuncay\\Desktop\\proje\\'
workbook = xlwt.Workbook()
worksheet = workbook.add_sheet('data')
#excel_file_path = EXCEL_FILES_FOLDER+'result.xlsx'
#workbook.save(excel_file_path)
#k=277.33
#n=51
#s=26.8

#k=1. satır 1. sütun
#n=1. satır 2. sütun
#n=1. satır 3. sütun
for i in range(132):
    k=float(sheet.cell_value(i, 0))
    n=float(sheet.cell_value(i, 1))
    s=float(sheet.cell_value(i, 2))

    x1=1/math.sqrt(2)*(math.cos(0.5777*1.6-42.6630*k)+math.sin(0.5777*1.6-42.6630*k))
    x2=1/math.sqrt(2)*(math.cos(4.2817*1.6-5.7567*n)+math.sin(4.2817*1.6-5.7567*n))
    x3=1/math.sqrt(2)*(math.cos(0.2493*1.6-98.7421*s)+math.sin(0.2493*1.6-98.7421*s))


    h1=5*x1*(1-x1)+(3-0.9999)*math.sin(math.pi*x1)/3
    h2=5*x2*(1-x2)+(3-0.9999)*math.sin(math.pi*x2)/3
    h3=5*x3*(1-x3)+(3-0.9999)*math.sin(math.pi*x3)/3

    i1, d1 = divmod(h1, 1)
    o1=round(d1,4)

    i2, d2 = divmod(h2, 1)
    o2=round(d2,4)

    i3, d3 = divmod(h3, 1)
    o3=round(d3,4)

    top=o1+o2+o3
    sonuc=math.pow(math.e,math.sin(math.pi*top))
    worksheet.write(i, 0,sonuc)
    workbook.save('result.xls')
```

Figure 3. Random number generator

The formation of random numbers with the available data is shown in Figure 4.

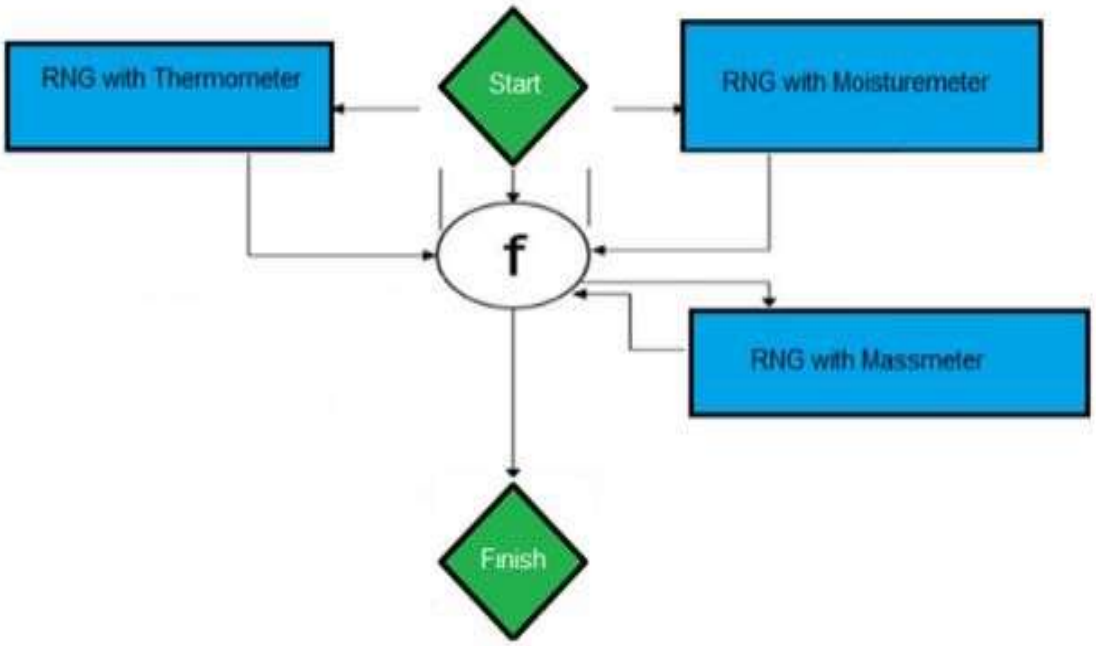

Figure 4. Combining Random Number Generators with the f Function

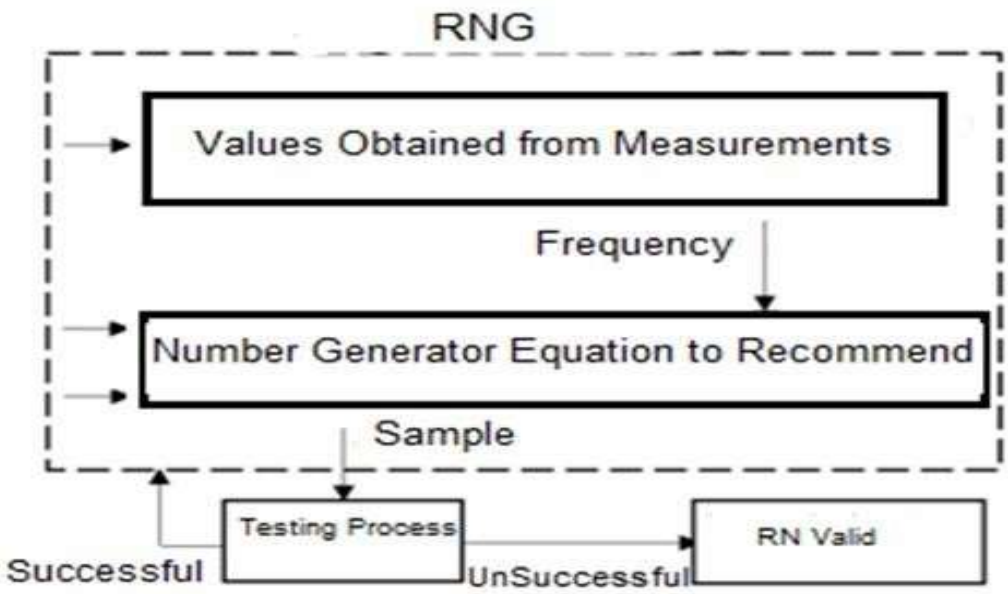

Figure 5. Overview of the Proposed Architecture

```
import random
import math
import statistics
import xlrd
import xlwt

def runsTest(l, l_median):

        runs, n1, n2 = 0, 0, 0

        # Checking for start of new run
        for i in range(len(l)):

                # no. of runs
                if (l[i] >= l_median and l[i-1] < l_median) or \
                                (l[i] < l_median and l[i-1] >= l_median):
                        runs += 1
                # no. of positive values
                if(l[i]) >= l_median:
                        n1 += 1

                # no. of negative values
                else:
                        n2 += 1

        runs_exp = ((2*n1*n2)/(n1+n2))+1
        stan_dev = math.sqrt((2*n1*n2*(2*n1*n2-n1-n2))/ \
                                        (((n1+n2)**2)*(n1+n2-1)))

        z = (runs-runs_exp)/stan_dev

        return z


loc = ("C:\\Users\\Tuncay\\Desktop\\test\\result.xls")
wb = xlrd.open_workbook(loc)
sheet = wb.sheet_by_index(0)
l = []
for i in range(132):
    k=float(sheet.cell_value(i, 0))
    print(k)
    l.append(k)

l_median= statistics.median(l)
Z = abs(runsTest(l, l_median))
print('Z-statistic= ', Z)
```

Figure 6. Run -Test Python Codes

4. CONCLUSIONS

In the proposed model, since the numbers obtained from different starting values are combined with an f function in the random number generation process; Even if a part of the number sequence is generated, it is impossible to obtain the other part. Since the number sequence does not contain periodic results, the generated sequences do not have hidden correlations within themselves. Therefore, it was concluded that the numbers found are not related to each other and have sufficient randomness.

The value obtained as a result of the run test is 0.524 and shows the reliability of the data. Some of the issues considered in the selection of a random number generator are cost, speed, installation, and performance values. It has been observed that the chemical reactions and real number generation presented in this study provide superiority over its competitors in terms of both cost and ease of use. The proposed method offers a different perspective that can be a source for future studies in this field. It is thought to be a guide for new research to be conducted in the future.

Various generators can be designed with data to be obtained from existing plants using appropriate

mechanisms. The technique used can be developed and placed in a cabin system, and a hybrid random number generator can be designed by obtaining more data in a shorter time.

ACKNOWLEDGMENTS

Muharrem Tuncay Gencoglu was supported by TUBİTAK (121E323).

Author Contributions

Writing – Original draft, conceptualization, and methodology were performed by the MTG. The software, experimental methods, results and outcomes were evaluated via TG.

Funding

This study was produced from the master's thesis titled "Quantum Wave Equation Based Real Random Number Generator with the Effects of Chemical Reactions", which is the output of the project supported by TÜBİTAK [121E323].

REFERENCES

Chaitin, GJ. (2001). Exploring Randomness, London, Springer.

Daemen, J., Rijmen V. (2013). The Design of Rijndael: AESThe Advanced Encryption Standard, New York, Springer Science & Business Media.

Robinson SO., Dessart, DJ. (1998). Teaching and Learning of Algorithms in School Mathematics, USA, National Council of Teachers of Mathematics.

Schoukens, J., Pintelon, R., van der Ouderaa, E., Renneboog. (1998) J. Survey of excitation signals for FFT based signal analyzers, IEEE Transactions on Instrumentation and Measurements, 37(3), 342-352.

Schindler, W., Killmann, W. (2002). Evaluation criteria for true (physical) random number generators used in cryptographic applications, Cryptographic Hardware and Embedded Systems.

Avaroğlu, E. (2017). LFSR soru girdisi ile puf tasarımının gerçeklenmesi, Fırat Üniversitesi Mühendislik Bilimleri Dergisi. 29(2), 15–21.

Tuncer, SA., Genç, Y. (2019). İnsan hareketleri tabanlı gerçek rastgele sayı üretimi. 8(1), 261–269.

Yalçın M., Suykens J., Vandewalle J. (2004). True Random Bit Generation from a Double Scroll Attractor, IEEE Trans. Circuits Syst.. 51(7), 1395-1404.

Koç, Ç. K. (2009). Cryptographic Engineering, SpringerVerlag.

Von Neumann, J. (1951). Various Techniques Used in Connection with Random Digits, National Bureau of Standards Applied Mathematics Series. 12, 36-38.

Wold, K. (2011). Security Properties of a Class of True Random Number Generators in Programmable Logic, Doctoral Degree, Gjøvik University College, Doctor of Philosophy in Information Security.

Sanguinetti, B., Martin, A., Zbinden, H., Gisin, N. (2014). Quantum random number generation on a mobile phone, Physical Review. 4(3), 031056.

Avaroğlu, E. (2014). Donanım Tabanlı Rastgele Sayı Üretecinin Gerçekleştirilmesi, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü.

Kapur, JN., Kesavan, HK. (1992). Entropy Optimization Principles and Their Applications, Netherlands, Springer.

Dichtl, M. (2007). Bad and good ways of post processing biased physical random numbers, International Workshop on Fast Software Encryption.

Yıldırım, S. (2012). A True Random Number Generator in FPGA for Cryptographıc Applications, Master's degree, Middle East Technical University, Graduate School of Natural and Applied Sciences.

Özkaynak, F. (2013). Security problems for a pseudorandom sequence generator based on the Chen chaotic system, Computer Physics Communications.184(9), 2178-2181.

Özkaynak, F. (2020). Cryptographically secure random number generator with chaotic additional input, Nonlinear Dynamics. 78, 2015-2020.

Özkaynak, F. (2015). Kriptolojik Rasgele Sayı Üreteçleri, Türkiye Bilişim Vakfi Bilgisayar Bilimleri ve Mühendisliği Dergisi. 8(2), 37-45.

Voris, J., Saxena, N., Halevi, T. (2011). Accelerometers and randomness: perfect together, Proceedings of the fourth ACM conference on Wireless network security, Hamburg, Germany.

Mitra, M. (2012). A Low-Cost Lightweight Random Number Generator Implementation, International Journal of Engineering Research & Technology. 1(10), 1-9.

Hennebert, C., Hossayni, H., Lauradoux, C. (2013). Entropy harvesting from physical sensors, Proceedings of the sixth ACM conference on

Security and privacy in wireless and mobile networks, Budapest, Hungary.

Bedekar, N., Shee, C. (2015). A Novel Approach to True Random Number Generation in Wearable Computing Environments Using MEMS Sensors. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics. 8957, 530-546.

Contassot-Vivier, S., Couchot, JF., Guyeux, C., Heam, P.C. (2017). Random Walk in a N-Cube Without Hamiltonian Cycle to Chaotic Pseudorandom Number Generation: Theoretical and Practical Considerations, International Journal of Bifurcation and Chaos. 27(1), 1750014.

Akgül, A., Arslan, C., Arıcıoğlu, B. (2019). Design of an Interface for Random Number Generators based on Integer and Fractional Order Chaotic Systems, Chaos Theory and Applications. 1(1), 1-18.

Rezk, A., Madian, A., Radwan, A., Soliman, A.M. (2019). Multiplierless Chaotic Pseudo Random Number Generators, AEU- International Journal of Electronics and Communications. 113, 152947.

Avaroğlu, E., Tuncer T. (2020). A novel S-box-based postprocessing method for true random number generation, Turk J Elec Eng & Comp Sci. 28, 288-301.

Khan, F. U., Bhatia, S. (2012). A Novel Approach to Genetıc Algorıthm Based Cryptography, International Journal of Research in Computer Science. 2(3), 7-10.

Hurley-Smith, D., Hernandez-Castro, J. (2020). Quantum Leap and Crash: Searching and Finding Bias in Quantum Random Number Generators, ACM Transactions on Privacy and Security. 23(3), 1-25.

Lin, X., Wang, S., Yin, Z.Q. (2020). Security analysis and improvement of source independent quantum random number generators with imperfect devices, Npj Quantum Information. 6(1), 100.

Kavulich, J., Van Deren, B., Schlosshauer, M. (2021). Searching for evidence of algorithmic randomness and incomputability in the output of quantum random number generators, Physics Letters; 2021. A (388), 127032.

Dutang, C., Wuertz. D. (2009). A note on random number generation, Overview of Random Generation Algoritms.

Gençoğlu, MT. (2021). Quantum cryptography, quantum communicatıon and quantum computing problems and solutions, Turkish Journal of Science and Technology. 16 (1), 97-101.

Gençoğlu, MT., Agarwal, P. (2021). Use of Quantum Differential Equations in Sonic Processes, Applied Mathematics and Nonlinear Science. 6(1), 21-8.

Gençoğlu, MT. (2013). Complex solutions for Burgers-Like equation, F.U. Türkish Journal of Science and Technology. 8(2), 121-123.

Bujang MA., Sapri, F. (2018). An Application of the Runs Test to Test for Randomness of Observations Obtained from a Clinical Survey in an Ordered Population, Malaysian Journal of Medical Sciences. 25, 146-151.

Yakut, S., Tuncer, T., Ozer, A. B. (2019). Secure and Efficient Hybrid Random Number Generator Based on Sponge Constructions for Cryptographic Applications. *Elektronika Ir Elektrotechnika*, *25*(4), 40-46. https://doi.org/10.5755/j01.eie.25.4.23969

Yakut, S., Tuncer, T., Ozer, A. B. (2020). A New Secure and Efficient Approach for TRNG and Its Post-Processing Algorithms, Journal of Circuits, Systems and Computers. 29:15.

Yakut, S. (2021). Random Number Generator Based on Discrete Cosine Transform Based Lossy Picture Compression. NATURENGS, 2(2), 76-85. https://doi.org/10.46572/naturengs.1009013

Yakut, S. (2022). Kayıplı Resim Sıkıştırma Algoritmalarını Temel Alan Rastgele Sayı Üreteci. Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi, 9(18), 571-580. https://doi.org/10.54365/adyumbd.1145590

He, D., Huang, W., Chen, L., Chan, S. (2024). A Secure and Efficient Software Random Number Generator Applicable to the Internet of Things, IEEE *Internet of Things Journal*, 1-12. doi: 10.1109/JIOT.2024.3468451.

Santa Cruz, L.J.M., Faina, L.F., Souza Pereira, J.H. (2025). Exploring quantum systems for pseudo-random number generation. *Quantum Stud.: Math. Found.* **12**, 3. https://doi.org/10.1007/s40509-024-00348-1

Cirauqui, D., Ángel, M., Guillem, G.M., Corominas, G., Graß, T., Grzybowski, P.R., Muñoz-Gil, G., Saavedra,J.R.M., Lewenstein, M. (2024). Comparing pseudo- and quantum-random number generators with Monte Carlo simulations. APL Quantum, 1 (3): 036125. https://doi.org/10.1063/5.0199568