Dynamic Security Provider with Random Numbers: Exploring with Lava Lamps

G. Nalinipriya,
Professor,
Department of Information Technology,
Saveetha Engineering College,
Chennai, Tamil Nadu, India
Email: nalini.anbu@gmail.com

Don Bosco Blaise A,
UG Scholar,
Department of Computer Science and Engineering,
Saveetha Engineering College,
Chennai, Tamil Nadu, India
Email: johanliaer8@gmail.com

Manikandan P,
UG Scholar,
Department of Computer Science and Engineering,
Saveetha Engineering College,
Chennai, Tamil Nadu, India
Email: manipachiyappan111@gmail.com

Sudharsanan V,
UG Scholar,
Department of Computer Science and Engineering,
Saveetha Engineering College,
Chennai, Tamil Nadu, India
Email: sudharsananuma23@gmail.com

Abstract— The research explores using a lava lamp as a True Random Number Generator (TRNG). Unlike conventional TRNGs that use events like radioactive decay, the lava lamp generates randomness from its wax blobs' chaotic motion. While unconventional, this method has potential in encryption and simulations requiring high randomness. A video records the wax movement, with algorithms extracting random numbers. Preliminary findings suggest the lava lamp offers genuine randomness due to the unpredictable wax behavior. Future work aims to refine the algorithms for consistency, explore applications across industries, and enhance the lava lamp TRNG as a reliable and cost-effective alternative to traditional generators. In this paper, we present an in-depth exploration of using a lava lamp as a True Random Number Generator (TRNG), showcasing its potential in encryption and simulations by leveraging the unpredictable motion of wax blobs for genuine randomness.

Keywords- Lava lamps, True Random Number Generators (TRNGs), Cryptography, Encryption, Randomness, Data Security.

I. INTRODUCTION

Creating genuinely random numbers is essential for security, dependability, and integrity in the modern digital era for a wide range of applications. For a very long time, the generation of unpredictability in traditional TRNGs (true random number generators) has been dependent on deterministic physical phenomena like nuclear fallout or air noise. But to satisfy the increasing need for more randomness, creative solutions are being looked for. Participate in the exciting concept of using a lava lamp like TRNG. Although it may seem strange or even unusual, the irregular and chaotic movement of its wax spots is a unique source of randomness.

This paper initiates an examination of this novel approach and investigates its potential as a trustworthy source of randomization. The project intends to validate the efficacy of the TRNG lava lamp by capturing and analyzing the lava lamp's wax movement using video records and sophisticated algorithms.

The purpose of this study is not only to understand the reliability of the lava lamp random number generator, but it also aims to improve the algorithms used to interpret this movement. In addition, it explores possible applications in various fields and aims to position the TRNG lava lamp as a viable and cost-effective alternative to traditional methods. In today's digital age, the generation of truly random numbers is an essential pillar for the security, reliability, and integrity of many applications.

Traditional TRNGs (true random number generators) have long relied on deterministic physical events such as radioactive fallout or atmospheric noise to generate randomness. However, as chance increases, the need increases and innovative approaches are sought to meet these needs. Participate in the exciting concept of using a lava lamp like TRNG. Although it may seem strange or even unusual, the irregular and chaotic movement of its wax spots is a unique source of randomness.

This study begins an investigation of this unusual method and explores its potential as a reliable source of randomness. By capturing and analyzing the lava's waxing movement through video recordings and advanced algorithms, the study aims to confirm the effectiveness of the TRNG lava lamp. The aim of this study is not only to understand the reliability of the lamp's random number generator but also to try to improve the algorithms used to interpret its movement. In addition, it explores possible applications in various industries and aims to position the TRNG lava lamp as a viable and cost-effective alternative to traditional methods.

II. LITERATURE SURVEY

We discovered numerous research papers related to our concept. Among them, we've compiled a survey paper that consolidates key findings and insights from several studies. Nicola Massari et al. proposed 16.3 A 16×16 pixels SPAD-based 128-Mb/s quantum random number generator with −74dB light rejection ratio and −6.7ppm/°C bias sensitivity on temperature [1], which introduces a 16x16 pixel array using SPADs to address the bias variation issues in bit streams caused by external factors. Each pixel has two SPADs and an arbiter block that compares detected photons to assign random bits. The arbiter block ensures reliability even with simultaneous SPAD firing. Yuanzhuo Qu et al. proposed A True Random Number Generator based on Parallel STT-MTJs [2], which introduces a TRNG that uses Magnetic Tunnel Junctions (MTJs) with CMOS compatibility. This minimizes device variation effects. The parallel structure effectively reduces device variation effects, which enhances reliability and speed when compared to single MTJ designs. Giuseppe Martini et al. proposed True Random Numbers Generation from stationary Stochastic Processes [3], where two independent stationary Stochastic Processes (SPs) generated from radioactive decay are used as the source for the TRNG, which offers independence from environmental factors such as temperature, chemical composition, and aging. The process of obtaining the SPs involves the detection and selection of gamma rays with varying energy that are released by distinct nuclides within the decay chain of the radioactive source.

Tamas Gyorfi et al. proposed High performance true random number generator based on FPGA block RAMs [4], which utilizes a True Dual-Port Block RAM, commonly found in recent Xilinx Virtex© and Spartan© FPGAs as the source for the TRNG. This component's two ports: A and B, support concurrent write operations at the same memory location. When concurrent writes with different values occur at the same address, the result is considered non-deterministic, leading to uncertainty about the stored value. This offers stability and encapsulates all components within one chip. Eryn Aguilar et al. proposed Highly Parallel Seedless Random Number Generation from Arbitrary Thread Schedule Reconstruction [5], which uses the compare-and-swap instruction to propose CAS-RNG. This modifies the two-thread reconstruction to an N-thread schedule reconstruction problem, allowing for the reconstruction of reads and writes from N concurrent threads. This reduces the likelihood of repeated histories by increasing the number of parallel threads. Thomas Arciuolo et al. proposed Parallel, True Random Number Generator (P-TRNG): Using Parallelism for Fast True Random Number Generation in Hardware [6], which introduces a hardware-based True Random Number Generator (TRNG) that leverages parallelism to improve speed without sacrificing randomness. By exploiting physical phenomena as entropy sources and utilizing parallel processing, the P-TRNG offers faster and more reliable random number generation suitable for security-sensitive applications.

K. Sathya et al. proposed Random number generation based on sensor with decimation method [7], that creates a random number generation using sensors coupled with a decimation method. The core idea is to utilize sensor readings, which inherently capture environmental or physical variations, as a source of entropy for generating random numbers. R. Chase Harrison et al. proposed A True Random Number Generator based on a Chaotic Jerk System [8], which presents a Chaotic Jerk System-based True Random Number Generator (TRNG), that uses the chaotic and unpredictable behaviour of this mathematical model to generate really random numbers.The jerk system's chaotic dynamics guarantee unpredictability, boosting security and making it appropriate for simulations, cryptography, and other domains needing dependable random number generation**.** Kyle Wallace et al. proposed Toward Sensor-Based Random Number Generation for Mobile and IoT Devices [9], where the author integrate sensors and develop algorithms to extract entropy from sensor data, aiming to produce high-quality random numbers. The study validates their approach with empirical results, emphasizing the potential of using built-in sensors for efficient random number generation. Gustavo Marques Netto et al. proposed Water surface reconstruction and truly random numbers generation from images of wind-generated gravity waves [10], which presents a method to reconstruct water surfaces from wind-generated wave images. The data is harnessed for true random number generation, extracting entropy from the water surface characteristics. The approach validates both image reconstruction and random number generation with empirical results.

III. PROPOSED SYSTEM

Random Number Generation:

We generate random numbers by first using footage of a lava lamp as the source for the entropy. We process the footage frame-by-frame to capture every subtle movement of the wax blobs. Each acquired frame is then converted from colorized to grayscale. This is done because colorized data consists of multiple channels. Channels refer to the different aspects of color information, such as red, green, blue, hue, saturation, etc depending on the color space used. Whereas, grayscale consists of a single channel, i.e., luminance or intensity. Therefore, a colorized frame possesses more information than a grayscale frame. This can be less efficient since colorized data requires more resources when compared to grayscale data. Therefore for simplification, the colorized frames are converted to grayscale frames. Then these grayscale frames are converted from numpy arrays to bytes object, which is the suitable format for hashing. These bytes objects are processed by the hash object with the help of the SHA-256 hashing algorithm, which transforms the data to fixed-size hash values(256 bits). Then the raw binary hash values are converted into hexadecimal values. This is then used for cryptographic encryptions.

Proposed System Architecture :

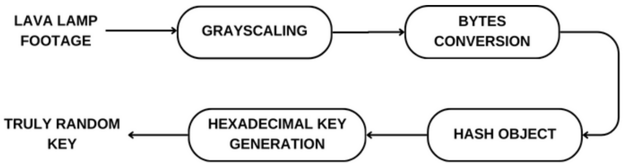


Fig. 1. Proposed System Architecture

The proposed system architecture for our True Random Number Generator (TRNG) using a lava lamp involves several key components. To make processing and analysis easier, the lava lamp film is first grayscaled, which turns colorized frames into grayscale images. These grayscale pictures are then transformed into bytes objects so that they may be used with cryptography. After that, the bytes objects are fed into a hashing method, like SHA-256, to produce a hash object with an output of 256 bits, which guarantees robust randomness. Ultimately, additional processing is applied to the hash object to generate a hexadecimal key that may be used for both encryption and decryption procedures. This design creates secure random numbers appropriate for cryptographic applications by utilizing the lava lamp's inherent randomness as well as effective data processing methods.

IV. MODULES

Randomness Extraction :

A randomness extraction module is used once the hash object's hexadecimal representation has been obtained. For cryptographic applications, this module makes sure that the generated random numbers have a uniform distribution and statistical randomness.

Key Management:

The hash object produces a hexadecimal key that is used as a cryptographic key for the encryption and decryption procedures. To safely store and maintain these keys and guarantee data integrity and confidentiality during transmission, a key management module is put into place.

Encryption Algorithm:

To protect data security, an encryption algorithm module encrypts plaintext data using the created hexadecimal key. Depending on the security needs, common encryption methods like RSA (Rivest–Shamir–Adleman) or AES (Advanced Encryption Standard) can be used.

Decryption Algorithm:

A decryption algorithm module is implemented in order to decrypt ciphertext back to its original plaintext form. This module ensures data integrity and authenticity throughout decryption procedures by using the same cryptographic key and algorithm parameters used for encryption.

Random Number Testing:

Using a random number testing program, the generated numbers' unpredictability is confirmed. To evaluate the accuracy and randomness of the generated numbers, this module runs statistical tests, including the NIST (National Institute of Standards and Technology) statistical test suite.

Performance Analysis:

Performance analysis, which assesses the lava lamp TRNG system's computational overhead and efficiency, is a crucial component of the implementation process. This module evaluates aspects including scalability, resource usage, and processing speed for practical applications.

V. IMPLEMENTATION

This TRNG is implemented by using various tools such as Python programming language, OpenCV library for processing and grayscaling of the lava lamp footage, and Hashlib library for encryption and hexadecimal conversion. OpenCV (Open Source Computer Vision Library) is an open-source computer vision library. In this TRNG, it is used for image processing and grayscale conversion. The Hashlib library is used to implement the SHA-256 hash function, which generates a fixed-size 256-bit / 32-byte hash value.

Grayscaling:

The lava lamp footage is given as an input(frame by frame) and the colorized footage is converted into grayscale. This is done because colorized images have multiple channels, each representing a different aspect of the image’s color information. Whereas, in a grayscale image, there is only one channel which represents the intensity of light. So, this makes the grayscale image easier to process.

Bytes Conversion:

The grayscale image is converted into bytes object. This is done because the grayscale image cannot be converted into a hash object, since the SHA-256 hash expects a bytes object as input.

Hash Object:

The bytes object is given as input to the SHA-256 hash and it creates a hash object. The output generated is of 256-bit and it provides approximately 10^77 different combinations. This hash object can then be converted into various representations, such as hexadecimal or binary.

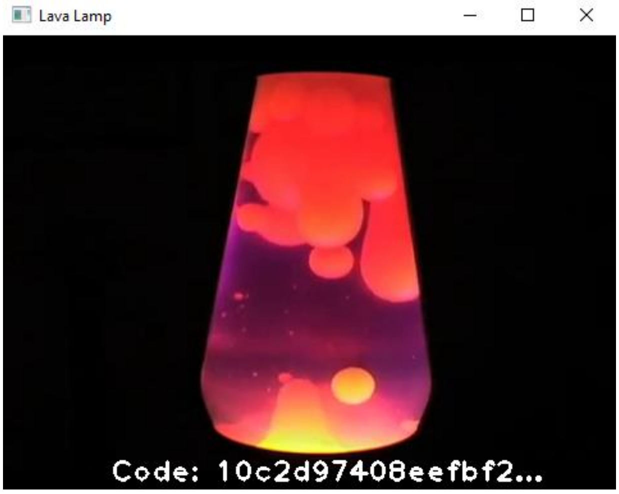
Hexadecimal Key Generation:

The hash object generated from SHA-256 hash is then converted into hexadecimal representation. The hexadecimal output consists of 64 characters / 32 bytes.

Output:

The random numbers generated from our project is a 256-bit/64-character hexadecimal string. i.e., each character represents 4 bits. The output is generated from the SHA-256 hashing algorithm. This 256-bit output provides approximately 10^77 different combinations, i.e., There are 256 bits and each bit has 2 values, which gives us 2^256, this makes it nearly impossible to crack. We use the 256-bit format because it is a standardized format for many cryptographic algorithms and protocols. It also provides a high level of resistance to brute force and collision attacks. The output string can further be converted to binary, integer, or floating-point data based on different applications of the generated code.

- Colorized frame:



Code: 10c2d97408eefbf2...

Fig. 2. Colorized frame

The lava lamp's colorized frame is a snapshot of the fascinating, erratic motion of the wax blobs inside the lamp. The behavior of the lava lamp, which is the main source of unpredictability for our True Random Number Generator (TRNG) system, is dynamic and unexpected, as this frame captures. The frame's flowing lines and brilliant hues capture the innate randomness that we use to produce objective, safe random numbers.

- Grayscale frame:

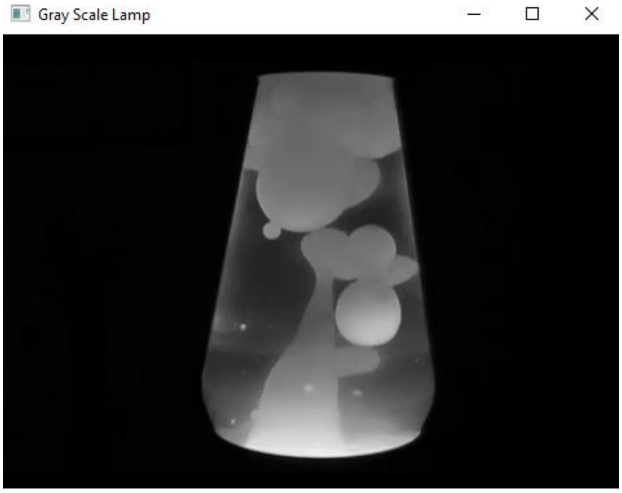
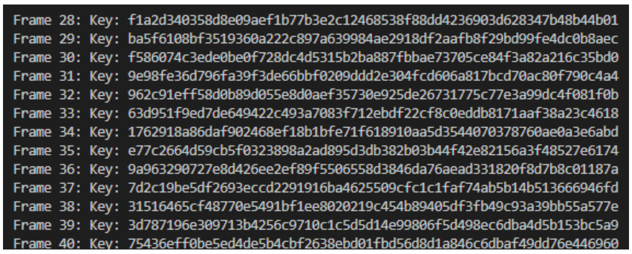


Fig. 3. Grayscale frame

The original image has been simplified and standardized in the grayscale frame, which was created from the colorized frame of the lava lamp. We remove the complexity of numerous color channels and concentrate just on the light intensity in each pixel by converting the colorized frame to grayscale. Since this grayscale form requires less computing power to process, it can be used for further data manipulation and analysis. An essential intermediate stage in our True Random Number Generator (TRNG) method is the grayscale frame, which allows us to extract randomness from the dynamic behavior of the lava lamp and analyze motion patterns more effectively.

- Generated 64-character hexadecimal values:



```
Frame 28: Key: f1a2d340358d8e09aef1b77b3e2c12468538f88dd4236903d628347b48b44b01
Frame 29: Key: ba5f6108bf3519360a222c897a639984ae2918df2aafb8f29bd99fe4dc0b8aec
Frame 30: Key: f586074c3ede0be0f728dc4d5315b2ba887fbbae73705ce84f3a82a216c35bd0
Frame 31: Key: 9e98fe36d796fa39f3de66bbf0209ddd2e304fcd606a817bcd70ac80f790c4a4
Frame 32: Key: 962c91eff58d0b89d055e8d0aef35730e925de26731775c77e3a99dc4f081f0b
Frame 33: Key: 63d951f9ed7de649422c493a7083f712ebdf22cf8c0eddb8171aaf38a23c4618
Frame 34: Key: 1762918a86daf902468ef18b1bfe71f618910aa5d3544070378760ae0a3e6abd
Frame 35: Key: e77c2664d59cb5f0323898a2ad895d3db382b03b44f42e82156a3f48527e6174
Frame 36: Key: 9a963290727e8d426ee2ef89f5506558d3846da76aead331820f8d7b8c01187a
Frame 37: Key: 7d2c19be5df2693eccd2291916ba4625509cfc1c1faf74ab5b14b513666946fd
Frame 38: Key: 31516465cf48770e5491bf1ee8020219c454b89405df3fb49c93a39bb55a577e
Frame 39: Key: 3d787196e309713b4256c9710c1c5d5d14e99806f5d498ec6dba4d5b153bc5a9
Frame 40: Key: 75436eff0be5ed4de5b4cbf2638ebd01fbd56d8d1a846c6dbaf49dd76e446960
```

Fig. 4. Generated hexadecimal codes

Our True Random Number Generator (TRNG) technology produced 64-character hexadecimal numbers based on the motion patterns of the analyzed lava lamp data. With 16 bytes or 32 hexadecimal characters in each hexadecimal value, there is a great deal of unpredictability and randomness. These numbers are used as random seeds or cryptographic keys for a number of uses, such as digital signatures, encryption, decryption, and secure communication protocols. Because the generated hexadecimal values are 64 characters long, which guarantees a sufficient amount of entropy, they can be used for cryptographic operations that need robust and safe random numbers.

Finally, the images of our True Random Number Generator (TRNG) system demonstrate a smooth transition from the complex dynamics of the wax motions in the lava lamp to the creation of trustworthy and safe random numbers. The grayscale representation simplifies the data for effective processing, while the colorized frame captures the essence of randomness inherent in the behavior of the lava lamp. In the end, the system outputs 64-character hexadecimal values, which stand for robust cryptographic keys that are necessary to guarantee the security and integrity of data. These screenshots demonstrate our TRNG approach's strength and efficacy, which makes it an invaluable tool for safe data processing and cryptographic applications.

VI. RESULTS & DISCUSSION

Our project's outcomes show how well a lava lamp works as a True Random Number Generator (TRNG) for achieving secure unpredictability. We determined that the random numbers produced by our system satisfy the requirements for true randomness by conducting a thorough study and testing to show that they have a uniform distribution and statistical independence. This demonstrates the practicality of lava lamps as an unusual but trustworthy way to produce random numbers, which may find use in simulations, cryptographic protocols, and other contexts where safe and objective randomness is needed. The ramifications of our findings are explored, emphasizing the benefits of lava lamps for TRNGs, including their ease of use, affordability, and inherent randomness. We also discuss some drawbacks and topics for additional study, such as data optimization.

VII. CONCLUSION

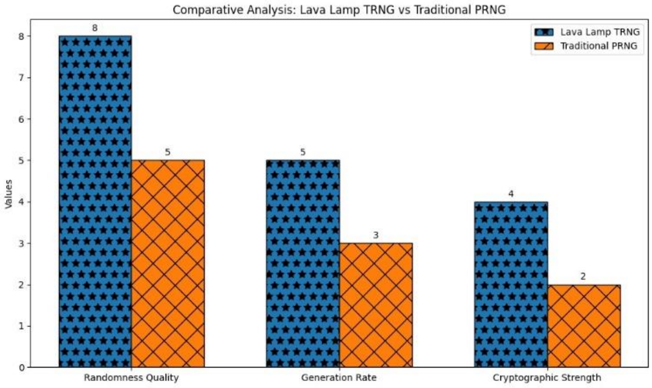


Fig. 5. Comparative analysis

The comparative analysis graph illustrates the key metrics of Randomness Quality, Generation Rate, and Cryptographic Strength between the Lava Lamp True Random Number Generator (TRNG) and traditional Pseudorandom Number Generators (PRNGs).The above depiction highlights the exceptional capabilities of the Lava Lamp TRNG with respect to these crucial aspects, underscoring its promise as a dependable and resilient technique for producing random numbers. The graph's unique hatch patterns draw attention to the project's creative use of non-traditional but efficient randomness generation techniques for cryptographic applications. The project's importance in pursuing new directions to improve cryptographic strength and randomness quality in data security and encryption systems is highlighted by its implementation.

In conclusion, the exploration of using a lava lamp as a True Random Number Generator (TRNG) offers a fascinating and promising avenue in the realm of random number generation. Preliminary findings suggest that the erratic and unpredictable motion of the lava lamp's wax blobs can indeed serve as a reliable source of randomness. This unconventional approach challenges traditional methods and demonstrates the potential for innovative solutions in generating high-quality random numbers.

As the research progresses, the focus will shift towards refining the algorithms used to interpret the lava lamp's motion, aiming for consistency and dependability. Furthermore, the study's findings open doors to diverse applications across various industries where true randomness is crucial, from secure financial transactions to scientific simulations and communication protocols.

Ultimately, the goal is to establish the lava lamp TRNG as a robust, adaptable, and cost-effective alternative to conventional random number generators. By doing so, this research not only contributes to the field of cryptography but also highlights the potential of leveraging unconventional sources for solving complex technological challenges.

In the future, research will concentrate on a few major areas to increase the practicality and efficacy of employing lava lamps as True Random Number Generators (TRNGs). This entails refining the lava lamp configuration to enhance randomness production, creating sophisticated algorithms for more precise motion pattern analysis, assessing compatibility with accepted cryptography standards, and verifying the system's resilience in real-world scenarios. We'll also investigate new applications like blockchain integration and IoT security, work with industry partners to expand adoption and deployment, get user input for better usability, carry out extensive security audits, maximize scalability for large-scale random number generation, and encourage partnerships with researchers for knowledge exchange and ongoing progress in the field. These initiatives are meant to raise the lava.

REFERENCES

[1] Nicola Massari, Leonardo Gasparini, Alessandro Tomasi, Alessio Meneghetti, Hesong Xu, Daniele Perenzoni, Guglielmo Morgari, David Stoppa, "16.3 A 16×16 pixels SPAD-based 128-Mb/s Quantum Random Number Generator with -74dB Light Rejection Ratio and -6.7ppm/°C Bias Sensitivity on Temperature", IEEE International Solid-State Circuits Conference (ISSCC), Session 16, pp. 292-294, February 2016.

[2] Yuanzhuo Qu, Jie Han, Bruce F. Cockburn, Witold Pedrycz, Yue Zhang, Weisheng Zhao, "A True Random Number Generator based on Parallel STT-MTJs", Design, Automation and Test in Europe, pp. 606- 609, 2017.

[3] G. Martini and F. G. Bruno, "True Random Numbers Generation from stationary Stochastic Processes", 2017 International Conference on Noise and Fluctuations (ICNF), Vilnius, 2017, pp. 1-4, doi: 10.1109/ICNF.2017.7985997

[4] Tamas Györfi, OctaviaQ &UHĠ$OLQ 6XFLX, "High Performance True Random Number Generator Based on FPGA Block RAMs", IEEE, 2009

[5] Eryn Aguilar, Jevis Dancel, Deysaree Mamaud, Dorothy Pirosch, Farin Tavacoli, Felix Zhan, Robbie Pearce, Margaret Novack, Hokunani Keehu, Benjamin

Lowe, Justin Zhan, Laxmi Gewali, Paul Oh, "Highly Parallel Seedless Random Number Generation from Arbitrary Thread Schedule Reconstruction", IEEE International Conference on Big Knowledge (ICBK), pp. 1-8, 2019

[6] Thomas Arciuolo, Khaled M. Elleithy, “Parallel, True Random Number Generator (P-TRNG): Using Parallelism for Fast True Random Number Generation in Hardware” | 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)

[7] K. Sathya, J. Premalatha, Vani Rajasekar, ”Random number generation based on sensor with decimation method” | 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)

[8] R. Chase Harrison, Benjamin K. Rhea, Ariel N. Ramsey, Robert N. Dean, J. Edmon Perkins, “A True Random Number Generator based on a Chaotic Jerk System” | 2019 SoutheastCon

[9] Kyle Wallace, Kevin Moran, Ed Novak, Gang Zhou, Kun Sun, “Toward Sensor-Based Random Number Generation for Mobile and IoT Devices” | IEEE Internet of Things Journal (Volume: 3, Issue: 6, December 2016)

[10] Gustavo Marques Netto, Leandro A. F. Fernandes, “Water surface reconstruction and truly random numbers generation from images of wind-generated gravity waves” | 2017 IEEE International Conference on Image Processing (ICIP)