Techno Science
International Scientific
Journal of Technical
Sciences

# *Random Number Generators and Their Application*

**MAID OMEROVIĆ, ALJO DELIĆ, AHMED PALIĆ**

FACULTY OF TECHNICAL STUDIES, UNIVERSITY OF TRAVNIK, BOSNIA AND HERZEGOVINA

**ABSTRACT**

This paper will address random number generators. At the beginning, we will define and classify them. Subsequently, we will introduce the methods, and then the algorithms for random number generation. It is important to note that, in this paper, we will state one of the random number generation algorithms, provide details about it, and then compare a histogram of the ideal generator with those stated in this paper. At the end of the paper, the highly important application of random number generators will be described through certain methods and algorithms.

**Keywords:** random number generators, algorithms, physical methods, computational methods, Lehmer LCG, entropy, histogram

## Random Number Generators

A random number generator *(RNG)* is a device or an algorithm which, as its output, provides statistically, completely independent and unpredictable values in the form of a number sequence. Random number generators are generally divided into: true random number generators and pseudo-random number generators.The fundamental difference between them is as follows: the output of pseudo-random number generators becomes repetitive after some time, while it is not the case with true random number generators. We will make use of pseudo-random number generators in the case when the system we are making is less complex. True random number generators are almost impossible to design for computing since computers are deterministic machines.

## Random Number Generation Methods

### Physical Methods

With the goal to generate (true) random numbers, it is necessary to observe and note some unpredictable natural phenomena. Designing a circuit device and/or a computer programme which would gather sampled values of a measurable quantity related to an observed phenomena, free from correlations and predictability, is an extremely difficult task. For cryptographic applications, it is also necessary that such a generator is resistant to possible observations and manipulations of the internal state.

Circuit random number generators use unpredictability appearing in the following physical phenomena: thermal noise of a semiconductor diode or an electric resistor, the time between two particle emissions during radioactive decay, free oscillator frequency instability, the charge of a capacitor after the set charging period, changes in the latent readout time caused by air turbulence within the hard disk drive enclosure, the sound from a microphone or an image from a camera. However, physical phenomena and the tools used for their measurements typically have both asymmetric and systematic deviations due

to which their outputs and not uniformly random. Another frequent and interesting source of entropy can be the behaviour of a human as the user of a system. Humans cannot be considered good random generators when it is demanded of them, but, when they use a system, their unconscious behaviour can surely be put to good use. The best example for this is a new trend in programmes dealing with computer security. They demand that their users move the mouse or press the keys on the keyboard for a time in order to create enough entropy to generate a random key or initialise a pseudo-random number generator.

There are also other processes upon which we can base software random number generators: system clock reading, the contents of input-output storage of certain computer components, RAM contents, contents of processor registers, and the state of the operating system.

**Computational Methods**

Due to the deterministic nature of computers, the term random number generation on a computer is usually attributed the prefix pseudo.
A pseudo-random number generator *(PRNG)* is a deterministic algorithm that, for the given random value of the k bit quantity, renders a number sequence of the value $l >> k$ which behaves as if it is random. The input value is usually called seed, while the output sequence is termed pseudorandom number sequence. It is important to mention the reproducibility of *PRNG* results, that is, for equal seed values, we receive identical output sequences, and, after a period of time, the results start repeating.

One of the first methods for random number generation was developed by ***John Von Neumann.***

It is called the middle-square method which is so simple that it can be calculated without the use of a computer, but its output is of extremely low quality. It should be noted that almost all programming languages offer ready-made functions and libraries for random number generation. It most frequently boils down to generating random bytes or words.

However, one should be cautious because such functions often have poor statistical properties and some will repeat patterns after only tens of thousands of trials.

That is why many operating systems offer built-in random number sources of much higher quality.

# Random Number Generation Algorithms

The following will list only one of the random number generation algorithms.

**Lehmer LCG**

It is sometimes also referred to as the Park–Miller random number generator since this variation is most frequently used. The algorithm belongs to the type of multiplicative linear congruential generator (LCG). LCGs are extremely easy for implementation.

There are several types of LCGs and each has distinct demands. Lehmer's algorithm requires a selection of two integers:

1. modulus $m$, which has to be a prime number;

2. a multiplier $a$, which is an integer from the interval $2, 3, ..., m - 1$.

A sequence of random numbers $\{zn\}$ is obtained by the following recursion relation:
$zn+1 = f(zn)\ n\ 2\ N$
$f(z) = az \bmod m$

The first number of the sequence is also the generator seed which can be selected from the interval $1, 2, ..., m - 1$. Likewise, the stated random number generator generates a number sequence $\{zn\}$ in the stated interval. That is why it is necessary to scale the sequence in order to obtain numbers distributed in the interval [0, 1]. The final sequence $\{un\}$ is defined by the expression $\{un\} = \{zn\}/m$.
Then, by selecting the parameters $a$ and $m$, we strive to accomplish the best possible generator properties.

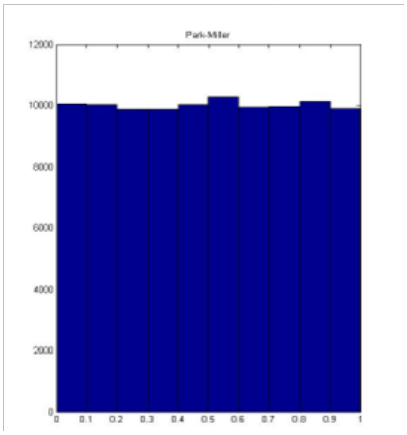


Image 1. Park–Miller LCG Histogram

### Entropy

In information theory, entropy is a measure of random variable unpredictability.
Most frequently, entropy signifies the notion of Shannon entropy. Shannon entropy expresses the expected value of information in a message. The message in our case is the realisation of the random variable, i.e.,a sequence of pseudo-random numbers.

Ideal random number generator entropy is H(X) = 8 [*bit/byte*].

| | Park-Miller | Mersenne twister |
|---|---|---|
| **H(X) [bit/byte]** | 7.9803 | 7.9823 |

Table 1. Park-Miller and Mersenne Twister Entropy

The following will list only one of the random number generation algorithms.

### Histogram

In statistics, a histogram is a graphic representation of data distribution. The data are distributed into classes, and the height of the individual bin signifies its frequency density. By displaying histograms (images 1. and 2.), we have reconfirmed the pseudo-random properties of the implemented generators.

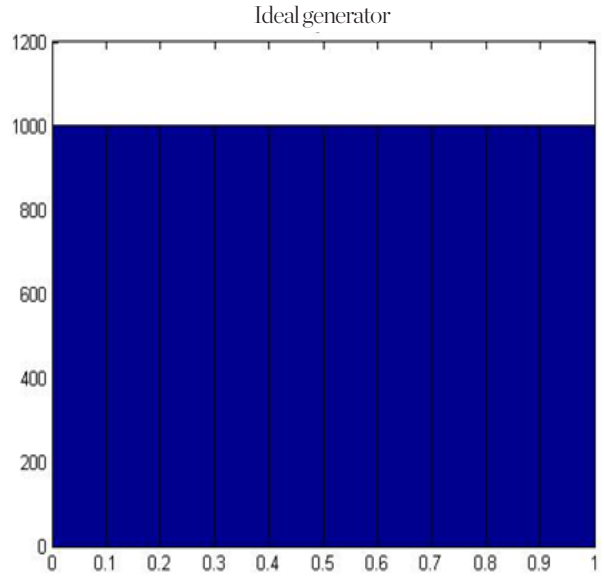


Image 2. Histogram of the Ideal Random Number Generator

# True Random Number Generator Circuitry

### Area of Application and the Purpose of the Circuitry

True random number generators have a great advantage in the fact that they create sequences which are impossible to predict since they use accidental physical phenomena as the source of their randomness.
They have a wide application for the following purposes:

***Encryption*** is a process in cryptography with which data are modified so that the message, i.e., information is made illegible for people who do not have the encryption key. Random numbers are used in both symmetric and asymmetric cryptography as a way to generate keys. Since integrity must be kept in communication between two parties, it is necessary to keep the confidentiality of the key in use. In this context, true random number generators are very important since they insure adequate randomness necessary for maintaining the security of the sent messages.

***Psychokinesis*** relates to moving or influencing certain things by the power of the mind. Within the limits of the generally recognised mechanical-physical laws, it deals with the impossible or, at least, unexplained phenomena.
In research related to psychokinesis, true random number generators are used, and the participants, through the concentrated guidance of their thoughts, strive to impact the output values of the generator. An insight into several million trials allowed the scientists to establish the existence of small, but statistically significant signs pointing at the fact that it could be possible to interact with machines through thoughts. However, in the absence of real evidence, scientists do not claim that it is precisely the thoughts of a human that cause the change in the output data of a generator.

### Application of Random Number Generators

Random number generators have applications in gambling, statistical sampling, various computer simulations, cryptography, and other areas where producing an unpredictable result is desirable.
It is important to note that in cases where randomness is of utmost importance, for example, when it comes to security systems, hardware implementation is preferred over pseudo-random algorithms.

Random number generators are extremely useful in developing Monte Carlo simulations, as well as for debugging since they provide the option of starting the same sequence of random numbers beginning with the same random seed. In addition, they are frequently used in cryptography as long as the seed is kept secret. Today, information security is an extremely important issue which is heavily invested in, with random number generators playing a key role, especially in generating passwords and asymmetric keys. Generating pseudo-random numbers is an important and frequent task in computer programming.

For example, while cryptography and other numerical algorithms require a high degree of randomness, most algorithms can still function properly with a significantly smaller one. We have already encountered such algorithms, and they are frequently used (search and sort algorithms). It is important that a generator is optimally adjusted to the system requirements, and it should be primarily ensured that the very generation procedure is fast, efficient and precise. In fact, it could be stated that we have made a quality generator if the manner in which numbers are generated cannot be detected.

The complexity of the very notion of randomness should be reemphasised.

One of the frequent applications which we encounter on a daily basis is on our mp3 player or iPod where we select the option *shuffle* instead of directly choosing the song we want to listen to. Even though the choice should be completely random, it is definitely not desirable that a song is reproduced two or several times in a row.

# Conclusion

In conclusion, it is important to state that a random number generator is a device or an algorithm which, as its output, provides statistically, completely independent and unpredictable values in the form of a number sequence. Random number generators are generally divided into: true random number generators and pseudo-random number generators.

Random number generators have applications in gambling, statistical sampling, various computer simulations, cryptography, and other areas where producing an unpredictable result is desirable. In addition, they are frequently used in cryptography as long as the seed is kept secret.

Today, information security is an extremely important issue which is heavily invested in, with random number generators playing a key role, especially in generating passwords and asymmetric keys.

# References

1. The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 2nd Edition, Donald E. Knuth, 1981.
2. Laboratorijska vježba 1. – Generiranje slučajnih brojeva (LCG), generiranje slučajnih procesa, uzoračka srednja vrijednost, devijacija te autokorelacijska i kroskorelacijska funkcija, Tomislav Petkovic, 2009
3. Testing Random Number Generators, Dr. John Mellor-Crummey, Department of Computer Science Rice University, 2005.
4. Pseudo-Random Number Generators, Vivien Challis, Part of the postgraduate journal club series, Mathematics, UQ, 2008.

Techno Science
International Scientific
Journal of Technical
Sciences

# *Generatori slučajnih brojeva i njihova primjena*

**SAŽETAK**

U ovom radu biti će riječi o generatorima slučajnih brojeva. Na početku ćemo definisati nevedeno, izvršiti određenu podjelu. Nakon svega toga biti će uvedeni postupci za genenerisanje slučajnih brojeva, a potom algoritmi za generisanje slučajnih brojeva. Bitno je napomenuti da ćemo mi u ovom radu navesti jedan od algoritama za generisanje slučajnih brojeva, navesti neke detalje o istom, a potom usporediti histogram idealnog generatora sa nekim navedenim u ovom radu. Na kraju rada, veoma važna, primjena generatora slučajnih brojeva biti će opisana kroz pojedine metode i algoritme.

**Ključne riječi:** Generatori slučajnih brojeva, fizički postupci, računarske metode, algoritmi, Lehmerov LCG, entropija, histogram

Received: November 29, 2018 / Accepted: December 03, 2018
Correspondence to: mr.Maid Omerović, University of Travnik, Faculty of Technical Studies, Travnik, Bosnia and Herzegovina
E-mail: maid.omerovic@gmail.com