Volume 6, Issue 1, January 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Random Number Generation Methods a Survey

Antu Annam Thomas*
Mar Thoma College,
Thiruvalla, Kerala, India

Varghese Paul
Department of Computer Science,
Cochin University, Kerala, India

Abstract— Random numbers are numbers that occur in a sequence such that the values are uniformly distributed over a defined interval or set, and it is impossible to predict future values based on past or present ones. This paper deals with the methods that generate such a sequence of numbers. It covers the two main groups of generators, true-random number generators and pseudo-random number generators.

Keywords— random numbers; pseudo random; true random.

I. INTRODUCTION

Random number generation techniques generate a sequence of numbers that cannot be reasonably predicted better than by a random chance. Dice, coin flipping, the shuffling of playing cards are some of the ancient methods for generating random sequences. Because of the mechanical nature of these techniques, generating large numbers of sufficiently random numbers requires a lot of work and time. Nowadays, after the advent of computational random number generators(RNGs), many application like gambling, statistical sampling, computer simulation, cryptography, government-run lotteries, and lottery games, are using computational RNGs instead of more traditional drawing methods. There are two principal methods used to generate random numbers. The first method known as true random number generators measures some physical phenomenon that is expected to be random and uses the measure to generate random number sequence. The second method known as pseudo random number generator uses computational algorithms that can produce long sequences of apparently random results, which are in fact completely determined by a shorter initial value, known as a seed value or key.

II. RANDOMNESS AND ITS GENERATION TECHNIQUES

A. *Randomness*

Oxford Dictionary defines Randomness as the lack of pattern or predictability in events. A random sequence of events will not have an order and does not follow an intelligible pattern or combination. Individual random events are thus unpredictable. Random numbers can be divided into three categories: true random numbers, pseudorandom numbers and quasirandom numbers. In true random number sequence there is no way to predict what the next random number is. Pseudo random number sequence is sequence of numbers which is algorithmically produced and not really random. The sequence can be fully repeated if the initial conditions and algorithm are known. Quasirandom number sequence act as random number sequence in some sort of simulations, but are well-ordered in some other types. Randomness has many uses in science, art, statistics, cryptography, gaming, gambling, and other fields. Each of these uses has different levels of requirements, which leads to the use of different methods for generating them.

B. *Random Number Generators*

Random number generators define methods that help in generating randomness in the generated sequence of numbers. It is difficult to design random number generators that generate ideal randomness. Due to this difficulty in many situations pseudo random number generators are used that generates pseudo random number sequences. But this does not mean that there aren't any true random number generators. Thus depending upon the requirement of the application we choose either pseudo random number generators or true random number generators.

III. TRUE RANDOM NUMBER GENERATORS

High-quality random number generation is essentially demanded for security. True random number generators capture random events in the real world to create its sequences. . Each bit of the bitstreams is independent from the other bits and the probabilities of 1/0 occurrences are identical. The generated sequence is such that it is impossible for anyone to predict the next number in the sequence. Because true random numbers cannot be predicted by computational methods, they are highly desirable for security purposes

A true random number generator uses entropy sources that already exist. Entropy refers to the amount of uncertainty about an outcome of an event. Real world events such as flipping coin and rolling of dice have a high degree of entropy, because it is almost impossible to predict accurately what the final result will be. It is the source of entropy that makes a true random number generator unpredictable.

Since mechanical actions, like flipping of coin and rolling of dice, is involved in such methods the rate at which random numbers could be produced would be restricted. Low production rate is thus a problem that plagues most true random number generators.

Since they use real world phenomena, some physical device capable of recording the event is needed. This gives way to the next disadvantage of these generators that is they rely on some sort of hardware. This can make true random generators a lot more expensive to implement, especially if the necessary device is not commonly used. The use of physical devices makes these generators vulnerable to physical attacks that can bias the number sequences.

Finally, even when there are no attackers present, physical devices are typically vulnerable to wear over time and errors in their construction that can naturally bias the sequences produced. To overcome bias, most true random number generators have some sort of post processing algorithm that can compensate for it.

Despite these disadvantages, many applications uses true random number generators itself because no mathematician can break a code that does not exist.

Some major true random generators are Random.org, Hotbits, lasers, and oscillators.[1]

A. Random.org

Random.org is a website created in 1998 by Mads Haahr. The website produces "true" random numbers based on atmospheric noise captured by several radios tuned between stations. In addition to generating random numbers in a specified range and subject to a specified probability distribution, it has free tools to simulate events such as flipping coins, shuffling cards, and rolling dice. It also offers paid services to generate longer sequences of random numbers and act as a third-party arbiter for raffles, sweepstakes, and promotions. Today, RANDOM.ORG is operated by Randomness and Integrity Services Ltd.[2]

B. HotBits

HotBits is an Internet resource that brings genuine random numbers, generated by a process fundamentally governed by the inherent uncertainty in the quantum mechanical laws of nature, directly to your computer in a variety of forms.HotBits are generated by timing successive pairs of radioactive decays detected by a Geiger-Müller tube interfaced to a computer. You order up your serving of HotBits by filling out a request form specifying how many random bytes you want and in which format you'd like them delivered. Your request is relayed to the HotBits server, which flashes the random bytes back to you over the Web. Since the HotBits generation hardware produces data at a modest rate (about 100 bytes per second), requests are filled from an “inventory” of pre-built HotBits. Once the random bytes are delivered to you, they are immediately discarded—the same data will never be sent to any other user and no records are kept of the data at this or any other site.

C. Lasers

Random.org and Hotbits though they are good generators the use of internet to satisfy the request for random number sequence give way to security threats like intruding, unauthorised access and so on. Physical entropy sources such as electronic and photonic noise can be used for generating random numbers but there is a large gap between the generation rates achieved with existing physical sources and the high data rates of many computation and communication systems. This is the fundamental weakness of these systems. Random number generation scheme that uses broadband measurements of the vacuum field contained in the radio-frequency sidebands of a single-mode laser is a good advancement in TRNG.[3] Study of Semiconductor lasers show that good quality random bit sequences can be generated at very fast bit rates using physical chaos in semiconductor lasers.[4] In laser-based generators, entropy can be obtained by several different means. Having two photons race to a destination is one method that is currently implemented. [5] The chaos in a semiconductor laser with optical feedback has been extensively studied from the viewpoint of basic research in random number generation. The most advantageous feature of using a semiconductor laser in random bit generation is its potential high-speed performance, which can be much faster than the rate achievable with electrical circuits. [6]

D. Oscillators

True random number generators based on sampling phase jitter in oscillator rings will generate provably random bits with some tolerance to adversarial manipulation and running in the megabit-per-second range.[9]

Oscillator-based TRNG[7], which utilizes random period jitter of oscillators as random source, is one of the popular circuits for generating truly random numbers.Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

Oscillator-based TRNG with jitter amplifier is a new advancement to improve the randomness of the generated sequence.[8]

Recently TRNG using the jitter of events propagating in a self-timed ring to generate random numbers at a high bit-rate has been developed. The self-timed ring allows to adjust the time lapse between two successive events as short as needed by simply increasing its number of stages and adjusting its number of events. This time lapse can be thus adapted to the jitter magnitude which depends on the selected technology and device. Therefore, the design allows to extract entropy from the jitter even if its magnitude is extremely low. Moreover, the designer can precisely tune the architecture based on his security, throughput, cost and power consumption requirements with a very low design effort.[10]

IV. PSEUDO RANDOM NUMBER GENERATORS

Random number generators that do not rely on real world phenomena to produce their streams are referred to as pseudo random number generators. These generators appear to produce random sequences to anyone who does not know the secret initial value. In a minimalistic generator, the initial value will be the only time entropy is introduced into the system. Unlike true random number generators that convert entropy sources directly into sequences, a pseudo random needs to find entropy to use to keep itself unpredictable. Classic tactics for accomplishing this include taking the time of day, the location of the mouse, or the activity on the keyboard.[1]

Middle-square method, , Linear Feedback Shift Registers, Xorshift generators are some of the early Pseudo Random Number Generators.

A. Middle-square method

One of the classic approaches in PRNGs is the middle square method in which with a simple mathematical model generating pseudorandom numbers in high speed and minimum correlation between output numbers. Middle Square Method is a one of the best methods of generating pseudorandom numbers. The method was first described in a manuscript by a Franciscan friar known only as Brother Edvin sometime between 1240 and 1250. It was reinvented by John von Neumann, and was described at a conference in 1949. To generate a sequence of 4-digit pseudorandom numbers, a 4-digit starting value is created and squared, producing an 8-digit number. If the result is less than 8 digits,leading zeroes are added to compensate. The middle 4 digits of the result would be the next number in the sequence, and returned as the result. This process is then repeated to generate more numbers. But in the some practical situations it is not a good method, since its period is usually very short and its output sequence almost always converging to zero. A recent advancement that proposed a novel approach of chaotic system based on logistic map gave an improved version of Middle Square method. The chaotic system that iterated independently starting from initial conditions could help to generate appropriate values [11]

B. Linear Congruential Generators

A Linear Congruential Generator (LCG) is an algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piecewise linear equation. The method represents one of the oldest and best-known pseudorandom number generator algorithms. The theory behind them is relatively easy to understand, and they are easily implemented and fast, especially on computer hardware which can provide modulo arithmetic by storage-bit truncation.

The generator is defined by the recurrence relation:

$$X_{n+1} = (a X_n + c) \bmod m$$

where, X, is the sequence of pseudorandom values, and

m, $0 < m$ - the modulus

a, $0 < a < m$ – the multiplier

c, $0 \leq c < m$- the increment

X_0, $0 \leq X_0 < m$ – the seed or start value

are integer constants that specify the generator.

If $c = 0$, the generator is often called a multiplicative congruential generator(MCG), or Lehmer RNG.

If $c \neq 0$, the method is called a Mixed Congruential Generator. [12]

A major problem with simple linear algorithms is that the period is limited by m.

C. Lagged Fibonacci generator

A Lagged Fibonacci generator (LFG or sometimes LFib) is an example of a pseudorandom number generator. This class of random number generator is aimed at being an improvement on the 'standard' linear congruential generator. These are based on a generalisation of the Fibonacci sequence.

$$X_n = X_{n-l} + X_{n-k} \pmod m; \text{ where } l > k > 0$$

Instead of two initial values, l initial values, X0, ... , Xl-1, are needed in order to compute the next sequence element. In this expression the “lags” are k and l, so that the current value of X is determined by the value of X k places ago and l places ago. In addition, for most applications of interest m, the modulus, is a power of two. That is, m = 2M.

A very popular and widely used example of feedback shift registers is the Mersenne Twister . [13] The Mersenne Twister can be classified as a twisted generalized feedback shift register (TGFSR), which has algorithms more tightly tied to matrices than strings. The benefits of this generator are rapid number generation, highly random sequences, and a large period.

V. PSEUDO RANDOM NUMBER GENERATORS VS. TRUE RANDOM NUMBER GENERATORS

Many techniques are adopted for making random number generators. On one end of the spectrum, true random number generators take random events from the real world as input and convert them into sequences of random numbers. But these generators have been labeled as slow and difficult to install.

However, the advent of laser-based generators is helping to solve the speed problem, while circuit-based generators are being designed that utilize existing hardware. True random number generators are constantly collecting feedback from outside phenomenon so care needs to be taken so that attackers do not disrupt the environment.

On the other end of the spectrum, mathematicians and cryptographers have developed many algorithms that are unpredictable under certain circumstances. The predetermined yet unforeseen sequences that result from these methods have been labeled pseudo random.

Pseudo random number generators are more commonly used since it is easy to set up and is able to produce values quickly. Normally, these generators need to keep their initial conditions and parameters (seed value) a secret, or else anyone could compute the same number sequence. Unless these pseudo random generators have some method of refreshing themselves with real world entropy, they will eventually repeat themselves.

Comparing the results of a Pseudo-RNG with that from a True Random Number Generator will prove useful. If you list down the results of a Pseudo Random Number Generator mimicking dice rolls the numbers will really appear as if they are random. But statistical analysis will prove that the numbers produced by a Pseudo-RNG is not really random but is rather predetermined. Thus its results can be measured and standardized, and, we can say, controlled.

True Random Number Generators behave differently since the results are truly random and unpredictable. If we try to get a computer to produce a really random sequence of numbers then the computer must base its numbers on a naturally occurring physical phenomenon, which may include the radioactive decay of isotopes, atmospheric noise and so on.

True Random Number Generators will not really be cost-efficient if you compare it to Pseudo Random Number Generators True Random Number Generators will be subject to wear and tear since all things natural will really be subject to entropy. A Pseudo Random Number Generator will not be subject to such physical phenomena.

Another efficiency of a Pseudo Random Number Generator is that you can reproduce the same sequence of numbers in another time by simply knowing the starting point of the sequence. This better facilitates inspections.

The method for selecting and appraising the most appropriate random number generator is highly dependent on context. Having the best randomness is not always relevant.

VI. CONCLUSION

Random number generators thus depending on the method the generated sequence can be either true or random. The source of entropy from the real world yields a sequence that is truly random. But most of the applications go for pseudo random number generators because it is cheap and easy to install. High cost and large time taken makes true random number generators less used. The growing demand for digital unpredictability has led the field of random number generation to grow rapidly in breadth and complexity. However, the basic types and techniques of random number generators have remained stable for decades.

REFERENCES

[1] David DiCarlo, “Random Number Generation: Types and Techniques,” A Senior Thesis submitted in partial fulfillment of the requirements for graduation in the Honors Program Liberty University Spring 2012.

[2] McNichol, Tom (2003-08-11). "Totally Random". Conde Nast Publications. p. 2. Retrieved 2009-10-23. Mads Haahr, a lecturer in computer science at Trinity College in Dublin, designed the system

[3] T. Simul, S.M. Assad, P.K. Lam “Real time demonstration of high bitrate quantum random number generation with coherent laser light”, Appl Phys Lett 98:231103-1-3

[4] Atsushi Uchida, Kazuya Amano, Masaki Inoue, Kunihito Hirano, Sunao Naito, Hiroyuki Someya, Isao Oowada, Takayuki Kurashige, Masaru Shiki, Shigeru Yoshimori, Kazuyuki Yoshimura & Peter Davis, “Fast physical random bit generation with chaotic semiconductor lasers”, Nature Photonics 2, 728 - 732 (2008)

[5] Stefanov, A.,”Optical quantum random number generator”, Retrieved 10/16/2011 from http://arxiv.org/pdf/quant-ph/9907006v1, 2008

[6] Kazuyuki Yoshimura, Susumu Shinohara, and Kenichi Arai ,“ Fast Physical Random Number Generation Using Semiconductor Laser Chaos” Feature Articles: Communication Science that Connects Information and Humans, Vol. 10 No. 11 Nov. 2012,pp. 1-5

[7] B. Jun and P. Kocher, “The Intel random number generator,” cryptography research inc., white paper prepared for Intel corp., Apr., 1999.

[8] Takehiko Amaki, Masanori Hashimoto and Takao Onoye, “An Oscillator-Based True Random Number Generator with Jitter Amplifier”, Circuits and Systems (ISCAS), 2011 IEEE International Symposium on 15-18 May 2011, Rio de Janeiro, ISSN : 0271-4302, pp 725-728

[9] Sunar, B., Martin, W.J., Stinson, D.R. “A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks”, Computers, IEEE Transactions on (Volume:56 , Issue: 1), Jan. 2007, pp. 109 - 119

[10] Abdelkarim Cherkaoui, Viktor Fischer, Alain Aubert, Laurent Fesquet’ “A Self-timed Ring Based True Random Number Generator”, International symposium on advanced research in asynchronous circuits and systems - ASYNC 2013, May 2013, Santa Monica - California, United States. pp.99-106, 2013.

[11] Hamed Rahimov, Majid Babaie, Hassan Hassanabadi, “Improving Middle Square Method RNG Using Chaotic Map”, Applied Mathematics, 2011, 2, 482-486

[12] Chan, H. “Random number generation”. Retrieved 10/16/2011 from http://fuchun00.dyndns.org/~mcmintro/random.pdf, 2009.

[13] Nishimura, T, “Tables of 64-bit mersenne twisters” , ACM Transactions on Modeling and Computer Simulation, 10(4), 348-357, 2000.

