International Journal of Scientific Research in Multidisciplinary Studies
Vol.8, Issue.9, pp.90-92, September (2022)

Research Paper
E-ISSN: 2454-9312
P-ISSN: 2454-6143

Applying Motion Sensor Data from Diverse Mobile Devices to Generate Pure Random Numbers

Deep Kumar Bangotra

Department of Higher Education, J&K Govt.India-180016

Author's Mail Id: deepbangotra.ap@gmail.com

Available online at: www.isroset.org

Received: 29/Apr/2022, Accepted: 22/Jul/2022, Online: 30/Sept/2022

Abstract- The random numbers have inherent susceptibility to be reverse engineered. Therefore, the production of pseudorandom numbers is not a completely reliable process. In anyway, random number generators should not employ a seed value and should produce unpredictably generated numbers. This paper presents a way for building a pure random number generator that is highly effective, modern, and impossible to reverse-engineer. Additionally, it does not require the installation of any special equipment. It can be used to meet the needs of any user and has a lower processing complexity.
In this paper, we make use of information gathered from several mobile motion sensor devices. After combining the data, an algorithm is run through it to produce a data entropy pool with a list of random values. Then, in order to generate random numbers in the appropriate range, a pseudo random number generation process is utilised with this pure random data set as a seed. Since modern smartphones are well-equipped with sensors, this solution did not require any specialised hardware.

Keywords- Pure random number generation, Reduced Complexity, Algorithm.

I. INTRODUCTION

As we are all aware, "Von Neumann Architecture" based on computers is intended to attain 100% efficiency. This is the reason why programmers and mathematicians have been creating complex algorithms with a long duration for the development of pseudo-random numbers in order to generate random numbers.

A true random number generator, as opposed to pseudo random number generators, uses any physical phenomenon to generate unpredictability and input it into a computing system. Generally speaking, real random number generators are appropriate for applications like data games, encryption, and gambling that pseudo random number generators are not suitable for such applications.

In general, they are less suitable for modelling and simulation applications because of their poor efficiency and non-deterministic nature, which frequently need for more data that is impractical to provide with real random number generators.

II. BACKGROUND

Based on the review of literature, it has been found that there are different types of random number generators. They all offer different complexity, efficiency, reliability and correctness. The different types of random number generators are explained below.

2.1 Intel's Digital Random Number Generator (DRNG)

An effective and cutting-edge hardware method for producing high-quality, high-performance entropy and random numbers is the "Digital Random Number Generator," or DRNG[1]. The "RDRAND" and "RDSEED" new Intel 64 Architecture instructions that are included, along with a hardware implementation of a digital random number generator. According to the above-mentioned random number generator taxonomy, the random number generator uses a processor-resident entropy source to systematically seed a hardware-implemented cryptographically secure pseudo random number generator. This value generator type is known as cascade construction. Unlike software techniques, it has a high-quality entropy source utilisation that can be sampled quickly enough to continuously supply high-quality entropy values to the Cryptographically Secure Pseudo Random Number Generators.

Additionally, it stands for a self-contained hardware unit that is protected from all software attacks on its internal state. As a result, a solution is produced that produces a Random Number Generator with numerous features.

2.2 Quantis Random Number Generator

Thermal noise and atomic- or sub-atomic-scale quantum mechanics are the two main sources of practical quantum mechanical unpredictability. According to quantum mechanics, many physical processes, including the nuclear decay of an atom, are essentially random in nature and cannot be precisely predicted. Since we are not at zero temperature, every system contains a small amount of random fluctuation. For instance, the air molecules constantly bounce off one another in an unpredictable way. Because it is a quantum phenomenon, its randomness is unpredictable [2].

2.3 True Random Number Generator with a Meta-stability-Based Quality Control

A real random number generator of the meta-stability type[3] that satisfies all NIST randomness tests and has high entropy. The resolution time meta-stability of this generator serves to indicate the likelihood of randomization independent of the output. The computer is tuned to produce a high chance of randomness after determining the initial level of random noise. The user can compromise between "output bit-rate" and "quality of the bit-stream" using the grading module. With a cross-sectional area of 0.145 mm2, a real random number generator with full functionality was incorporated into a 0.13 mm bulk complementary metal-oxide semiconductor.

2.4 A truly random number generator based on thermal noise

True random values are produced using a straightforward circuit that relies on the resistor's thermal noise[4], which is used to generate genuinely random numbers. This circuit is coupled to a computer system to produce pure random numbers and can be incorporated using conventional complementary metal-oxide-semiconductor technology.

2.5 True Random Number Generator Based on ROPUF Circuit

The circuit that is primarily made for ring oscillators is used to produce real random values[5]. This study set out to demonstrate that it is feasible to design a cryptosystem that may be put to a variety of uses. It can be used for salt generation, ephemeral session keys, asymmetric cryptography, computing asymmetric keys, etc.

III. PURE RANDOM NUMBER GENERATION: A METHODOLOGY

The methodology to generate pure random numbers consists of four important steps and these steps are followed in sequence to obtain the desired results. The steps are mentioned below.

3.1 Step1: Data Collection

Data acquisition from the mobile sensors initiates the process of data gathering. The data is then saved in csv format on the mobile device before being transferred to the server for analysis and manipulation. After processing, the data is transferred to the entropy pool, which serves as the source for generating random numbers. The flowchart that shows how data is transferred from portable sensors to the entropy pool is shown in figure 1.

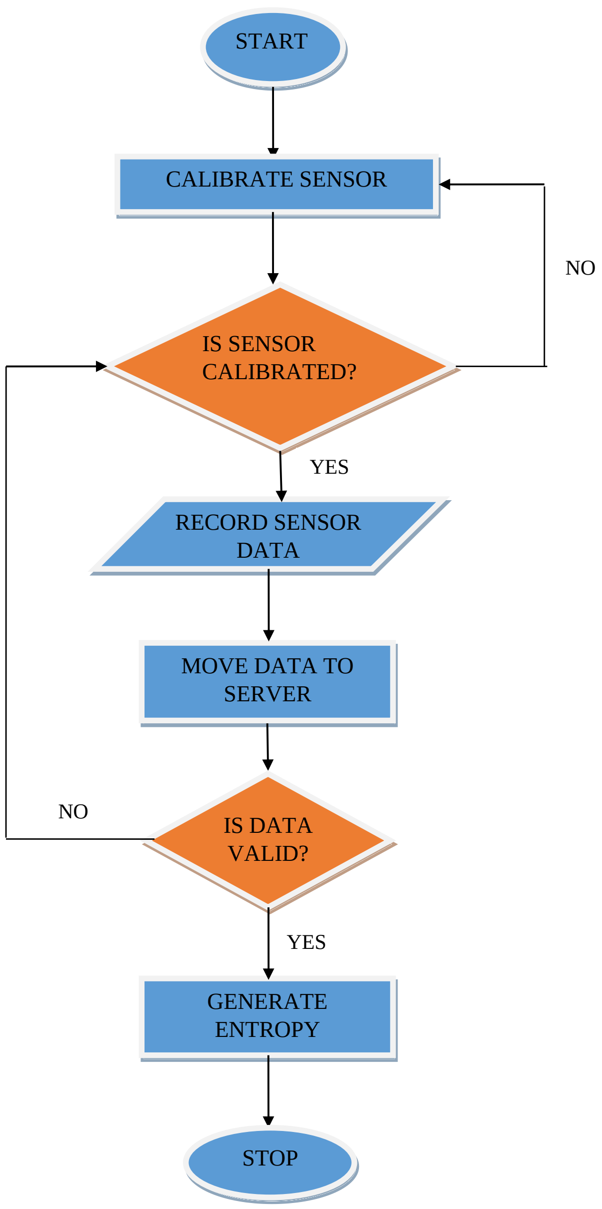

Figure 1 Data accumulation through sensors

3.2 Step2: Moving Database to Server

The A MySQL Server is then given this data set to process. The user's requests for processing and passing the data shall be followed after this data accumulation and transition to the server.

3.3 Step3: Extraction of meaningful information

A server side language utilised by this algorithm is PHP. The algorithm on the server separates the sensor's decimal value into three to seven digits, which is then used to generate a random number for that sensor at that moment.

3.4 Step4: Range Optimization

A ranging algorithm is used to process this random integer, forcing it to lie within a predetermined range while maintaining its unpredictability. Now that this randomly generated number has been placed on the graph, the outcomes are displayed in figure 2.

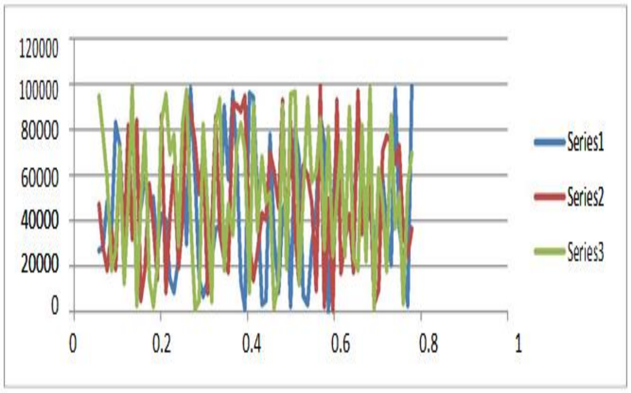

Figure 2: Randomness plot of collected data

IV. PERFORMANCE ANALYSIS

The performance of the proposed algorithm is analysed in terms of its complexity and time taken by the program during its execution.

4.1 Algorithmic Complexity

For n random numbers, every algorithm utilised in any module of the project is O(n). This demonstrates that the algorithms used to generate random numbers aren't CPU-intensive. Additionally, because the server is multithreaded, it can handle several requests at once. With this capability, the disadvantage of low data rate of existing Pure Random number generators is eliminated.

4.2 Resource Consumption at runtime

According to this stress testing, an entropy collection algorithm's processing requirements is depicted when mobile device used is under some load. As can be seen, 20% of RAM is needed to collect data for 10 seconds at a very high sampling rate of 0.002 seconds for each sample. A batch of 4000 random numbers is produced after 10 seconds of data collecting. This assortment of arbitrary numbers is displayed in figure 3.

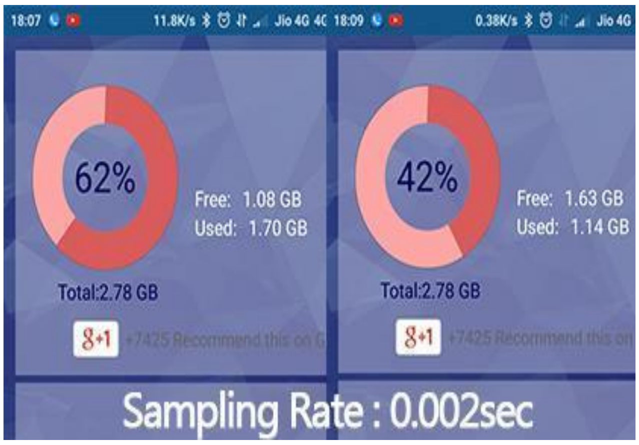

Figure 3. Stress test at sampling rate of 0.002 seconds

V. CONCLUSION

For any given finite range, the above-developed model generates pure random numbers with success. The physical phenomena, or random numbers, can be generated at any moment and is not entirely dependent on online connectivity. The model also demonstrates the capability of using these random numbers as desired. Due to the O(1) complexity of the technique utilised, the model also addresses the significant disadvantage of low bit rate without adding any additional processing overhead.

REFERENCES

[1] *Intel ® Digital Random Generator Generat or (DRNG)*, Revision 1. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.intel.com/content/dam/develop/external/us/en/documents/441-intel-r-drng-software-implementation-guide-final-aug7.pdf, 2012.

[2] "Quantum Random Number Generation (QRNG) - ID Quantique." [Online]. Available : https ://www.idquantique.com/random-number-generation/overview/. [Accessed: 08-Oct-2022].

[3] C. Tokunaga, D. Blaauw, and T. Mudge, "True Random Number Generator With a Metastability-Based Quality Control," *IEEE Journal of Solid-State Circuits*, **vol. 43, no. 1, pp. 78–85, 2008.**

[4] Z. Huang and H. Chen, "A truly random number generator based on thermal noise," *International Conference on ASIC, Proceedings*, **pp. 862–864, 2001.**

[5] S. Buchovecka, R. Lorencz, F. Kodytek, and J. Bucek, "True Random Number Generator Based on ROPUF Circuit," *Proceedings - 19th Euromicro Conference on Digital System Design, DSD 2016*, **pp. 519–523, 2016.**