BULGARIAN ACADEMY OF SCIENCES
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2023 • Vol. **80**, pp. 3-12
p-ISSN: 2738-7356; e-ISSN: 2738-7364
https://doi.org/10.7546/PECR.80.23.01

Entropy Test Degradation After Random Numbers Scaling

Petar Tomov, Gergana Mateeva, Dimitar Parvanov

Institute of Information and Communication Technologies
Bulgarian Academy of Sciences, Sofia, Bulgaria
petar.tomov@iict.bas.bg, gergana.mateeva@iict.bas.bg, dimitar.parvanov@iict.bas.bg

Abstract: The process of generating random numbers involves creating a series of numbers that possess randomness, devoid of any recognizable pattern or predictability. Typically, algorithms or physical processes, known as Random Number Generators (RNGs), are employed to produce these numbers. These generated random numbers are seldom used in their original form; instead, they undergo various scaling or mapping procedures. These transformations can potentially impact the generated random numbers' uniformity, distribution, and statistical attributes. If the scaling process is executed incorrectly or introduces bias, it can compromise the overall randomness and quality of the resulting numbers. Industries like gambling heavily rely on random numbers, and in certain regions such as Italy, the quality of the scaled numbers is required to match that of the raw random numbers. This stringent requirement raises a fundamental question: To what degree does the scaling of random numbers influence their statistical properties? Instances of poor-quality scaled numbers have been well-documented. To address this, the current study employs the Mersenne Twister pseudo-random number generator and a hardware true random number generator to produce raw random numbers. After this, a series of scaling operations are applied to these raw numbers, enabling a comparison of the statistical characteristics of the scaled results.

Keywords: entropy, randomness, scaling

1. Introduction

Digital transformation requires the transformation of various business processes and models, which is also related to the introduction of new information systems

with authorized access [1, 2]. That is why many authors involve proper schemes to protect the users [3, 4]. To provide such kind of authorized access for secure identification and encryption, random number generation techniques are often applied [5]. Random numbers find applications across diverse fields such as statistics, cryptography [6, 7], simulations and detection [8], gaming [9], information systems access [10], scientific inquiry [11], etc. Randomness is an important issue for the Internet of Things to generate suitable random numbers for IoT devices due to the limited resources and cryptographic protocols [12, 13]. The generation of random numbers involves two primary methodologies:

A. Pseudo-Random Number Generators (PRNGs)

PRNGs deploy deterministic algorithms to produce sequences of numbers with statistical attributes akin to genuine randomness. They necessitate an initial seed and employ mathematical operations to generate successive numbers. PRNGs can cycle through their sequences after a certain cycle, termed the period length. Prominent PRNG algorithms encompass the Mersenne Twister [14] and the Linear Congruential Generator [15].

B. True Random Number Generators (TRNGs)

TRNGs generate random numbers by exploiting inherently unpredictable physical processes. They draw upon external sources of entropy like atmospheric noise, radioactive decay, or electronic noise. TRNGs confer a heightened level of randomness compared to PRNGs but might necessitate specialized hardware or sensors for capturing the random sources.

Scaling of Random Numbers

Scaling raw random numbers involves metamorphosing the generated numbers from their original range into an alternative range or adapting their attributes in line with specific requisites. Some prevalent techniques for scaling encompass:

- Linear Scaling: This technique entails mapping the raw random numbers onto a desired range through linear transformations.
- Non-linear Scaling: Non-linear scaling entails applying mathematical functions or transformations to the raw random numbers for modifying their distribution or attributes. Familiar functions used for non-linear scaling include logarithmic, exponential, or trigonometric functions. These functions can be harnessed to create specific distributions, like the normal distribution (via the Box-Muller transform) or the exponential distribution (using the inverse transform method).

Scaling for Specific Objectives: On certain occasions, scaling may be executed to adhere to particular requirements or constraints. For instance, in cryptography, random numbers might necessitate scaling to align with the key size or to fit within a specific modular arithmetic range.

The selection of a scaling technique hinges on the desired attributes and scope of the scaled numbers. It is of paramount importance to ensure that the scaling process does not instill bias, correlation, or other anomalies that could jeopardize the randomness or statistical traits of the random numbers. Furthermore, rigorous testing and analysis should be performed to validate the quality and randomness of the scaled numbers.

Overview of Research

This study undertook a sequence of number scaling operations on sequences generated by both PRNGs and TRNGs. The raw random numbers underwent assessment through the ENT [16] statistical test, available in the Linux operating system. Subsequently, a comparative analysis was executed to discern the influence of scaling on the Entropy parameter gauged by the ENT tool.

The subsequent sections of the paper are structured as follows: Section 2 delineates the statistical tools utilized, Section 3 presents the experiments and resultant outcomes, and the concluding section wraps up the paper while proffering recommendations for future research endeavors.

2. Statistical Tools

In this research, the tools employed for statistical analysis include the Mersenne Twister pseudo-random number generator, the TrueRNG v3 hardware random number generator, and the ENT tool for evaluating random number sequences.

2.1. Mersenne Twister

The Mersenne Twister stands as a widely utilized pseudo-random number generator algorithm. Its conception dates back to 1997, credited to Makoto Matsumoto and Takuji Nishimura, and it boasts an extended period and commendable statistical attributes. The name "Mersenne" is derived from its foundation on Mersenne primes, which are prime numbers representable as $(2^p – 1)$.

Operating as a linear feedback shift register (LFSR) generator, the Mersenne Twister engages a shift register and bitwise operations to yield random numbers. It showcases a substantial period, specifically $(2^{19937} – 1)$, signifying its capability to generate a sequence of $(2^{19937} – 1)$ distinct numbers prior to repetition.

Central to the Mersenne Twister is the notion of employing a considerable internal state comprising 624 32-bit integers to yield the pseudo-random sequence. These integers undergo deterministic updates using a recurrent relationship, yielding the appearance of randomness. The transitions of the state rely on modular arithmetic, bitwise operations, and non-linear transformations.

To generate a pseudo-random number using the Mersenne Twister, a subset of the internal state is harnessed to compute a value, subsequently adjusted to fit within the desired range. This transformation usually involves shifting, masking, and scaling operations to ensure alignment with the desired output range.

An advantage of the Mersenne Twister lies in its protracted period, resulting in a considerable time span before the sequence repeats. This renders it suitable for applications necessitating numerous independent random values, such as simulations, statistical sampling, and cryptography.

It is worth noting, however, that the Mersenne Twister is a deterministic algorithm, implying that given the same internal state, it unfailingly generates the same sequence of pseudo-random numbers. Depending on the context, this attribute can be both beneficial and disadvantageous. For cryptographic applications where unpredictability is critical, alternative algorithms like cryptographic-strong PRNGs are generally recommended.

2.2. TrueRNG v3

The TrueRNG v3, introduced in 2013, is a hardware-based true random number generator conceived by ubld.it [17]. Diverging from pseudo-random number generators reliant on deterministic algorithms, TRNGs derive random numbers from inherently unpredictable physical processes, such as electronic noise or radioactive decay.

TrueRNG v3 adopts a hardware architecture that captures atmospheric noise via an avalanche diode [18]. This diode engenders random fluctuations in voltage due to the uncertain nature of quantum processes. These voltage fluctuations undergo amplification and digitization to yield a stream of random bits.

Functioning as a peripheral device, TrueRNG v3 interfaces with computers via a USB connection. It generates real-time random numbers as long as it remains powered and linked to a compatible computer. Access to the generated random data is facilitated through standard interfaces like USB.

TrueRNG v3 encompasses various features to ensure the quality and security of the random numbers it produces:

- *Entropy*: True randomness emerges from the capture of entropy sourced from physical origins. The harnessed entropy seeds a random number generator within the device, yielding high-quality random numbers.

- *Noise Source Conditioning*: Employing conditioning techniques, the captured noise undergoes processing to eliminate biases or patterns. This enhances the quality of the generated random numbers.
- *Bit Rate*: TrueRNG v3 boasts a high bit rate, typically generating random data at rates of several megabits per second. This is suitable for applications necessitating a substantial volume of random numbers.
- *Compliance*: The device conforms to various standards and protocols, ensuring compatibility with diverse operating systems and applications.
- *Open Source*: The open-source design of TrueRNG v3 permits users to scrutinize and validate both the hardware and firmware for transparency and security.

TrueRNG v3 is widely employed in applications demanding authentic randomness, including cryptography, scientific simulations, and gaming. It presents a dependable source of random numbers capable of enhancing the security and performance of diverse systems and applications.

2.3. Scaling of Random Numbers

The ENT tool, a software package tailored for assessing the entropy of files and byte sequences, finds common usage in evaluating data's randomness and informational content, encompassing random number sequences.

At its core, the ENT package features a command-line tool named ENT, which calculates various statistical metrics pertaining to randomness and entropy. This tool subjects input data to a battery of tests, analyzing the distribution of byte values and other characteristics. Outputs include entropy measurements, results from chi-square tests, mean values, and assorted statistical parameters.

Primarily oriented toward evaluating the statistical attributes and entropy of data, the ENT tool proves valuable in gauging the quality of random number generators, encryption algorithms, compression methodologies, and other applications reliant on randomness and information density. The ENT tool focuses on five statistical parameters:

- *Entropy*: Measuring the unpredictability of a byte sequence, entropy quantifies the information contained within the data. In the realm of random number generators, higher entropy signifies a sequence of numbers exhibiting greater randomness and diminished predictability.
- *Chi-square Test*: Employed to discern significant deviations between anticipated and observed data distributions, the chi-square test determines whether byte values within input data follow a uniform distribution or display biases and patterns.

- *Arithmetic Mean*: This computes the average value of a number set. Within the context of the ENT tool, it calculates the mean of byte values in input data, offering insights into the dataset's center or typical value.
- *Monte Carlo Value for Pi*: Employing the Monte Carlo method, the ENT tool estimates the value of Pi (π) by generating random points within a unit square and assessing their distribution within a unit circle. This estimation indicates the randomness of the input data.
- *Serial Correlation Coefficient*: Reflecting the correlation between consecutive sequence values, the serial correlation coefficient measures the intensity and direction of this correlation. In the context of the ENT tool, it computes the serial correlation coefficient for byte values in input data. Values near zero denote minimal correlation, while values approaching (+1) or (– 1) signify strong positive or negative correlations, respectively.

These parameters furnish insights into randomness, distribution, and statistical attributes of input data. By assessing these metrics, the ENT tool aids in evaluating the quality and randomness of random number generators, encryption algorithms, and other applications where randomness and information density hold significance.

3. Experiments & Results

The empirical data, each consisting of 1MB, originates from two distinct sources. The initial dataset is derived from the Mersenne Twister implementation within the NumPy library, while the subsequent dataset is procured from the TrueRNG v3 hardware generator. The primary objective of this study is to discern the impact of scaling on raw random numbers with regard to entropy measurement for both pseudo and authentic random numbers. This rationale underscores the investigation of two separate raw data collections.

The raw data undergo a scaling process that transforms their range from $(0 - 1)$ to a $(0 - 32)$ span. This particular range holds widespread usage within the domain of the gambling industry. Post the execution of the scaling operation, the ENT tool is invoked, thereby yielding measured parameters that are subsequently recorded.

Fig. 1 and Fig. 2 illustrate the variations in entropy as influenced by the scaling range. In both test sets, it is evident that entropy measurements are notably more impacted by scaling in smaller ranges compared to larger ones. The outcomes distinctly indicate that employing conventional tools such as ENT yields reduced efficacy in assessing randomness when random number scaling is implemented.

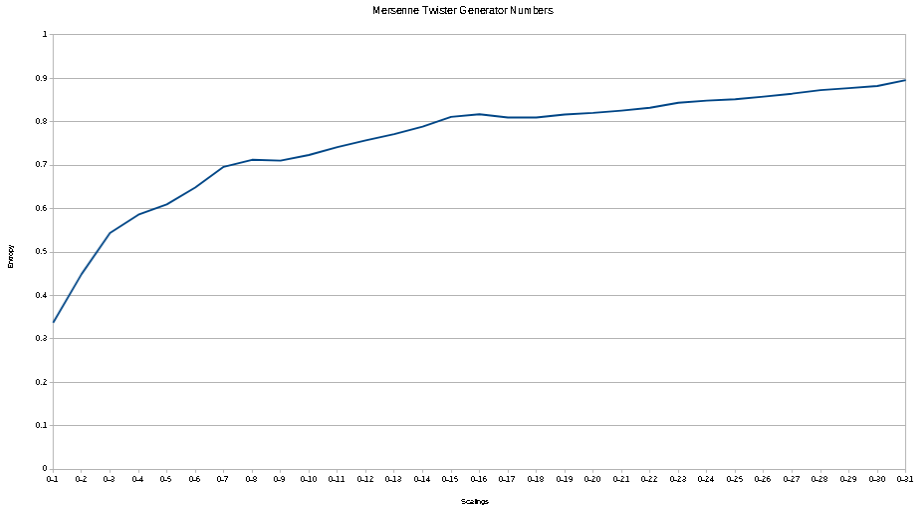

Fig. 1. Mersenne Twister entropy

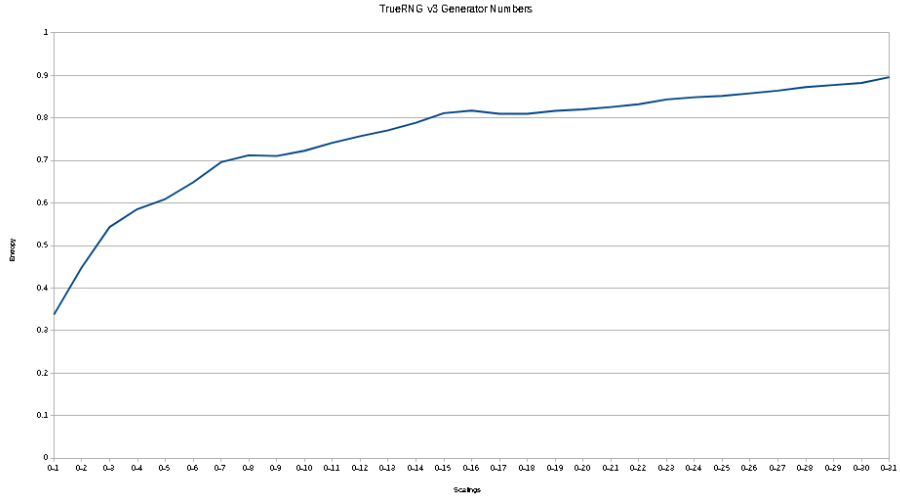

Fig. 2. TrueRNG v3 entropy

Fig. 3 depicts the disparity between the two entropy curves. This discrepancy is relatively minor and remains within the scope of statistical fluctuations.

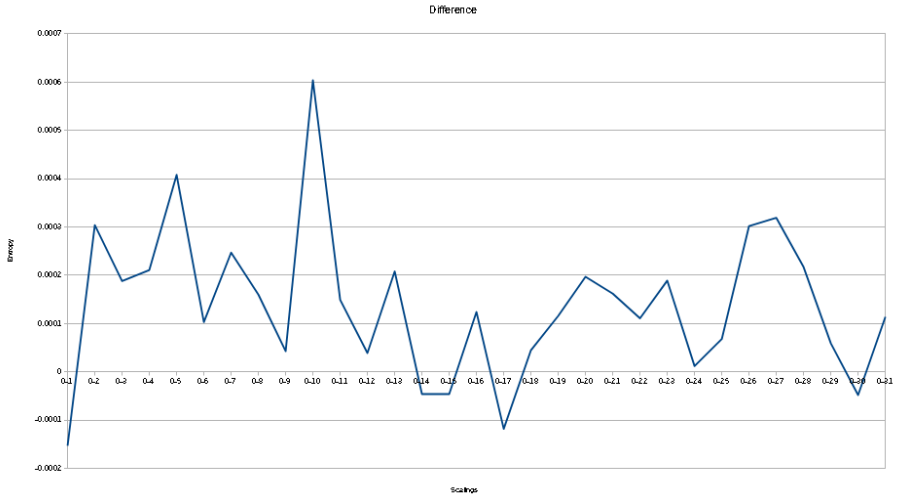

Fig. 3. Entropy difference

4. Conclusions

This study tackles the challenge of scaling raw random numbers, particularly in the context of gambling regulations that emphasize the scaling and mapping of these numbers. However, the empirical findings do not substantiate these legal apprehensions.

The constraints of limited research time and resources have hindered a more extensive exploration of various scaling ranges. As a future avenue of investigation, it would be beneficial to significantly expand the scope of scaling ranges. Additionally, relying solely on 1MB of raw data imposes limitations on the representativeness of the samples. In forthcoming experiments, the exploration of larger binary files could be pursued.

The analysis conducted using the ENT tool is circumscribed by its statistical capabilities. Future analyses could consider employing more robust tools, such as Dieharder [19], to enhance the depth of statistical assessment. While this paper delves into the intricacies of the scaling problem, the related mapping challenge remains unaddressed.

Acknowledgments

This research is supported by the Ministry of Education and Science of the Republic of Bulgaria under the National Science Program INTELLIGENT ANIMAL HUSBANDRY, grant agreement No. D01-62/18.03.2021.

References

1. Yoshinov, R., Iliev, O.: Sharing local resources within a community by enhancing the potential of Eduroam and EduVPN with mobile application for remote and local resources and through secure user identification over the network (MARLIN). Problems of Engineering Cybernetics and Robotics 79, 3–36 (2023), https://doi.org/10.7546/PECR.79.23.01.
2. Borissova, D., Dimitrova, Z., Dimitrov, V., Yoshinov, R., Naidenov, N.: Digital transformation and the role of the CIO in decision making: A comparison of two modelling approaches. In: Saeed, K., Dvorský, J. (eds) Computer Information Systems and Industrial Management. Lecture Notes in Computer Science, vol. 13293, 93–106, (2022), https://doi.org/10.1007/978-3-031-10539-5_7.
3. Gaidarski, I.: Model Driven Development of Information Security System. Problems of Engineering Cybernetics and Robotics 76, 47–62, (2021), https://doi.org/10.7546/PECR.76.21.04
4. Dimitrova, Z., Borissova, D., Dimitrov, V.: Design of web application with dynamic generation of forms for group decision-making. In: Saeed, K., Dvorský, J. (eds) Computer Information Systems and Industrial Management. CISIM 2021. Lecture Notes in Computer Science, vol. 12883, pp. 112–123 (2021), https://doi.org/10.1007/978-3-030-84340-3_9.
5. Boneva, A., Boneva, Y.: An approach for encrypted exchange of information in corporate networks based on Tcl/Tk. Problems of Engineering Cybernetics and Robotics 78, 5–22, (2022), https://doi.org/10.7546/PECR.78.22.02.
6. Blagoev, I.: Method for evaluating the vulnerability of random number generators for cryptographic protection in information systems. In: Dimov, I., Fidanova, S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence, vol. 902, pp. 391–397, (2021), https://doi.org/10.1007/978-3-030-55347-0_33.
7. Blagoev, I.: Application of time series techniques for random number generator analysis. In: Proceedings of XXII International Conference DCCN 2019, September 23-27, Moscow, Russia, pp. 437–446, (2019).
8. Borissova, D.: Methods for NVG visual acuity determination. Cybernetics and Information Technologies 3(2), 25–33 (2003).
9. Staneva, A., Ivanova, T., Rasheva-Yordanova, K., Borissova, D.: Gamification in education: Building an escape room using VR technologies. In: 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, pp. 678-683, (2023), https://doi.org/10.23919/MIPRO57284.2023.10159923.
10. Rajamäki, J., Lahdenperä, J., Shalamanov, V.: Design science research towards ECHO governance and management information system. In: 12 International Conference on Dependable Systems, Services and Technologies (DESSERT), Athens, Greece, pp. 1–7, (2022), https://doi.org/10.1109/DESSERT58054.2022.10018802
11. Borissova, D., Barzev, I., Yoshinov, R., Kotseva, M.: Group decision-making models for selection of virtual machine software for malware detection purposes.

In: Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, pp. 1–5, (2023), https://doi.org/10.1109/MECO58584.2023.10155084.

12. Borissova, D., Danev, V., Garvanova, M., Garvanov, I., Yoshinov, R.: Key indicators to measure student performance in IoT and their teamwork ability. In: Auer, M.E., Tsiatsos, T. (eds) New Realities, Mobile Systems and Applications. IMCL 2021. Lecture Notes in Networks and Systems, vol. 411, pp. 711–720, (2022), https://doi.org/10.1007/978-3-030-96296-8_64.
13. Garvanov, I., Garvanova, M., Borissova, D., Garvanova, G.: A model of a multi-sensor system for detection and tracking of vehicles and drones. In: Shishkov, B. (eds) Business Modeling and Software Design. BMSD 2023. Lecture Notes in Business Information Processing, vol. 483, pp. 299–307, (2023), https://doi.org/10.1007/978-3-031-36757-1_21.
14. Matsumoto, M.; Nishimura, T.: Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation, 8(1), 3–30, (1998).
15. L'Ecuyer, P.: Tables of linear congruential generators of different sizes and good lattice structure. Mathematics of Computations, 68(225), 249–260, (1999).
16. Walker, J.: ENT – A Pseudorandom Number Sequence Test Program. 2008 https://www.fourmilab.ch/random/, last visited 20 August 2023.
17. TrueRNG v3 – Hardware Random Number Generator. 2013. https://ubld.it/truerng_v3, last visited 20 Aug 2023.
18. Ewert, M.: A random number generator based on electronic noise and the xorshift algorithm. In VII International Conference on Network, Communication and Computing, pp. 357–362, (2018).
19. Brown, R.: Dieharder: A Random Number Test Suite. 2023, https://webhome.phy.duke.edu/ ~rgb/General/dieharder.php, last visited 20 Aug 2023.