Performance evaluation of chaotic random numbers generated from responses of integer logistic maps

Shiki Kanamaru\textsuperscript{1a)}, Yutaka Shimada\textsuperscript{2}, Kantaro Fujiwara\textsuperscript{3}, and Tohru Ikeguchi\textsuperscript{1,4}

\textsuperscript{1} Department of Management Science, Graduate School of Engineering, Tokyo University of Science
6-3-1 Niijuku, Katsushika, Tokyo 125-8585, Japan

\textsuperscript{2} Department of Information and Computer Sciences, Graduate School of Science and Engineering, Saitama University
255 Shimo-Okubo, Sakura-ku, Saitama City, Saitama 338-8570, Japan

\textsuperscript{3} International Research Center for Neurointelligence, The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo 113-0033, Japan

\textsuperscript{4} Department of Information and Computer Technology, Faculty of Engineering, Tokyo University of Science
6-3-1 Niijuku, Katsushika, Tokyo 125-8585, Japan

\textsuperscript{a)} shikimaru@hisenkei.net

Received January 7, 2021; Revised March 19, 2021; Published July 1, 2021

Abstract: One of the engineering applications of chaotic nonlinear dynamical systems is a pseudorandom number generator. Pseudorandom numbers generated from chaotic dynamical systems are called chaotic random numbers. A logistic map which exhibits a chaotic response can be used as such a chaotic random number. However, an important issue exists when we use such dynamical systems as a pseudorandom number generator by a digital computer: when the chaotic response of a logistic map is reproduced numerically, the number of iterations that the chaotic response is sustained depends on the precision of the numerical calculation, because the precision of the numerical calculation affects the size of the numerical error. In this paper, we extended the logistic map to an integer logistic map to reduce such numerical errors. We investigated the performance of chaotic random numbers obtained from the integer logistic map with varying numerical precisions and transforming them into binary random numbers. We then used NIST SP 800-22 to evaluate the performance of the random numbers. The results show that a numerical precision of 20 orders of magnitude or more is desirable to the generation of a well-performing chaotic random numbers from the response of the integer logistic map.

Key Words: Chaotic random number, numerical calculation precision, and integer logistic map
1. Introduction

A random number sequence is an irregular and unpredictable sequence of numbers, each element of which is called a random number. One property expected of random numbers in any situation is unpredictability. Then, random numbers are employed in a wide range of applications, for example, setting initial values for cryptographic communications and numerical simulations.

On the other hand, chaos is a phenomenon that can be observed from deterministic nonlinear dynamical systems. One of the properties of deterministic chaos is sensitive dependence on initial conditions, wherein a small error in the initial conditions would expand over time. Then, it is difficult to predict responses produced from deterministic chaos in the long term due to such sensitive dependence on the initial conditions.

The long-term unpredictability of deterministic chaos is therefore potentially beneficial to the unpredictability required for random numbers, which has led to a number of research efforts on random number generation using deterministic chaos [1–12]. The random numbers generated using chaos, as in these references, are called chaotic random numbers. There are two main types of random number generation methods. One is physical random number generation, which is based on physical phenomena. For example, Uchida et al. proposed a high-speed physical random number generation method using laser chaos [13] that can greatly improve the random number generation speed, which is otherwise a drawback of physical random number generation methods. The other is pseudorandom number generation, which uses numerical calculations, such as the Mersenne Twister [14], which is a well-known pseudorandom number generation method.

In this paper, we discuss a chaotic random number generator that is classified as a pseudorandom number generator. The random numbers involved in this study are binary random numbers, i.e., 0 and 1. We focused on the generation of chaotic random numbers using a logistic map [15], which can be used to observe chaotic phenomena using a very simple mathematical formula. Although several studies [7, 16] have already been conducted on the generation of chaotic random numbers using the logistic map, they require post-processing methods, such as exclusive summation of multiple sequences, to improve the performance of the generated chaotic random numbers. One of the reasons why post-processing is necessary is that the numerical precision is insufficient for common double-precision floating-point operations. In a numerical simulation, it can be difficult to accurately reproduce the chaotic response due to the influence of numerical errors. In addition, the response converges to a periodic response or a fixed point because of the numerical calculation error. Therefore, we aim to reduce this error by increasing the precision of the numerical calculation. Namely, we generate chaotic random numbers based on the response of an integer logistic map [4, 17], which converts the decimal part to an integer part.

In [4], Ishida et al. evaluated the performance of chaotic random numbers generated from the responses of integer logistic maps from the viewpoint of the power spectrum [4]. However, this evaluation from the power spectrum perspective may not be sufficient from a statistical perspective. This is because the evaluation criterion is decided as follows: if the power spectrum of the generated chaotic random number is less than or equal to 0.6 for all the frequencies obtained, the number is considered as white noise and can be used as a pseudorandom number. In [6], Araki et al. proposed pseudorandom number generators using the logistic map and the integer logistic map with renewing the initial value. The renewal process of the initial value is introduced to avoid the convergence of the response to a periodic orbit caused by a finite numerical precision. In [11], Wang et al. evaluated the performance of a pseudorandom number generator that updates the bifurcation parameters of the logistic map. This process improved the confidentiality of the cipher by updating the bifurcation parameters.

In this study, we evaluated the performance of a pseudorandom number generator in a simple, natural and straightforward way: we only give an initial value and set numerical precision without any processing, such as the renewal of the initial condition or changing the bifurcation parameter. Then, we evaluated the performance of random numbers using the NIST [18] Special Publication 800-22 provided by the U.S. National Institute of Standards and Technology. The NIST test is an international standard statistical test method that is widely used and suitable for evaluating the
performance of fair random numbers.

In addition, we investigated the case that the bifurcation parameter $a$ of the logistic map is negative, because it has already been shown that the chaotic response can be obtained even when the parameter $a$ is negative [19]. Further, we investigated the performance of the pseudorandom number generator in relation to the values of the Lyapunov exponent by evaluating the number of passed NIST test items.

Therefore, this study has two objectives. The first is to investigate the performance of chaotic random numbers generated from an integer logistic map when the numerical calculation precision is changed. The second is to clarify which factors, other than numerical calculation precision, affect the random number performance.

2. Random number generators

The logistic map is a simple one-dimensional discrete time dynamical system [15]. The equation of the logistic map is described as:

$$ x_{t+1} = f(x_t) = ax_t(1-x_t), $$

where $a$ is a bifurcation parameter and $x_t$ is the state value at time $t$.

Figure 1 shows the bifurcation diagram of $x_t$ when the value of $a$ is changed in the range $-2.0 \leq a \leq 4.0$ [19]. The horizontal axis is for $a$, the left vertical axis is for the state value $x_t$ (purple dots), and the right vertical axis is for the Lyapunov exponent $\lambda$ (green crosses) which is defined to be

$$ \lambda = \frac{1}{T} \sum_{t=1}^{T} |f'(x_t)|, $$

where $T$ is the length of the time series. Figure 1 shows that $\lambda$ can be greater than 0 regardless of the positive or negative value of $a$. The symmetric property in the Lyapunov exponent of the logistic map is discussed in [19], which showed that the Lyapunov exponents are symmetric about $a = 1.0$. Thus, $\lambda$ for $a = 4.0$ and $\lambda$ for $a = -2.0$ are equal and have the largest values ($\lambda = \ln 2$).

Figure 2 shows the invariant measures when $a = 4.0$ and $-2.0$. The horizontal axis represents the state value $x_t$, whereas the vertical axis represents the appearance probability $P(x_t)$. Figure 2 shows that $P(x_t)$ is symmetric about $x_t = 0.5$, whereas Fig. 3 shows the frequency proportion for $x_t < 0.5$ and $x_t > 0.5$, using the data from Fig. 2. In Fig. 3(a), the proportion of $x_t < 0.5$ is 0.499160, whereas the proportion of $x_t > 0.5$ is 0.500840. Meanwhile, in Fig. 3(b), the proportion of $x_t < 0.5$ is 0.499583, whereas the proportion of $x_t > 0.5$ is 0.500417. These values are expected to make the frequency distributions of 0 and 1 uniform when the random numbers are binarized.
Fig. 2. Invariant measures of the logistic map for bifurcation parameters (a) $a = 4.0$ and (b) $a = -2.0$. In these cases, responses are chaotic, and the Lyapunov exponents are $\lambda = \ln 2$.

Fig. 3. Frequency proportion of binary values of the logistic map for (a) $a = 4.0$ and (b) $a = -2.0$.

In terms of the density function, we show that the frequencies of 0 and 1 are almost equal based on a binarization of the response of the logistic map. The invariant measure of the response for $a = 4.0$ in the logistic map is calculated using

$$P(x_t) = \frac{1}{\pi \sqrt{x_t(1-x_t)}}. \tag{3}$$

The product of $x_t$ and $(1 - x_t)$ appears in the square root of the denominator of Eq. (3), indicating that the probability density $P(x_t)$ of the logistic map, which is an interval dynamical system of $[0, 1]$, has symmetry with $x_t = 0.5$ as the center. Thus, if we binarize the response of the logistic map with $x_t = 0.5$ as the threshold, we can obtain a random number with an equal probability of occurrence.

On the other hand, because the numerical calculation precision is finite, the chaotic response of the
logistic map cannot be reproduced in the numerical simulation. Therefore, we extended the logistic map to an integer logistic map in decimal notation using [4]. The integer logistic map can be derived from Eq. (1). The procedure for extending the logistic map to the integer logistic map is as follows. First, multiplying the left term and the right term of Eq. (1) by $10^N$, we obtain

$$10^N \times x_{t+1} = 10^N \times ax_t(1 - x_t).$$

(4)

With $X_t = 10^N \times x_t$, Eq. (4) is rewritten to

$$X_{t+1} = aX_t(10^N - X_t) \times 10^{-N}.$$  

(5)

We use the floor function $\lfloor \rfloor$ to extract the integer part as follows:

$$X_{t+1} = \lfloor aX_t(10^N - X_t)10^{-N} \rfloor.$$  

(6)

In Eq. (6), $X_t$ is the state value of the integer logistic map at time $t$, and $N$ represents the numerical precision.

We then explain how to make pseudorandom numbers from the integer logistic map. In this study, the parameter $a$ is set to 4.0 and -2.0, at which the Lyapunov exponent becomes $\ln 2$ and chaotic responses can be observed. We generate a time series from the integer logistic map (Eq. (6)) and convert it into binary random numbers using Eq. (7):

$$\tilde{X}_t = \begin{cases} 0 & (X_t \leq \theta), \\ 1 & (X_t > \theta), \\ \end{cases}$$

(7)

where $\theta$ is a threshold. When $a = 4.0$, the range of the state values of the logistic map (Eq. (1)) is $0 \leq x_t \leq 1$. Thus, in the case of an integer logistic map (Eq. (6)), the range is $0 \leq X_t \leq 10^N$. If $a = -2.0$, the range of the integer logistic map is $-0.5 \times 10^N \leq X_t \leq 1.5 \times 10^N$. Thus, we set $\theta$ to 0.5 $\times 10^N$, which is the median of the distributions of state values of the integer logistic maps in case that $a = 4$ or $a = -2$.

Figure 4 shows the results of generating chaotic random numbers from the response of the integer logistic map (Eq. (7)). In Fig. 4, the size of each image is $m \times m$ pixels, where $m = 800$. The $(i, j)$th pixel in Fig. 4(a)–(f) $(i, j = 0, 1, \cdots, m - 1)$ corresponds to $X_{mi+j}$, where the $(i, j)$th pixel is colored white if $X_{mi+j} \leq \theta$, and black otherwise. In Fig. 4, the upper row shows the generated random numbers when $a = 4.0$, and the lower row shows them when $a = -2.0$. The numerical calculation precision is set to $N = 7, 15, \text{and } 20$ from left to right.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{random_numbers.png}
\caption{Examples of binary random number sequences displayed on two-dimensional plane.}
\end{figure}
When \( N = 7 \), the generated random numbers exhibit a regular pattern, as shown in Fig. 4(a) and (d). On the other hand, when the precision of the numerical calculation is increased, the generated random numbers seem to be more random. However, to verify their randomness, the obtained random numbers have to be evaluated by a random number test method.

3. NIST SP 800-22

In this section, we describe the NIST test [18], which statistically evaluates whether a sequence of numbers is really random or not. The NIST test consists of 15 test items, which are listed with their respective test numbers, names, and descriptions in Table I. The NIST test in Table I is provided by the National Institute of Standards and Technology (NIST). Each test item in Table I tests properties required in random numbers.

| No. | Items                                                     | Descriptions                                                                 |
|-----|-----------------------------------------------------------|-----------------------------------------------------------------------------|
| 1   | Frequency Test                                           | Test whether the occurrence probabilities of 0 and 1 are close to \( \frac{1}{2} \). |
| 2   | Frequency Test within a Block                            | For a given block length, test whether the occurrence probabilities of 0 and 1 are close to \( \frac{1}{2} \). |
| 3   | Cumulative Sums Test                                     | Convert the binary random numbers to \((+1, -1)\) and test for the maximum deviation from 0 of the random walk defined by the cumulative sum. |
| 4   | Runs Test                                                 | Test for the number of consecutive 1s (runs).                               |
| 5   | Test for Longest Run of Ones in a Block                  | For a given block length, test for the longest runs.                        |
| 6   | Binary Matrix Rank Test                                  | Test for the ranks of sparse matrices across several columns.              |
| 7   | Discrete Fourier Transform Test                          | To investigate the periodicity, test for the height of the peak due to discrete Fourier transformation of the sequence. |
| 8   | Non-overlapping Template Matching Test                   | Test for the numbers of occurrences of bit string prepared in advance, without overlap. |
| 9   | Overlapping Template Matching Test                       | Test for the numbers of occurrences of bit string prepared in advance, with overlap. |
| 10  | Maurer’s “Universal Statistical” Test                    | Detect and test matching patterns within a sequence, to see if they can be compressed. |
| 11  | Approximate Entropy Test                                 | Test for the frequencies of occurrence of patterns of a given bit length.  |
| 12  | Random Excursions Test                                   | Test for the total number of times \( K \) is visited by cumulative sum random walks. |
| 13  | Random Excursions Variant Test                           | Test for the sum of the number of visits to a specific state with cumulative sum random walks. |
| 14  | Serial Test                                               | Test for the occurrence frequencies of all overlapping bit patterns of a certain bit length. |
| 15  | Linear Complexity Test                                   | Test for the length of the linear feedback register to generate a random number. |

A random number sequence with a length of at least 1 Gbits is necessary to conduct the NIST test. If the random number sequence has a length of 1 Gbits, it is then divided into shorter segments with lengths of 1,000,000 (= 1 Mbits) each. Each of these 1,000 segments are then subjected to the 15 NIST test items. As a result, we obtain 1,000 \( P \)-values from each test.

The following two sub-tests are then conducted. In the first sub-test, the \( P \)-value, which indicates the uniformity of distribution of the 1,000 \( P \)-values, is obtained. Meanwhile, in the second sub-test, for each of the 1,000 \( P \)-values, we evaluate the probability that the \( P \)-value is greater than \( \alpha (= 0.01) \), where the significance level \( \alpha \) is the probability that the random number sequence is not random in the sense of each NIST item. A random number sequence is considered to have passed a certain test item if and only if it passes the first and second sub-tests for that item.

On the other hand, the parameter rules of the NIST test are important for obtaining an appropriate verification result according to the length of the random number sequence. Table II outlines the parameters used in this study. The values of the parameters in Table II were determined based on the method provided by NIST for setting the parameters [18].
Table II. Parameters used in NIST test [18].

| Parameters                                           | Values |
|------------------------------------------------------|--------|
| Significance level $\alpha$                         | 0.01   |
| Number of segments $m$                               | 1,000  |
| Frequency test within a block: Length of each block  | 16,384 |
| Non-overlapping Template Matching Test: Length in bits of each template | 9      |
| Overlapping Template Matching Test: Length in bits of the template | 9      |
| Approximate Entropy Test: Length of each block       | 10     |
| Serial Test: Length in bits of each block            | 16     |
| Linear Complexity Test: Length in bits of a block    | 500    |

4. Experiments

In this study, we used the number of passed NIST test items, out of a total of 15 items, as an index of the performance evaluation of the generated random numbers. In addition, the binary random number sequences evaluated in this study were generated using $2^{30}$ ($= 1,073,741,824$) responses obtained from an integer logistic map after the first 100,000 responses were omitted as transient states.

Figure 5 shows the relationship between the number of passed NIST test items and the positive Lyapunov exponent $\lambda$ of the logistic map. The horizontal axis is for the Lyapunov exponent $\lambda$, whereas the vertical axis represents the number of passed NIST test items. For the result shown, we had set the initial value $X_0 = 0.0002 \times 10^N$ and the numerical calculation precision $N = 20$. Then, we set the threshold $\theta$ to $\bar{x} \times 10^N$, where $\bar{x}$ is the average value calculated by

$$\bar{x} = \frac{1}{2^{30}} \sum_{t=1}^{2^{30}} x_t. \tag{8}$$

Figure 5 shows that the larger the value of $\lambda$, the larger the number of NIST test items that are passed, indicating that the orbital instability of the random number generator affects the performance of the generated random numbers. However, even when $\lambda$ is positive, the number of passed NIST test items is particularly large when $a = 4.0$ and $a = -2.0$, at which $\lambda$ is maximized ($\lambda = \ln 2$). From these results, the performance of chaotic random numbers generated from the integer logistic map is expected to improve only when $a$ is 4.0 or -2.0.

Meanwhile, Fig. 6 shows the results of examining whether the number of passed NIST test items changes between $a = 4.0$ and $a = -2.0$. The initial value was $X_0 = 0.0002 \times 10^N$, and the numerical

![The number of passed NIST test items vs. Lyapunov exponents](image_url)

**Fig. 5.** Relationship between positive Lyapunov exponents $\lambda$ and the number of passed NIST test items. The square marks denote the number of passed NIST test items, and the cross marks denote the average value $\bar{x}$. 

495
The number of passed NIST test items for generated chaotic random numbers with respect to numerical calculation precision. Two bifurcation parameters were used: \( a = 4.0 \) (circles) and \( a = -2.0 \) (triangles).

The horizontal axis represents \( N \), whereas the vertical axis represents the number of NIST test items that were passed. The results for \( a = 4.0 \) are shown as circles, whereas the results for \( a = -2.0 \) are shown as triangles. According to Fig. 6, the number of passed NIST test items is likely to be large when the numerical calculation precision is in the range \( N \geq 20 \) for both \( a = 4.0 \) and \( a = -2.0 \). To clarify the change in quality of generated random numbers with respect to the value of \( a \), we further investigated the ratio of the number of passed NIST test items to the total number of items. The horizontal axis corresponds to the NIST test names listed in Table I, whereas the vertical axis represents the percentage of 150 random numbers with different \( N \) that passed these NIST test items. From Fig. 7, it can be seen that, for each test item, there is no difference in the distributions of the percentage due to the difference between \( a = 4.0 \) and \( a = -2.0 \). In addition, it is clear that both parameters occasionally fail in the Non-Overlapping Template Matching test, probably because of the nature of the logistic map used as the source of the random numbers.

On the other hand, Fig. 8 shows the results of generating chaotic random numbers as the precision of the numerical calculation is further increased. The horizontal axis represents the numerical calculation precision \( N \), whereas the vertical axis represents the number of passed NIST test items. Because there was no difference in the performance of the random numbers generated when \( a = 4.0 \) and \( a = -2.0 \), we performed numerical experiments focusing on \( a = 4.0 \). Figure 8 shows that the higher the numerical precision \( N \), the higher the performance of the generated chaotic random numbers.

However, when \( N = 455 \), the number of passed NIST test items dropped to 12. In addition, there are many cases wherein one item was rejected. These results imply that there are other factors than numerical calculation precision that may improve the number of passed NIST test items. If the performance of the chaotic random number is degraded even if the numerical precision is large, the initial value may be the cause.

Finally, we report the results of the investigation based on differences in the initial value \( X_0 \). The numerical calculation precision \( N \) was investigated in the range \( 6 \leq N \leq 30 \). We prepared 98 initial values \((X_0)\) for each value of \( N \), where we increased the value of \( X_0 \) in increments of 0.005 from 0.005 \( \times 10^N \) to 0.495 \( \times 10^N \). Because the response of the integer logistic map converges to a fixed point when \( X_0 = 0.25 \times 10^N \), we excluded \( X_0 = 0.25 \times 10^N \) in this experiment. We investigated the performance of chaotic random numbers generated based on these values of \( N \) and \( X_0 \). Figure 9 shows the results. The horizontal axis represents the numerical calculation precision \( N \), whereas the vertical axis represents the average number of passed NIST test items based on the initial values for the 98 patterns. The error bars in Fig. 9 show the maximum and minimum numbers of NIST test items that were passed, whereas the histograms show the standard deviation.
Fig. 7. Proportion of generated random numbers that passed each NIST test item for 150 values of $N$, with (a) $\alpha = 4.0$ and (b) $\alpha = -2.0$.

Fig. 8. The number of passed NIST test items when chaotic random numbers are generated as numerical calculation precision is varied from $155 < N \leq 1000$.

In the range $N \leq 9$, both the mean and standard deviation of the number of passed NIST test items were zero. Therefore, in this case, it is difficult to generate high-performance chaotic random numbers. On the other hand, when $N = 19$, the average number of passed NIST test items was high, but the minimum value was 6. Such performance differences depend on the initial value, suggesting that both the numerical calculation precision and the initial value are key factors that affect the performance of chaotic random numbers.

In the range $N \geq 20$, the average number of passed NIST test items was close to 15, and the standard deviation did not exceed unity, which means that if $N \geq 20$, the performance of pseudorandom
numbers is high. Namely, to generate high-performance chaotic random numbers from the response of the integer logistic map, the precision of the numerical calculations must be at least 20.

5. Conclusion
In this study, we investigated the performance of chaotic random numbers and the factors that affect their performance improvement. To mitigate numerical calculation errors, we extended the logistic map to an integer logistic map and used it as a random number generation source. The integer logistic map allows calculations with arbitrary numerical precision, where the numerical precision is defined as the number of digits used in the numerical calculations of the state of the integer logistic map. To generate chaotic random numbers, we binarized the responses of the integer logistic map.

The performance of the generated chaotic random numbers was evaluated using NIST SP 800-22. The results of the numerical experiments indicate that the precision of the numerical calculation is a leading factor that affects the performance of the generated chaotic random numbers. In addition, both the numerical calculation precision and the initial value were inferred to affect the performance of the generated chaotic random numbers. In particular, it was suggested that the performance of the chaotic random numbers could be stabilized if the number of digits used for the numerical calculations was set to 20 or higher.

In this paper, we used the NIST test for evaluating the performance of pseudorandom numbers. However, it is also important to introduce other test suites, such as TestU01 [20]. Then, it is an important future work to verify the performance using these methods.

Acknowledgments
This research is partially supported by the JSPS Grants-in-Aid for Scientific Research (Nos. JP15KT0112, JP17K00348, JP18K18125 and JP20H00596).

References
[1] S. Oishi and H. Inoue, “Pseudo-random number generators and chaos,” Transactions of IEICE, vol. E65, no. 9, pp. 534–541, 1982.
[2] A. Arneodo and D. Sornette, “Monte Carlo random-walk experiments as a test of chaotic orbits of maps of the interval,” Physical Review Letters, vol. 52, no. 21, pp. 1857–1860, 1984.
[3] S. C. Phatak and S. Suresh Rao, “Logistic map: A possible random-number generator,” Physical Review E, vol. 51, no. 4, pp. 3670–3678, 1995.
[4] K. Ishida, A. Tsuneda, and T. Inoue, “Properties of chaotic sequences generated by finite-bit operations,” Technical Report of IEICE, vol. 98, no. 140, pp. 9–14, 1998, in Japanese.
[5] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[6] S. Araki, T. Miyazaki, S. Uehara, and K. Kakizaki, “A study on precision of pseudorandom number generators using the logistic map,” *2012 International Symposium on Information Theory and its Applications (ISITA)*, pp. 740–744, 2012.

[7] M. François, T. Grosges, D. Barchiesi, and R. Erra, “A new pseudo-random number generator based on two chaotic maps,” *Informatica*, vol. 24, no. 2, pp. 181–197, 2013.

[8] A. Saito and A. Yamaguchi, “Pseudorandom number generation using chaotic true orbits of the Bernoulli map,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 6, pp. 063122-1–063122-8, 2016.

[9] K. Cho and T. Miyano, “Stream cipher using augmented Lorenz map and the evaluation of its performance,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. J101, no. 8, pp. 210–218, 2018, in Japanese.

[10] A. Tsuneda, “Orthogonal chaotic binary sequences based on Bernoulli map and Walsh functions,” *Entropy*, vol. 21, no. 10, pp. 930-1–930-9, 2019.

[11] L. Wang and H. Cheng, “Pseudo-random number generator based on logistic chaotic system,” *Entropy*, vol. 21, no. 10, pp. 960-1–960–12, 2019.

[12] S. Eguchi, T. Miyazaki, S. Araki, S. Uehara, and Y. Nogami, “A study on randomness of sequences obtained from piecewise logistic map over integers,” *2020 International Symposium on Information Theory and its Applications (ISITA)*, pp. 451–455, 2020.

[13] A. Uchida, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya, I. Oowada, T. Kurashige, M. Shiki, S. Yoshimori, K. Yoshimura, and P. Davis, “Fast physical random bit generation with chaotic semiconductor lasers,” *Nature Photonics*, vol. 2, no. 12, pp. 728–732, 2008.

[14] M. Matsumoto and T. Nishimura, “Mersenne twister: a 623-dimensionally equidistributed uniform pseudorandom number generator,” *ACM Transactions on Modeling and Computer Simulations*, vol. 8, no. 1, pp. 3–30, 1998.

[15] R. M. May, “Simple mathematical models with very complicated dynamics,” *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[16] V. Patidar, K. K. Sud, and N. K. Pareek, “A pseudo random bit generator based on chaotic logistic map and its statistical testing,” *Informatica*, vol. 33, no. 4, pp. 441–452, 2009.

[17] J. Dong, “Random numbers generation method and the application by integer logistic map,” *Doctoral dissertation, The University of Electro-Communications*, 2012, in Japanese.

[18] L. Bassham, A. Rukhin, J. Soto, J. Nechralal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Hecket, and D. Banks, “A statistical test Suite for random and pseudorandom number generators for cryptographic applications,” *National Institute of Standards and Technology, Special Publication 800–22*, 2010.

[19] Y. Shimada, E. Takagi, and T. Ikeguchi, “Symmetry of Lyapunov exponents in bifurcation structures of one-dimensional maps,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, no. 12, pp. 123119-1–123119-2, 2016.

[20] P. L’Ecuyer and R. Simard, “TestU01: A C library for empirical testing of random number generators,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, pp. 1–40, 2007.