Paper

Design of the nonlinear look-up table in the chaotic pseudorandom number generator based on augmented Lorenz map

Kiyotaka Miyauchi¹ᵃ), Yoshihiko Horio¹ᵇ), Takaya Miyano²ᶜ), and Kenichiro Cho²ᵈ)

¹ Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan
² Department of Mechanical Engineering, Ritsumeikan University
1-1-1 Noji-higashi, Kusatsu, Shiga 525-8577, Japan

ᵃ) bmitkz@gmail.com
ᵇ) horio@riec.tohoku.ac.jp
ᶜ) tmiyano@se.ritsumei.ac.jp
ᵈ) kcho@fc.ritsumei.ac.jp

Received January 31, 2020; Revised May 20, 2020; Published October 1, 2020

Abstract: A chaos-based stream cipher using an augmented Lorenz map has been proposed. It was shown through numerical simulations that this chaotic map can generate statistically secure pseudorandom numbers, although high-speed hardware is necessary for practical use. One of the problems in digital hardware realization is implementation of the nonlinearities included in the map. In this paper, we propose a nonlinear look-up table (LUT) technique to implement the sine function considering the distribution of the argument variable. We experimentally demonstrate through field programmable gate array (FPGA) prototyping, high-speed and small-sized digital hardware implementation of the pseudorandom number generator based on the augmented Lorenz map using the proposed nonlinear LUT method.

Key Words: pseudorandom number generator, augmented Lorenz map, chaotic cryptography

1. Introduction

Recently, progressing information technologies as exemplified by advances, such as AI and IoT, accelerated the demand for high-speed communication and small devices. Correspondingly, the amount of information communicated via networks has become extremely high, and thus cryptography has

*K. Miyauchi is currently with R & D Dept., Hosiden Corporation, 4-33, Kitakyuhoji 1-Chome, Yao-City, Osaka 581-0071, Japan.
become increasingly important. Additionally, in the future, quantum computers will decrypt current cryptosystems, which are largely based on the difficulty of prime factorization and discrete logarithm problems. Therefore, it is important to develop a cryptosystem that is resistant to quantum computers. In addition, the cryptosystems should exhibit high security and be characterized by high speed, small size, and low power consumption.

A chaotic cryptosystem is one of the cryptosystems which can be used to counter quantum computers on account of its complex dynamical behavior. However, some weaknesses of the chaos-based cryptosystems have been pointed out [1], which prevent the chaos-based cryptosystems from being practical. In order to overcome the problems in conventional chaos-based encryption, a chaotic stream cipher based on the augmented Lorenz equations has been proposed [2]. The pseudorandom number generated from the augmented Lorenz system is shown to be statistically secure [3] in the sense that the generated pseudorandom numbers pass all of the statistical tests in NIST SP 800-22 [4] and TestU01 BigCrush [5]. However, the generating speed of the pseudorandom number through numerical integration of the augmented Lorenz equations is slow compared to modern stream ciphers.

To accelerate the pseudorandom number generation, the augmented Lorenz map has been proposed [6]. The pseudorandom number generated from this map is also proved to be statistically secure. However, the generating speed of the pseudorandom number from the augmented Lorenz map is still not sufficiently high. The current stream cipher operates with a generating speed of several dozens of Gbps [7, 8]. Furthermore, a generation speed of several hundreds of Gbps can be theoretically achieved [9]. Therefore, it is necessary to implement a chaotic cipher based on the augmented Lorenz map in high-speed, small-size and low-power dedicated hardware.

In general, analog circuitry with continuous variables is a natural choice for efficient chaotic hardware systems, mainly because the complexity in chaos comes from the complexity in real-numbers [10, 11]. However, one of disadvantages of the analog implementation is inevitable circuit noise, by which we cannot accurately determine the initial condition for each time. This results in different pseudorandom number sequences for each time even we use the same circuit parameters every time. To guarantee the reproducibility, which is one of the required property for encryption hardware, we use digital circuitry sacrificing original mathematical chaotic dynamics resulting in “digital” chaos [10–12]. It is important, however, to guarantee the statistical security of generated pseudorandom numbers even with quantized chaotic dynamics through the digital implementation.

One of the problems in digital hardware implementation of chaotic maps is how to implement nonlinearities in the map; it is generally difficult to faithfully implement nonlinearities in digital hardware. In particular, the augmented Lorenz map has two sine functions. In the case of software implementation, the calculation of the sine function often uses a coprocessor. However, the coprocessor may be different if the computer architecture is different. In this case, due to the nature of chaos, small errors will be expanded exponentially, and the generated pseudorandom number sequences will also differ considerably depending on the hardware. Therefore, it is important to guarantee that the output value does not depend on hardware.

Another salient problem of the digital hardware implementation of chaotic systems is quantization error, which causes dynamical degradation of digital chaos [10–12]. Digital arithmetic such as floating-point and fixed-point, and the number of bit-length will strongly affect the chaotic dynamics. We already evaluated such effects on the quality of pseudorandom numbers generated from the augmented Lorenz map using numerical simulations and hardware implementation [13, 14]. As a consequence, we found interesting but unintuitive observations which would be resulting from the digital chaos. For example, reducing the bit-length does not necessarily degrade the quality of pseudorandom numbers. To understand the observed strange phenomena, we evaluated the dynamics of digital augmented Lorenz map using some measures for dynamics and complexity such as Lyapunov spectrum, the largest Lyapunov exponent, permutation entropy [16], and string entropy [17]. However, the effects of bit-length and fixed-point arithmetic on the quality of pseudorandom numbers are very complex, so that we need further comprehensive investigations. Therefore, in the rest of the paper, we concentrate

\[\text{We obtained similar results from a digital chaotic neural network [15].}\]
only on the effects of digital implementation of the nonlinearities mentioned above.

In this paper, we examine the implementation of the sine function as a nonlinearity. Several methods for implementing the sine function have been proposed [18, 19]. For example, iterative calculations such as the coordinate rotation digital computer algorithm [20] are often used to calculate elementary functions with high accuracy and relatively small size. However, the iterative calculation is slow when compared with other implementation methods. Polynomial approximation is another example, but its arithmetic circuit is relatively large.

In contrast, a look-up table (LUT) method is simple and fast because calculated values are stored in memory in advance. However, a sufficiently large table is required to obtain high accuracy. In this study, we propose a method for reducing the size of the LUT while maintaining the performance of the hardware in pseudorandom number generation. To concentrate on the implementation of the sine function only, the other parts of the pseudorandom number generation are kept identical to those in [6]. Hence, we implement the sine function using various LUTs; however, 64-bit floating-point arithmetic is used for all calculations and LUTs to avoid unwanted effects of arithmetic and bit-length. We use NIST SP800-22 statistical tests to evaluate the resulting pseudorandom numbers.

In order to confirm the effectiveness of the proposed LUT technique, we use a field programmable gate array (FPGA), which is a reconfigurable hardware system. It allows us a rapid prototyping to explore a number of hardware architectures in a short time. In experiments, we use Xilinx FPGA VCU118 evaluation board with Xilinx Vivado2018.3 tool kit [21] as an evaluation vehicle to demonstrate that the proposed LUT technique improves the performance of the pseudorandom number generation circuit.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the augmented Lorenz map, and its method of generating pseudorandom numbers. In Section 3, we illustrate the problems associated with hardware implementation. In Section 4, we implement the sine function in the map using various LUTs, including a nonlinear LUT. We then evaluate the output sequences with those LUTs. In Section 5, we confirm the performance of pseudorandom number generator with the proposed nonlinear LUT using FPGA.

2. Stream cipher based on the augmented Lorenz map

The augmented Lorenz map [6] forms a star network centered on the variable $y_i$, and is defined as

$$y_{i+1} = \frac{1}{M_n} \sum_{n=1}^{N} x_{n,i},$$

$$x_{n,i+1} = x_{n,i}y_i - z_{n,i},$$

$$z_{n,i+1} = \sin(w_n y_i),$$

$$w_n = R \sin(M_n \phi),$$

$$M_n = n + \varepsilon Q_{n-1},$$

where $x_{n,i}, y_i,$ and $z_{n,i}$ are variables, $i$ is the time step, $n$ is an integer from 1 to $N$, $\varepsilon$ is a small positive real number. In the following, we set $\varepsilon = 10^{-4}$ according to [6]. $Q_{n-1}$ is the secret key with $Q_0 = 0$, and $M_n$ generated from $Q_{n-1}$ characterizes the augmented Lorenz map. The settings of $M_n$, that is, $Q_{n-1}$, may not strongly affect the dynamical properties of the augmented Lorenz map. In contrast, the bifurcation parameters $R$ and $\phi$ determine the dynamical properties of the augmented Lorenz map; but their values are fixed, and open to all users. In this paper, we use $R = 3.2$ and $\phi = 0.36$ [rad.] according to [6]. The initial values of $x_{n,0}, y_0$, and $z_{n,0}$ are also fixed, and open to all users. The size of the key space amounts to $2^{N-1}$, where the length of the secret key $N - 1$ can be set to an arbitrary positive integer. We use $N = 129$ for NIST SP800-22 tests as in [6].

The method for generating binary pseudorandom numbers from the chaotic time series of $x_{n,i}$ is as follows [3, 6]. We first transform $x_{n,i}$ into the corresponding binary number $b_{n,i}$ using the following equation.

$$b_{n,i} = (10^\alpha x_{n,i}) \mod 2,$$
where $10^\alpha$ is an amplification factor [3, 6], and $\alpha$ is a positive integer. According to [3, 6], we use $\alpha = 6$ in this paper. Subsequently, $b_{n,i}$ are concatenated into a binary sequence $b$ [3, 6] defined as

$$b = \{b_{2,1}, \ldots, b_{N,1}, b_{2,2}, \ldots, b_{N,2}, \ldots, b_{2,T}, \ldots, b_{N,T}\},$$

where $T$ is an appropriate positive integer which determines the length of $b$. In Sections 4 and 5, we evaluate the randomness of $b$ using the NIST SP800-22 tests. We use $T = 11,000$ throughout the paper, but $b$ obtained for $T \geq 1,000$ is used for the statistical tests.

3. Problems in hardware implementation

In this section, we summarize the problems associated with digital hardware implementation of the augmented Lorenz map. The first problem involves implementing nonlinearities included in the map defined in Eqs. (1)–(5). In particular, the augmented Lorenz map has the two sine functions in Eqs. (3) and (4). These sine functions need to be treated separately because the values assigned to them are different. At each time step, Eq. (3) calculates the variable $z_{n,i}$ for each $n$. Because the values of $y_i$ assigned to the sine function in Eq. (3) represent time-dependent variables, the behavior is complex. On the other hand, the sine function in Eq. (4) calculates $w_n$ from $M_n$, and two bifurcation parameters $R$ and $\phi$. The values of $R$ and $\phi$ are fixed, and $M_n$ is also fixed once the secret key $Q_{n-1}$ is given. Therefore, in the following sections, we focus on the effects of the sine function in Eq. (3) to construct an effective method to treat nonlinearity in the map.

The second problem is associated with digital arithmetic and bit-length. In the case of fixed-point calculations, operations such as subtraction in Eq. (2) and division in Eq. (1) involve numerical errors. For example, the subtraction of almost the same values will incur digit cancellation, thereby inducing a large error. Therefore, it is necessary to estimate the precision required for such operations and LUTs in advance to devise implementation. As mentioned in Section 1, the fixed-point calculations with reduced bit-length induce complex effects on the performance of pseudorandom numbers generated from the digital augmented Lorenz map [13, 14]. Therefore, we need further investigation on these effects, and results will be discussed elsewhere.

4. Implementation of the sine function using an LUT

As mentioned in Section 3, the effect of the sine function in Eq. (3) can be quite large. Hence, we examine the effects of the sine function implementation scheme in Eq. (3) in the rest of the paper.

We use the statistical tests in NIST SP800-22 for evaluation purposes. The simulation conditions are identical to those in [6] as stated in Sections 1 and 2. In the following tables, “PASS” indicates that all tests in NIST SP800-22 were passed (i.e., there is no statistical weakness in the output sequences), while “NOT PASS” indicates that the sequences failed for one or more tests.

First, as the simplest LUT to implement the sine function in Eq. (3), we uniformly divide the segment $[0, \pi/2]$ with various numbers of equal partitions. Table I summarizes the results of the NIST tests. As shown in the table, the resulting output sequences passed the tests for 1024 and 814 partitions. The output sequences with 813 partitions did not pass the Non-overlapping Template test in NIST SP800-22.

To further reduce the size of the LUT, we consider the distributions of the values of the variable $w_n y_i$, which is substituted in the sine function in Eq. (3). Figure 1 shows the distributions when $N$
Fig. 1. Distributions of variable $w_n y_i$ which is assigned to the sine function in Eq. (3) for (a) $N = 129$, (b) $N = 257$, (c) $N = 513$, and (d) $N = 1025$. The shape of each distribution is centered around zero irrespective of $N$. We apply this trend in the distribution to the partition methods.

From the above observations, we first simply reflect this distribution shape to the uniformly divided partition method, but with multiple segments (piece-wise linear (PWL) partition). Tables II and III show the PWL partitions, and results of the tests. In Table II, different numbers of uniform partitions are applied to two segments $[0, \pi/4]$ and $[\pi/4, \pi/2]$. In Table III, we apply different numbers of uniform partitions to four segments $[0, \pi/8]$, $[\pi/8, \pi/4]$, $[\pi/4, 3\pi/8]$, and $[3\pi/8, \pi/2]$. As shown in Table II, the PWL partition with 781 parts, which is less than the 814 in Table I with one segment, passes the tests. Therefore, the PWL partition method with two segments would reduce the size of LUT. In addition, from Table III, 769 partitions are sufficient if we use four segments. Therefore, the PWL partition method is efficient. By the way, in Tables II and III, the Non-overlapping Template test is the main cause of failure.

To further reduce the size of the LUT, we consider the distribution more accurately using nonlinear partitions based on the method in [19]. Figure 2 shows an example of this type of a nonlinear partition in the segment $[0, \pi/2]$ with 32 partitions. Table IV lists the number of partitions used in the simulations, and the results of the tests for these partitions. As shown in the table, it is possible to reduce the total number of partitions down to 670 parts using the nonlinear partition method.

in Eq. (1) is 129, 257, 513, and 1025. As shown in the figure, all the distributions are concentrated around zero, and the shapes of the distributions do not change drastically when $N$ changes.
Table II. Results of the statistical tests of NIST SP800-22 for the PWL partitions with two segments of $[0, \pi/4]$ and $[\pi/4, \pi/2]$.

| Total number of partitions | Number of partitions in each segment | Result of NIST tests |
|----------------------------|-------------------------------------|----------------------|
| 1024                       | 612, 412                            | PASS                 |
| 814                        | 457, 357                            | PASS                 |
| 781                        | 441, 340                            | PASS                 |
| 780                        | 440, 340                            | NOT PASS             |

Table III. Results of the statistical tests of NIST SP800-22 for the PWL partitions with four segments of $[0, \pi/8]$, $[\pi/8, \pi/4]$, $[\pi/4, 3\pi/8]$, and $[3\pi/8, \pi/2]$.

| Total number of partitions | Number of partitions in each segment | Result of NIST tests |
|----------------------------|-------------------------------------|----------------------|
| 1024                       | 512, 256, 128, 128                  | PASS                 |
| 780                        | 512, 128, 92, 48                    | PASS                 |
| 769                        | 512, 128, 88, 41                    | PASS                 |
| 768                        | 512, 128, 88, 40                    | NOT PASS             |

Fig. 2. The example of the nonlinear partition of segment $[0, \pi/2]$ with 32 number of partitions, using the method in [19].

Table IV. Results of the statistical tests of NIST SP800-22 with nonlinear LUTs, considering the distribution of the argument variable in the sine function.

| Total number of partitions | Result of NIST tests |
|----------------------------|----------------------|
| 1024                       | PASS                 |
| 768                        | PASS                 |
| 670                        | PASS                 |
| 669                        | NOT PASS             |

This decreases the size of the LUT, thereby leading to fast and small-sized hardware implementation of the augmented Lorenz map.

5. FPGA implementation results

In this section, we use FPGA (Xilinx FPGA VCU118 with Vivado2018.3 [21]) to experimentally confirm the effectiveness of the proposed LUT scheme. The block diagram of the FPGA implementation is shown in Fig. 3. To consider only the effect of the LUT of sine function, other conditions are set identical to those in Section 4.

Figures 4 and 5 show the results of FPGA implementation. Figure 4 shows the generation speed of
Fig. 3. The block diagram of FPGA implementation. To avoid unwanted effects of arithmetic and bit-length, 64-bit floating-point expressions are used in all calculations including the LUT.

the pseudorandom numbers when we change the number of partitions in the nonlinear LUT. Figure 5 shows the hardware size of the pseudorandom number generator according to the number of partitions in the LUT. In the figure, the hardware size denotes the sum of the number of LUTs and flip-flops used in the FPGA by Vivado2018.3. In the figures, the circles indicate that all tests in NIST SP800-22 are passed, while the crosses show failure cases. As shown in the figures, decrease in the size of the LUT increases the generation speed, and at the same time, decreases hardware size. In the case of the sine function implemented with a fifth-order Maclaurin expansion as in [6], the generation speed and hardware size were reported as 171.166 [Mbps] and 331,436, respectively. In contrast, the proposed nonlinear LUT technique with 670 partitions resulted in 269.8 [Mbps] in speed, and 316,311 in size. Therefore, the nonlinear LUT with 670 partitions improves the generation speed about 1.58 times, and reduces the hardware size about 0.95 times. These results experimentally confirm that the nonlinear LUT is effective for hardware implementation of the augmented Lorenz map, in particular, for acceleration.

In Figs. 4 and 5, decreases in the size of the LUT caused some low resolution LUTs failed the NIST tests. However, contrary to intuition, even further decreases in the size of the LUT, the pseudorandom numbers passed in some cases. The reason of this reversal may be considered as follows. The approximation error of the sine function due to decreases in the size of the LUT may help to increase the complexity in the augmented Lorenz map to improve the randomness resulting better

Fig. 4. The generation speed of the pseudorandom numbers obtained from FPGA implementation with the nonlinear LUT. The circles indicate that all tests in NIST SP800-22 are passed, while the crosses show the failures.
6. Conclusions

For efficient implementation of nonlinearities in the augmented Lorenz map using digital circuits, we examined the method of designing the LUT for the sine function in the map. As a result, we confirmed that the nonlinear partition method using the distribution of the argument variable in the sine function is effective in reducing the size of the LUT without degrading the quality of pseudorandom number. Through FPGA implementation of the proposed nonlinear LUT, we confirmed with statistical tests of NIST SP800-22 the high-speed and small-sized implementation of the pseudorandom number generator based on the augmented Lorenz map hardware.

As a future problem, we will further ensure the statistical security of the resulting random sequences through TestU01 BigCrush tests. In addition, we will evaluate and analyze the effects of fixed-point arithmetic and bit-length reduction to further improve the speed and size, in particular, of the digital augmented Lorenz map hardware [13, 14]. Finally, we will fully implement the dedicated digital chaotic stream encryption hardware using the proposed LUT technique and specific fixed-point arithmetic.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 18H03307. Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University.

References

[1] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” Int. J. Bifurcation & Chaos, vol. 16, pp. 2129–2151, 2006.
[2] K. Cho and T. Miyano, “Chaotic cryptography using augmented Lorenz equations aided by quantum key distribution,” IEEE Trans. Circuits Syst. I, vol. 62, no. 2, pp. 478–487, 2015.
[3] K. Cho and T. Miyano, “Design and test of pseudorandom number generator using a star network of Lorenz oscillators,” Int. J. Bifurcation & Chaos, vol. 27, no. 12, pp. 1750184-1–1750184-14, 2017.
[4] A. Rukhin et al., “A statistical test suite for random and pseudorandom number generators for cryptographic applications,” NIST Special Publication 800-22 Revision 1a, April 2010.
[5] P. L’Ecuyer and R. Simard, “TestU01: A C library for empirical testing of random number generators,” ACM Trans. Math. Soft., vol. 33, Article 22, pp. 1–40, 2007.

[6] K. Cho and T. Miyano, “Stream cipher using augmented Lorenz map and the evaluation of its performance,” IEICE Trans. Fundamentals, vol. J101-A, no. 8, pp. 210–218, 2018. (in Japanese)

[7] S. Kiyomoto et al., “Fast implementation of KCipher-2 for software and hardware,” IEICE Trans. Inf. & Syst., vol. E97-D, no. 1, pp. 43–51, 2014.

[8] P. Ekdahl and T. Johansson, “SNOW - a new stream cipher,” in Proc. of First Open NESSIE Workshop, KU-Leuven, 2001.

[9] P. Ekdahl et al., “A new SNOW stream cipher called SNOW-V,” in IACR Cryptology ePrint Archive, 2018. http://eprint.iacr.org/2018/1143.

[10] Y. Horio and K. Aihara, “Analog computation through high-dimensional physical chaotic neurodynamics,” Physica-D, vol. 237, no. 9, pp. 1215–1225, 2008.

[11] L. Blum, F. Cucker, M. Shub, and S. Smale, Complexity and Real Computation, Springer-Verlag, New York, 1997.

[12] S. Li, G. Chen, and X. Mou, “On the dynamical degradation of digital piecewise linear chaotic maps,” Int. J. Bifurcation & Chaos, vol. 15, no. 10, pp. 3119–3151, 2005.

[13] K. Miyauchi, Y. Horio, T. Miyano, and K. Cho, “Consideration of bit length reduction in pseudorandom number generator hardware,” in Proc. IEICE Society Conf., N-1-2, p. 147, 2019. (in Japanese)

[14] K. Miyauchi, “Digital hardware implementation of pseudorandom number generator based on augmented Lorenz map,” Master’s Thesis, Grad. School of Info. Sci., Tohoku Univ., 2020. (in Japanese)

[15] N.M. Triet and Y. Horio, “Hopfield-type chaotic neural network: Bit-length effect on solving traveling salesman problems,” in Proc. IEICE Society Conf., N-1-18, p. 163, 2019.

[16] C. Bandt and B. Pompe, “Permutation entropy: A natural complexity measure for time series,” Phys. Rev. Lett., vol. 88, no. 17, pp. 174102-1–174102-4, 2002.

[17] T. Miyano and K. Cho, “String entropy as a measure of complexity in chaotic time series,” in Proc. Int. Symp. on Nonlinear Theory and Its Appl., pp. 133–136, 2016.

[18] J.M. Muller, Elementary Function: Algorithms and Implementation, Birkhauser, Boston, 1997.

[19] T. Sasao, S. Nagashima, and J.T. Butler, “Numerical function generators using LUT cascades,” IEEE Trans. Computers, vol. 56, no. 6, pp. 826–838, 2007.

[20] R. Andraka, “A survey of CORDIC algorithms for FPGAs,” in Proc. ACM/SIGDA Conf., pp. 191–200, 1998.

[21] Xilinx, Inc., http://www.xilinx.com/.