Sensors and Materials, Vol. 35, No. 5 (2023) 1687–1700
MYU Tokyo

S & M 3280

Hybrid Chaotic Keystream Generator Based on Dawson's Summation Generator

Shyi-Tsong Wu*

Department of Electronic Engineering, National Ilan University,
No. 1, Sec. 1, Shennong Rd., Yilan City, Yilan County 26047, Taiwan

(Received January 7, 2023; accepted March 16, 2023)

Keywords: IoT, stream cipher, hybrid chaotic system, Dawson's summation generator

The IoT consists of a vast number of interconnected sensors, and ensuring secure communications between these sensors is a crucial concern. In this study, we apply chaos theory to the security of real-time communications in the IoT. Chaos theory has been increasingly utilized in cryptography in recent years, with the proposal of numerous stream ciphers based on chaos. We merge different 1D chaotic maps and construct a hybrid keystream generator. To increase the randomness and linear complexity of the output sequence, we use a nonlinear Dawson's summation generator (DSG) to mix different chaotic outputs. The proposed keystream generator is suitable for real-time IoT communications as it features a low computational load and an improved throughput. Our analysis of the proposed keystream generator includes an evaluation of its resilience against known attacks, a measurement of its linear complexity, and statistical random number tests on the output keystream. The proposed keystream generator achieved a pass rate of 100% for the statistical test of the FIPS PUB 140-1 standard and a pass rate of at least 96% for that of the NIST SP800-22 standard.

1. Introduction

Pseudorandom binary sequences are widely used in various applications such as spread spectrum communications, digital communications, and cryptography. In cryptography, the security of a stream cipher largely depends on the quality of the pseudorandom sequences it generates. Chaotic systems have gained significant attention in recent years as a promising source of high-quality pseudorandom sequences.

The sensitive dependence on initial conditions and the good statistical properties of chaotic systems have made them attractive for use in cryptography.[1–7] This has led to the development of various stream ciphers based on chaos theory, which offer high-speed encryption and are suitable for applications of real-time communication in the IoT. The IoT is composed of numerous sensors, and the security of communications among the sensors is an important issue. This paper is about the security of real-time communications within the IoT.

*Corresponding author: e-mail: stwu@niu.edu.tw
https://doi.org/10.18494/SAM4316

ISSN 0914-4935

A keystream generator that utilizes a high-dimensional Chen chaotic system has been proposed.[8] High-dimensional chaotic systems are more secure and have more complex parameters. Moreover, one chaotic iteration only produces one bit of the output sequence, which is inefficient.[1] The output data rate can be increased to meet the requirement of real-time communications in the IoT. Some reported sequences based on two 1D chaotic maps with different parameters or different chaotic maps use a linear XOR operation to mix the two sequences.[9] Another consideration in the combination of chaotic components is to use a nonlinear operation to increase the resistance against attacks.[10]

The focus of our paper is to present a novel hybrid stream cipher that is based on chaos. We use 1D chaotic maps to reduce the operation load. We increase the output data rate through the output of a 64-bit binary sequence with each iteration. In addition, to increase the linear complexity of the output keystream and its security, we apply Dawson's summation generator (DSG) as a nonlinear combining function to construct the compound maps. We also investigate the security, linear complexity, and randomness of the proposed scheme and analyze it for robustness against known attacks. The results of statistical tests show that the outputs of the proposed scheme are improved.

The contributions of the article are as follows:

- The proposed keystream generator merges hybrid 1D chaos to ensure the security of real-time communications among the sensors in IoT systems.
- A nonlinear element is used to combine different chaotic components, thus increasing security and resistance against attacks.
- The utilized nonlinear element is DSG, for which the correlation probabilities between inputs and outputs are all 1/2. This desirable characteristic makes it possible to resist correlation attacks and obtain good randomness.

The paper is structured as follows. In Sect. 2, we provide the background knowledge and related works on chaotic stream ciphers. In Sect. 3, we present the proposed hybrid chaotic keystream generator with DSG. To assess the quality and randomness of the proposed keystream generator, we conducted statistical tests using both Federal Information Processing Standards Publication 140-1 (FIPS PUB 140-1) and NIST Special Publication 800-22 (SP800-22) standards. In Sect. 4, experimental results are given for the proposed hybrid keystream generator. Furthermore, we analyzed the resistance of the generated sequences against known attacks. Finally, in Sect. 5, we conclude the findings obtained from our proposed keystream generator.

2. Background and Related Works

In this section, we introduce chaos theory and some related stream ciphers based on chaos. We also derive some characteristics of these stream ciphers.

2.1 Chaos theory

Chaos theory is an area of mathematics that focuses on the study of complex systems. Henri Poincaré was one of the early proponents of chaos theory, and in the 1880s, he demonstrated that

the orbits of the three-body problem are nonperiodic and do not converge to a fixed point.[11–13] Today, chaos theory is a significant research area with broad applications across many disciplines, including physics, biology, economics, engineering, and philosophy. Chaotic behavior has been observed and studied in a range of systems, and the high sensitivity to initial conditions exhibited by chaotic systems has made them valuable in various applications, including cryptography.[2,14]

2.1.1 Logistic map

The logistic equation, a 1D chaotic map, was introduced by Pierre Verhulst in the 19th century.[15] This equation is a nonlinear chaotic system that achieves chaos through period-doubling bifurcation. Logistic maps can be classified on the basis of bifurcation diagrams, which display the properties of a dynamical system as a function of a control parameter. The most widely used logistic map features a nonlinear recurrence relation with a single control parameter μ and the variable value x_n. The map is

$$x_{n+1} = \mu x_n (1 - x_n). \tag{1}$$

The logistic map has an initial value x_0 in the interval [0, 1], and x_n represents the value of the map at iteration n. The map has a branch parameter μ, which ranges from 0 to 4. The system exhibits chaotic behavior when $3.5699456 < \mu \leq 4$.[15]

2.1.2 Tent map

Equation (2) gives a piecewise definition of the parameterized tent map.[16]

$$x_{n+1} = \begin{cases} ux_n, & 0 \leq x_n < 0.5 \\ u(1 - x_n), & 0.5 \leq x_n < 1 \end{cases} \tag{2}$$

The tent map is a commonly used chaotic map in the literature of nonlinear discrete dynamical systems. It is often used as an initial example to illustrate chaotic maps. The tent map can be used to describe the behavior of a 1D elastic band, where each iteration of the map results in band stretching to twice its original length and then folding back onto itself. The value of the parameter u is between 0 and 2, and the map maps the unit interval [0, 1) into itself.

2.1.3 Sine map

Owing to the special range of the sine map function, we can easily control the value within the range [−1, 1]. This map is defined as[17]

$$x_{n+1} = a\sin(\pi x_n), \tag{3}$$

where a is a control parameter and x_n is a real value in the interval (0, 1). When a is in the interval [0, 1], the sine map enters a chaotic state.

2.1.4 Dawson's summation generator

DSG is known for its ability to produce a good input–output correlation probability. It was proposed by Dawson and is shown in Fig. 1, where D is a D-type flip-flop. The symbols are defined as follows:[18]

a_j: input bit at clock cycle j,

b_j: input bit at clock cycle j,

c_j: carry bit at clock cycle j, assuming an initial carry value of $c_{-1} = 0$,

z_j: sum output at clock cycle j, $z_j = a_j \oplus b_j \oplus c_{j-1}$.

According to Table 1, all the input–output correlation probabilities and the carry–output correlation probability for DSG equal 1/2.[19] These features can prevent a correlation attack.

2.2 Related stream ciphers based on chaos

Many stream ciphers have been generated by chaos theory. A new stream cipher was proposed by Kanso and Smaoui,[9] whose two algorithms were based on logistic chaotic maps. Their first algorithm produced a sum of output sequences based on a single 1D logistic map, and the second algorithm produced a combination of output sequences based on two 1D logistic maps with different parameters. The second algorithm defined a bit of the binary sequences from every chaotic transformation with a threshold value of 0.5. Finally, both output binary sequences were obtained by using the XOR operation to mix the two sequences. The production of only one bit from every chaotic iteration was inefficient, and the linear combining element can be replaced by a nonlinear element to increase its security.

Hu *et al.* proposed a new stream cipher.[8] Their algorithm was based on a Chen chaotic system. They proposed a binary sequence generator that utilizes the high-dimensional Chen chaotic system. After conducting a large number of experiments, they generated the final output binary sequence by alternating between the 3D sequences $x(i)$, $y(i)$, and $z(i)$. Their statistical tests and security analysis demonstrated that the binary sequences produced by the proposed cipher

Table 1
Correlation probability of DSG.

a_j	b_j	c_{j-1}	c_j	z_j	Correlation probability
0	0	0	0	0	
0	0	1	0	1	Input–Output
0	1	0	1	1	prob($z_j = a_j$) = 1/2
0	1	1	0	0	prob($z_j = b_j$) = 1/2
1	0	0	0	1	prob($z_j = c_{j-1}$) = 1/2
1	0	1	1	0	Output–Carry
1	1	0	1	0	prob($z_j = c_j$) = 1/2
1	1	1	1	1	

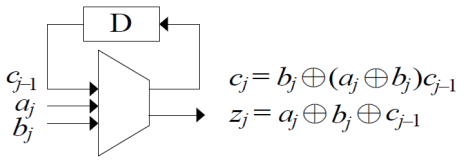

Fig. 1. Dawson's summation generator.

exhibited strong pseudorandom characteristics and were highly resistant to attacks. However, the computation of high-dimensional equations is complex, increasing the computational load.

Abderrahim *et al.* developed a chaos-based stream cipher that leveraged the simplicity of symbolic dynamic description and synchronization.[1] The proposed stream cipher was based on discretizing chaotic sequences generated from 1D chaotic maps. The two discrete chaotic systems used were a Bernoulli map and a skewed tent map. The final encryption/decryption process was based on a binary additive stream cipher principle that utilizes the XOR operation between the confidential data and the keystream. This proposed cipher utilized symbolic dynamic sequences generated from 1D chaotic maps. The novelty of this cipher was its use of a combination of chaotic sequences. However, it also produced only one random bit in each round, limiting its throughput and efficiency.

François *et al.* proposed a secure pseudorandom number generator that combined three chaotic maps. Their algorithm employed a digressive modulo to progressively index the positions of an initial vector, followed by the permutation of the associated elements using the XOR operation.[20] The advantage of this scheme is that it permutes and shuffles the elements constantly, increasing the period and complexity. However, its operation was complicated, which might increase the system load.

3. Proposed Keystream Generator

This section introduces a novel keystream generator based on hybrid chaos. It uses 1D chaotic maps to reduce the operation load. To enhance the linear complexity and security of the output sequence, the output keystream of the generator is merged from the chaotic maps with a nonlinear combination. We present the hybrid keystream generator based on chaos with DSG as a nonlinear combining function to merge different chaotic maps.

We propose three basic hybrids of chaotic maps based on the DSG component. They are as follows:
the hybrid chaos of the logistic map with the tent map,
the hybrid chaos of the logistic map with the sine map, and
the hybrid chaos of the tent map with the sine map.

3.1 Hybrid chaos of logistic map with tent map using DSG

In this subsection, we present the hybrid chaos of the logistic map with the tent map using DSG, as shown in Fig. 2. Here, l_n is the output keystream of the logistic map and t_n is the output keystream of the tent map. c_n is the carry bit from DSG and z_n is the output binary sequence.

The generator is implemented as follows. First, we select a 128-bit key randomly. Then, the 128-bit key is divided into two 64-bit numbers r_i to serve as the initial value and key of the chaotic map.

Second, because the initial value x_0 of the logistic map must be in the interval (0, 1), we essentially map the 64-bit r_i to the real interval (0, 1). A simple way to transform a 64-bit hexadecimal number to a real number is to use the following equation:

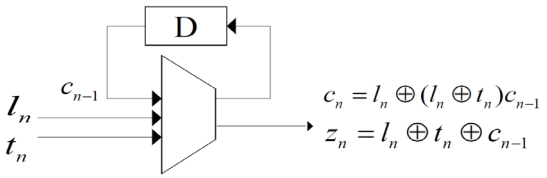

Fig. 2. Hybrid chaos of logistic map with tent map using DSG.

$$F_1(r_i) = r_i / 1.8447 \times 10^{19}. \tag{4}$$

The initial value x_0 of the logistic map in the interval (0, 1) is thus obtained. We use the same method to map another r_i to the parameter m in the same interval in the chaotic logistic map. The equation is as follows:

$$F_2(r_i) = 3.569 + (r_i / 1.8447 \times 10^{19}) \times (4 - 3.569). \tag{5}$$

Then, the binary sequence l_n of the logistic map output is produced using

$$l_n = x_n \times 2^{64}, \tag{6}$$

where x_n is the chaotic output.

The parameter u of the tent map is between 0 and 2, and it maps the unit interval [0, 1] into itself. For the tent map, we use another key and also map r_i to the initial value using Eq. (4), and μ is mapped to the interval (0, 2) using

$$F_3(r_i) = (r_i / 1.8447 \times 10^{19}) \times 2. \tag{7}$$

The output sequence t_n of the tent map is produced similarly to Eq. (6), $t_n = x_n \times 2^{64}$. Here, t_n denotes a 64-bit binary sequence and x_n is the chaotic output. Finally, the binary sequences t_n and l_n are mixed bitwise using DSG to produce the output sequence z_n.

3.2 Hybrid chaos of logistic map with sine map using DSG

Figure 3 shows the similar hybrid of the logistic map with the sine map, where l_n is the output sequence of the logistic map and s_n is the output keystream of the sine map. z_n is the binary keystream output from the hybrid keystream generator.

3.3 Hybrid chaos of logistic map with sine map using DSG

We propose a hybrid chaotic keystream generator with the same architecture that uses the tent and sine maps with DSG, as shown in Fig. 4, where s_n is the output sequence of the sine map and t_n is the output sequence of the tent map. z_n is the binary sequence output from this combination.

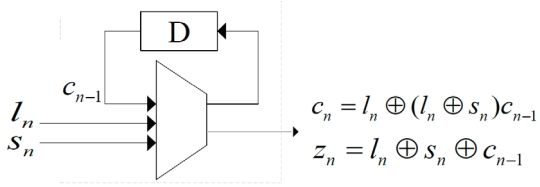

Fig. 3. Hybrid chaos of logistic map with sine map using DSG.

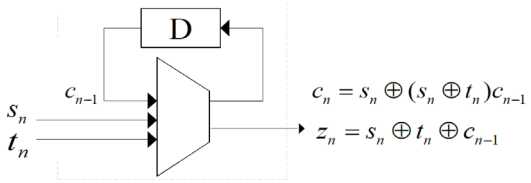

Fig. 4. Hybrid chaos of sine map with tent map using DSG.

4. Security Analysis and Experimental Results

To evaluate the proposed hybrid keystream generator, we conducted simulations using MATLAB. Specifically, we measured the generator's linear complexity as well as its performance in randomness tests based on the FIPS PUB 140-1 and NIST SP800-22 standards. In addition, we analyzed the resistance of the proposed hybrid keystream generator against various security attacks.

This section is divided into three parts. Section 4.1 provides experimental results of the linear complexity for the proposed hybrid keystream generator. Section 4.2 presents an analysis of the security of the proposed generator. Section 4.3 presents the results of statistical random number tests on the proposed generator.

4.1 Linear complexity of the proposed keystream generator

The linear complexity of a pseudorandom number sequence is a crucial factor in stream cipher systems as it directly impacts the strength of the output sequence and the difficulty in cracking the system. The linear complexity is defined as the minimum order of the linear feedback shift register (LFSR) that can produce the same sequence as the given sequence $\{s_n\}$. Figure 5 illustrates the linear complexity of the proposed hybrid keystream generator, which combines linear operations. The length of the input sequences is denoted by N, with the maximum N being 10^5 in our experiment. The simulation results indicate that the linear complexity of the proposed system is close to $N/2$.

4.2 Attack analysis

In the proposed hybrid keystream generator, the secret keys are utilized to determine the initial values and system parameters, which are essential components for generating the keystream. Malicious attackers attempt to determine these values. In this section, we evaluate the ability of the generated keystream to withstand attacks such as brute-force, chosen ciphertext, and resynchronization attacks.

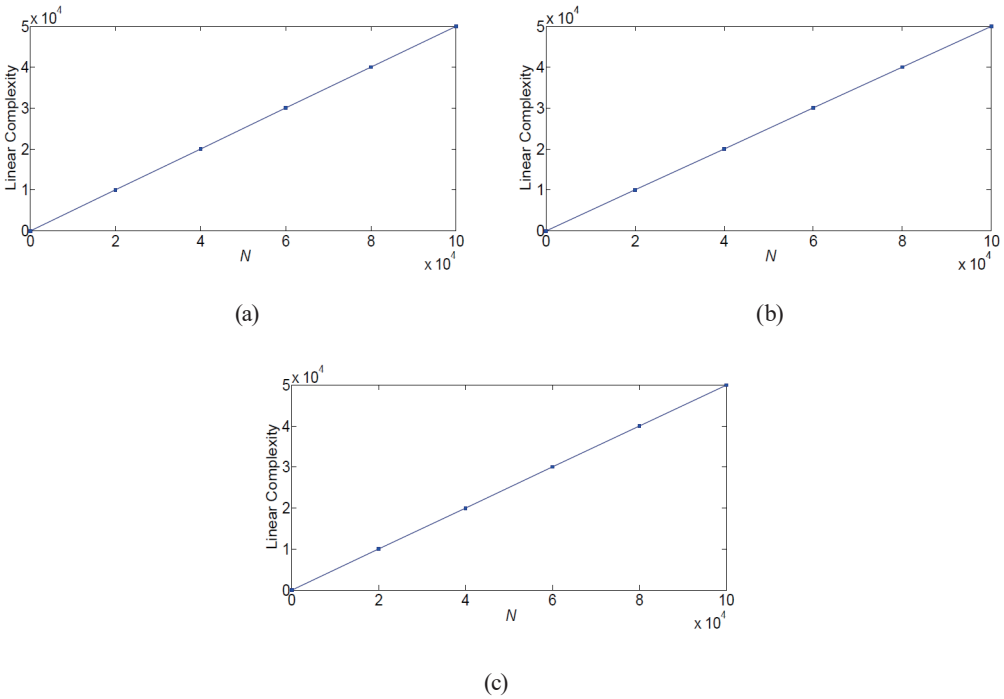

Fig. 5. (Color online) Linear complexity of the hybrid chaos using DSG. (a) Linear complexity of hybrid chaos of logistic map with tent map using DSG, (b) linear complexity of hybrid chaos of tent map with sine map using DSG, and (c) linear complexity of hybrid chaos of logistic map with sine map using DSG.

4.2.1 Brute-force attack

A brute-force attack is a commonly used method to break a cryptographic cipher, which involves trying all possible keys until the correct one is found.(1) This approach can be time-consuming as it requires every combination to be tested, and it is typically used when the size of the key space is small. To prevent brute-force attacks, it is important to have a large key space, typically at least 2^{128}, to ensure the security of the cipher.

For the proposed hybrid keystream generator combined with DSG, there are two initial values and two system parameters. All the initial values and system parameters come from the random 64-bit number r_i and a total 256-bit key in the system. Thus, the key space of the proposed keystream generator is about 2^{256}, which is larger than 2^{128}. The key space of the proposed keystream generator is about 2^{384}, which is also larger than 2^{128}, enabling it to avoid brute-force attacks.

4.2.2 Chosen ciphertext attack (CCA)

A CCA is a cryptanalysis attack in which the attacker can select a ciphertext of their choice and obtain its corresponding decryption using an unknown key.(6) By analyzing the obtained information, the attacker can try to deduce the secret key used for decryption. Essentially, the attacker can input certain ciphertexts into the system and observe the resulting plaintexts, which

they can then use to gain insights into the encryption process and potentially break the system's security.

A CCA is ineffective against the proposed DSG-based hybrid chaos scheme. Firstly, this is because we combine two chaotic algorithms with a nonlinear structure. The output sequences that the attacker eavesdrops on are from DSG, and the attacker cannot obtain the original chaotic sequences. DSG has a good correlation probability of 1/2 and nonlinearity. These features can prevent such an attack. Moreover, DSG has two input sequences with two initial values and two system parameters, giving it a total of four values that constitute the key space for the hybrid keystream generator. For this reason, the attacker cannot obtain the original chaotic sequence, making it infeasible to crack the secret keys.

4.2.3 Resynchronization attack

When using synchronous stream ciphers, it is important to have a way to resynchronize the ciphers in case synchronization is lost. If this situation occurs, the ciphers may be vulnerable to attack. To prevent this attack, a common practice is to encrypt the stream in fixed data blocks, known as frames or packets. Although the same secret key is used for all frames, a new initial or frame-counter value is incorporated in each frame, which helps to maintain synchronization and prevent attacks.[21]

This type of attack involves manipulating the synchronization signal of a system to gain information about its chaotic dynamics, which can compromise its security. However, the proposed keystream generator cryptosystem makes effective use of the two key features of chaotic systems, namely, their sensitivity to initial conditions and unpredictability, which enables it to resist such attacks.

The proposed keystream generator uses the hybrid chaotic maps to enhance security. Two chaotic maps are combined in the proposed hybrid chaotic DSG scheme. The complexity of the resulting keystream makes it difficult for attackers to synchronize without exact knowledge of the chaotic map parameters. Moreover, to prevent malicious attacks and collisions, the initial values must be changed regularly. The high sensitivity of chaos to initial conditions leads to significant divergence in the keystream when nearby values are changed, thereby preventing attackers from predicting the keystream and obtaining the secret keys.

4.2.4 Differential cryptanalysis

Differential cryptanalysis is used to analyze block ciphers, stream ciphers, and cryptographic hash functions. It involves studying how changes in input can affect the output. Specifically, for block ciphers, it refers to a set of methods that can be used to trace differences through the sequence of transformations, identifying areas where the cipher exhibits nonrandom behavior, and leveraging these properties to uncover the secret key.[22]

The proposed keystream generator is resistant to a certain type of attack based on the manipulation of the synchronization signal. This is due to its high sensitivity to small input changes and its high diffusion level. To verify this, a sensitivity analysis of the keystream

generator is conducted by altering a single bit of the secret key and calculating the keystream difference rate (*kdr*) using the following equation:[22]

$$kdr(k) = \frac{Diff(k, k_1) + Diff(k, k_2)}{2 \times N} \times 100\%. \quad (8)$$

Here, N represents the length of the binary output keystream and $Diff(k, k_1)$ and $Diff(k, k_2)$ denote the numbers of differing bits between the output keystreams generated by the secret keys k, k_1 and k, k_2, respectively, with each keystream having a length of N bits. For our experiment, we set the binary length of the output keystream to $N = 10^7$ and the secret keys k, k_1, and k_2 to 0x0000000000000000, 0x0000000000000001, and 0x0000000000000010, respectively. The secret keys are mapped as the parameters of the chaotic system to generate the output sequences.

Table 2 presents *kdr* of the proposed keystream generator based on the nonlinear DSG combination. The values of *kdr* for the combinations of logistic and tent maps, logistic and sine maps, and tent and sine maps are 43.56, 44.27, and 47.23%, respectively.

4.3 Statistical random number tests

In this section, we give the results of statistical random number tests on the output keystreams. To avoid attacks on the initial value and system parameter data, we discard the first 200 bits of the output sequence. We assess the randomness properties of the output keystreams using two tests in FIPS PUB 140-1 and SP800-22.

4.3.1 FIPS PUB 140-1

We first describe the results for the four FIPS PUB 140-1 tests. If the keystream fails any of the tests, then we consider it to have failed the FIPS PUB 140-1 test.[23]

For the FIPS PUB 140-1 randomness test, we created 100 different keystreams using random keys and initial values, with each keystream having a length of 20000 bits. Table 3 displays the

Table 2
Keystream difference rate (*kdr*) of the proposed chaotic nonlinear hybrid using DSG.

Proposed nonlinear hybrid chaos using DSG	*kdr* (%)
Logistic and tent maps	43.56
Logistic and sine maps	44.27
Tent and sine maps	47.23

Table 3
Results of FIPS PUB 140-1 randomness test under 20000 bits/sample.

	Logistic and tent maps (%)	Logistic and sine maps (%)	Tent and sine maps (%)
Monobit test	100	100	100
Poker test	100	100	100
Runs test	100	100	100
Long run test	100	100	100

results of the FIPS PUB 140-1 random test for the hybrid keystream generator with the DSG nonlinear component. The results indicate that the keystream generator has excellent statistical properties, with a pass rate of 100% for each test.

4.3.2 NIST SP800-22

The NIST SP800-22 standard has 15 tests,[24] and if the computed p-value for the test is <0.01, then it is concluded that the sequence is non-random.

To perform the random tests specified in NIST SP800-22, we generated 100 keystreams using different random keys and initial values, each with a length of 10^7 bits. To ensure security, we discarded the first 200 bits of each keystream. Table 4 summarizes the pass rates of these keystreams for each test. The results show that the proposed keystream generator with the

Table 4
Statistical test results for NIST SP800-22.
(a) Logistic and tent maps

No	Statistical test	p-value	Pass rate under 10^7 bits/sample (%)
1	Frequency	0.599789	100
2	Block frequency	0.559488	99
3	Runs	0.614362	99
4	Longest runs of ones	0.587471	98
5	Rank	0.401284	99
6	Discrete Fourier transform	0.574241	98
7	Non-overlapping templates matching	0.489086	99
8	Overlapping templates matching	0.577766	99
9	Universal statistical	0.615446	98
10	Linear complexity	0.752259	99
11	Serial	0.724171	99
12	Approximate entropy	0.549099	98
13	Cumulative sums	0.498781	100
14	Random excursions	0.563970	96
15	Random excursions variant	0.693043	96

(b) Logistic and sine maps

No	Statistical test	p-value	Pass rate under 10^7 bits/sample (%)
1	Frequency	0.610623	100
2	Block frequency	0.518483	99
3	Runs	0.719435	99
4	Longest runs of ones	0.600454	99
5	Rank	0.477268	99
6	Discrete Fourier transform	0.550107	98
7	Non-overlapping templates matching	0.592170	98
8	Overlapping templates matching	0.770031	99
9	Universal statistical	0.698853	98
10	Linear complexity	0.603548	98
11	Serial	0.636348	98
12	Approximate entropy	0.668394	99
13	Cumulative sums	0.674290	99
14	Random excursions	0.659713	97
15	Random excursions variant	0.709708	98

Table 4 (Continued)
(c) Tent and sine maps

No	Statistical test	*p*-value	Pass rate under 10^7 bits/sample (%)
1	Frequency	0.634401	100
2	Block frequency	0.776840	99
3	Runs	0.527110	98
4	Longest runs of ones	0.525258	99
5	Rank	0.626294	98
6	Discrete Fourier transform	0.505455	97
7	Non-overlapping templates matching	0.519561	98
8	Overlapping templates matching	0.694232	98
9	Universal statistical	0.617590	98
10	Linear complexity	0.605597	98
11	Serial	0.681793	99
12	Approximate entropy	0.652069	99
13	Cumulative sums	0.610212	97
14	Random excursions	0.541178	98
15	Random excursions variant	0.620784	97

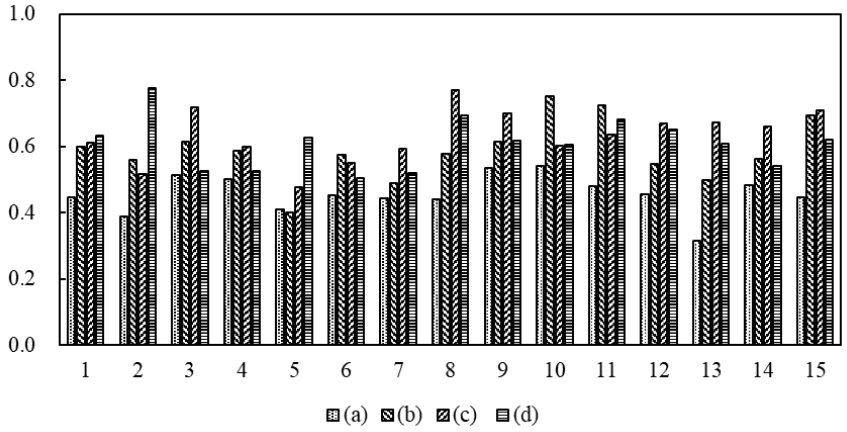

Fig. 6. Comparison of *p*-values between (a) Kanso's scheme and the proposed scheme, (b) logistic and tent maps, (c) logistic and sine maps, and (d) tent and sine maps.

nonlinear component DSG achieved a pass rate of at least 96% in all tests. In the statistical random number tests from FIPS PUB 140-1 and NIST SP800-22, the *p*-value is computed to determine the strength of randomness in the sequence, where a *p*-value of 1 indicates ideal randomness in the sequence. In Fig. 6, the *p*-values of Kanso's scheme and the proposed scheme are compared.[9] It can be observed that the proposed scheme outperforms Kanso's scheme in the statistical tests.

5. Conclusions

In the realm of IoT, the communication channels among sensors are of paramount importance. To ensure the secure and reliable transmission of sensors, chaos theory can be applied to real-

time communication security in the IoT environment. In this paper, we propose a chaotic hybrid keystream generator that uses DSG as the nonlinear combining function to enhance its security. The proposed generator is analyzed against known attacks, and its security is experimentally verified through tests of linear complexity, statistical randomness, and resistance to attacks such as brute-force, chosen ciphertext, and resynchronization attacks. The results show that the proposed generator outperforms the linear combining function and has a pass rate of 100% for the statistical test of FIPS PUB 140-1 and at least about 96% for the NIST SP800-22 test. To further increase the security, the proposed hybrid chaos with a nonlinear combining element can be merged into a multimode chaotic structure. A nonlinear combining element utilizing cellular automata and a session key also merits research.

Acknowledgments

The author expresses his gratitude to Mr. Chun-Chen Liu for his essential contribution and to the anonymous reviewers for their valuable feedback, which has greatly enhanced the quality of this paper.

References

1 N. W. Abderrahim, F. Z. Benmansour, and O. Seddiki: Nonlinear Dyn. **78** (2014) 197. https://doi.org/10.1007/s11071-014-1432-z
2 Y. Feng, J. Li, and X. Yang: Proc. Symp. Photonics and Optoelectronics (2009) 1–4.
3 J. Fridrich: Int. J. Bifurcat Chaos **8** (1998) 1259. https://doi.org/10.1142/S021812749800098X
4 M. Goresky and A. Klapper: IEEE Trans. Inf. Theor. **48** (2002) 2826. https://doi.org/10.1109/TIT.2002.804048
5 A. Klapper and M Goresky: Fast Software Encryption, R. Anderson, Ed. (Springer, Berlin and Heidelberg, 1994) pp. 174–178. https://doi.org/10.1007/3-540-58108-1_21
6 L. M. Pecora and T. L. Carroll: Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 1992) 137–140.
7 K. W. Wong: Phys. Lett., A **298** (2002) 238. https://doi.org/10.1016/S0375-9601(02)00431-0
8 H. Hu, L. Liu, and N. Ding: Comput. Phys. Commun. **184** (2013) 765. https://doi.org/10.1016/j.cpc.2012.11.017
9 A. Kanso and N. Smaoui: Chaos Solitons Fractals **40** (2009) 2557. https://doi.org/10.1016/j.chaos.2007.10.049
10 K. Wang, W. Pei, X. Hou, Y. Shen, and Z. He: Phys. Lett., A **374** (2009) 44. https://doi.org/10.1016/j.physleta.2009.10.021
11 H. Hu, X. Wu, and B. Zhang: Chaos Solitons Fractals **22** (2004) 359. https://doi.org/10.1016/j.chaos.2004.02.008
12 E. N. Lorenz: J. Atmos. Sci. **20** (1963) 130.
13 H. Nagashima and Y. Baba: Introduction to Chaos, Physics and Mathematics of Chaotic Phenomena (CRC Press, Boca Raton, 1998). https://doi.org/10.1201/9780429187001
14 G. Alvarez and S. Li: Int. J. Bifurcat. Chaos **16** (2006) 2129. https://doi.org/10.1142/S0218127406015970
15 P. Collett and J.-P. Eckmann: Iterated Maps on the Interval as Dynamical Systems (Birkhäuser, Boston, 1980). https://doi.org/10.1007/978-0-8176-4927-2
16 Y. Chen, L. Zhang, and Y. Weng: Proc. 2010 Int. Conf. Computer Application and System Modeling (ICCASM 2010) 431–435.
17 H. Zhang and R. CaiImage: Proc. Intelligent Computing and Integrated Systems (ICISS 2010) 113–117. https://doi.org/10.1109/ICISS.2010.5656735
18 E. Dawson: Advances in Cryptology, J. Seberry and Y. Zheng, Eds. (Springer, Berlin and Heidelberg, 1993) pp. 209–215. https://doi.org/10.1007/3-540-57220-1_63
19 M.-H. Lim, B.-M. Goi, S. Lee, and H. Lee: Proc. 2007 Int. Conf. Convergence Information Technology (IEEE ICCIT 2007) 1395–1407.
20 M. François, T. Grosgesa, D. Barchiesia, and R. Erra: Commun. Nonlinear Sci. Numer. Simul. **19** (2014) 887. https://doi.org/10.1016/j.cnsns.2013.08.032

21 A. Biryukov: Encyclopedia of Cryptography and Security, H. C. A. van Tilborg and S. Jajodia, Eds. (Springer, New York, 2011) pp. 1042–1043. https://doi.org/10.1007/978-1-4419-5906-5
22 S. Lian, J. Sun, and Z. Wang: Chaos Solitons Fractals **26** (2005) 117. https://doi.org/10.1016/j.chaos.2004.11.096
23 Federal Information Processing Standards Publication 140-1: Security Requirements for Cryptographic Modules (National Institute of Standards and Technology, 1994).
24 A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo: Special Publication 800-22 Revision 1, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (National Institute of Standards and Technology, 2008).

About the Author

Shyi-Tsong Wu was born in Jiaoxi, Yilan, Taiwan. He received his Ph.D. in electronic engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2005. He is now an associate professor at the Department of Electronic Engineering, National Ilan University, Yilan, Taiwan. He has passed the National Higher Examination in electronic engineering and the National Telecommunication Special Examination in electrical engineering. His research interests include IoT security and applications, cryptography, and electronic circuits.